universitätbonn

Rheinische Friedrich-Wilhelms-Universität Bonn
Mathematisch-Naturwissenschaftliche Fakultät

Cognitive Approaches for Mobile Vision Systems

Kumulative Habilitationsschrift

zur Erlangung der Venia Legendi im Fach Informatik

vorgelegt von

Dr. Simone Frintrop

Akademische Rätin

am Institut für Informatik III

28. März 2014

Acknowledgments

I would like to thank all the people that contributed to this thesis. First, I want to thank Armin B. Cremers, who gave me the opportunity to work in his group and who kept me as free as possible from other duties, so that I could concentrate strongly on my research. He always supported me, financially and with his advice.

Chapter 3 and 4 evolved from close cooperations with Germán Martín García and Dominik Klein. I thank you for all the fruitful discussions, your implementations, and all the other effort you have put into our work. I was especially happy to have found in Germán someone who shares my enthusiasm for the great work from cognitive psychology and I enjoyed our vivid discussions about various topics. Also, our student assistants, especially Thomas Werner and Bernd Wendt, contributed strongly by programming useful tools and evaluating data, data, and even more data. I guess Thomas is already dreaming of precision-recall plots.

Dirk Schulz, Achim Königs, and Frank Hoeller worked with me on chapter 5, and I remember well our experiments with the robot Blücher that sometimes behaved more autonomously than we wished. It was fun working with you on this. Chapter 6 was developed mainly during my postdoc year in Stockholm, mostly in cooperation with Patric Jensfelt. This was a great year, I want to thank all my Swedish and non-Swedish friends I met there for a wonderful time, and Henrik I. Christensen for the opportunity to work in his lab. He had always an open door and was willing to discuss my work. Patric was a great colleague, and I learned a lot from him about robotics, SLAM, coding, and paper writing.

I also want to thank all the people that proof-read this summary, especially Henrik Grosskreutz, Germán Martín García, and Jens Behley. I am also very grateful to John Tsotsos, Markus Vincze, Ute Schmid, and Henrik Christensen, who all took the time to comment on my classification of cognitive systems.

When it came to the end of this thesis, it turned out I had strongly underestimated the amount of work it would take to "quickly write a summary". A cumulative habilitation could not be much work, could it? Oh yes, it could! I am very thankful for my family and friends who supported me during this time. First of all, Henrik, who always encouraged me, discussed my work with me until late in the evening, took care of the kids when I had to work, and much more. This would not have been possible without you. I also thank my mother and my parents in law for always supporting me, and for being available in emergency Kita-is-closed or kids-are-sick cases. Last but not least, I want to thank my great little kids Robin and Jana for showing me every day that there is more to life than work. This helped me more than once to put everything back into perspective.

Publications that form this cumulative habilitation thesis

[1] Simone Frintrop, Germán Martín García, and Armin B. Cremers. A cognitive approach for object discovery. *International Conference on Pattern Recognition (ICPR) (accepted)*, Stockholm, Sweden, 2014.

[2] Germán Martín García and Simone Frintrop. A computational framework for attentional 3D object detection. In *Proceedings of the Annual Conference of the Cognitive Science Society*, Berlin, Germany, 2013.

[3] Germán Martín García, Simone Frintrop, and Armin B. Cremers. Attention-based detection of unknown objects in a situated vision framework. *German Journal of Artificial Intelligence, Springer*, 27, 2013.

[4] Simone Frintrop. Towards attentive robots. *PALADYN Journal of Behavioral Robotics, Springer*, 2(2), 2011.

[5] Dominik A. Klein and Simone Frintrop. Center-surround divergence of feature statistics for salient object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011.

[6] Simone Frintrop. Computational visual attention. In Albert A. Salah and Theo Gevers, editors, *Computer Analysis of Human Behavior*, Advances in Pattern Recognition. Springer, 2011.

[7] Simone Frintrop, Erich Rome, and Henrik I. Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception*, 7(1), 2010.

[8] Simone Frintrop. General object tracking with a component-based target descriptor. In *Proceedings of the IEEE International Conference Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010.

[9] Simone Frintrop, Achim Königs, Frank Hoeller, and Dirk Schulz. A component-based approach to visual person tracking from a mobile platform. *International Journal of Social Robotics, Springer*, 2(1), 2010.

[10] Simone Frintrop and Armin B. Cremers. Visual landmark generation and redetection with a single feature per frame. In *Proceedings of the IEEE International Conference Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010.

[11] Simone Frintrop and Patric Jensfelt. Attentional landmarks and active gaze control for visual SLAM. *IEEE Transactions on Robotics, Special Issue on Visual SLAM*, 24(5), 2008.

[12] Simone Frintrop and Armin B. Cremers. Top-down attention supports visual loop closing. In *Proceedings of European Conference on Mobile Robotics (ECMR)*, Freiburg, Germany, 2007.

Zusammenfassung

Sehen ist der wichtigste Sinn des Menschen und visuelle Wahrnehmung fällt uns so leicht, dass wir üblicherweise gar nicht darüber nachdenken. Dagegen hat sich die Computerinterpretation von Bilddaten in den letzten Jahrzehnten als ausnehmend schwierig erwiesen. Obwohl es viel Fortschritt im Bereich Computer Vision gegeben hat und schon sehr gute Systeme für Einzelanwendungen existieren, sind maschinelle Bildverarbeitungssysteme immer noch weit von den Fähigkeiten des Menschen entfernt. Insbesondere Aufgaben wie die automatische Erkennung von beliebigen Objekten und die Interpretation und Analyse von Szenen stecken noch in den Kinderschuhen.

In dieser Arbeit verfolgen wir den Ansatz, das menschliche Sehsystem besser zu verstehen und dessen Mechanismen zu modellieren, um verbesserte technische Systeme zu erstellen. Wir glauben, dass viel Potenzial in der Ausnutzung der Konzepte liegt, die das menschliche Sehen so mächtig machen. Die Industrie, die Forschung, und nicht zuletzt der Endanwender brauchen Systeme, die robust, flexibel, effizient, und intuitiv zu bedienen sind. All diese Eigenschaften hat der Mensch, optimiert durch Jahrtausende von Evolution, und eine eingehende Analyse dieser Fähigkeiten und Ausnutzung ihrer Eigenschaften kann für technische Systeme von großem Nutzen sein.

Die in dieser Habilitationsschrift vorgestellten Arbeiten folgen diesem Ansatz und wenden Erkenntnisse aus Psychophysik, Neurobiologie und den Kognitionswissenschaften auf verschiedene Fragestellungen der Bildverarbeitung an. Diese können in vier Themenbereiche gruppiert werden: *Object Discovery*, *Saliency Detection*, *Visual Tracking* und *Visual SLAM (Simultaneous Localization and Mapping)*. Hierbei bezeichnet Object Discovery die automatische Detektion von vorher unbekannten Objekten in Bild und Videodaten, ohne vorherige Trainingsphase. Saliency Detection befaßt sich damit, auffällige Regionen in Bildern zu finden, die als Kandidaten für weitere Verarbeitung dienen können. Visual Tracking behandelt das Verfolgen von Objekten und Personen in Bildern. Und Visual SLAM zielt darauf ab, automatisch eine Karte einer unbekannten Umgebung zu erzeugen und einen Roboter oder eine mobile Kamera in dieser Karte zu lokalisieren. Die vorgestellten Arbeiten basieren auf Erkenntnissen über die Funktionsweise des menschlichen Sehsystems und nutzen diese aus, um echtzeitfähige und kompetitive Systeme zu erstellen. Wir zeigen anhand von Experimenten mit Bilddaten aus realistischen Umgebungen, dass die Systeme, die wir entwickelt haben, robust und effizient sind, und signifikante Verbesserungen des aktuellen Forschungsstands darstellen.

Contents

Part I

Summary

Chapter 1

Introduction

Computer vision has advanced tremendously during the last two decades and reliable solutions are available now for many tasks. However, there is still a big gap between computer vision algorithms and the capability of humans concerning the interpretation of visual data. Even two year olds outperform computer systems easily in detecting and recognizing objects as well as in finding their way in their environment based on visual perception. In a recent survey by Andreopoulos and Tsotsos (2013), it was pointed out that "artificial recognition systems are still far removed from the elegance and generalization capabilities that solutions based on the organic brain are endowed with".

These capabilities are so natural to us and achieved so effortlessly that we usually do not realize how complex the tasks are that our visual system accomplishes in every moment. Think of the ability of humans to interpret images. A description might sound like this: "The picture shows a large celebration, most likely a wedding. Most people seem to be in their forties, and the location is probably northern Europe, maybe Sweden. The atmosphere is relaxed and it seems to be already late in the evening". No current computer vision system comes close to such an image analysis.

To better understand and exploit the mechanisms that make human vision so powerful, several research groups model the mechanisms of the visual system computationally (Krüger et al., 2013; Tsotsos, 2011; Serre et al., 2007; Riesenhuber and Poggio, 1999; Liao et al., 2013). This interdisciplinary research area involves many fields: scientific findings from psychology, neuroscience, and cognitive sciences are combined with well-approved methods from computer vision, artificial intelligence, and robotics. Depending on the community and the focus of the work, there are different names for this research field, e.g., *Biologically-inspired Computer Vision* or *Computational Neuroscience*. We denote it here as *Cognitive Computer Vision*, to emphasize our interest in modeling higher-level cognitive abilities of human vision, e.g., object detection and visual attention. Figure 1.1 illustrates the field and its related disciplines.

This cumulative habilitation thesis presents our research in the field of Cognitive Computer Vision. The publications that are the basis of this work have been listed at the beginning of this thesis. They can be divided into four areas that form the chapters of this summary: object discovery, saliency computation, visual tracking, and visual robot localization and mapping. Before we give an overview of these chapters in Section 1.2, we will first define terms such as "cognition" and "cognitive (vision) systems" in

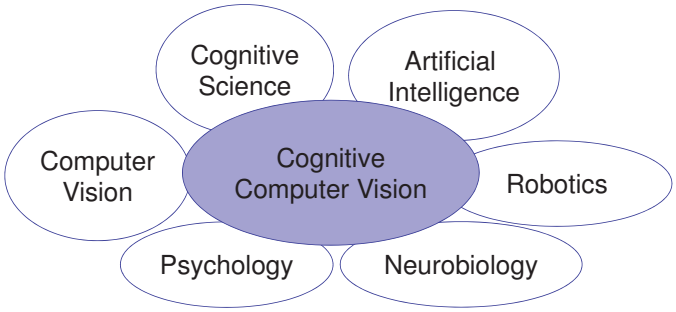


Figure 1.1: Cognitive Computer Vision and related fields

Section 1.1, and locate the work of this thesis within the related work. Finally, Section 1.3 summarizes the contributions of this thesis.

1.1 Cognition and Cognitive Systems

"Cognition" and "Cognitive System" are terms that have been used frequently during the last two decades; they appear in many different fields and contexts, and each of them interprets the terms slightly differently. This makes it of special importance to define the terms precisely, to discuss their meaning in different fields and research directions, and to locate the here presented work within the related literature.

According to Neisser, who is often referred to as the father of cognitive psychology (Hyman, 2012), the term **'Cognition'** refers to "all the processes by which the sensory input is transformed, reduced, elaborated, stored, recovered, and used" (Neisser, 1967). This includes mental processes such as attention, perception, memory, language processing, learning, awareness, reasoning, problem solving, and decision making. In science, cognition is mostly investigated in the field of cognitive psychology, but also disciplines such as neuroscience, linguistics, and philosophy contribute to the understanding of these mechanisms. **Visual Cognition** is the subfield that is concerned with the visual aspects of cognition, for example object detection and recognition, visual attention, visual search, scene recognition and categorization, visual memory, and learning based on visual perception.

Cognitive Systems are computational approaches that aim to *achieve cognitive behavior* that is similar to the capabilities of humans. The European Union has put several hundred millions of Euros into the funding of such systems in their work programs on "Cognitive Systems" and "Cognitive Systems and Robotics"[1]. The joining goal to achieve cognitive behavior can be addressed in many different ways and for different purposes. We classify the approaches for cognitive systems therefore according to two dimensions that span a space which we will call *Cognitive-System-Space (CS-space)* and which is visualized in Figure 1.2. The area of **Cognitive Vision Systems** is a subfield of Cognitive Systems, which is concerned with the interpretation of visual data. This subfield has the same dimensions as the CS-space.

[1] http://cordis.europa.eu/fp7/ict/programme/challenge2_en.html

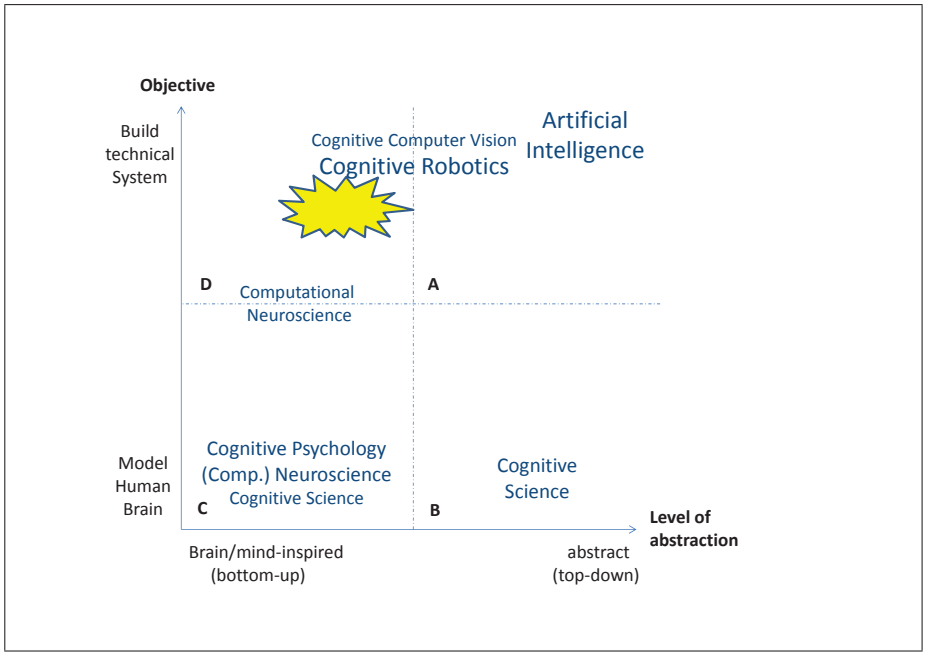


Figure 1.2: The Cognitive Systems Space (CS-space) defines systems according to their objective and their level of abstraction from the human brain and mind. The names denote research communities and they are placed where most of the work of the community is located (font size shall correspond approximately to community sizes). The yellow star denotes an approximate location of the work presented in this thesis.

The first dimension of the CS-space addresses the *objective* a system pursues. While one group of systems is designed to model and better understand human cognition, another direction is the development of technical systems that perform well in various applications. The systems that belong to the first category contribute usually to research communities such as Cognitive Psychology, Cognitive Science, and Neuroscience. The systems with a technical objective contribute mostly to engineering fields such as Cognitive Robotics, Cognitive Computer Vision, or Artificial Intelligence. A field that bridges these two extremes is Computational Neuroscience, which mostly aims at modeling the human brain, but in which many recent groups build also systems that are competitive in technical applications.

The second dimension addresses the *level of abstraction* from the human brain that is chosen to build a cognitive system. At one end of this dimension are the biologically plausible, bottom-up approaches that build models based on neurobiological and psychophysical findings. We do not distinguish here between physical brain and mind related findings, thus, at this end of the CS-space are models that simulate the neural hardware of the brain as well as systems that base on psychological models. At the other end of this dimension are the top-down approaches that regard the inside of the brain as a black box and aim at modeling cognitive behavior with engineering methods. In these models, the only brain-related inspiration is the overall goal of the system, for example

object recognition. In between can be found many approaches on different abstraction levels that simplify some concepts while retaining enough information to model a specific cognitive behavior.

We want to note that, although we think this dimension of the level of abstraction is useful to classify systems, we do not believe that one level is superior to another. Research has to take place in all of these areas of the CS-space to gain a better understanding of the principles of the human brain as well as to come up with better solutions for intelligent cognitive systems. In our belief, hybrid approaches, which take into consideration findings from different abstraction levels, are of special interest.

In the following, we separate the space into its four quadrants and discuss different approaches of cognitive systems that fall in each of these areas. It should be noted however that the distinctions are not hard and systems might fall in between quadrants. Especially the placement of scientific communities concerns only the majority of approaches and there are usually also systems belonging to a community that fall into one of the other quadrants. Additionally, we want to point out that this overview does by no means attempt to give an exhaustive overview of the field of Cognitive Systems. The mentioned projects and research groups have been chosen to illustrate the dimensions of the CS-space and are often related to the work in this thesis. Hopefully, these examples enable the reader to place also other related work into the CS-space.

A) Objective: building a technical system; level of abstraction: high (top-down).

This quadrant describes approaches with the objective to build technical systems, which address the problem mainly from an engineering perspective. Work in this field designs solutions based on techniques from artificial intelligence, logic, machine learning, or computer vision. The inside of the brain is often considered as a black box and the relation to performance of the human brain is only given by the fact that the system shall achieve similar behavior.

This interpretation of cognitive systems is strongly related to artificial intelligence and machine learning and includes many approaches in the fields of robotics and computer vision. Correspondingly, the relevant journals and conferences can be found in these fields, for example the journals *"IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)"*, *"Journal of Computer Vision and Image Understanding (CVIU)"*, *"IEEE Transactions on Robotics"*, *"Artificial Intelligence"*, *"Springer Lecture Notes in Artificial Intelligence"*, and the conferences AAAI, IJCAI, ECAI, RSS, ICRA, IROS, CVPR, ICCV, ECCV, and many more.

An example of a research direction that is at the very right side of the CS-space is the area of Cognitive Robotics that develops methods for automated, logic-based reasoning, for example the Cognitive Robotics group at the University of Toronto[2] or the Knowledge-Based Systems Group at RWTH Aachen[3]. An international event that summarizes work in this area is the international workshop of cognitive robotics that is held every two years since 1998 at the highly reputable conferences in artificial intelligence AAAI and ECAI.

[2] http://www.cs.toronto.edu/cogrobo/
[3] http://www.kbsg.rwth-aachen.de/

The research within the German national cluster of excellence CoTeSys (Cognition for Technical Systems) is located more within the center of this quadrant. CoTeSys, which was funded from 2006 to 2011, took inspiration from the human brain to develop systems such as vehicles, robots, and factories with the expectation that these would be "much easier to interact and cooperate with, and will be more robust, flexible, and efficient."[4]

Also in machine vision, many people work in areas that fall into this quadrant. For example, the ECVision network[5] (European Research Network for Cognitive Computer Vision Systems) was a forum to expose AI-oriented and systems-oriented areas of cognitive computer vision, such as knowledge representation, learning, reasoning, recognition and categorization, as well as goal specification and achievement. It was funded from 2002 to 2005 by the European Commission. One example project following this directive was the CogViSys (Cognitive Vision Systems) project[6].

B) Objective: modeling the human brain/mind; level of abstraction: high (top-down).

Interestingly, there is also a large group of approaches that follow a top-down approach to model human cognition and base on logic and probability theory. While inspired from human cognition, this inspiration is on a higher level of abstraction than the systems in quadrant C. Most of this work is located in the cognitive science community, but some works can also be found in the field of artificial intelligence. A whole research area has developed that models cognitive functions based on Bayesian theory (Knill and Pouget, 2004; Friston, 2012). The essence of the underlying "Bayesian brain hypothesis" is that the brain tries to "infer the causes of our sensations based on a generative model of the world" (Friston, 2012). Important journals in this field are *"Cognitive Science"*, *"Topics in Cognitive Science"*, and *"Trends in Cognitive Sciences"*.

C) Objective: modeling the human brain/mind; level of abstraction: low (bottom-up; brain/mind-inspired).

Cognitive systems that aim to model and better understand human cognitive behavior are often located within this quadrant. Here, the level of abstraction is low and systems aim to follow the findings of research on the human visual system closely. Correspondingly, the research communities interested in these topics are mainly Cognitive Psychology and Neuroscience. Important journals in this field include *"Cognitive Psychology"*, *"Attention, Perception and Psychophysics"*, and, in the field of vision, the *"Journal of Vision"* and *"Vision Research"*. Work in this area often constructs theoretical models to explain human cognitive behavior, but more and more groups also implement their ideas. However, these systems are mostly designed rather as a proof-of-concept and often applicable only to artificial data instead of being useful for technical applications. It should be mentioned

[4] http://www.cotesys.org/
[5] http://www.vernon.eu/ECVision
[6] http://cogvisys.iaks.uni-karlsruhe.de/

however that many cognitive models from this area serve as inspiration and basis for the biologically inspired systems of quadrant D.

The largest project in this area is certainly the *Human Brain Project* that started in 2013 and aims at building a complete computer simulation of the human brain. It is funded for 10 years by the European Union and involves more than 80 partners. Other related projects are FACETS[7] and its successor BrainScaleS[8].

Also work of the field *Computational Neuroscience* falls mostly into this quadrant, but partly also into quadrant D. According to Sejnowski et al. (1988), the goal of computational neuroscience is "to explain how electrical and chemical signals are used in the brain to represent and process information". While computational neuroscience includes also aspects that do not model cognition but rather low-level behavior, e.g., modeling single neurons, there are also approaches that model cognitive behavior such as object recognition (Serre et al., 2007; Riesenhuber and Poggio, 1999), visual attention (Kirkland and Gerstein, 1999; Itti et al., 2005; Hamker, 2004; Itti, 2003), and visual search (Hamker, 1999; Navalpakkam et al., 2004). A well-known journal that addresses these topics is the *"Journal of Computational Neuroscience"*.

D) Objective: building a technical system; level of abstraction: low (bottom-up; brain/mind-inspired).

Finally, work in this quadrant follows a biologically inspired approach based on findings from psychology and neuroscience, but aims at building and improving technical systems that are useful in applications, for example for service robots or mobile vision devices. During the last decade, there has been increasing interest in building high quality systems based on biologically inspired methods (Krüger et al., 2013; Tsotsos, 2011; Serre et al., 2007; Riesenhuber and Poggio, 1999; Liao et al., 2013). Work of this area is mostly published in the corresponding journals and conferences of engineering fields that are the same as in quadrant A. Additionally, some groups publish also in journals from quadrant C. A journal that emphasizes explicitly the connection of computer science and human perception is the *ACM Transactions on Applied Perception*.

A German cluster of excellence that deals with building Cognitive Systems of this characteristics is CiTeC[9] (Cognitive Interaction Technology) at Bielefeld University. Work in this cluster is highly interdisciplinary and deals with the key topics Motion Intelligence, Attentive Systems, Situated Communication, and Memory and Learning. Much of the research in CiTeC is dealing with Human Robot Interaction (HRI), a field that integrates research from artificial intelligence, robotics, natural language understanding, design, and social sciences. Thus, most work of HRI fits perfectly into this quadrant.

In Computational Neuroscience, there are several groups that aim not only at modeling human brain behavior, but also at building technical systems. Much work

[7] http://facets.kip.uni-heidelberg.de/index.html
[8] http://brainscales.kip.uni-heidelberg.de/
[9] http://www.cit-ec.de/

in this direction is done by the iLab[10] around Laurent Itti. They summarize their directive vividly on their webpages[11]: "to be proven truly useful and insightful, computational neuroscience models should not only be tested against neural or behavioral data in the context of specialized laboratory experiments, but should also be exercised in the context of more general applications which confront the models to the real world." Some work in the field of Computational Neuroscience is Itti et al. (1998); Walther et al. (2005); Rutishauser et al. (2004); Navalpakkam and Itti (2006); Siagian and Itti (2007, 2009); Serre et al. (2007); Tsotsos et al. (1998); Rotenstein et al. (2007). One of these examples is the PlayBot project that develops a smart wheelchair to support disabled children (Tsotsos et al., 1998; Rotenstein et al., 2007). In this project, a wheelchair locates toys in the surrounding based on biologically-inspired visual attention methods.

The work of this thesis is also located in this quadrant, and its approximate location is visualized by the yellow star in Figure 1.2.

Generally, there are many **hybrid approaches**. Especially the systems with a technical objective can be more or less biologically plausible and systems fill the whole range of quadrants C and D. Many example projects can be found in the Cognitive Systems and Robotics portfolio of the European Union, for example CoSy, VAMPIRE, CogSys, Cogniron, PACO-PLUS, MACS, BACS, CogX, ROBOT-CUP, and SEARISE.[12]

To the best of our knowledge, our classification of cognitive systems is the first that classifies systems in such a broad way and allows to place and distinguish different research directions that deal with cognitive systems. It covers viewpoints from Cognitive Psychology, Computational Neuroscience, and Cognitive Sciences as well as those from Artificial Intelligence, Robotics, and Computer Vision. One of the dimensions of our CS-space, the abstraction-level, has some relations in the literature.

For example, Vernon (2006) introduces a space of cognitive vision. He distinguishes the *cognitivist approach* and the *emergent approach*, which might correspond roughly to the top-down and bottom-up approaches of our abstraction-level dimension. Note however that our dimension focuses explicitly on the abstraction from human perception, while Vernon focuses rather on the manner in which cognition is achieved. Furthermore, our CS-space defines Cognitive Systems in general, whereas Vernon's classification is restricted to Vision Systems. The abstraction-level dimension is also similar to the "levels of analysis" that O'Reilly and Munakata (2000) mention for Computational Neuroscience models. But since their models are concerned with modeling the brain, they contain always some sort of biological inspiration. On the other hand, we explicitly include systems with a purely technical objective and no brain-inspiration besides the goal. Both definitions, Vernon's and O'Reilly's, lack the dimension of the objective of a system. We think this dimension is essential to classify approaches since the objective a group pursues influences strongly the methods, the evaluation, and the interpretation of the systems they build.

[10] http://ilab.usc.edu
[11] http://ilab.usc.edu/research/
[12] http://cordis.europa.eu/fp7/ict/cognition/projects_en.html

1.2 Overview of this Habilitation Thesis

This thesis presents our contributions in the field of Cognitive Computer Vision. In our research, we aim to build technical systems inspired by mechanisms of the human visual system; thus, most of our work can be located within quadrant D of the CS-space from the previous section (yellow star in Figure 1.2). Parts of our work could also be placed into quadrants A (Frintrop and Jensfelt, 2008b) and C (García and Frintrop, 2013).

We follow this approach since we believe that there are great opportunities for technical systems in better understanding and exploiting the concepts that make human vision so powerful. Especially the generalization capabilities of human perception as well as the ability to cope with noise, clutter, and with new situations still outperform current machine vision systems considerably. Furthermore, there is a large interest in robotics for systems that are robust, flexible, efficient, and intuitive to interact with — properties that humans have and that are worth being investigated in depth.

Following this directive, we have addressed four main topics in this thesis that will be approached in the four main chapters of this summary: object discovery, visual saliency detection, visual tracking, and visual SLAM (Simultaneous Localization and Mapping). Additional chapters are this introduction, a brief summary of foundations (Chapter 2), and a conclusion.

In the following, we summarize the content of the four main chapters. Numbers in brackets [x] refer to the number of the corresponding publication in the list at the beginning of this thesis.

- **Object discovery:** Chapter 3 presents our work on object discovery (Frintrop et al., 2014; García and Frintrop, 2013; García et al., 2013) [1, 2, 3] and (Horbert et al., 2014). The method detects unknown objects in a scene without any prior knowledge or training phase and thus answers the question "What is an object?". Our object discovery system is applicable to web images, real-world video data, as well as RGB-D data. The approach follows the principles of human perception: 1) Color and depth information are processed in parallel as in the ventral and dorsal pathway of the human visual system. 2) A segmentation method clusters similar pixels as the grouping mechanisms in human perception. 3) The order in which to analyze a scene is determined by visual attention mechanisms that direct the processing in a Saccade-Fixate cycle to regions of most potential interest. 4) Spatial Inhibition-Of-Return mechanisms, inspired from human perception, enable to remember already visited locations and facilitate the investigation of new areas. We have shown that our method outperforms several state-of-the-art methods for object discovery.

- **Distribution-based saliency:** Chapter 4 introduces our work on distribution-based saliency (Klein and Frintrop, 2011; Frintrop et al., 2014)[5, 1]. Saliency detection is a concept of human perception and is used to draw human attention to regions of potential interest. In a similar way, computational saliency methods compute a saliency value for each pixel and are useful to determine which image regions might be worth investigating in more depth. Saliency computation is useful for many applications and is used throughout this thesis. In contrast to

traditional saliency approaches, the saliency method in this chapter captures the statistics of features by probability distributions. Then, saliency is computed in an information-theoretic way by measuring the Kullback-Leibler divergence between two distributions of a center and a surround region in the image. We show that certain types of saliencies can be found with this approach that are not detected in traditional saliency systems. The representation of distributions by integral histograms makes the system real-time capable and applicable to real-world scenarios. The approach is a mathematically sound way to compute saliency, enables real-time computation, and outperforms other state-of-the-art methods in terms of accuracy. At the end of the chapter, we compare the distribution-based approach to traditional methods and discuss which method is preferable in which settings.

- **Multi-component tracking:** Chapter 5 presents our work on visual, multi-component tracking (Frintrop, 2010; Frintrop et al., 2010a)[8, 9]. We construct a flexible, component-based descriptor of the target object that is computed online from a single frame (Frintrop, 2010)[8]. The components are locally salient regions since these are especially stable and thus good candidates for tracking; additionally their extraction is quick. The number of components is chosen automatically dependent on the appearance of the target object. The component-based descriptor is integrated into the observation model of a visual tracker based on the Condensation algorithm. The method captures the structure and appearance of the target in a flexible way and is largely robust to illumination and viewpoint changes. The tracking method extends our previous work on tracking (Frintrop and Kessel, 2009, 2008) in which we presented a simple but fast way for general object tracking based on the visual search mode of a computational attention system. In (Frintrop et al., 2010a)[9], we have extended the multi-component tracker to a person tracking module as part of a mobile robot for the task of tracking individual persons.

- **Attentive visual SLAM:** In Chapter 6, we summarize our work on visual SLAM (simultaneous localization and mapping) (Frintrop and Jensfelt, 2008b; Frintrop and Cremers, 2007, 2010)[11, 12, 10]. We have presented a novel approach for visual simultaneous localization and mapping (SLAM) called "Attentive visual SLAM"(Frintrop and Jensfelt, 2008b)[11]. It proposes a new landmark selection scheme which allows the robot to reliably estimate its pose based on a sparse set of especially discriminative landmarks. In (Frintrop and Cremers, 2010)[10], we showed that salient landmarks are particularly well suited for robot localization and map generation since they have a high repeatability and are easily redetected. Using our SLAM approach, the robot can reliably estimate its location with a very sparse set of landmarks. We have also presented a new approach for active gaze control on a mobile robot to improve the previously introduced SLAM system (Frintrop and Jensfelt, 2008b)[11]. We show that the new method results in better landmarks, more frequent loop closings, and a more uniform distribution of landmarks. Using the active SLAM system, the robot is able to maintain a correct pose estimate also in difficult cases in which the passive approach fails.

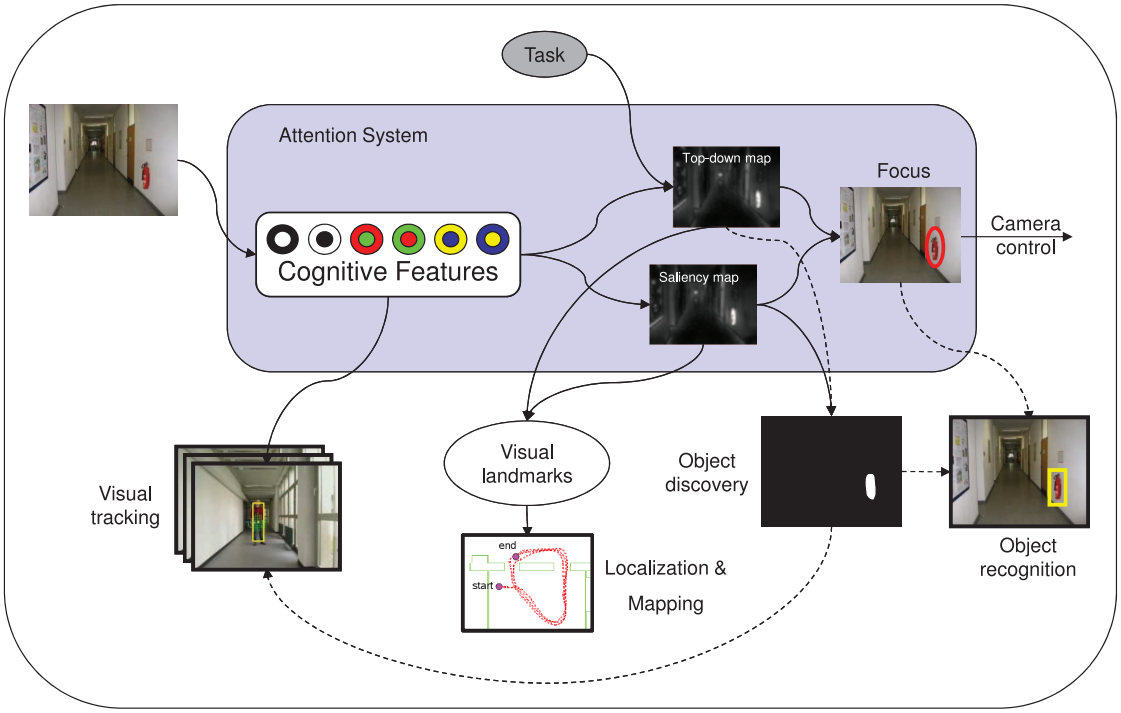


Figure 1.3: Overview of the topics of this thesis. Dashed lines indicate future directions to extend the current work by linking the topics addressed in this thesis.

A joining element of the topics mentioned above is the concept of visual attention and the sub-problem of saliency detection. In the human brain, attention pervades all our perception. It selects what is of current interest and disregards what is irrelevant. Attention does not only enable us to deal with the huge complexity of the perceptual input, it also enables us to act, since being able to focus attention on things of interest is the prerequisite for decisions. In this thesis, attentional mechanisms enable to select the candidates for the detection of objects, choose the parts of objects that are most promising for tracking, and select the landmarks for robot localization and mapping. Because of this omnipresence of attention in this work, we introduce the foundations of visual attention and saliency in Chapter 2.

While we investigated the above topics separately in the publications of this thesis, they can all be part of a larger cognitive system. In Figure 1.3, we show in a diagram how all these components could be combined into one system. Object discovery can be a pre-processing step for object recognition and thus improve speed and accuracy (most recognition methods work best on pre-segmented images). We are currently investigating this extension in our cooperation with RWTH Aachen (Horbert et al., 2014). Additionally, the visual tracker can be initialized with the object candidates that the discovery method delivers, similar as in the work in our on-going work in (Horbert et al., 2014). Up to now, most methods in this work, except (Frintrop and Cremers, 2007)[12], utilize only bottom-up saliency. Using top-down information, for example prior knowledge about the target and the scene, context information, etc., can help all methods

Habilitation paper	Chapter	Topic	Additional related publications from the author
[1]	3	2D Object Discovery	Horbert et al. (2014)
[2]	3	3D Object Discovery and Spatial Inhibition of Return	
[3]	3	3D Object Discovery and Situated Vision	
[4]	2	Survey Article	
[5]	4	Distribution-based Saliency	Klein and Frintrop (2012)
[6]	2	Survey Article	
[7]	2	Survey Article	
[8]	5	Multi-component Tracking	Frintrop and Kessel (2009) Klein et al. (2010) García et al. (2012)
[9]	5	Person Tracking	Frintrop et al. (2009)
[10]	6	Localization, Landmark Stability	Frintrop (2008)
[11]	6	Attentive Visual SLAM and Active Gaze Control	Frintrop and Jensfelt (2008a) Frintrop et al. (2007) Frintrop et al. (2006a) Frintrop et al. (2006b)
[12]	6	Top-down Landmark Detection	Frintrop (2007)

Table 1.1: Relation of the papers of this cumulative habilitation thesis to the chapters of this summary and to other related publications of the author. The references in the first column refer to the publications that form this cumulative habilitation (listed at the beginning of this thesis).

considerably to focus on objects of interest. Finally, all modules can benefit from active camera control. Currently, this is only integrated into the active visual SLAM method (Frintrop and Jensfelt, 2008b)[11], but also object discovery, object recognition, and visual tracking can benefit from high-resolution images obtained by focusing and zooming into the region of interest. However, when competing for resources and hardware access in such a complex system, an additional control and planning module is essential that decides how to use the resources. An attention module can support this by providing information about how relevant or promising certain perceptions are.

1.3 Contributions of this Habilitation Thesis

In this section, we outline the contributions of the publications that form this cumulative habilitation thesis. The numbers refer to the list of publications at the beginning of this thesis. Table 1.1 relates the papers to the chapters of this summary and to other related publications of the author.

[1] **Topic: 2D object discovery**

This work is part of the DFG project "Situated Vision to Perceive Object Shape and Affordances", a cooperation project with the universities in Vienna (TUW, Prof. Dr. Markus Vincze) and Aachen (RWTH, Prof. Dr. Bastian Leibe) and the research institute IDIAP in Martigny (Prof. Dr. Barbara Caputo). The aim of the project is to detect, recognize, and categorize objects, and one of the subtasks of our group is to detect candidate objects.

According to this, we present in (Frintrop et al., 2014)[1] a new approach for object discovery in 2D images, based on concepts of the human visual system. We compute perceptually coherent proto-objects which were assembled by saliency. We show that our method clearly outperforms several state-of-the-art methods for object detection in terms of precision and recall (more in our current work Horbert et al. (2014)). Additionally, we improved the saliency system from Klein and Frintrop (2012) and show that it outperforms 7 state-of-the-art saliency methods. The saliency system is able to work on web images as well as on real-world images from a moving camera.

[2, 3] **Topic: 3D Object Discovery**

This topic is also part of the above mentioned DFG project. The work in (García and Frintrop, 2013; García et al., 2013)[2, 3] addresses a second subtask of our group: building a 3D map of the environment that integrates object information over time. According to this, we have performed Object Discovery in 3D data from an RGB-D sensor. Our main contributions are:

- **Color-Depth Stream and Saccade-Fixate Cycle:** We generate 3D objects models with an attention-guided object discovery method on RGB-D data. Following the concepts of human vision, we separate color and depth processing and fixate object candidates sequentially in a Saccade-Fixate Cycle. This accumulates object information over time and results in 3D object models. To our knowledge, our approach is the first that performs attentional scene exploration in 3D data.
- **Spatial Inhibition of Return:** In our 3D object discovery method, we simulate the human mechanisms of inhibition of return (IOR) to enable orientation towards novelty. In contrast to previous work, we root the inhibition information not in image but in spatial coordinates which corresponds to findings from human vision and enables us to deal with dynamic scenes.

While (García and Frintrop, 2013)[2] focuses stronger on the spatial inhibition of return and the 3D map generation, (García et al., 2013)[3] gives emphasis on rooting the 3D object discovery in a Situated Vision paradigm.

[4] **Topic: Survey on Attentive Robots**

The interest for attentional modules has strongly increased in robotics within the last decade. Systems are now mature enough to enable more complex behaviors and

interaction of different modules. Therefore, more and more groups are interested in attentional modules that prioritize the processing. To reflect this interest and to provide an overview of approaches in this field, we present in (Frintrop, 2011b)[4] a survey on attentive robots. These are systems, which have a visual attention module that directs their resources to the most promising parts of the sensory input. We have outlined the similarities of the needs and requirements of robots and humans, and discussed the chances and challenges in this field.

[5] **Topic: Distribution-based Saliency**

We introduce in (Klein and Frintrop, 2011) a new method to compute saliency based on probability distributions. Feature statistics are represented by distributions and compared in an information-theoretic way by the Kullback-Leibler divergence between distributions of center and surround regions. To maintain real-time performance despite the computational complexity of comparing distributions of large image regions, the approach uses integral histograms to represent the distributions. The approach is a mathematically well-founded way to compute saliency, enables real-time computation, and outperforms other state-of-the-art methods in terms of accuracy. Since its appearance in November 2011, it was already cited 43 times (Google Scholar, March 2014).

[6] **Topic: Textbook Introduction to Attention**

We present in (Frintrop, 2011a) a book chapter in a textbook for graduate students that explains the concepts of computational visual attention for newcomers to the field. Methods and background are explained in a comprehensive way and illustrated with many examples. A list of available source code and of benchmarking databases completes the overview.

[7] **Topic: Interdisciplinary Survey on Visual Attention**

Due to the increasing interest in computer vision and robotics in attentional mechanisms, it is essential to make the background of visual attention in human perception accessible to technically and mathematically educated researchers. This need is addressed by our survey on computational attention systems and their cognitive foundations in the interdisciplinary journal "ACM Transactions on Applied Perception" (Frintrop et al., 2010b). It aims to bridge the gap between researchers from different disciplines, namely psychology, neurobiology, and cognitive science on the one hand, and computer vision, robotics, and artificial intelligence on the other hand. It introduces the cognitive foundations required to understand the biological basis of visual attention, explains computational approaches, and gives an overview of existing methods. Since its appearance in 2010, it was cited already 152 times (Google Scholar, March 2014).

[8] **Topic: Multi-component Object Tracking**

We present in (Frintrop, 2010) a novel approach for general object tracking that is based on a multi-component target representation. It can be applied to arbitrary objects without previous training phase, the method learns the appearance of the

target online from a single frame. The main idea of the approach is to build a flexible, component-based descriptor of the target object which can be used for tracking. The components are locally salient regions since these are especially stable and thus good candidates for tracking; additionally their extraction is quick. The component-based descriptor is integrated into the observation model of a visual tracker based on the Condensation algorithm. The method captures the structure and appearance of the target in a flexible way and is largely robust to illumination and viewpoint changes. With these properties and its real-time capability, the method meets all requirements that have to be fulfilled to use the method on mobile vision systems, e.g., on autonomous service robots.

[9] **Topic: Person Tracking**

This work (Frintrop et al., 2010a) was performed in cooperation with the group of Dr. Dirk Schulz at the Fraunhofer FKIE institute, and robot experiments were performed at the FKIE institute. Here, we extend the multi-component tracker from Frintrop (2010) to a person tracking module as part of a mobile robot for the task of tracking individual persons. In contrast to shape-based people trackers, our method aims to distinguish individuals. This is of special importance for service robots, for example for guiding a specific person through a museum or hospital. Depending on the appearance of the person (clothing, hair color, skin color etc.), the system determines a flexible number of components, each representing a discriminative part with respect to a certain feature dimension. Because of its flexibility, the approach is also able to deal with unusual appearances, for example with people wearing a backpack or carrying a large object. Our method is able to run in real-time on a mobile platform and can track persons in real-world environments under varying lighting conditions and backgrounds. We have compared the method with other tracking approaches and show that it outperforms other methods clearly.

[10] **Topic: Localization, Landmark Stability**

In (Frintrop and Cremers, 2010), we show that salient landmarks are especially well suited for robot localization and map generation since they have a high repeatability and are easily redetected. We compare salient regions with other feature detectors and show that we obtain a higher repeatability in tracking as well as in viewpoint change situations. Our experiments show that visual landmark generation and redetection is possible with a single feature per frame because of the high redetectability of the salient landmarks. This can be exploited for topological robot localization, since in this application, it is in principle sufficient to have one landmark every few meters, it is not necessary to see a landmark in each frame as long as the scene is recognized every few seconds. We show in an office environment with different floors that a reliable scene localization can be performed with a single feature per frame.

[11] **Topic: Attentive Visual SLAM**

This work (Frintrop and Jensfelt, 2008b) on attentive visual SLAM (Simultaneous Localization and Mapping) was performed during my postdoctoral stay at KTH

Stockholm within the EU project NEUROBOTICS. The article in the highly reputable journal "IEEE Transaction on Robotics" was cited already 84 times (Google Scholar, March 2014). The main contributions are the following:

- **Attentive Visual SLAM with Salient Landmarks:**
  We present a novel approach for visual simultaneous localization and mapping (SLAM) called "Attentive visual SLAM". It proposes a new landmark selection scheme which allows the robot to reliably estimate its pose based on a sparse set of especially discriminative landmarks. A visual attention system detects salient features that are highly discriminative because of the uniqueness property of salient regions. Therefore, they are ideal candidates for visual landmarks that are easy to redetect. Features are tracked over several frames to determine stable landmarks and to estimate their 3D position in the environment. Matching of current landmarks to database entries enables loop closing. In real-world robot experiments, we show that reliable pose estimation is possible with a very sparse set of landmarks (below 100 per environment) which is in contrast to other visual SLAM approaches that usually require hundreds of features per frame and several thousands of landmarks per environment.
- **Active Gaze Control:**
  We present a new approach for active gaze control on a mobile robot to improve the attentive SLAM system. The active gaze control module controls the camera according to three behaviors: redetection of landmarks, tracking of landmarks, and exploration of unknown areas. These behaviors make it possible to, first, observe landmarks for a longer time resulting in better landmark representations, second, actively redetect landmarks to enable more frequent loop closings, and finally, achieve a more uniform distribution of landmarks in the environment. We present several experiments showing that the active SLAM approach enables to regain a correct robot pose in difficult cases in which the passive approach fails.

[12] **Topic: Top-down Landmark Detection**

We extended in (Frintrop and Cremers, 2007) the landmark redetection module of our attentive SLAM approach with top-down information about expected landmarks. This results in explicitly supporting the features of expected landmarks. Information about which landmarks are expected is provided by the SLAM module, based on the estimated robot pose and the map. In several real-world experiments, we show that while bottom-up matching shows advantages in easy matching situations like tracking features in consecutive frames, the top-down matching outperforms the bottom-up strategy considerably in difficult matching situations with changing viewpoints. Therefore, the method is especially suited for loop-closing situations in which the robot returns to a previously visited location.

Chapter 2

Foundations

In this chapter, we will introduce some of the basic foundations required as background in the following chapters. We start in Section 2.1 with a brief overview of human vision. In Section 2.2, we introduce then the basic concepts of computational attention and saliency systems, since these will play a role in all of the following chapters.

2.1 The Human Visual System

In this section, we will provide a brief overview over the basic principles of human perception. The overview is by no means exhaustive but focuses on the aspects of perception that are relevant for the work in this thesis. For more details, we point the reader to (Zeki, 1993; Kandel et al., 1996) or the comprehensive interactive website about the human brain and behavior (Dubuc, 2014).

2.1.1 Overview

Vision plays a primary role in human perception: the eyes process more information (10^9 bits/sec (Itti and Koch, 2001a)) than any other sense and about "50% of the human neocortex responds to changes in the visual environment" (Ettinger and Klein, 2014). An overview of the areas involved in visual processing is shown in Figure 2.1. Visual information processing in the human brain starts already in the eye: the light falls onto the retina, where the photoreceptors convert the light into nerve impulses. Two types of photoreceptors exist: the rods respond only to brightness, whereas the cones are color sensitive. Color perception is thereby performed by three types of cells: the L-cones react mainly to red, the M-cones to green, and the S-cones to blue light.

The photoreceptors are connected via bipolar cells with the ganglion cells. These cells are still part of the retina, and they transform the analog signal to a discrete one. These cells will later be important for our computational systems, since they are the first place in the human visual system to compute contrasts. This is performed by their center-surround (On-Off or Off-On) structure: They respond excitatorily to light at the center of their receptive field[1] and inhibitorily to light at the surround or vice versa.

[1] The receptive field of a cell is the collection of other cells that influences the output of the cell.

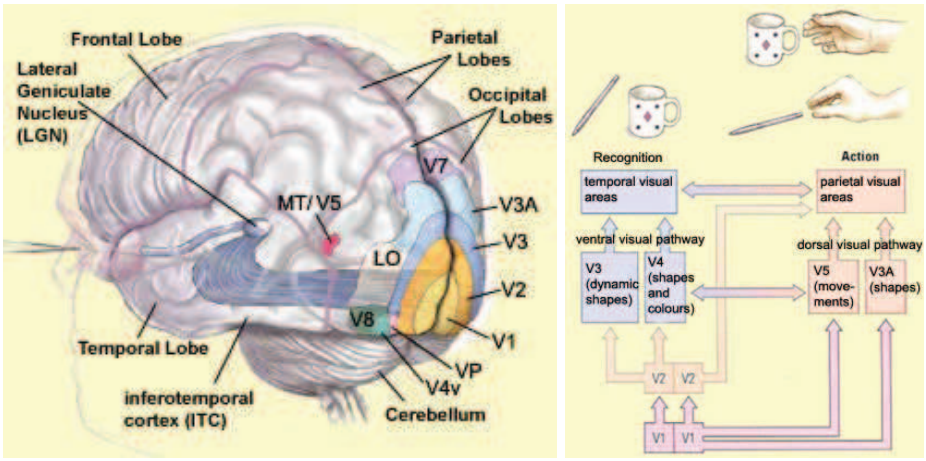


Figure 2.1: Left: The most important areas involved in visual processing in the human brain. Right: the two streams involved in visual processing: the ventral stream, responsible for object recognition, and the dorsal stream for locating objects in space. (Figures from Dubuc (2014)).

This means, they have the strongest response if the center is bright and the surround dark (On-Off cells) or vice versa (Off-On cells). Similar but increasingly complex cells are found in higher brain areas: the Lateral Geniculate Nucleus (LGN), primary visual cortex (V1), and the areas of the *extrastriate cortex*[2].

Ganglion cells are divided into three types, organized in three channels which are also called "the cardinal directions of color space" (Krauskopf et al., 1982): the luminance channel, the red-green channel, and the blue-yellow channel (Gegenfurtner, 2003). These channels lead from the retina to higher brain areas. Cells exist with concentric receptive fields and with elongated ones. It has been shown that the concentric fields are modeled best with a two-dimensional Difference-of-Gaussian (DoG) function (Rodieck, 1965), while the elongated fields are model-led best with Gabor filters (Jones and Palmer, 1987).

From the retina, most of the visual information is transmitted to the LGN, and from there to the primary visual cortex (V1). These areas are retinotopically organized, which means that adjacent locations contain information about adjacent locations on the retina (and thus in the visual scene). From V1, the information travels on to higher brain areas in the extrastriate cortex. The most important areas of the extrastriate cortex are V2, V3, V4, V5 (or MT), the infero-temporal cortex (IT/ITC), and the posterior-parietal cortex.

Important for our work, especially in chapter 3, is the functional separation of the visual processing that starts already in the retina and becomes more apparent in higher brain areas. The processing is separated into *two pathways* (Ungerleider and Mishkin, 1982): first, the *ventral stream (or "what pathway")* which is strongly involved in color and form processing and is responsible for object detection and recognition, and, second,

[2]The extrastriate cortex consists of all visual areas in the cortex, except V1 which is also called striate cortex. Thus, extrastriate means basically "beyond the striate cortex".

the *dorsal stream (or "where pathway")* which processes mainly motion and depth cues and is responsible for object localization (cf. Figure 2.1). Goodale and Milner pointed out that these streams perform "vision for perception" and "vision for action" respectively, highlighting that it is the functional difference between the brain areas rather than their visual input that matters most (Goodale and Milner, 1992; Milner and Goodale, 2008).

In this thesis, we are mostly interested in object detection and thus in the ventral visual pathway (Grill-Spector, 2003). This pathway starts its processing as early as the retina, where it derives information mainly with help of the cones, which are responsible for color perception. Blob and edge structures are then recovered by the ganglion cells, and later on, by the P-cells of LGN, and the simple and complex cells of the primary visual cortex (V1). Information then travels on over the extrastriate visual areas V2 and V4 to IT, which is responsible for object recognition.

In the following, we will cover in more detail two aspects of human perception that are especially important for the work of this thesis: visual attention (Section 2.1.2) and object perception (Section 2.1.3).

2.1.2 Human Visual Attention and Saliency Perception

Attention belongs to the most important capabilities of human perception since it guides the processing capacities of the brain to the parts of the perceptual input that are of most potential interest (Pashler, 1997). This concept of *selective attention* enables the brain to deal with the huge complexity of the incoming sensory data: the information entering the eye is estimated to be on the order of 10^9 bits per second (Itti and Koch, 2000) which exceeds by far the processing capabilities of the brain. Thus, attention mechanisms prioritize the perceptual data and direct attention in a serial manner to regions of interest. However, the purpose of attention is not limited to pure complexity reduction: ignoring irrelevant information is essential to concentrate on what is relevant and to enable an understanding of the perceptual input, or as Carrasco put it: "Attention [...] is the mechanism that turns looking into seeing" (Carrasco, 2011).

Human visual attention has been investigated since a long time; already Aristotle was fascinated by its capabilities (Aristotle, BCE). During the last decades, research on attention has increased strongly: a PubMed[3] search on visual attention retrieved 33338 articles (February 3rd, 2014), of which only 62 were published before 1970 and more than 2000 each year between 2008 and 2013 (cf. Figure 2.2). An early finding in the field of attention was the discovery that visual processing takes place in two steps: a pre-attentive stage analyzes the visual field quickly in parallel and an attentive stage processes these regions serially (Neisser, 1967). That means, complex processes such as object recognition operate not on the whole visual input but only on the currently attended region.

The selectivity of visual attention is deeply rooted in the physiology of the human visual system and strongly connected to eye movements: the fovea — the center of the retina which is responsible for sharp vision — covers only about 1° of the retinal size, but accounts for 50% of the visual information that is transferred from the eye to the visual cortex (Ettinger and Klein, 2014). Thus, by moving our eyes to a target of interest, we

[3] http://www.ncbi.nlm.nih.gov/pubmed

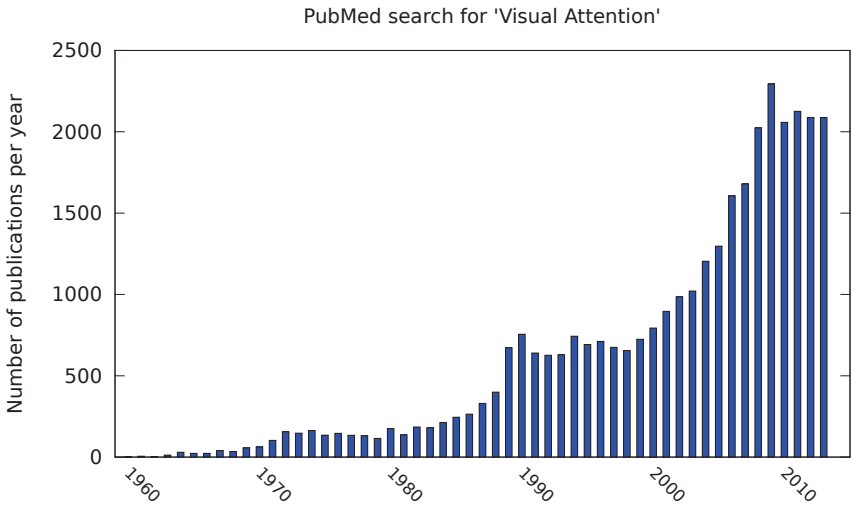


Figure 2.2: Result of a PubMed search for "visual attention": 33338 articles were found between 1960 and 2013.

enable a detailed processing of this area. Humans scan their environment permanently with such quick eye movements, also called saccades[4], which are separated by fixations. We will call this *Saccade-Fixate Cycle* in the following. However, attention can also be directed to a region without moving the eyes to this place. This aspect is called *covert attention*.

The mechanisms of visual attention can be separated into *bottom-up* and *top-down processes* (Connor et al., 2004). Bottom-up attention is based purely on the sensory data and is passive and automatic. Saliency is an important aspect of bottom-up attention. It guides the gaze to regions that stick out of the surrounding and that automatically attract our attention. Top-down attention on the other hand is usually volitional and "under control of the intentions of the observer" (Theeuwes, 2010). If a person is for example looking for a specific object (*visual search*), the attention is guided by knowledge about the appearance of the object, about likely locations in a scene, and so on. Whether aspects such as emotions, expectations, and previous experience belong to bottom-up or top-down cues is controversial. Some people classify such aspects as top-down cues, because they originate from the mind of the observer (Corbetta and Shulman, 2002), while others define them as bottom-up as long as they do not correspond to the intentions of the person (Theeuwes, 2010). While top-down attention is an important aspect in human attention, such cues are not always available in a machine vision system and many computational methods profit from purely determining the bottom-up saliency.

Many psychological models of visual attention have been proposed during the last decades (see Bundesen and Habekost (2005) for a survey). The most influential for computational attention systems have been the Feature Integration Theory (FIT) (Treisman

[4]Saccade: rapid, irregular eye movement that occurs when changing focus from one point to another (Random House Kernerman Webster's College Dictionary, 2010)

and Gelade, 1980) and the Guided Search model (Wolfe, 1994). They state that early in the visual processing stream, features are processed in parallel and compete for selective attention. Opinions differ about which basic features guide human attention, but it is undoubted that color, orientation, motion, and size are among them (Wolfe and Horowitz, 2004). Although neurobiological findings show that there is no such clear segregation of feature computations (Gegenfurtner, 2003), there is a bias of several brain areas for specific features, for example color processing in V4 and motion processing in V5/MT (middle temporal area).

For each of these feature channels, pattern recognition is performed by basic blob and edge detection cells. As mentioned in Section 2.1.1, many of these cells have a center-surround structure and thus respond to contrasts. These contrasts can be based on intensity, color, orientation, depth, or motion. Important is that these contrasts measure how much a region differs from its (spatial or temporal) neighborhood, which is the essential aspect of saliency computation.

The salient stimuli of all feature dimensions compete for attention. Therefore, Treisman has introduced the concept of a *master map of locations*, denoting a map that "collects" saliencies from different features and indicates where salient regions in the field of view are. Later on, this map has been called *saliency map*. Recent work in psychology and neurobiology has indicated that in human vision, this bottom-up saliency computation might take place in the primary visual cortex (Zhang et al., 2012).

Finally, an important aspect in attention-guided processing is *inhibition of return (IOR)*. This mechanism was discovered by Posner and Cohen (1984) and suppresses the processing of locations and objects that have recently been the focus of attention. It enables to withdraw attention from fixated regions and orient towards novelty. It has been shown that IOR happens in spatial, not in retinotopic coordinates and that it can be environment-based (inhibiting a spatial location) as well as object-based (inhibit an object, even if it moves) (Tipper et al., 1994). We will come back to this aspect in chapter 3, where we use IOR for attention-based scene exploration.

For more details on the cognitive foundations of visual attention and how they relate to computational attention systems we point the reader to our survey in (Frintrop et al., 2010b).

2.1.3 Human Object Perception

Object perception is deeply rooted in the human visual system and enables a fast and effortless detection of objects. Even objects of completely unknown appearance are easily recognized as objects. Already young infants that are only a few months old can reliably detect objects (von Hofsten and Spelke, 1985; Spelke, 1990). It is not yet completely understood how object perception works in the human brain, but many findings are well known. We will concentrate here on the findings which are important for our computational object discovery framework that will be presented in chapter 3.

Physiologically, object detection and recognition take place in the ventral visual stream that was mentioned in Section 2.1.1. It starts as early as in the retina and goes up to IT. The dorsal stream on the other hand is responsible for locating the ob-

jects in space. Both of these aspects will be important in our computational model of object perception.

Furthermore, there is evidence that the individuation of objects, which addresses the question of what is an object, takes place before object recognition (Pylyshyn, 2001). The decision of which parts of the visual scene belong to objects results from perceptual organization rules, especially from segmentation processes that bundle parts of the visual input. This bundling is based on concepts such as similarity, proximity, and other processes that have been described already early by the Gestalt principles (Wertheimer, 1922). A recent review about the history of the Gestalt laws and new findings of their rooting in the human visual system can be found in (Wagemans et al., 2012). Such segmentation mechanisms that individuate objects are believed to exist on all levels of the visual system (Scholl, 2001).

The result of these segmentation processes are so called "proto-objects" (Rensink, 2000). They describe the local scene structure of a spatially limited region and often correspond to objects, but they can be also object parts or collections of several objects. Rensink describes them as volatile structures of limited spacial and temporal coherence, meaning that they are regenerated constantly and not stored in visual memory. Later on, proto-objects are combined by focused attention to form coherent objects. This is an important step, since it enables to decide which segments an object consists of.

2.2 Computational Attention and Saliency Systems

Computational visual attention systems aim to find regions of interest in images, that mean, regions that attract the attention of humans. In general, attention systems can consist of a bottom-up part that is automatically guided by properties of the visual data (usually by salient regions) and a top-down part that is guided by intentions and goals of the observer. In this work, we focus mainly on the bottom-up part which is especially of interest if top-down knowledge is not available. More on top-down attention can be found in our previous work (Frintrop et al., 2005; Frintrop, 2006).

Computational attention systems usually follow a structure that is motivated from psychological models of visual attention, such as the Feature Integration theory (Treisman and Gelade, 1980) or the Guided search model (Wolfe, 1994), which were mentioned in the previous section. These models suggest an independent feature computation for features such as color or orientation and the fusion of saliencies in a single map, which was called "master map of locations" by Treisman, and "activation map" by Wolfe. One of the first computational models that was constructed according to these findings was the Koch-Ullman model from 1985 (Koch and Ullman, 1985) and later implementation by Milanese et al. (1994) and the group around Laurent Itti (Itti et al., 1998).

An overview of the structure of such systems is shown in Figure 2.3. The basic components are a separation of feature channels (1), a hierarchical investigation of different scales, usually with an image pyramid (2), a center-surround method to capture feature contrast (3), a fusion of feature conspicuities to a saliency map (often including a uniqueness weighting of channels) (4), a method to find the most salient region (originally a winner-take-all network) (5), and an inhibition of return mechanism that prevents the focus of attention from returning to previously visited areas (6). An additional important

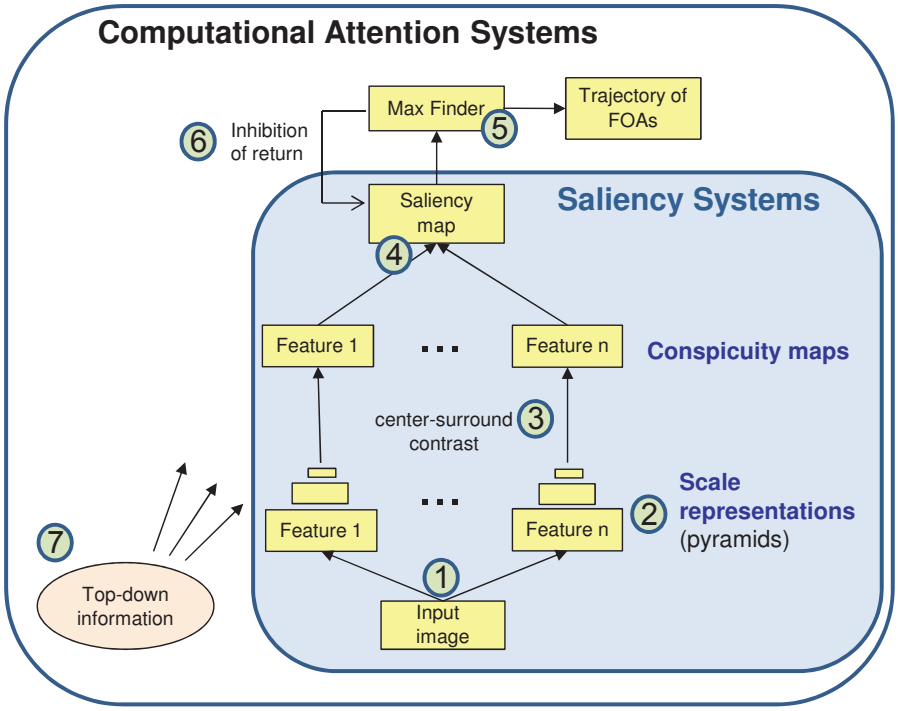


Figure 2.3: General structure of computational visual attention systems

factor is the top-down information (7) which can influence the processing in different ways. Since in this thesis we are mainly interested in bottom-up attention and saliency, we do not tackle this aspect here. The result of an attention system is then a trajectory of foci of attention (FOAs), ordered by their saliency. The basic structure shown in this figure can still be found in modified forms in most existing attention systems. Saliency systems mostly focus on steps (1) – (4) and end with the computation of a saliency map.

The most essential element of saliency methods and bottom-up attention is most likely the mechanism to compute the center-surround contrast between an image region and other parts of the image (number (3) in Figure 2.3). A high contrast in some feature dimension is an intrinsic property of a salient item (by definition it "stands out relative to its neighbors"[5]) and basically all saliency methods compute such a value. Thus, one important element when distinguishing saliency methods, maybe the most important one, is the way this contrast is computed. The traditional method to compute the center-surround contrast is to apply Difference-of-Gaussian (DoG) or Gabor filters (Itti et al., 1998; Frintrop, 2006), since these are known to model best the concentric and elongated cells of the human visual system (Rodieck, 1965; Jones and Palmer, 1987). In Chapter 4, we will discuss other methods to compute the center-surround contrast and introduce an alternative to the DoG approach.

In practice, another important element is the scale-space structure (number (2) in Figure 2.3). It is commonly used in many computer vision areas and also for saliency computation such a structure is necessary to find salient objects of different sizes. Thus,

[5]Wikipedia: "Salience (neuroscience)", Jan. 2014.

most methods operate on image pyramids or vary the size of the center-surround filter, which has the same effect.

The separation of feature channels (1) is also essential to detect different types of saliency. A red item among green ones can only be detected with a color channel, a vertical bar among horizontal ones requires an orientation channel, and moving elements a motion channel. Nevertheless, many recent saliency models compute saliency only based on color features (e.g. Achanta et al. (2009)). This already leads to good performance for specific applications and benchmarks, where images are often strongly color-based (e.g. web images in the MSRA database (Liu et al., 2009)). However, depending on the application, other feature channels are essential.

The fusion of channels to a single saliency map (4) is of course only necessary if several feature channels are computed. There are however also approaches that propose to regard the feature channels separately and not fuse them in the end (Draper and Lionelle, 2003). The fusion of channels often includes a weighting step, which we call uniqueness weight (Frintrop, 2006). This function gives more emphasis to outliers than to frequently occurring elements. Such a weighting can be realized by a non-linear function that considers the number of local peaks in a feature map (Itti et al., 1998; Frintrop, 2006) or by lateral inhibition in the map (Itti and Koch, 2001b). Additionally, such a weighting makes only sense if there are many elements in an image, which is the case for most real-world images, but not for many photographs and web images. For example, benchmarks as the MSRA database (Microsoft Research Asia Salient Object Database) (Liu et al., 2009), on which most current saliency methods are evaluated, contain often only a single object per image.

The two final steps, i.e., selecting the most salient item (5) and inhibition of return (6), are often ignored in recent saliency systems (Achanta et al., 2009; Liu et al., 2009; Hou and Zhang, 2008; Marchesotti et al., 2009; Goferman et al., 2011; Zhu et al., 2013). Instead, they end with the computation of the saliency map. This is sufficient for evaluating the quality of saliency maps, but if regions shall be selected from the map for further processing and if image sequences instead of still images are considered, these steps are important. While determining the order of regions according to their saliency is simple, step (5) involves as well determining the size of the region. This is related to segmentation, which is still not satisfyingly solved in the general case. IOR (6) is of special interest in dynamic scenes in which it is necessary to remember which items have already been focused to enable fixations on novelty. We will address (5) and (6) in more detail in Chapter 3.

The quality of a saliency method is evaluated by comparing the saliency maps of a collection of images to corresponding ground truth. Ground truth for an image is given by a ground truth map G that has the same dimensions as the corresponding input image I. The ground truth map can be either obtained from human eye movements as in (Kootstra et al., 2008) or from user labelings of salient objects as in (Liu et al., 2009; Achanta et al., 2009). One example of user labeled data is the MSRA database of salient objects that was introduced by Liu et al. (2009) and is nowadays in the computer vision community the most frequently used benchmark for evaluating saliency systems (Achanta et al., 2009; Liu et al., 2009; Hou and Zhang, 2008; Marchesotti et al., 2009). We will also use this database for some of our experiments in Chapter 3 and 4.

Given a benchmark dataset, a saliency method is evaluated by computing the difference $D = d(S, G)$ of the saliency map S with respect to the ground truth map G, where d is a distance measure. Borji and Itti (2010) propose different ways for comparing saliency maps with ground truth. The one which is used most frequently in the computer vision domain (e.g. Achanta et al. (2009)), and which we also used for our experiments in (Klein and Frintrop, 2011) and (Klein and Frintrop, 2012), is to threshold the saliency map S, consider it as a binary classifier, and use analyzing methods from signal detection theory (Receiver Operating Characteristics (ROC curves) or precision-recall curves and Area Under Curve (AUC) metrics) to evaluate this classifier.

Borji and Itti (2010) have defined the goal of attention modeling as finding a saliency function f_{best} that minimizes the error on eye fixation prediction, which we extend here to general ground truth that could, e.g., also result from user labelings:

$$f_{\text{best}} = \arg\min_{f} \sum_{j=1}^{N} d(f(I_j), G_j), \tag{2.1}$$

for an image collection $\mathbf{I} = \{I_j\}_{j=1}^{N}$ and corresponding ground truth maps $\mathbf{G} = \{G_j\}_{j=1}^{N}$. Since f can be in principle any possible function, this equation is in its general form of limited use. However, for a given saliency method with k parameters, these can be determined by minimizing the error according to this equation on a training dataset.

This brief overview of the main concepts of computational attention systems gives the basis for the following chapters. For more details, we point the reader to our survey in (Frintrop et al., 2010b) and to the book chapter (Frintrop, 2011a) which is written for graduate students and contains many useful details about implementation and evaluation of attention systems. Additional reviews of methods for salient object detection and benchmarks can be found in (Borji and Itti, 2012b; Borji et al., 2013), and a review of applications of attentional approaches in robotics is presented in (Frintrop, 2011b).

Chapter 3

Object Discovery

Object discovery is the task to find all objects in a scene without knowing how they might look like or what category they belong to. In contrast to object recognition or classification, the types of objects are not known in advance, there is no training phase, and the system starts without any prior knowledge. Following the phrasing of Alexe et al. (2010), such a system addresses the question "what is an object?". This topic is of interest for many applications, ranging from automatically cropping the most interesting thumbnail from your holiday pictures up to collecting a database of objects with an autonomous service robot that explores a new environment. Some typical images in which finding objects is of interest are shown in Figure 3.1.

In this chapter, we present our work on object discovery that follows several principles of human vision. We show that our object discovery system achieves good results on web images, real-world video data, as well as on RGB-D data. The work was performed together with Germán Martín García within the DFG project "Situated Vision to Perceive Object Shape and Affordances", a cooperation project with the universities in Vienna, Austria (TUW, Prof. Dr. Markus Vincze) and Aachen, Germany (RWTH, Prof. Dr. Bastian Leibe), and the research institute IDIAP in Martigny, Switzerland (Prof. Dr. Barbara Caputo). The aim of the project is to detect, recognize, and categorize 3D objects, and the subtask of our group is to detect candidate objects and to build a 3D map of the environment that integrates the object information over time. Both parts are addressed in the publications that are the basis of this chapter (García and Frintrop, 2013; García et al., 2013; Frintrop et al., 2014; Horbert et al., 2014).

The contributions presented in this chapter are first, a new method to discover objects based on concepts of human perception that clearly outperforms other state-of-the-art methods for object discovery (Frintrop et al., 2014; García and Frintrop, 2013), second, an extension of attention-based scene exploration to RGB-D data which is to the best of our knowledge the first attention method that operates directly on 3D data (García et al., 2013; García and Frintrop, 2013), and finally, a spatial inhibition of return method that roots the inhibition information in 3D voxels and enables thus to deal with dynamic scenes (García et al., 2013; García and Frintrop, 2013).

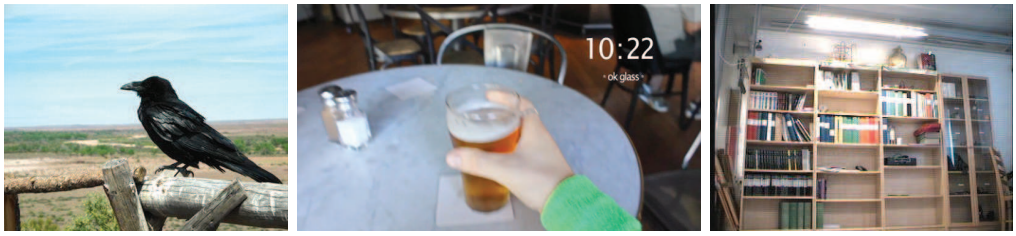


Figure 3.1: Some examples of different image types in which object discovery is of interest: a typical web image (MSRA database (Liu et al., 2009)), an image obtained with Google Glass[1], and one obtained with a mobile robot (lab at KTH, Stockholm).

3.1 Object Discovery – An Overview

Before we address the problem of finding objects, let us clarify what we mean by an "object". Although the term is so commonly used, a clear definition is not easy. Most definitions come from philosophy and are often not suitable for practical purposes, for example, the definition by Charles S. Peirce who is sometimes called the father of pragmatism: "By an object, I mean anything that we can think, i.e., anything we can talk about" (Peirce). This is counter-intuitive to our usual understanding of objects that would not consider things like 'hope' or 'mathematics' as objects. Instead, we follow here a definition from psychology: according to von Hofsten and Spelke (1985), objects are "manipulable units with internal coherence and external boundaries". This is a practical definition that is also useful for applications in computer vision and robotics. Similar, and also suitable, is a definition from the computer vision area: "Objects are standalone things with a well-defined boundary and center, such as cows, cars, and telephones, as opposed to amorphous background stuff, such as sky, grass, and road" (Alexe et al., 2010).

The notation for the task to detect and localize unknown objects in a scene varies strongly among communities. While the robotics community calls the problem *object discovery* or *general object detection*, in computer vision the problem is rather known as *object proposal detection*. Literature in cognitive science and psychology speaks usually about *object detection* or *object perception*. In this thesis, we call the problem "object discovery" since we think that the term best describes the fact that objects are not known in advance. Also the items that result from the segmentation steps as well as the finally resulting object candidates have different names in the communities. To disambiguate the notation, we list the terminology in Table 3.1.

While objects are discovered easily and effortlessly by humans, it is a challenging task for machine vision and belongs to the open problems in the field. The reason is the 'chicken-and-egg property' of the problem: how to search for an object before knowing how it looks like? To get an impression of the complexity of the task, let us regard the problem formally. In that sense, object discovery means we are interested in an algorithm that can decide whether a given pixel set corresponds to an object or not.

[1] http://www.androidmag.de/news/technik-news/neue-google-glass-parodie-google-glass-ermoglicht-ein-schnelleres-leben/

Community:	Computer Vision	Robotics	Cognitive Psychology
Task:	Object proposal detection	Object discovery/ General object detection	Object detection Object perception
Results of segmentation:	Segments/ Superpixels	Segments	Proto-objects
Final results:	Object proposals Object candidates Object hypotheses	Object candidates Object hypotheses	Proto-objects Object candidates Object hypotheses

Table 3.1: Disambiguation of terminology in different communities.

But even if we had a method to answer this question reliably, the problem would still be complex: an image of $w \times h = n$ pixels consists of 2^n possible subsets that could potentially form an object (due to partial occlusions, object parts do not necessarily have to be connected). Tsotsos has proven that the related problem of unbounded visual search, i.e., search for an object whose features are unknown, is NP hard (Tsotsos, 1990). And even when restricting the problem to a rectangular bounding box, the problem is still demanding: $O(n \cdot w \cdot h) = O(w^2 \cdot h^2) = O(n^2)$ subwindows have to be tested for their objectness, since at each pixel, subwindows of all possible sizes have to be tested. While a quadratic running time is per se not necessarily impractical, one has to consider that at each of these locations, the objectness measure has to be applied, which can be expensive. Especially, if each subwindow has to be sent to a server to be classified, as for example necessary for mobile devices such as Google Glass, investigating thousands of subwindows becomes quickly unfeasible.

While difficult for machines, detecting objects is effortlessly, even unconsciously, done by humans. Already infants that are only a few months old can reliably discover objects (von Hofsten and Spelke, 1985; Spelke, 1990). Thus, it is worth investigating how the human visual system achieves this task and whether we can improve vision systems by considering these concepts. While not yet completely understood, many findings from psychology and neurobiology describe the processes involved in object perception in the brain and we have outlined several of them in Chapter 2. A brief summary of the most important findings which are relevant for the work of this chapter are the following **object principles**:

O1 In the human brain, two different pathways process the visual data: the "what pathway" (ventral stream) that processes color and form and is responsible for object detection and recognition, and the "where pathway" (dorsal stream) that processes mainly motion and is responsible for spatially localizing objects (Ungerleider and Mishkin, 1982).

O2 Detection (discovery) of objects takes place before object recognition. This is an important aspect of the Situated Vision Theory that states that it is important that a visual system is able to individuate objects without previous knowledge of them (Pylyshyn, 2001).

O3 Segmentation processes bundle parts of the visual input based on Gestalt principles, such as similarity or proximity (Scholl, 2001). The results of the segmentation are called "proto-objects" (Rensink, 2000).

O4 Proto-objects are combined by focused attention to form coherent objects (Rensink, 2000).

O5 Visual attention mechanisms direct the processing to the regions of most potential interest, resulting in a sequential investigation of a scene by fixations and saccades (Pashler, 1997).

O6 Inhibition of return (IOR) mechanisms inhibit cells that correspond to previously fixated locations and objects, which supports orienting towards novelty and enables scene exploration (Posner and Cohen, 1984). IOR is encoded in spatial (not retinotopical) coordinates and can be environment-based as well as object-based (Tipper et al., 1994).

Our object discovery system utilizes these perceptual concepts in the following way:

1. Color and depth information are processed in parallel (if depth information is available) (corresponds to object principle **O1**).
2. Object principle **O2** is not explicitly modeled, but justifies the approach to deal with the object discovery problem before object recognition and use it as a preprocessing step for advanced perception modules. This is in contrast to the traditional approach in computer vision that applies classifiers to many subwindows of an image in a sliding window approach. It is especially important in a Situated Vision Framework in which the whole visual perception is situated in its environment (see García et al. (2013)).
3. A segmentation method clusters similar pixels to perceptually coherent regions (proto-objects), related to the grouping mechanisms in human perception (corresponds to **O3**).
4. An attention mechanism (saliency method) selects proto-objects to form object hypotheses (corresponds to **O4**).
5. The order in which to analyze a scene is determined by visual attention mechanisms that direct the processing to regions of most potential interest. This is done by switching between a fixate and a saccade behavior (corresponds to **O5**).
6. Inhibition-of-return mechanisms enable to remember already visited locations and facilitate the investigation of new areas. We encode the IOR data in a spatial 3D map and implement environment-based as well as object-based IOR (corresponds to **O6**).

Table 3.2 states which of the object principles has been addressed in which of our publications. We are currently working on a unified version that integrates all aspects in one coherent system.

	García and Frintrop (2013)[2]	García et al. (2013)[3]	Frintrop et al. (2014)[1]	Horbert et al. (2014)
O1	+	+	–	–
O2	+	+	+	+
O3	–	–	+	+
O4	–	–	+	+
O5	+	+	–	–
O6	+	+	–	–

Table 3.2: Relation of object principles O1 – O6 to our publications. '+' denotes a principle that was addressed in the corresponding publication, '–' means it was not addressed.

Related work on object discovery focuses on different aspects of the problem. Much of the work in computer vision tackles, for example, the problem of automatically discovering the category of a given, pre-segmented image by finding and clustering similarities in large datasets (overview in (Tuytelaars et al., 2010)). The task is different than in our work, since the result of the method is a separation of categories instead of the detection and segmentation of an unknown object. More related to our understanding of object discovery is the work of Alexe et al. (2012) and Manén et al. (2013) that captures the objectness of regions in single static web images. In contrast to their work, our approach does not only provide bounding boxes but pixel-precise segmented object boundaries, and we show that our method clearly outperforms their work in terms of precision and recall (see Section 3.2 and (Horbert et al., 2014)).

Recently, especially with the upcoming RGB-D sensors, several groups have investigated object discovery in 3D data. Karpathy et al. (2013) find objects on the 3D meshes obtained from RGB-D data. Johnson-Roberson et al. (2010) perform object segmentation on full point clouds. These method exploit the fact that objects mostly show a strong depth difference to their surrounding. Other approaches observe a scene over time and consider regions that change as object candidates (Herbst et al., 2011). In robotics, several approaches use interactive perception to verify their object hypotheses. That means they manipulate (push, poke, grasp, etc.) things in the real-world to figure out if an entity is a single object or consists of several objects (Katz et al., 2013).

Our approach differs from the above methods in that it is applicable to single static images and, with little extensions, also to videos and RGB-D data. It is independent of temporal and 3D data, although it can exploit the advantages of these elements if available. Our method is directly applicable to complex real-world scenes with a high degree of clutter, without requiring a previous training phase or any pre-knowledge about the objects of interest.

In the following, we introduce first our method for object discovery in 2D images (Section 3.2), which is for example useful to analyze web images (Frintrop et al., 2014), and an extension to 2D sequences, which adds a temporal component. This is, for example, relevant to find objects in video streams from devices such as Google Glass (Horbert et al., 2014). In Section 3.3, we extend the method to 3D data from an RGB-

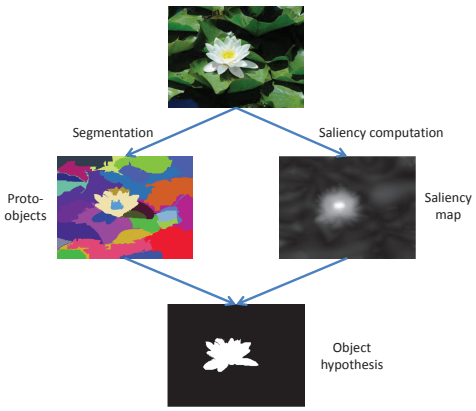


Figure 3.2: Simplified overview of the object discovery approach for web images: saliency (right) selects the segments (left) that compose an object hypothesis (bottom).

D sensor, treating depth and color information separately before finally fusing them to obtain 3D object models. And finally, we show in Section 3.4 how to detect objects in 3D sequences, which involves attentional scene exploration and spatial inhibition of return.

3.2 Computational Object Discovery in 2D images

This section describes our approach for object discovery in 2D images that was published in (Frintrop et al., 2014; Horbert et al., 2014). This is of interest for many methods that operate on web images or on photo collections, for example image thumbnailing (Marchesotti et al., 2009), retargeting (Goferman et al., 2011), or object classification (Liu et al., 2009).

Our method bases on the idea of proto-objects that originates from psychological research where it was introduced by Rensink (2000) (cf. Section 2.1). As mentioned there, proto-objects are object candidates, which correspond to visual structures that result from early segmentation processes. According to Rensink, attention then "acts as a hand to grasp proto-objects to form coherent objects" (cf. object principle **O4**). Following this idea, we find objects in a two step approach: first the image is segmented into perceptually coherent parts; second, a saliency map is computed and segments are selected depending on their saliency. The concept is visualized in Figure 3.2.

First, the input image is segmented using the approach of Felzenszwalb and Huttenlocher (2004). This is a graph-based segmentation method that is based on two important Gestalt principles: the similarity and proximity of pixels (cf. object principle **O3**). The method is a well established segmentation method in computer vision and creates, as the authors state, "perceptually important regions". The resulting segments, also called superpixels, are our perceptually coherent proto-objects (cf. Figure 3.2, left). For selecting the proto-objects that form an object hypothesis, attention comes into play. This is done by computing a saliency map that highlights regions of potential interest.

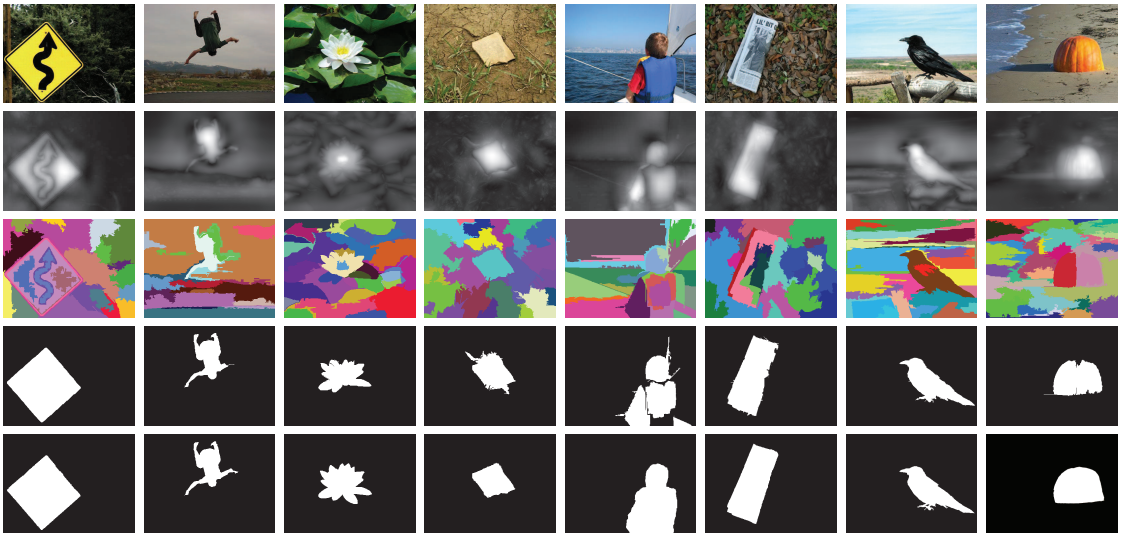


Figure 3.3: Several examples of our object discovery approach for web images. From top to bottom: original images (from MSRA database (Liu et al., 2009)), saliency maps, segmentations, salient segments, ground truth.

In principle, any saliency system can be used that produces precise saliency maps on an object level (in contrast to systems that simulate eye movements which produce much sparser saliency maps). We use the *simple CoDi* saliency system (Frintrop et al., 2014), since it is real-time capable and has shown to outperform many other saliency methods. Simple CoDi is an adaption of the CoDi Saliency system (Klein and Frintrop, 2012) and will be described in more detail in Chapter 4. Here, it is sufficient to know that the system computes saliency by measuring center-surround contrasts in different feature dimensions and on different scale levels and fuses them into a single saliency map. An example saliency map can be seen in Figure 3.2, right.

Selecting proto-objects based on saliency is then done by combining all segments in which at least $k\%$ of the pixels are above a saliency threshold. These selected segments form an object hypothesis. The concept is visualized in Figure 3.2. This simple method can directly be used to detect objects in internet images. We have evaluated the approach on images from the MSRA database of salient objects (Liu et al., 2009; Achanta et al., 2009) that was already introduced in Chapter 2.2. We have shown in (Frintrop et al., 2014) that the method produces precise object candidates and clearly outperforms the original CoDi system and 7 other saliency methods, of which some also include segmentation steps. Some examples of discovered object candidates are shown in Figure 3.3.

The MSRA database contains a comparably simple selection of images: each image contains only one object that is especially salient, the objects are rather large, often centered, and they usually do not intersect with the image borders. While this is a simplification, these properties can actually often be found in internet images: mostly, such images are taken by a human photographer, who has already solved part of the object detection problem: he or she has focused on the object and zoomed in, thus,

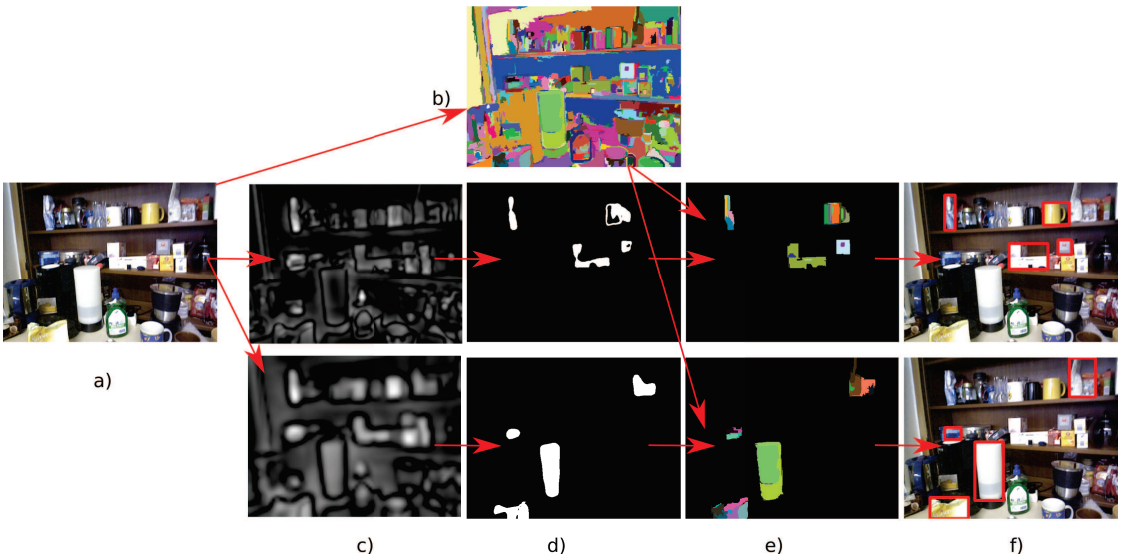


Figure 3.4: Object discovery in real-world images: a) original image; b) segmentation into proto-objects (superpixels); c) specific saliency maps for octaves 2 (middle row) and 3 (bottom row); d) some exemplary salient blobs obtained by region-growing, e) object candidates, obtained by combining proto-objects with help of the salient blobs, and e) bounding boxes of the object candidates. (Figure adapted from (Horbert et al., 2014))

many images show close-up views of objects, which are often centered (an effect known as the photographer bias (Tseng et al., 2009)).

In other application areas, such as interpreting data from an autonomous mobile robot or a mobile device, e.g., Google Glass, images are much more complex in content because they contain more objects and clutter. To enable the detection of several objects per image, simple thresholding is not enough. Instead, we have to determine which proto-objects belong to which object hypothesis. Therefore, we have extended our approach for object discovery as follows (see Figure 3.4).

We start with the same saliency computation as before but treat the pyramid layers (octaves) of the system independently, resulting in several octave-specific saliency maps, each favoring a specific object size (Horbert et al., 2014). To detect salient blobs in the saliency map, we first find local maxima within each octave-specific saliency map. After ranking the maxima by their saliency, seeded region growing (Adams and Bischof, 1994) is applied at each of the maxima, starting from the most salient one.[2] We repeat this process for different region-growing thresholds, where the threshold is set with respect to the value of the corresponding local maximum (Horbert et al., 2014). Finally, the overlap of each proto-object with these salient blobs is determined and all proto-objects that are covered by at least $k\%$ of a salient blob are chosen to belong to the current object candidate. Thus, each salient component results in an object proposal and the precise boundaries are obtained by the segmentation process.

[2]We used adaptive thresholding before (García and Frintrop, 2013; Frintrop et al., 2014), but the region growing produces less artifacts and improved our results considerably.

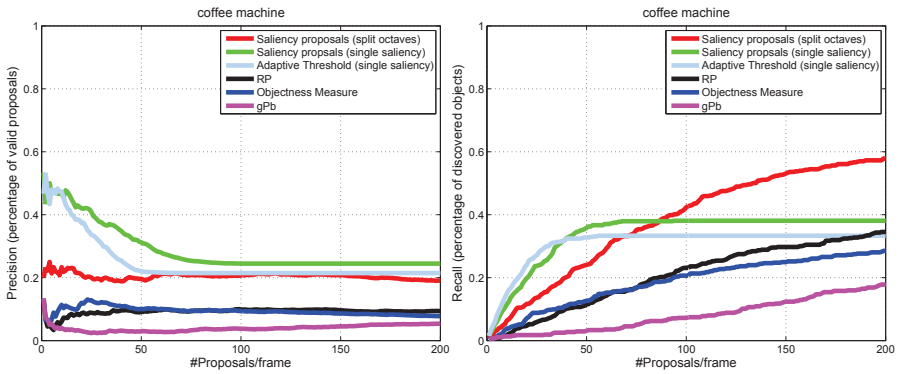


Figure 3.5: Comparison of three versions of our object discovery method (red, green, pale blue) with the objectness measure from Alexe et al. (2012), the Prime Object Proposals (RP) of Manén et al. (2013) and the contour detector (gPb) of Arbelaez et al. (2011). Left: the percentage of valid proposals per frame (precision); right: the percentage of discovered objects per frame (frame-recall). Performance is plotted depending on the number of object proposals that were considered (best N proposals per frame) (more results in (Horbert et al., 2014)).

In (Horbert et al., 2014), we show that our approach for object discovery clearly outperforms several recent methods that have shown good performance for object proposal detection and for which source code is available, namely the objectness measure of Alexe et al. (2012), the Prime Object Proposals of Manén et al. (2013) and the contour detector of Arbelaez et al. (2011). The experiments were performed on a new dataset that we provided for sequence-level object discovery and which consists of several sequences of indoor home environments. All sequences include a high degree of clutter and many (30-50) objects per frame.

Figure 3.5 shows the results for the coffee machine sequence that was also used in (García and Frintrop, 2013) and (Frintrop et al., 2014). Results for the other sequences can be found in (Horbert et al., 2014). It can be seen that our method outperforms the other approaches clearly in terms of precision (percentage of valid proposals) as well as recall (percentage of discovered objects). The pale blue curve represents our object discovery method with adaptive thresholding that was presented in (Frintrop et al., 2014), the green curve represents the replacement of adaptive thresholding by region growing, and the red curve treats the octave levels independently instead of computing a single saliency map. While region growing turned out to be always superior to adaptive thresholding, the advantage of the split octaves depends on the application. If only a few object candidates per frame are of interest, the single saliency map version achieves a higher precision and recall. However, it starts to saturate at about 50 proposals/frame, and for more than 90 proposals/frame the recall is considerably higher for the split-octave version. This is important for applications with many objects per frame in which it is desired to detect as many of the visible objects as possible. Especially the objects that are difficult to detect are retrieved rather with the split-octave version.

In (Horbert et al., 2014), we have extended this approach to video sequences and track the retrieved object candidates over time. This enables to group object candidates

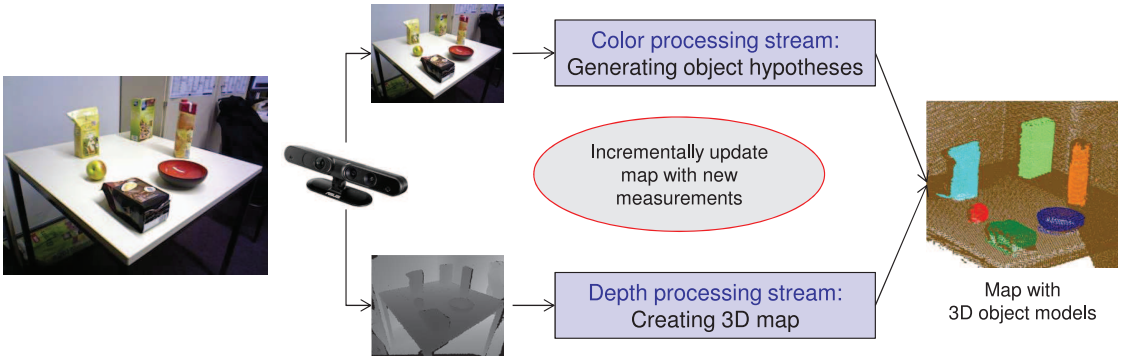


Figure 3.6: Sensor data is analyzed in two streams: a color stream processes the RGB image and generates object hypotheses and a depth stream processes the depth data of the sensor and produces a 3D map. Object hypotheses are then projected into the 3D map and data is incrementally improved over time when new measurements arrive.

that belong to the same object and to automatically filter out inconsistent regions. We have shown that this results in a significant reduction of the number of object candidates, while keeping a consistently high recall.

3.3 Object Discovery in 3D: Color and Depth Stream

While the previous sections dealt with 2D images, we extend the approach here to 3D data obtained with an RGB-D camera. Depth information provides important information for human perception as well as for machine vision. Especially object discovery profits from such data since objects often stand out not only visually but also spatially from their surrounding. Moreover, depth data facilitates to build a spatial map of the environment, which will show to be very useful in the next section. The content presented in this section was published in (García and Frintrop, 2013; García et al., 2013).

In our system, we obtain depth as well as color information from the ASUS Xtion PRO Live sensor RGB and Depth sensor. The sensor as well as example images are shown in Figure 3.6. This data separation enables directly a separated processing of color and depth data, similar as in the visual streams in the human brain (object principle **O1**). The color processing stream finds object hypotheses (corresponding roughly to the ventral pathway), while the depth processing stream builds a 3D scene map. The latter serves to locate objects in space, similar as in the dorsal pathway of the human visual system. Finally, both streams are fused by projecting proto-objects into the 3D map. An overview of the two-stream approach is shown in Figure 3.6.

The color processing stream follows the strategy to detect object proposals that was described in Section 3.2. The computations in the depth stream are based on the KinectFusion algorithm (Newcombe et al., 2011), which builds a 3D map of the environment by integrating multiple range scans from a moving depth sensor. The result is a 3D scene map consisting of voxels. To fuse color and depth stream, the 2D object hypotheses obtained from the color stream are projected into this 3D map (details in (García and Frintrop, 2013; García et al., 2013)).

While this approach can be used to detect objects in static 3D scenes, its real advantage reveals when applying the method to temporal data. In this case, different views of an object can be subsequently integrated into the same 3D model. Thus, the obtained models improve over time and contain at each moment all the available object information. The next section covers this extension.

3.4 Scene Exploration in 3D: The Saccade-Fixate Cycle

While we have investigated static scenes up to now, we extend the approach here to data sequences (see García and Frintrop (2013); García et al. (2013)). Since a system that analyzes such sequences usually should operate in real-time, it is even more important in such a setting to prioritize the processing. As in human vision, we use an attention-guided exploration behavior that simulates the human strategy to analyze a scene: attention guides the processing to the region of most potential interest, this region is fixated and analyzed, and, finally, the attended region is inhibited and the attentional beam switches to the next region of interest (cf. object principle **O5**). Thus, the scene is analyzed sequentially.

To simulate this attentional scene exploration, our system operates in two behaviors: the *saccade* behavior and the *fixate* behavior. When the system starts, it first finds the most salient object hypothesis, which is then attended for several frames (fixate behavior), allowing other modules to analyze the attended region and project it to the 3D scene. After fixating an object for a while, the saccade behavior takes over to determine the next focus of attention.[3]

When exploring a scene over time with help of an attention system, one problem occurs: how do we make sure that we do not stick to the most salient object, but switch attention from one hypothesis to the next? In computational attention systems this aspect is usually solved by inhibition of return (IOR) mechanisms (cf. Section 2) that withdraw attention from fixated regions and orient towards novelty (object principle **O6**).

In traditional computational attention systems (Itti et al., 1998; Frintrop, 2006), IOR is usually realized by inhibiting values in the saliency map (practically, they are often simply set to zero) However, this results in problems when the scene is dynamic and camera and/or objects move. Then, suddenly, the attended object does not cover the same region in the image anymore and the inhibited region does not correspond anymore to the object region. Backer et al. have addressed this problem by tracking all previously attended regions and inhibiting their new position in the saliency map (Backer et al., 2001). However, this method relies on the quality of the tracking and adds computational effort. Instead, we follow an idea from human perception. So, let us have a closer look at how human vision deals with this problem.

Human vision faces exactly the same problem when maintaining coherent positions of already attended objects over time. Most areas in the human visual system are organized retinotopically, that means, the neighboring cells in the retina correspond to neighboring

[3]Note, that in our case the saccades do not correspond to real camera movements, but to virtual shifts of the processing focus within the scene. It can however equally well be extended to active camera movements (cf. our visual SLAM approach in Chapter 6).

cells in other visual areas, e.g., LGN, V1, V2, etc. Since the input of retinotopically organized maps changes with each eye movement, the inhibition of cells would result exactly in the same problem as described above: as soon as the eyes move, the inhibited cells do not correspond to the previous object anymore. But, Posner and Cohen (1984) found that not the retinal location of a cue is inhibited, but the location of the cue in the environment. This is a very interesting finding, because it indicates that IOR is encoded in a spatial coordinate frame, not in a retinotopic one.

We follow this idea and encode inhibition information not as previous approaches in the saliency map (image coordinates, corresponding to retinotopic coordinates), but in the spatial map, namely our 3D map (cf. Figure 3.7). Thus, each voxel stores the information when it was attended last and whether it should be still inhibited at the current time. This is done with help of two variables: $I_t[v]$ is a binary flag that denotes whether voxel v should be inhibited at time t, and $IW_t[v]$ is a weight that determines the duration of inhibition. Each time a voxel is observed, weight $IW_t[v]$ is increased and as soon as a threshold is reached, the flag $I_t[v]$ is activated. On the other hand, the weight of not attended voxels is decreased continuously and as soon as it reaches 0, the flag of the voxel is set inactive. As in human vision, we distinguish environment-based and object-based IOR (Tipper et al., 1994) (cf. Section 2.1.2) which means that the spatial region surrounding the object can be a source of the inhibition as well as the object itself. In our system, this is implemented by inhibiting the object region stronger than its neighborhood. More details can be found in (García and Frintrop, 2013; García et al., 2013).

The values of the inhibition flags of each voxel can be raycasted at every time to a 2D inhibition map (cf. middle of Figure 3.7). Note that the 3D map integrates measurements over time, thus the 2D inhibition map contains inhibition information from frames 1 to $t-1$. The inhibition map can be directly used to determine which of the salient regions shall be fixated next. This is done by computing the overlap between all salient components in the object candidate map and the values from the inhibition map, and selecting the component with the highest value $sal \cdot (1-o)$, where sal is the average saliency of the component and o is the overlap.

In (García and Frintrop, 2013; García et al., 2013), we have shown that our system is able to find many objects, even in cluttered real-world scenes, and that the detection precision[4] is mostly very high (more than 90% for 17 out of 25 objects). An example can be seen in Figure 3.8. In this quite complex scene, 19 object proposals have been discovered after 438 frames (13 sec.). However, it can also be seen that many objects are still missed. More objects could be found by observing the scene longer. Note also that since the publication of García and Frintrop (2013), we improved the 2D object discovery considerably (Frintrop et al., 2014; Horbert et al., 2014) and we expect more and better object candidates when integrating the improvements into the 3D framework.

To our knowledge, the here presented approach is the first computational system that encodes IOR information in spatial coordinates and that thus enables a spatial attention-based scene exploration for detecting unknown objects and creating 3D models of these objects.

[4]Note that in contrast to Section 3.2, precision is here defined as the percentage of voxels that are correctly assigned to their corresponding ground truth object.

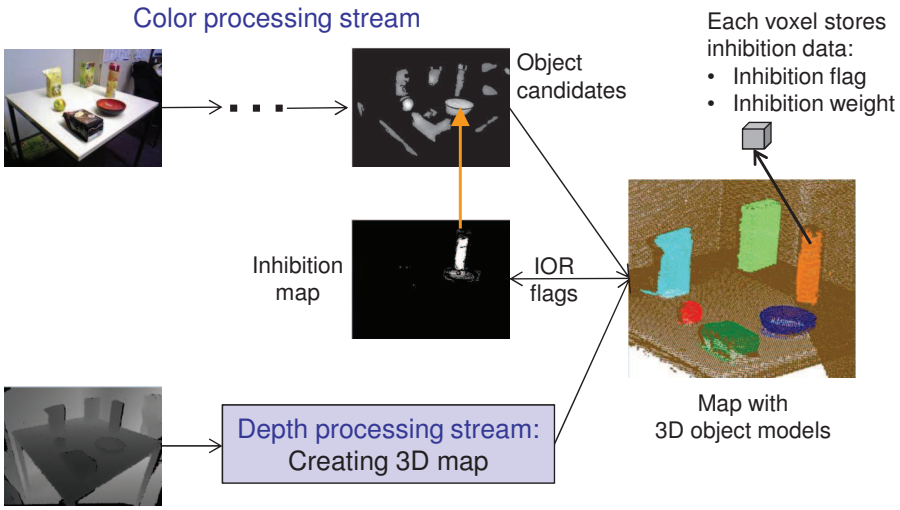


Figure 3.7: Inhibition of return in our object discovery system: the inhibition flags that are stored in the voxels of the 3D map (right) are raycasted to a 2D inhibition map. This map is used to inhibit values in the object candidate map and enables to fixate candidate objects that have not recently been attended (see text for details).

Figure 3.8: Discovered objects in one of our sequences. Left: original scene. Right: 3D map with 19 discovered objects after 13 sec. The rectangles show the automatically obtained 2D object candidates from the color processing stream.

3.5 Conclusion

In this section, we have presented our work on object discovery. We have shown that the same approach is applicable to static web images, to video sequences, and to RGB-D data. Our method delivers pixel-precise object proposals which is an advantage over bounding-box approaches, especially when projecting the candidate objects into a 3D map. The 3D discovery approach that roots the inhibition data in spatial coordinates is to our knowledge the first attention system that operates directly on 3D data.

Our object discovery method exploits concepts from human perception and combines saliency and segmentation. In contrast to many other approaches, we are independent of temporal data and 3D data (although both can be exploited if available). We also do not require a database of training images with similar objects, nor any interaction with the objects. The approach achieves good results in all settings and outperforms state-of-the-art methods for object discovery.

The current method can be extended in many ways. We are currently working on using Gestalt principles that can serve as further consistency checks to select the most promising proposals (Horbert et al., 2014). Interesting is also to integrate top-down knowledge to guide the attention to regions of current interest, for example searching for a specific object or concentrating on regions of the environment where an object is typically located (e.g., cups on horizontal surfaces such as tables). Finally, a natural extension of the 3D attentional framework is to include active camera control and thus move from a system for covert attention to one for overt attention. Actively focusing on objects of interest enables to obtain better viewpoints of objects and to zoom in to obtain images with higher resolution.

Chapter 4

Distribution-based Saliency

Visual saliency is the property of an image region to automatically attract human attention. That means, humans attend to such regions although their content is not necessarily relevant for the intentions or goals of the person. Regions that attract attention differ usually strongly from their neighborhood, for example a white toy on a red sofa, the waving hand of a person in a crowd, or an empty chair in an otherwise filled classroom. The ability to perceive saliency is one of the main components of the human visual attention system and it is also of large interest for computer vision systems. Applications range from analyzing web images and large photo collections (Marchesotti et al., 2009; Grundmann et al., 2010) up to driver assistance systems (Michalke et al., 2008) and service robotics (Schillaci et al., 2012; Frintrop, 2011b).

In this chapter, we summarize our work on saliency computation based on feature distributions that capture the statistics of features in a center and a surround area. Our approach combines the general structure of psychological attention models with a sound mathematical foundation and additionally enables an efficient computational implementation. We show that the system is able to outperform 8 state-of-the-art saliency methods in terms of precision and recall. Most of the work in this chapter was performed together with Dominik A. Klein within the DFG project "Saliency-based image matching for mobile systems". The publications that form the basis of this chapter are (Klein and Frintrop, 2011; Frintrop et al., 2014).

The chapter is structured as follows. After introducing the topic of saliency and its cognitive foundations in more detail (Section 4.1), we explain the concept of distribution-based saliency in Section 4.2. Then, we introduce our saliency system BITS (Bonn Information-Theoretic Saliency) that integrates these findings into a system structure based on psychological attention models (Section 4.3). In Section 4.4, we present an extension of this model from the discrete to the continuous case, the CoDi-Saliency system, and an adaption which is faster and more precise: the "Simple CoDi" system. At the end of this chapter, we compare the distribution-based saliency measure with the traditional Difference of Gaussians approach (Section 4.5).

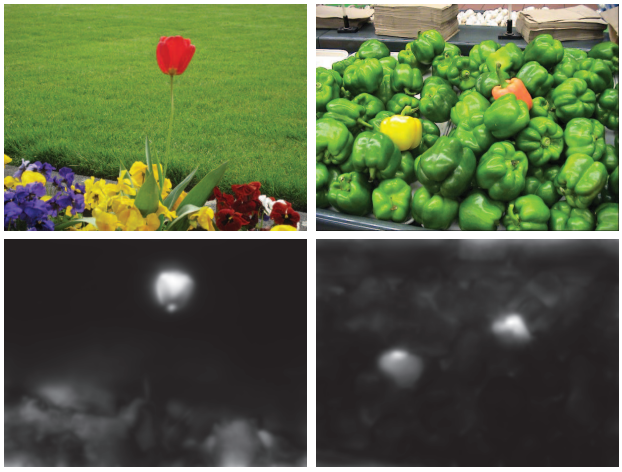


Figure 4.1: Two images and the corresponding saliency maps computed with our system BITS (Klein and Frintrop, 2011) (image sources: own image and Bruce dataset (Bruce and Tsotsos, 2009)).

4.1 Saliency – An Overview

According to a common definition, a salient region "stand[s] out relative to its neighbors" (cf. Wikipedia: "Salience (neuroscience)", Jan. 2014). This means, saliency is a relative property that relates an item to its background, not a property of an object itself. An object is never salient per se, it is only salient in a specific context. While a traffic sign is usually salient (it was designed to attract attention), it is not salient among other traffic signs. Thus, a saliency method always has to relate the properties of an image region to its surrounding. The output of a saliency method is a *saliency map*, which is a grayscale image in which the brightness of a pixel indicates its saliency. Two images with salient regions and corresponding saliency maps are shown in Figure 4.1.

During the last decade, many new saliency methods have been proposed and the number of approaches is strongly increasing every year (see surveys (Frintrop et al., 2010b) and (Borji and Itti, 2012b)). There are approaches that are based on the spectral analysis of images (Hou et al., 2012; Schauerte and Stiefelhagen, 2012), models that base on Bayesian theory (Itti and Baldi, 2009; Zhang et al., 2008), or on decision theory (Gao and Vasconcelos, 2007; Gao et al., 2009), and those that use machine learning techniques (Liu et al., 2009; Alexe et al., 2010). Because of the overwhelming number of different approaches, it is hard to keep an overview and to see the differences, and, more importantly, the similarities of the methods.

To find the roots of saliency, let us briefly review the most important findings on saliency and attention from psychology and neuroscience that were introduced in Chapter 2. We will call them **attention principles** in the following:

A1 Visual attention directs the processing in the brain to the regions of most potential interest (Pashler, 1997). This enables to deal with the large complexity of the

sensory input and to concentrate on relevant data, which is essential for scene understanding (Carrasco, 2011).

A2 Attentional processes consist of *bottom-up* (data-driven) and *top-down* (model-driven) aspects (Connor et al., 2004). Saliency is part of bottom-up attention and denotes the quality of a region to attract attention automatically.

A3 The most important element of saliency detection is the ability to detect *center surround contrast*. In the brain, such contrasts are detected by cells with a receptive field that has an excitatorily center and an inhibitorily surround (*On-Off cells*) or vice versa (*Off-On cells*). Examples are retinal ganglion cells or simple cells in V1 (Kuffler, 1953; Hubel and Wiesel, 1959).

A4 Cells with concentric receptive fields are modeled best with a two-dimensional *Difference-of-Gaussian (DoG)* function (Rodieck, 1965), while elongated fields are modeled best with *Gabor filters* (Jones and Palmer, 1987).

A5 On-Off and Off-On cell types are organized into three channels: a luminance, red-green, and blue-yellow channel (color opponency) (Gegenfurtner, 2003).

A6 Features are processed in parallel in different brain areas (Livingstone and Hubel, 1987) and compete for selective attention (Treisman and Gelade, 1980).

A7 Basic features that guide visual attention are color, orientation, motion, and size. Other features (e.g., intensity) are discussed in the literature, but there is less evidence for them (Wolfe and Horowitz, 2004).

A8 A *master map of location* or *saliency map* collects the conspicuities of the different feature channels. There is evidence that such a saliency map exists in V1 (Zhang et al., 2012).

Note that the object principle **O6** from the previous chapter, which describes the inhibition of return behavior, is also related to visual attention and could be another attention principle. However, since we tackled this case already in chapter 3, we omit it here.

Many psychological attention models have been created based on these findings, such as the Feature Integration theory (Treisman and Gelade, 1980) or the Guided search model (Wolfe, 1994). In these models, the main idea was to separate feature computations into different feature channels, such as color or orientation, and finally fuse the conspicuities of the channels into a single saliency map (attention principles **A6, A7, A8**).

Traditional computational attention systems followed these findings closely (Milanese et al., 1994; Itti et al., 1998; Frintrop, 2006), but also *most current saliency systems still inherit the basic structure*[1] which is visualized in Figure 4.2 (number (1) and (4)). As shown in this figure, additional components of most saliency systems are a scale

[1]Some systems restrict processing to a single feature such as color (Achanta et al., 2009) or motion (Vijayakumar et al., 2001), but this does not capture all types of saliencies and is only sufficient for specific applications.

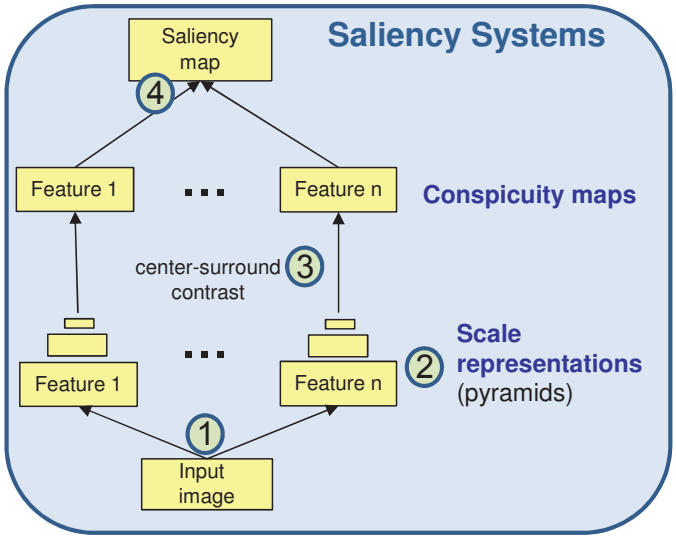


Figure 4.2: General structure of visual saliency systems (part of Figure 2.3). Basic elements: separation of feature channels (1), scale representation (2), center-surround contrast (3), fusion of feature conspicuities to a saliency map (4).

representation (2) to enable the detection of differently sized saliencies (addressing the size feature of attention principle **A7**) and a center-surround method (3) that captures the difference between image regions and their neighborhood (attention principle **A3**). In Table 4.1, we outline explicitly, which of the attention principles have been realized in which of the saliency systems BITS, CoDi, and Simple CoDi that we present in this chapter. For completeness, we include as well our previously presented attention system VOCUS.

As we have outlined in Chapter 2, the center-surround method is probably the most essential element of a saliency method, since a high contrast in some feature dimension is an intrinsic property of a salient item: by definition it "stands out relative to its neighbors". Basically all saliency methods compute such a value (although not always in a center-surround manner), and the most important difference between methods is the way this contrast is computed.

Cognitive models compute the center-surround contrast usually by Difference-of-Gaussian (DoG) or Gabor filters (Itti et al., 1998; Frintrop, 2006), since these are known to model best the concentric and elongated cells of the human visual system (attention principle **A4**). Also other approaches as the Bayesian surprise model (Itti and Baldi, 2009) or the decision-theoretic model of Gao and Vasconcelos (2007) use DoG and Gabor filters to compute contrasts. Some approaches compute the contrast not based on pixels but on patches (Sun et al., 2012; Borji and Itti, 2012a) or on previously segmented regions, e.g., superpixels (Perazzi et al., 2012; Zhu et al., 2013). Instead of computing local contrasts, some approaches compute global contrasts by considering the whole image as surrounding region, e.g., Achanta et al. (2009) or Bruce and Tsotsos (2009). Note however that while global contrasts are quicker to compute, they are not able to capture local saliencies that are important in human perception (cf. example in Figure 4.3). The

	VOCUS (Frintrop, 2006)	BITS (Klein and Frintrop, 2011)	CoDi (Klein and Frintrop, 2012)	Simple CoDi (Frintrop et al., 2014)
A1	+	+	+	+
A2	bu and td	bu	bu	bu
A3	+	+	+	+
A4	+	–	–	+
A5	+	–	+	+
A6	+	+	+	+
A7	i, c, o, m, (s)	i, c, o, (s)	i, c, (s)	i, c,(s)
A8	+	+	+	+

Table 4.1: Relation of the attention principles A1 – A8 to our saliency systems. For completeness, we include also our previously published system VOCUS. '+' denotes a principle that was addressed in the corresponding publication, '–' means it was not addressed. 'bu' and 'td' stand for 'bottom-up' and 'top-down' part implemented. For A7, the letters denote the features that are implemented: 'i' = intensity, 'c' = color, 'o' = orientation, 'm' = motion, 's' = size; the 's' are in parentheses since size is considered in all systems by a scale representation, but a size–pop-out is not implemented.

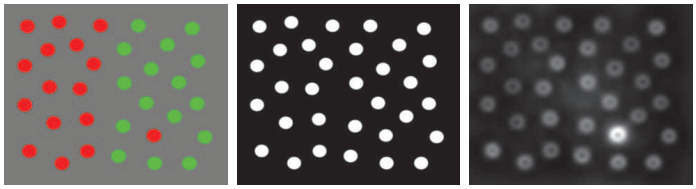


Figure 4.3: Global versus local saliency on an example of an item which is only locally salient (red item among green ones). Saliency maps from a method that computes only global contrast (Achanta et al., 2009) and from our BITS system that computes local contrasts (Klein and Frintrop, 2011).

contrast computation can be also extended to the spatial domain by computing depth contrasts (Maki et al., 2000; Björkman and Eklundh, 2007) or to the temporal domain, where it computes the change of the visual data over time (Itti and Baldi, 2009).

In this chapter, we propose a different way to compute the center-surround contrast. The idea here is to represent center and surround regions by feature distributions and measure the contrast by a comparison of the distributions. This captures more information about the corresponding image regions than the typical DoG method. Similar approaches have been presented in (Itti and Baldi, 2009) and (Bruce and Tsotsos, 2009), but while these approaches are computationally very expensive, we present a solution that is real-time capable and that performs very well in state-of-the-art benchmarks. We will discuss differences to these models in more detail in Section 4.3.

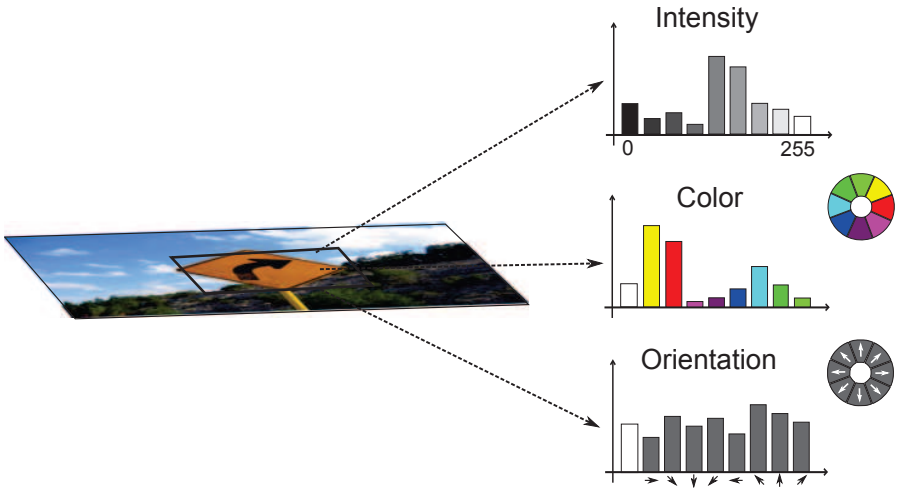


Figure 4.4: Three different types of discrete feature distributions: intensity, color, and gradient orientations. The circles next to the color and orientation distribution shall indicate that the feature values are obtained in a circular, 2D space, e.g., in case of color from the HS-plane in the HSV color space.

4.2 Distribution-Based Saliency

In this section, we introduce the foundations of distribution-based saliency. We start by describing how to represent image regions by feature distributions, before we show how these can be used to measure information content and information-based contrast in images.

4.2.1 Feature Distributions for Representing Image Regions

As mentioned before, the traditional way to measure saliency computes feature contrasts by Difference-of-Gaussian or Gabor filters (attention principle **A4**). These compute in principle a weighted average of the center and the surround region and subtract one from the other. Instead, we suggest to represent these image regions by feature distributions since these keep more information about the statistics of features than a simple average. A distribution captures a certain property of the image, e.g., the distribution of intensity or color values. Arbitrary properties can be regarded and represented by a distribution. Thus, we represent an image region R by k probability distributions R_k. The distributions can be based either directly on pixel values or on pre-processed features, for example on gradient orientations. Some examples of feature distributions are shown in Figure 4.4.

Since the distributions originate from pixel values, they are discrete. However, working directly on the original distributions with, e.g., 255 intensity values, makes a method sensitive to noise and is computationally expensive. Therefore, distributions are usually approximated by histograms with fewer bins. We will follow this strategy for our BITS saliency system that will be introduced in Section 4.3. An alternative is to approxi-

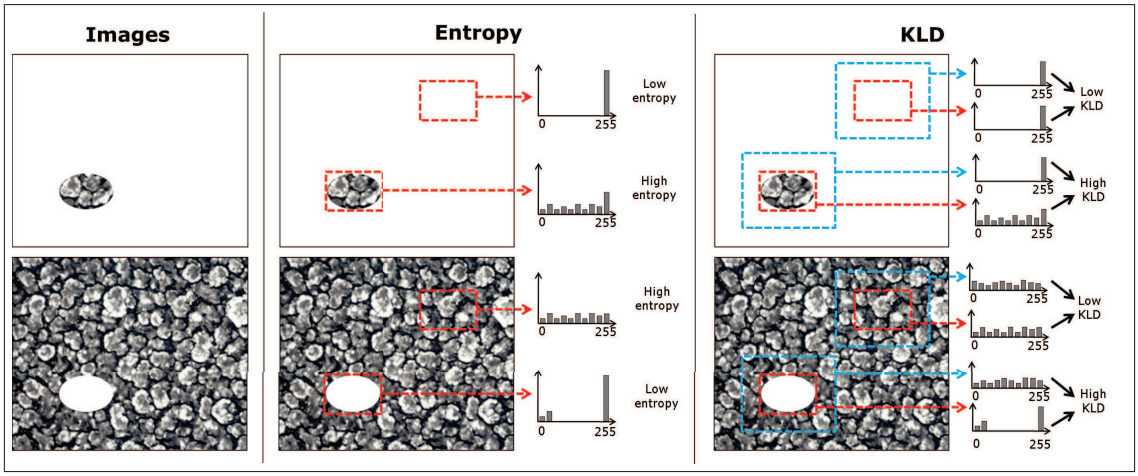


Figure 4.5: Entropy versus information gain (KLD). In the upper example, both entropy and KLD assign the highest value to the salient ellipse. However, in the example below, entropy assigns high values to the structured background, while the KLD captures the saliency of the bright ellipse by considering the surround.

mate the feature distribution by normal distributions, as in the CoDi saliency system (Section 4.4).

4.2.2 Entropy versus Information Gain

The classic way to measure information content in signals is the Shannon entropy. For a discrete feature distribution R_k of an image region R, it is defined as:

$$H(R_k) = -\sum_{i=1}^{b} R_k(i) \log(R_k(i)), \tag{4.1}$$

where b is the number of bins of the histogram that represents R_k. Since homogeneous image regions have a peaked histogram they have a low entropy, whereas cluttered image regions with an equally distributed histogram have a high entropy (cf. Figure 4.5). Some people have suggested using entropy as saliency measure (Kadir and Brady, 2001; Kadir et al., 2004), and it is indeed in many cases a useful measure. However, this is only the case if the salient image region is structured, whereas the background is homogeneous, as in the example in the top row of Figure 4.5. If on the other hand the image region itself is homogeneous and the background cluttered, the region is salient especially because of the absence of structure (bottom row of Figure 4.5).

Thus, we propose instead to compute saliency not directly by the entropy of the feature distribution of a region, but by the difference of entropy of an image region with respect to its surround. This difference can be computed by the *Kullback-Leibler Divergence (KLD)*, which is also called *information gain* or *relative entropy*. Given two feature distributions C_k and S_k that represent a center and a surround region in an image, KLD is defined as

Figure 4.6: The advantage of distribution-based saliency: since the average intensity of the background and the gray disk is exactly the same, the traditional Difference-of-Gaussians approach results in a black saliency map (middle, computed with the VOCUS system (Frintrop, 2006)). The distribution-based BITS system is able to capture the difference (right).

$$D_{KL}(C_k \| S_k) = \sum_{i=1}^{b} C_k(i) \log \frac{C_k(i)}{S_k(i)}. \tag{4.2}$$

Again, each distribution is represented as a histograms with b bins. Figure 4.5, right, visualizes the region of C_k as red and the region of S_k as blue rectangle. The figure shows that, in contrast to entropy, KLD is able to capture saliency of both types: structured regions on homogeneous backgrounds (top), but also the opposite case (bottom). Figure 4.6 shows the advantage of KLD-saliency over the traditional DoG approach: while the DoG method cannot detect any saliencies, since the gray patch and the checkerboard background have the same average intensity, the KLD captures the difference due to the different distributions of the intensity values.

Thus, by computing the Kullback-Leibler Divergence for center and surround distributions centered at each pixel, we obtain a measure for saliency. While the KLD on distributions of image properties is the core of the information-theoretic saliency computation, it has to be integrated into a structure that enables real-time computations and competitive performance on real-world data. In the following, we will describe two approaches for this, first a discrete approach using histograms, second a continuous approach using normal distributions.

4.3 BITS: Saliency Based on Information Gain

In (Klein and Frintrop, 2011), we have presented the BITS (Bonn Information-Theoretic Saliency) system, an information-theoretic approach to compute saliency based on the Kullback-Leibler divergence between histograms. This mathematically sound way to compute the center-surround contrast is integrated into the basic structure of saliency systems based on psychological findings. This allows a consistent computation of saliency for different feature channels as well as a well-founded fusion of feature channels.

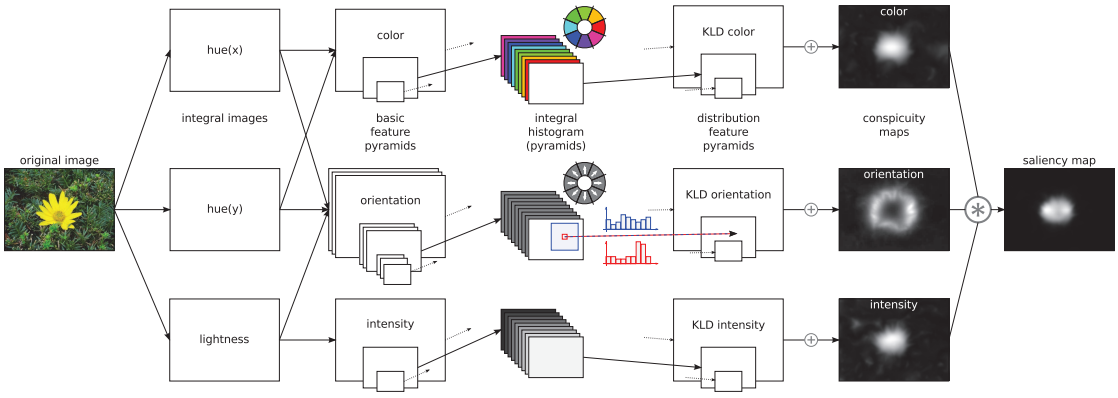


Figure 4.7: Overview of our saliency system BITS (Figure from (Klein and Frintrop, 2011)).

In a saliency system with different feature channels and many scales, many histograms of large image regions have to be computed and compared. This becomes quickly too demanding for real-time computations. On the other hand, approximations, such as computing global instead of local contrasts or restricting computations to one scale (Bruce and Tsotsos, 2009), affect the precision of the system.

To obtain a real-time system while maintaining high precision computations, we first use scalable features based on integral images (Viola and Jones, 2004), that allow an efficient computation for arbitrarily large, rectangular image regions. Second, we use integral histograms to represent the distributions, which is an extension of integral images to histograms (Porikli, 2005). The idea is to represent each bin by an integral image and to accumulate information for each bin separately. Since integral images allow to compute the average of an image region of arbitrary size in constant time, also integral histograms of such image regions can be computed in constant time. In our approach, we compute integral histograms for each of the features intensity, color, and orientation.

To integrate the concept of information-theoretic saliency into a complete model of saliency computation, the KLD measure replaces the center-surround measure that is based on Difference-of-Gaussians in traditional models. That means, the KL differences are computed on different scales for three different feature channels, intensity, color, and orientation (attention principles **A6** and **A7**), and, finally, the scales and feature channels are fused to a saliency map (attention principle **A8**). An overview of the BITS system is shown in Figure 4.7.

Related to our work is the model of surprise of Itti and Baldi (2009) that measures temporal and spatial surprise in a Bayesian probability framework. In contrast to our work, they first compute feature maps with the traditional saliency model approach by computing the center-surround contrast with across-scale differences (which approximates DoG filters) and then base the surprise measure on these values. Instead, we compute the feature maps directly in an information-theoretic way.

Bruce and Tsotsos (2009) have computed saliency by the self-information of image regions with respect to their surround. In contrast to our work, they base their feature detection on ICA coefficients that are learned from a large collection of images. Due to

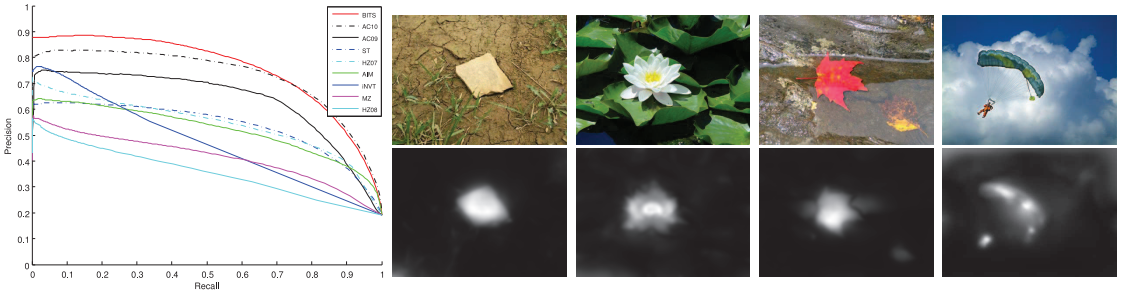


Figure 4.8: Results of the BITS saliency system on images from the MSRA dataset (Liu et al., 2009). Left: precision-recall curves for the methods iNVT (Itti et al., 1998), ST (Walther and Koch, 2006), HZ07 (Hou and Zhang, 2007), HZ08 (Hou and Zhang, 2008), AIM (Bruce and Tsotsos, 2009), MZ (Ma and Zhang, 2003), AC09 (Achanta et al., 2009) and AC10 (Achanta and Süsstrunk, 2010). Right: some example images (top) and the corresponding saliency maps (bottom) (Figures from (Klein and Frintrop, 2011)).

the computational complexity, they use a global surround from the whole image and only a single scale. Instead, our scalable feature detectors and the use of integral histograms enables us to compute features on several scales in a computationally feasible way, to use local instead of global surrounds, and makes us independent of a training set.

The performance of the BITS saliency system is shown in Figure 4.8. A comparison with 8 state-of-the-art saliency methods showed that the BITS system outperformed all other methods in terms of precision and recall (left). Some example images with corresponding saliency maps computed by BITS are shown in Figure 4.8, right. The performance of our BITS system is close to real-time (about 0.5 sec on a 320×240 pixel image, on a 2.66 GHz quad-core PC with double precision computation) and could be easily sped-up by code optimizations and implementation on a GPU.

4.4 Extensions: CoDi and Simple CoDi

In follow-up work, we have extended the idea of computing saliency based on probability distributions to the continuous case, resulting in the CoDi-Saliency system (**Co**ntinuous **Di**stributions) (Klein and Frintrop, 2012). While the CoDi paper itself is not part of this cumulative habilitation thesis, we briefly explain the key ideas here for completeness and for comprehensibility of the Simple CoDi saliency system (Frintrop et al., 2014) that contains extensions of CoDi and will be described at the end of this section.

Instead of using histograms, in CoDi the feature distributions are approximated by normal distributions (cf. Figure 4.9). These distributions are computed by maximum-likelihood estimates of the center or the surround region, weighted by a Gaussian integration window. The normal distributions of center and surround can be compared with the Kullback-Leibler divergence, as in the BITS system, or with other methods, for example the W_2-distance (Wasserstein metric based on the Euclidean norm) as in (Klein and Frintrop, 2012). The W_2-metric has the advantage that it treats the problem as a transportation of mass problem which considers the distance of feature values in feature

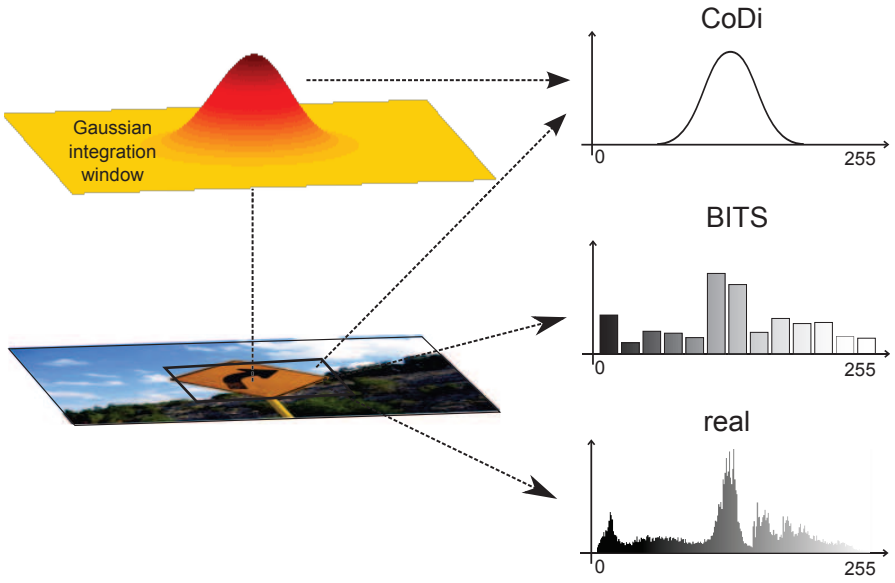


Figure 4.9: Difference between the BITS and CoDi saliency systems shown exemplarily for intensity distributions: while the BITS system represents the real feature distribution (bottom) by a histogram (middle), CoDi approximates it by a normal distribution (top). In CoDi, the intensity values are additionally weighted by values from a Gaussian integration window (top left).

space. This is relevant for saliency computation since here, the similarity of features plays an important role, and in our experiments we obtained better performance with the W_2 metric than with KLD.

Similar as in BITS and other saliency systems, this center-surround concept is embedded into a scale-space structure to enable the detection of objects of different sizes. The computations are performed for intensity and color features. The color distributions are computed in an opponent-color space with one red-green and one blue-yellow axis, corresponding to the color cells in the human visual systems (attention principle **A5**).

In (Klein and Frintrop, 2012), it was shown that the CoDi saliency was able to outperform all nine competitors on the MSRA database (including BITS and the methods used in (Klein and Frintrop, 2011)). Furthermore, with 82 ms per image (400×300 pixels, Intel Core i7-2600), it is real-time capable and, since it does not use priors such as "objects are central and do not intersect with the image borders", it is applicable not only to web images but also to data from a moving camera.

In (Frintrop et al., 2014), we have presented some improvements of the CoDi system. Since the new version is faster and easier to implement, we call this version "Simple CoDi". The improvements are, first, an adaption of the center-surround filter sizes to a ratio that corresponds better to human perception, and, second, an exchange of the Difference-of-Gaussians pyramid to a Gaussian pyramid (details in (Frintrop et al., 2014)). Additionally, we have changed the distance measure from W_2 to the simpler Manhattan distance. While W_2 measures the distance of two distributions based on their means μ and standard deviations σ, the Manhattan distance uses only the mean

values μ. Although this computation uses less information about the image regions, it achieves the same performance in terms of precision and recall with less computational effort (Frintrop et al., 2014). Apparently, the variance of the distributions does not have a large effect in practice. Interestingly, this change transformed the system back to a Difference-of-Gaussians approach as in the traditional saliency systems (more details in next section). Since this similarity allows an interesting direct comparison of distribution-based and traditional, biologically inspired saliency, we will explain this aspect in more detail in the following section.

4.5 Distribution-based versus DoG-based Saliency: A Comparison

In this section, we will address some of the similarities and differences of the distribution-based saliency computation to the traditional, biologically-inspired saliency based on Difference-of-Gaussians. Generally, comparing different saliency methods is difficult, because each group proposes a complex saliency system that is composed of many modules and parameters and is usually highly optimized for a specific task or benchmark. Methods use different color spaces, different numbers of features, and different filter sizes. Comparing whole saliency systems generally says little about a specific method. However, as mentioned in the previous section, the CoDi-Saliency system enables a direct comparison of distribution-based saliency to the biologically-inspired DoG approach within the same framework. The reason is the following:

A Difference-of-Gaussian filter D is simply a digital filter, obtained by subtracting two Gaussian filters G_1 and G_2 with different variances: $D = G_1 - G_2$. Because of the linearity of convolution, convolving an image with a DoG filter is the same as applying two Gaussian filters with different variance to the image separately and subtracting the resulting images:

$$O = F * D = F * [G_1 - G_2] = F * G_1 - F * G_2, \tag{4.3}$$

for output image O and input image F (in our case a feature map). In other words, each pixel in a DoG filtered image results from subtracting two weighted mean values obtained from windows of different sizes:

$$\begin{aligned} O(x,y) \quad = \quad & \left[\sum_{i=-k_1}^{k_1} \sum_{j=-k_1}^{k_1} w_1(x-i, y-j) F(x-i, y-j) \right] - \\ & \left[\sum_{i=-k_2}^{k_2} \sum_{j=-k_2}^{k_2} w_2(x-i, y-j) F(x-i, y-j) \right], \end{aligned} \tag{4.4}$$

where w_1 and w_2 are the weights of the Gaussians G_1 and G_2 with Gaussian integration windows of size $k_1 \times k_1$ and $k_2 \times k_2$ respectively.

In CoDi, the center-surround difference is computed as the W_2 distance between a center distribution C and a surround distribution S. Both distributions are given as

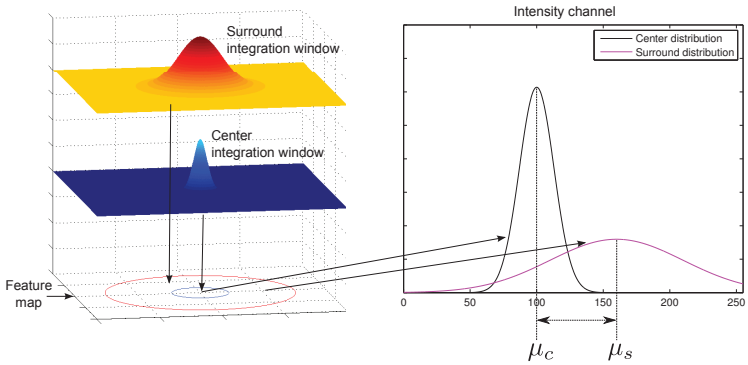


Figure 4.10: Computation of feature distributions in the CoDi saliency system and correspondence to Difference-of-Gaussians contrast. Left bottom: a feature map with center (blue) and surround (red) area, and above the corresponding Gaussian integration windows. Right: corresponding intensity distributions as computed by CoDi. Mean values correspond to the weighted means of the values in the corresponding feature map region. Thus, the difference of mean values corresponds to a Difference-of-Gaussians contrast.

normal distributions that are one dimensional for intensity, $N(\mu, \sigma^2)$, and two dimensional for the color channel, based on a red-green and a blue-yellow axis, $N(\mu, \mathbf{\Sigma})$. In the following, we concentrate for simplicity on the one dimensional case. It is not important here how the W_2 distance is computed, just that it is based on the two parameters μ and σ. In CoDi, these values are obtained by weighting the feature values by a Gaussian integration window that is centered at the current center or surround region (cf. Figure 4.9 and 4.10). Thus, the mean of a normal distribution of a region in feature map F centered at pixel position (x,y), is estimated as

$$\hat{\mu}(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(x-i, y-j) F(x-i, y-j), \tag{4.5}$$

for a Gaussian integration window w of size $k \times k$, centered at (x,y). Now, it can be easily seen that subtracting the two estimated mean values of the center and the surround distribution $\hat{\mu}_c(x,y) - \hat{\mu}_s(x,y)$, using Gaussian integration windows w_c and w_s, results directly in the DoG equation 4.4. This idea is visualized in Figure 4.10. While the contrast computation is straight forward for the intensity channel, we have two dimensional distributions with two dimensional mean vectors for the color feature channel. In this case, we can either compute the Euclidean distance of mean values, or the simpler Manhattan distance. In the first case, the dimensions of the color distribution are treated jointly, in the second case independently. In this case, the contrast computation based on the Manhattan distance corresponds to computing the Difference-of-Gaussian contrast independently on two feature maps for the two color dimensions and adding the resulting contrast maps.

Figure 4.11: Saliency maps for different distance measures of center and surround distributions. From left to right: Original image, W2-distance, Euclidean distance, Manhattan distance (DoG contrast).

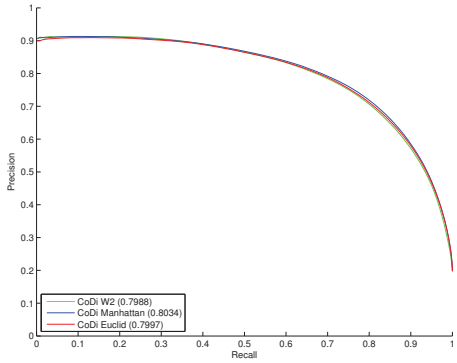


Figure 4.12: Evaluation of saliency computation with different distance measures (W_2-distance, Euclidean distance, Manhattan distance (DoG contrast)) for computing the center-surround contrast. The numbers in parentheses denote the AUC values. The simple Manhattan distance, that corresponds to DoG computation in traditional saliency systems, achieves the same performance as the distribution-based W_2 metric.

Of course, computing DoG contrast within the CoDi system is unnecessarily complicated compared to the direct way to compute it. However, it enables us to compare the DoG and the distribution-based contrast measures in exactly the same framework so that we can make sure that a change in performance really results from the distance measure and not from other design issues.

We have compared the saliency computation for three different center-surround measures: (i) the W_2-distance as in the original CoDi paper (Klein and Frintrop, 2012), (ii) the Euclidean distance of mean values, and (iii) the Manhattan distance of mean values, which corresponds to the DoG contrast. Some example images of saliency maps computed with the different distance measures are shown in Figure 4.11. It can be seen that the maps that compute the contrast only based on the μ values (Euclidean and Manhattan distance) are cleaner and less blurry. A quantitative evaluation on the MSRA dataset (Liu et al., 2009) is shown in Figure 4.12.

Interestingly, W_2, Euclidean, and Manhattan distance achieved almost the same results. Thus, although the distribution-based saliency operates on a richer feature representation based on mean *and* variance of feature distributions, the additional variance information seems to have little effect in practice. This shows that the traditional Difference-of-Gaussians method can, with well-chosen filter sizes, still achieve state-of-the art performance when computing saliency.

4.6 Conclusion

In this chapter, we have introduced our work on distribution-based saliency, implemented in the saliency systems BITS and CoDi. We have shown that the systems obtain state-of-the-art performance on the common MSRA dataset in a real-time capable framework. We have also shown exemplarily in Figure 4.6 that using distributions instead of traditional Difference-of-Gaussian filters captures certain saliencies that can not be detected with traditional saliency systems.

However, in our final experiments it has turned out that in practice, the variance of the feature distribution has little effect: the distribution-based CoDi and the DoG approach in Simple CoDi (implemented as Manhattan distance) achieve about the same values in terms of precision and recall. Of course, the implementation within CoDi is only reasonable for comparison purposes, otherwise it can be directly implemented in the traditional way as described in (Itti et al., 1998) and Frintrop (2011a).

In my opinion, the most interesting outcome of these experiments is that the traditional saliency model that was originally proposed by Itti and colleagues more than 15 years ago (Itti et al., 1998) can, if it is cleanly implemented and the parameters are optimized for the task, still achieve state-of-the art performance on current benchmarks. Because of its applicability for saliency detection as well as for simulating eye movements, and because of its clean and comprehensive structure that can easily be implemented in a real-time framework, it might be still favorable in practice over many newer methods.

On the other hand, the probability distributions that were computed in the saliency systems BITS and CoDi can be exploited also in other settings, e.g., for representing and comparing superpixels, as we proposed in a superpixel-based saliency approach in (Zhu et al., 2013), or for finding and matching keypoints as in (Klein and Cremers, 2013). In the latter approach, the normal distributions of CoDi were used both to detect keypoints and to build a feature descriptor. This work enables an interesting extension that we consider for future work: if saliency operator, detector, and descriptor base on the same concepts of normal distributions, it is possible to discover, describe, and re-detect objects within the same framework, based on the same feature computations and difference measures.

Chapter 5

Attentive Visual Tracking

Visual tracking is the task to estimate the state of an object in an image sequence, where the state can include position, extent, velocities, or other properties of the target. During tracking, either the object or the camera (or both) move, resulting in location changes of the target object in the image plane. While the tracking task is usually effortlessly solved by humans, the task can become very challenging for machines. Reasons include noise in the data, partial or full occlusions of the target, abrupt motion of object or camera, and strong variations in object appearance for example due to illumination changes, viewpoint changes, or deformations of the target.

Many good tracking approaches have been proposed in the past. However, the applicability depends strongly on the task and the setting. In our work, we are interested in an approach that is able to operate on a mobile vision system, e.g., a robot or a head-mounted camera, such as Google Glass. Thus, the system shall be able to run in real-time and to deal with background changes, varying illuminations, etc. Furthermore, we are interested in tracking arbitrary, previously unknown objects. That means, the appearance of the object has to be learned online from one or a few frames.

Among the most crucial parts of a visual tracker is the representation of the target object. If the target of interest is known in advance and a model can be learned, tracking can even be performed by recognizing the desired target in each frame. If the target is not known in advance, feature-based approaches are often used that represent target objects by features such as color or gradient distributions. For feature-based approaches, it is favorable to detect discriminative features that distinguish the target well from the background.

An ideal candidate to determine discriminative features is a saliency system. It can determine the most salient parts of an object that best distinguish it from its surrounding. Focusing tracking on these parts facilitates the task and results in more stable position estimates. We have developed two tracking approaches. First, we present in Section 5.1 the *most salient region tracking* which is based on the idea of visual search: target features, obtained from a visual attention system, are boosted in a top-down way to find the target. Second, we present in Section 5.2 a new *component-based tracking approach*. The method determines for each target object a flexible number of salient components, each representing a discriminative part of the object with respect to a certain feature channel. The resulting components form a template that is utilized for the observation

model of a particle-filter-based tracking scheme. In Section 5.3 we use the component-based tracker for person tracking on a mobile robot.

Both tracking approaches assume that we have an initial estimate of the target, given by a bounding box. In our experiments, we initialized the system manually in the first frame, but the region can as well be provided by a module of a larger system, e.g., by an object detection module as the one presented in Chapter 3.

The observation models of the trackers in this chapter are based on the visual attention system VOCUS (Frintrop, 2006). In principle, any attention system can be used. However, for the work in Section 5.1, a top-down component is required that is able to extract a feature descriptor from the feature maps and use it in a top-down manner to excite target-relevant features. For the work in Section 5.2, any bottom-up attention or saliency method can be used that computes several feature maps, which highlight different visual aspects of the target.

Finally, it should be said that the here presented work concentrates on the representation of the target and the observation model of the tracker. It does not address the problem of adapting the target representation if the target appearance or the background appearance change over time. This has been addressed in our subsequent work on adaptive object tracking (Klein et al., 2010) that uses a Boosting approach to learn and adapt target appearance over time and the extensions of this work (Klein and Cremers, 2011; García et al., 2012). These ideas could also be used to adapt the cognitive target model that is presented here.

5.1 Most Salient Region Tracking

Although the most salient region tracker (MSR tracker) and its related paper (Frintrop and Kessel, 2009) is not part of this habilitation thesis, it will be briefly described here since it was the predecessor of the component-based tracker and some ideas base on this work.

The idea of the MSR tracker is simple and builds on a visual attention system with a top-down mode for visual search. The appearance of the target is learned quickly from the initialization frame, according to the learning mode described in (Frintrop, 2006). Then, the resulting target descriptor is used to perform visual target search by exciting target-relevant features and inhibiting target-irrelevant features according to the visual search strategy proposed in (Frintrop, 2006). The concept is visualized in Figure 5.1, details of the method can be found in (Frintrop and Kessel, 2009).

Advantages of the method over other appearance-based tracking methods are that it first concentrates on especially discriminative regions. Second, it combines the output of several feature channels, so that the system can use the features that fit best for a specific target. And finally, feature contrasts rather than absolute feature values are considered, resulting in a higher invariance to illumination changes. We have shown in (Frintrop and Kessel, 2009), that the method outperforms other feature-based tracking approaches, such as the Camshift tracker based on color histograms (Bradski, 1998).

The MSR tracker focuses tracking always on the most salient region of an object. In the following section, we extend this idea to detect and track multiple salient regions per object.

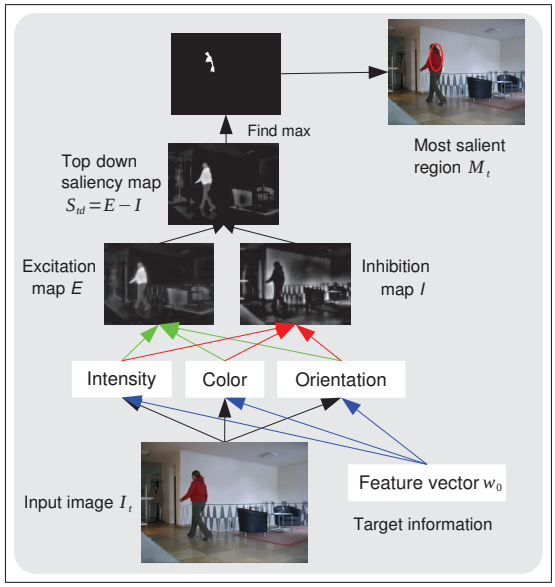


Figure 5.1: Visual search component of the MSR tracker: the feature maps for intensity, color and orientation are weighted to obtain an excitation map from target-relevant features and an inhibition map from target-irrelevant ones. Target-specific weights are obtained from the feature vector $\vec{w}_0$ that has been learned in advance. A top-down saliency map S_{td} is computed which highlights the target-specific regions of interest and the most salient region M_t determines the position of the object (Figure from (Frintrop and Kessel, 2009)).

5.2 Multi-Component Tracking

While the tracking approach of the previous section focuses on the most salient part of the target, the work in (Frintrop, 2010) extends this idea to represent a target by multiple salient components. This is especially favorable for complex objects that consist of several parts that are visually distinctive. The component-based template that represents the target is flexible in the sense that the number and position of components per target is not fixed but depends on the target and is determined automatically during runtime. This distinguishes our approach from previous work (Pérez et al., 2002, 2004; Adam et al., 2006; Beuter et al., 2009) that also represented targets by different parts, however, with rigid layouts. To the best of our knowledge, our work was the first to represent target objects for tracking by automatically determining discriminative parts of a target in a flexible and object-dependent way.

The multi-component tracking learns the target appearance from an initialization frame. The target appearance is represented by a template that consists of several components. Each component corresponds to a peak within the target region in one of the feature maps from a visual attention or saliency system. The six feature maps F_i that we obtain from the attention system VOCUS are two intensity maps for bright-dark and dark-bright contrasts, as well as four color maps for red-green, green-red, blue-yellow,

Figure 5.2: An example of a target object and the corresponding parts in the feature maps F_i. From left to right: original image, intensity bright-dark map, intensity dark-bright map, red-green contrast map, green-red contrast map, blue-yellow contrast map, yellow-blue contrast map. Each local maximum in these maps corresponds to one component in the template.

and yellow-blue contrasts (cf. Figure 5.2) and the components are obtained by detecting local maxima in these maps and segmenting a region surrounding each maximum. Each component is then approximated by the smallest surrounding rectangle. We call the resulting components $m_{i,j}$, where i denotes the feature and j the number of the peak.

The positions of the regions $m_{i,j}$ are stored relative to the center of the target region $\vec{R^*}$ and represent a template $\vec{M_{R^*}} = \{m_{i,j} | i \in \{1,..,6\}, j \in \{1,..,l_i\}\}$, where l_i is the number of components detected in feature map F_i (cf. Figure 5.3, left). Next, we derive a descriptor vector from the components $m_{i,j}$ by computing the ratio of the mean saliency value within $m_{i,j}$ and the mean value of the background. The saliency values are obtained from the corresponding feature maps of the attention system. Thus, the target descriptor that we obtain is $\vec{d^*} = \{\rho_{i,j} | i \in \{1,..,6\}, j \in \{1,..,l_i\}\}$.

In order to match the target descriptor $\vec{d^*}$ to an image region $\vec{R'}$ of arbitrary size and dimensions, the template is first adapted in size to the estimated size of the target in the current frame. In our setup, the size estimation is obtained from the particle estimates of the visual tracker. The adapted template is then used to compute a descriptor vector $\vec{d'}$ for the current image region and the two descriptors $\vec{d^*}$ and $\vec{d'}$ are matched by computing the similarity of the vectors with the Tanimoto coefficient (cf. Figure 5.3).

The component-based template is integrated into the observation model of a particle-filter-based tracker (Isard and Blake, 1998). The approach maintains a set of weighted samples (particles) over time using a recursive procedure based on the following three steps: first, the system draws particles randomly from the particle set of the previous time step, where each particle is drawn with a probability proportional to the associated weight of the particle. Second, the particles are transformed (predicted) according to a motion model. Finally, all particles are assigned new weights according to an observation model and the object state is estimated.

The most crucial step of a visual tracker is the observation model since it is responsible for which particles will survive. It therefore has the strongest influence on the estimated position of the target. In our approach, the set of particles is defined as $\mathbf{\Phi}_t = \{\phi_t^1, ...\phi_t^J\}$ with

$$\phi_t^j = (\vec{s}_t^j, \pi_t^j, \vec{d}_t^j), \quad j \in \{1, ..., J\}. \tag{5.1}$$

Here, $\vec{s}_t^j = (x, y, v_x, v_y, w, h)$ is the state vector that specifies the particle's region with center (x, y), width w, and height h; v_x and v_y specify the current velocity of the particle; π_t^j is a weight that determines the relevance of the particle with respect to the target,

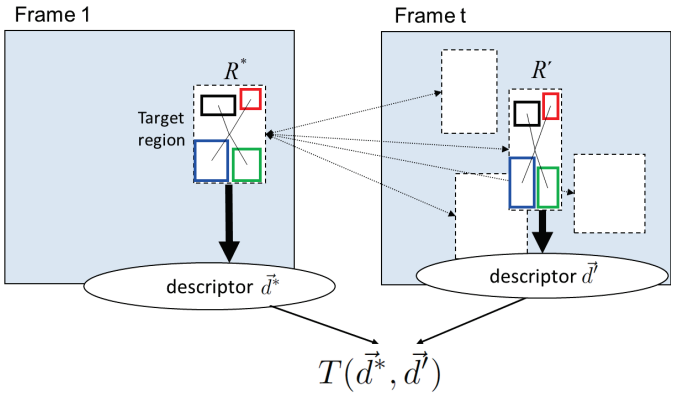


Figure 5.3: Multi-component tracking: in the first frame (left), a template is computed that represents the target in region R^*; it consists of several components, visualized here by the colored rectangles (the color indicates the corresponding feature map). The components are stored relatively to the center of the region and a descriptor $\vec{d}^*$ is computed for the target. In subsequent frames t (right), the template is matched to each particle region (dashed rectangles). It is first adapted in size, then, a descriptor $\vec{d}'$ is computed, and finally, $\vec{d}'$ is matched to the target descriptor $\vec{d}^*$ by the Tanimoto coefficient T. The similarity determines the weight of the particle.

and the component-based descriptor $\vec{d}_t^j$ describes the appearance of the particle region. The weight π_t^j is set according to the similarity of the current particle region to the target vector and is computed by

$$\pi_t^j = c \cdot e^{\lambda \cdot T(\vec{d}^*, \vec{d}_t^j)}, \tag{5.2}$$

where T is the Tanimoto coefficient, $\lambda = 14$ prioritizes particles very similar to the target and c is a normalization factor.

Finally, the current state of the object can be estimated as a weighted average of the particles by

$$\vec{x}_t = \sum_{j=1}^{J} \pi_t^j \cdot \vec{s}_t^j. \tag{5.3}$$

We have evaluated the multi-component tracking on several video sequences in real-world indoor and outdoor settings. Some examples of these sequences are shown in Figure 5.4. The method was compared to other feature-based tracking methods, such as the probabilistic tracking based on color histograms from Pérez et al. (2002), and different challenges were tested, such as illumination changes, scale changes, fast object motion, or temporal object occlusion. It has shown that the component-based tracker outperformed the other methods considerably, with a detection rate of 81% and an average error of the target position of 22 pixels, compared to a detection rate of 56% and a target position error of 41 pixels for the histogram tracking. The computation time was 80 ms per frame on a 2.5 GHz dual core PC with non-optimized code.

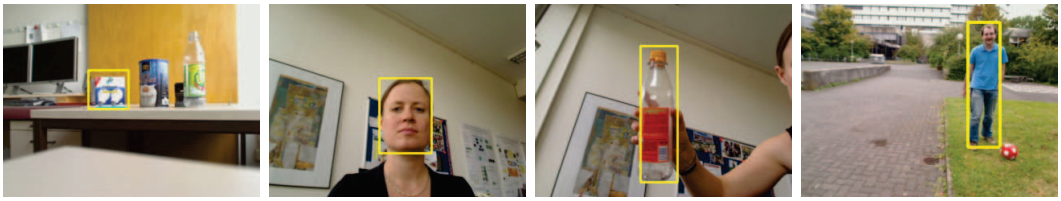


Figure 5.4: Some example frames from our tracking experiments. The yellow rectangle denotes the target object. See also the video on http://ivs.informatik.uni-bonn.de/research/tracking/

5.3 Person Tracking on a Mobile Robot

Visual tracking plays an important role in many areas, for example in service robotics. Here, especially the tracking of individual humans is of interest to distinguish the client from other people. While model-based vision approaches, trained to detect the shape of people, are well suited to detect people in general, an approach that is able to distinguish individuals has to capture the individual properties of humans, such as clothing, hair, or skin color.

In cooperation with the group of Dr. Dirk Schulz from the Fraunhofer institute FKIE, we have proposed such an individual person tracker (Frintrop et al., 2010a, 2009), based on the multi-component tracker from (Frintrop, 2010). Depending on the appearance of the person (clothing, hair color, skin color, hat, backpack, etc.), the system determines a flexible number of components, each representing a discriminative part with respect to a certain feature channel from a saliency system. Since the system is feature-based and learns the appearance of the person online, it is also able to cope with people wearing backpacks or carrying large objects.

We ran several experiments on the RWI B21 robot *Blücher*, equipped with a USB web camera mounted on a pan-tilt unit (Figure 5.5, left). Our component-based tracker was integrated into the tracking module of the robot. The module uses the position estimate of the tracker to compute a heading direction relative to the robot and steers the pan-tilt unit in order to center the person and commands the robot to follow the person. Experiments were carried out within the robot experimentation hall and the hallways at the Fraunhofer institute FKIE during normal working hours, with people walking around.

We have compared the component-based tracking method with three other color-based trackers and shown that the component-based tracker performed best, with an average detection rate of 90%, whereas the other methods ranged between 33% (CamShift) and 77% (simplified version of our method without components). In Figure 5.5 we display some of the tracking results.

Figure 5.5: Left: the RWI B21 robot *Blücher*. The images were taken using the small pan-tilt mounted webcam on top of the robot. Middle and right: tracking results. points: particles. The rectangles show the estimated target state.

5.4 Conclusion

In this chapter, we have summarized our work on visual object and person tracking. We have introduced a new approach for tracking based on a component-based descriptor. The method grabs the appearance of an object or a person together with a rough spatial layout which is quickly learned from a single training image. It can deal with different objects and settings, works in real-time, and is applicable on a moving platform. We have shown that, on average, it clearly outperforms other methods.

Interesting for future work is to adapt the target descriptor automatically to new backgrounds and new object appearances. We have introduced in (Klein et al., 2010) an adaptive method for tracking objects in video data. It is based on a classifier-based approach that trains weak classifiers on features which are boosted to select and combine the most discriminative ones into a strong classifier. In (García et al., 2012), this method was extended to color and depth data from an RGB-D sensor. While this work operated on simple Gradient features, it would be an interesting extension to use instead the saliency-based components which were presented in this chapter as a basis for boosting.

Chapter 6

Attentive Robot Localization and Mapping

A common and widely investigated problem in robotics is SLAM, which stands for *Simultaneous Localization And Mapping*. SLAM is the task of an autonomous system to automatically build a map of an unknown environment based on sensor data, while localizing itself within it. This is of interest not only for mobile robots, but also for systems that do not navigate autonomously such as cars or hand-held cameras.

The SLAM problem is a "chicken-and-egg problem": the robot needs a map to localize itself while on the other hand it requires an accurate pose estimate to build this map. The solution is to successively build the map while permanently using new sensor data to update the map. The process can be compared with a human that explores an unknown area, for example a new city. While walking through the streets, she/he obtains successively a clearer picture of the city, especially of the arrangement of streets and their connections. Especially when the streets are narrow and winding this can be difficult, and one might be surprised when coming to a previously seen location, not to be where one expected. Based on this new information, the internal picture of the world is updated and corrected. The same is done on a robot. The key idea for this update is that the information about the robot pose and the information about all sensor observations (e.g., landmarks) are correlated. If the position of a single observation is corrected due to better measurements, this can influence the complete map data, that means all other observations as well as the robot position itself.

During the SLAM process, the computations take place in two steps: first, the robot moves, which increases the pose uncertainty of robot and landmarks. Then, the robot processes its new sensor data, which decreases the uncertainty. The largest correction of uncertainty, and therefore the most useful one, takes place during so called *loop closing* situations. When the robot comes back to a region that it had already visited previously, it sees the same observations again and can correct its estimates accordingly. A precondition is that the robot recognizes the position and that the measurements belong to the same observations as previously seen ones. This step is not trivial and belongs to the biggest challenges in (visual) SLAM.

Traditionally, robots use range sensors such as laser range finders to create a map, and SLAM based on such sensors has reached a rather mature level. Range sensors are

especially well suited for map building and localization, since they offer exact information about the distance of obstacles and the layout of buildings. On the other hand, laser scanners are expensive, heavy, and require much energy. Therefore, other approaches aim to solve the SLAM problem with cameras as sensors. This is especially of interest for small robots that cannot carry heavy laser scanners. Camera-based SLAM is usually called *visual SLAM*. The main difference in visual SLAM is that first, images contain a huge amount of data, which poses a challenge for real-time processing, and second, the 3D position of image regions is not available instantaneously, but has to be estimated from stereo data or by structure-from-motion. Therefore, the standard approach in visual SLAM is to extract 2D *features* from the images (e.g., corners or blob-like regions) and to estimate their 3D position, resulting in so called *landmarks*. Recently, several groups have investigated SLAM based on data from RGB-D cameras (e.g. Engelhard et al. (2011)). This facilitates the 3D localization of landmarks, but the challenges of feature and landmark detection remain mostly the same.

A key competence in visual SLAM is to select landmarks of high quality to enable stable tracking and loop closing. These two tasks rely on different properties of landmarks. For tracking, it is especially important to reliably redetect features in subsequent frames. Matching of features between subsequent frames is usually easy since frames do not differ strongly. The computation should be fast but the descriptor matching does not have to be very powerful. The recently introduced ORB (Rublee et al., 2011) features are perfect candidates for such tasks. They are fast to compute and sufficient in their quality for tracking situations. Loop closing on the other hand requires more sophisticated matching capabilities. Landmarks on images viewed from different viewpoints and captured several minutes, hours, or even days later than the reference view, are much harder to match than landmarks between subsequent frames. In these cases, we are looking for landmarks that are easily redetected and that have a high discriminability. Perfect candidates for such landmarks are salient regions that stick out of their surrounding. By definition, they have a high saliency, resulting in a high repeatability (Frintrop and Cremers, 2010; Frintrop, 2008) and making them easy to distinguish from their environment.

In this chapter, we summarize our work on "attentive visual SLAM". Our visual SLAM system is based on salient landmarks, obtained from the visual attention system VOCUS. These landmarks are especially distinctive, which enables a stable loop-closing even with a sparse set of landmarks. Thus, although the computation of the salient features is slower than computation of features such as ORB, the system is efficient since dealing with a sparse set of landmarks pays off by strongly reduced matching and tracking efforts. Figure 6.1 gives an overview over the system. Most of the work in this chapter was performed together with Patric Jensfelt within the EU project NEUROBOTICS at KTH, Stockholm, in the group of Henrik Christensen. Some work has been done later in the group of Armin B. Cremers. The basis of this chapter are the publications (Frintrop and Cremers, 2010; Frintrop and Jensfelt, 2008b; Frintrop and Cremers, 2007); also related are (Frintrop et al., 2006b,a, 2007; Frintrop and Jensfelt, 2008a).

The contributions summarized in this chapter are first, a landmark selection scheme which allows a reliable pose estimation with a sparse set of especially discriminative landmarks, second, a precision-based loop-closing procedure based on SIFT descriptors,

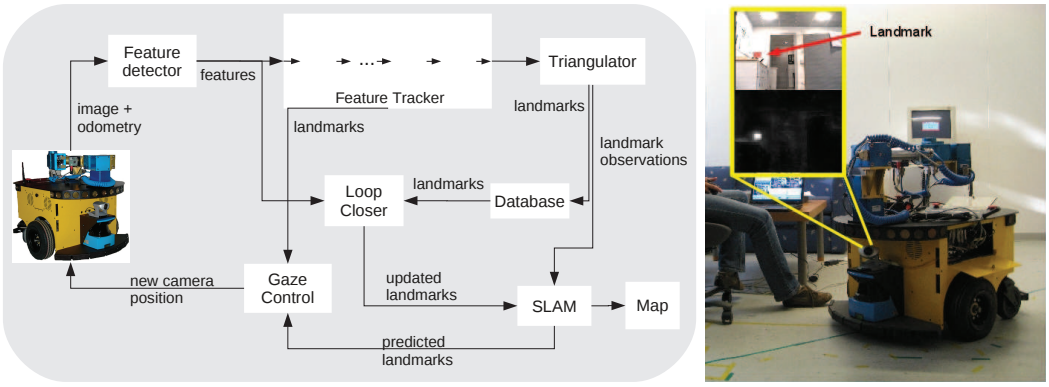


Figure 6.1: Attentive Visual SLAM: robot Dumbo builds a map of the environment and corrects its position estimates by detecting and tracking salient landmarks. Landmark detection is done with the attention system VOCUS. Left: overview of the architecture. Right: example view of the robot during operation (yellow rectangle); it shows an image with a landmark and the corresponding saliency map (Figures from (Frintrop and Jensfelt, 2008b) and http://www.iai.uni-bonn.de/~frintrop/research.html)

and, finally, an active gaze control strategy to obtain a better baseline for landmark estimations, a faster loop closing, and a more uniform distribution of landmarks in the environment. In the following, we briefly sketch the ideas of the attentive SLAM approach.

6.1 Salient Feature Detection and Landmark Selection

An ideal candidate for selecting a few, discriminative regions in an image is a visual attention system. In our attentive visual SLAM system, we detect salient landmarks with the visual attention system VOCUS that was presented in (Frintrop, 2006). VOCUS determines feature contrasts for intensity, orientation, and color features on 3 different scales with image pyramids. The peaks from the saliency map are extracted as *regions of interest (ROIs)*. We have shown that salient ROIs have a higher repeatability than standard detectors (Frintrop and Jensfelt, 2008b; Frintrop and Cremers, 2010; Frintrop, 2008), which makes them especially suited as landmark candidates. When using features with high repeatability, it is possible to reduce the overall number of features and deal with a sparse landmark set. This speeds up the tracking and matching procedures significantly.

In (Frintrop and Cremers, 2010), we have shown that it is even possible to create landmarks and to redetect them when re-visiting the same scene by considering only one (the most salient) feature per frame. This is in strong contrast to traditional approaches that consider hundreds or even thousands of features per frame. While this does not mean that using one feature is the optimal solution (it certainly is not, since it reduces the ability of a system to cope with occlusions), it shows nicely how focusing on salient

landmarks can reduce the processing time and memory requirements while keeping a high quality of landmark detection and matching.

While the extracted ROIs are regions in a 2D image, landmarks are parts of the 3D world. To obtain landmarks from ROIs, these are tracked over several frames to remove unstable observations and to obtain different viewpoints of a landmark. For tracking ROIs, a simple descriptor is sufficient since consecutive frames usually do not differ strongly. Instead of using standard descriptors that have to be computed additionally, we suggest a simple and effective solution that comes almost without cost: An attentional descriptor v can be obtained directly from the feature and conspicuity maps of the attention system. In other words, an entry v_i of the descriptor denotes the feature saliency of the ROI with respect to feature channel i (details in (Frintrop and Jensfelt, 2008b)). We matched two attentional descriptors $\vec{v}$ and $\vec{w}$ by calculating the similarity $d(\vec{v}, \vec{w})$ according to a distance measure that we introduced in (Frintrop et al., 2007).

After the ROI was successfully tracked over several frames, its observations are triangulated and the obtained 3D position is integrated as new landmark observation into the 3D map.

6.2 Landmark Redetection / Loop Closing

Loop Closing belongs to the essential capabilities of a SLAM system, because it enables the reduction of large position errors. In loop closing situations, a scene is re-visited from a different viewpoint. Thus, landmarks appear under strong transformations. Additionally the scene might have changed since visiting it the last time, the illumination can be different, and objects might have appeared or disappeared. To cope with such challenges, a powerful landmark matching is required.

The loop closing scheme that we proposed in (Frintrop and Jensfelt, 2008b) is especially suited for this purpose. It is based on the salient landmarks presented in the previous section, which enables to obtain a sparse, but discriminative landmark representation. For matching an observed salient region to a previously seen landmark, we use the SIFT descriptor (Lowe, 2004) that belongs to the most powerful and most frequently used image descriptors. Figure 6.2 shows some examples of correctly matched landmarks.

For matching landmarks, we introduced in (Frintrop and Jensfelt, 2008b) a new precision-based matching strategy that learns the dependence of ROI distances and matching precision from training data and enables to directly set a threshold for the desired matching precision. This is in contrast to the standard threshold-based matching that thresholds directly on the feature distances.[1] The precision-based matching has several advantages over the usual thresholding. First, it is possible to choose an intuitive threshold like "98% matching precision". Second, linear changes on the threshold result in linear changes on the matching precision which is not the case for thresholding distances. Finally, for every match a precision value is obtained. This value can be directly

[1] Alternative matching strategies are nearest neighbor and nearest neighbor distance ratio matching. Mikolajczyk and Schmid show that these strategies are more powerful than threshold-based matching, but also point out that they are difficult to apply when searching in large databases (Mikolajczyk and Schmid, 2005).

Figure 6.2: Some examples of correctly matched landmarks, displayed as rectangles. Top: current frame. Bottom: frame from the database.

used by other components of the system to treat a match according to the likelihood that it is correct. For example, a SLAM subsystem able to deal with more uncertain associations could use these values.

When a match is detected, the coordinates of the matched ROI in the current frame are provided to the SLAM system and used to update the coordinates of the corresponding landmark. We performed several experiments to validate the attentive visual SLAM system. One example can be seen in Figure 6.3, where the robot drove three loops in an office environment. This setting allows to investigate the loop closing behavior well since the same areas are re-visited several times. While the robot gets lost when relying only on its odometry (left), the SLAM system allows it to correct its errors (right). Note the sparse landmark representation on the right. While standard approaches for visual SLAM obtain hundreds or even thousands of features per frame, resulting in huge amounts of landmarks, we are able to maintain a correct robot position with very few landmarks. After three runs, the robot generated 17 landmarks, 10 of them were redetected when returning to the same area. Most of them were redetected several times, resulting in 21 matches over the whole sequence. More experiments and an extensive evaluation can be found in (Frintrop and Jensfelt, 2008b).

6.3 Active Gaze Control

Landmarks can only be detected and redetected if they are in the field of view of the robot's sensor. By actively controlling the viewing direction of the sensors it is possible to strongly improve the performance of a SLAM system. In (Frintrop and Jensfelt, 2008b) we presented a new approach for active gaze control of our attentive SLAM system that makes it possible first, to see landmarks for a longer time resulting in better landmark representations, second, to actively redetect landmarks to enable more frequent loop closings, and finally, to achieve a more uniform distribution of landmarks.

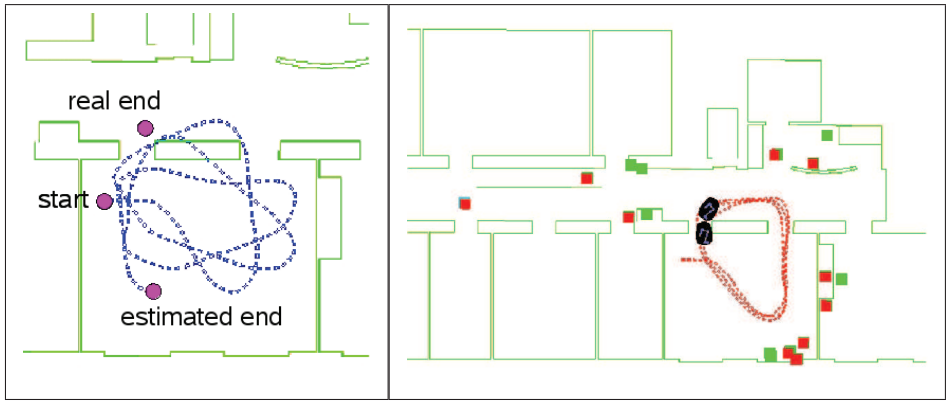


Figure 6.3: Attentive visual SLAM in an office environment: the robot trajectory was estimated once only from odometry (left) and once from the SLAM system (right). The walls (green lines) are only overlaid for visualization purposes and are not known by the robot. Right: Green dots are landmarks, red dots are landmarks which were redetected in loop-closing situations. While errors accumulate in odometry mode, resulting in a wrong pose estimate (left), the SLAM approach on the right allows the robot to correct these errors, based on a very sparse set of landmarks.

The active gaze control module controls the camera according to three behaviors:

- Redetection of landmarks to close loops
- Tracking of landmarks
- Exploration of unknown areas

The redetection mode uses information about the position of the landmarks and the robot to determine landmarks that are likely to be visible and directs the camera into their direction. This facilitates the detection of loop closure. The tracking behavior follows landmarks with the camera so that they stay longer within the field of view. The exploration behavior directs the camera to unseen areas to obtain a more uniform landmark distribution. Details about the behaviors can be found in (Frintrop and Jensfelt, 2008b).

The strategy to decide which behavior to choose is as follows: Redetection has the highest priority, but it is only chosen if there is an expected landmark in the possible field of view. If there is no expected landmark for redetection, the *tracking* behavior is activated. Tracking should only be performed if more landmarks are desired in this area. As soon as a certain amount of landmarks is obtained in the field of view, the *exploration* behavior is activated. In this behavior, the camera is moved to an area without landmarks. Most times, the system alternates between tracking and exploration; the redetection behavior is only activated every once in a while. Figure 6.4 shows two example runs of the robot, one obtained in passive, one in active camera mode. It is clearly visible that in this case, only the active mode allows loop closing and thus the

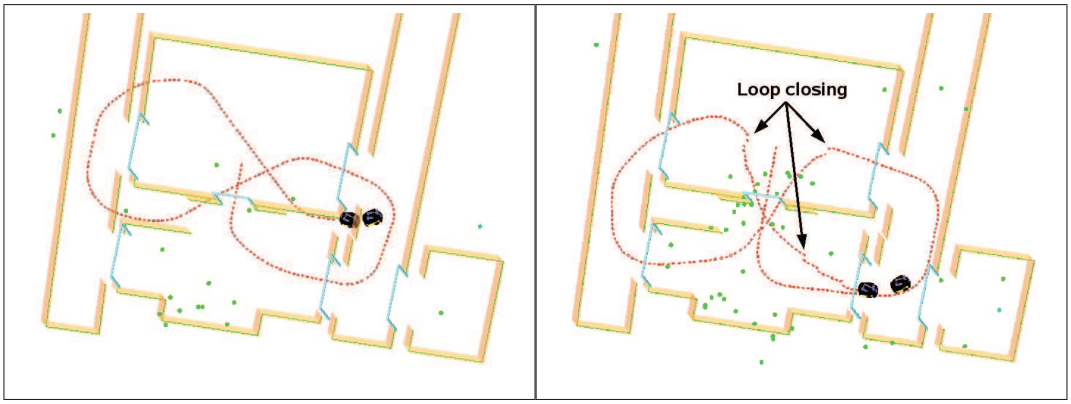


Figure 6.4: Atrium environment: the estimated robot trajectory in passive (left) and active (right) camera mode. The walls are only overlaid for visualization purposes and not known by the robot. Landmarks are displayed as green dots. In passive mode, the robot is not able to close the loop. In active mode, loop closing is clearly visible in the trajectory and results in an accurate pose estimation.

correct estimation of the robot position. More experiments can be found in (Frintrop and Jensfelt, 2008b).

6.4 Conclusion

In this chapter, we have summarized our work on attentive visual SLAM. The system includes feature detection, tracking, loop closing, and active camera control. Landmarks are selected based on biological mechanisms which favor salient regions, an approach which enables focusing on a sparse landmark representation. We have shown that the repeatability of salient regions is considerably higher than the one of regions from standard detectors.

The active gaze control module presented here enabled to obtain a better distribution of landmarks in the map and to redetect considerably more landmarks in loop closing situations than in passive camera mode. In some cases, loop closing is actually only possible by actively controlling the camera.

While we obtain a good pose estimation and a high matching rate, further improvements are always possible and planned for future work. For example, determining the salience of a landmark not only in the image but in the whole environment would help to focus on even more discriminative landmarks. While the SIFT descriptor is very powerful and achieves good results, it works best on textured regions while the salient region detector favors homogeneous blobs. The same is true for most existing detector-descriptor pairs, for example the most common combination of SIFT with a Difference-of-Gaussian detector (Lowe, 2004). Using a detector-descriptor pair that works on the same basic features will most likely result in an increase of speed and accuracy. In the future, we plan to use the recently developed detector-descriptor pair (Klein and Cremers, 2013) that is based on ideas of the CoDi saliency system to solve the visual SLAM problem.

Adapting the system to deal with really large environments could be achieved by removing landmarks which are not redetected to keep the number of landmarks low, by database management based on search trees, indexing (Sivic and Zisserman, 2003; Nister and Stewenius, 2006), and by using hierarchical maps as in (Clemente et al., 2007).

Chapter 7

Conclusion

In this summary, we have presented an overview of our research in the field of Cognitive Computer Vision. We have outlined the different directions that exist in this broad field, ranging from psychological and neurobiological work up to engineering approaches in robotics and computer vision. Our own approach to Cognitive Vision Systems is to seek for inspiration from findings about human vision and to build vision systems that profit from these findings. We believe that there are great opportunities in understanding the human visual system and in exploiting these ideas to improve machine vision systems. This will be especially important in the future, when cognitive systems become more and more advanced, and mobile vision devices will pervade our life even more than it is the case already now. Regardless if we consider cameras in smartphones and tablets, wearable devices such as Google Glass, driver assistance systems, or autonomous mobile robots that help us in our household, it is important that such devices are robust, efficient, and flexible, that they quickly adapt themselves to new situations by learning, and that humans can interact intuitively with them. All these properties are inherent in human perception and technical systems can learn from nature how to achieve them. Especially concepts of the human brain such as parallelization, hierarchical organization, and prioritization are equally important for machine vision and enable dealing with large amounts of data that have to be processed in real-time.

In this work, we have concentrated on four topics: object discovery, saliency detection, visual tracking, and visual SLAM. While we investigated these topics separately up to now and some of them make sense on their own for specific applications, they can also be part of a larger cognitive system that integrates these and other modules. We have outlined some of the possible connections between these modules already in the introduction, but there is still much more to a cognitive vision system. Some examples are the ability to recognize and categorize objects and scenes, to learn and integrate new knowledge continuously over time, and to deal with short and long term memory, including the important aspect of what to store and what to 'forget'. When leaving the subspace of Cognitive Vision Systems and regarding the much broader area of Cognitive Systems, the systems become even more complex. They include navigation behavior, planning and reasoning, speech recognition and production, manipulation of objects, and many more. One of the biggest challenges is to integrate all these modules to a robust, adaptive, and flexible system.

Bibliography

Achanta, R., Hemami, S., Estrada, F., and Süsstrunk, S. (2009). Frequency-tuned salient region detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Achanta, R. and Süsstrunk, S. (2010). Saliency Detection using Maximum Symmetric Surround. In *IEEE International Conference on Image Processing (ICIP)*.

Adam, A., Rivlin, E., and Shimshoni, I. (2006). Robust fragments-based tracking using the integral histogram. In *Proceedings Conference Computer Vision and Pattern Recognition (CVPR)*.

Adams, R. and Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 16(6):641 – 647.

Alexe, B., Deselaers, T., and Ferrari, V. (2010). What is an object? In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alexe, B., Deselaers, T., and Ferrari, V. (2012). Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(11):2189–2202.

Andreopoulos, A. and Tsotsos, J. (2013). 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding (CVIU)*, 117(8).

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916.

Aristotle (350 B.C.E.). *On Sense and the Sensible*. The Internet Classics Archive, 350 B.C.E., Translated by J. I. Beare.

Backer, G., Mertsching, B., and Bollmann, M. (2001). Data- and model-driven gaze control for an active-vision system. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(12):1415–1429.

Beuter, N., Lohmann, O., Schmidt, J., and Kummert, F. (2009). Directed attention - a cognitive vision system for a mobile robot. In *IEEE International Symposium on Robot and Human Interactive Communication*.

Björkman, M. and Eklundh, J.-O. (2007). Vision in the real world: Finding, attending and recognizing objects. *International Journal of Imaging Systems and Technology*, 16(2):189–208.

Borji, A. and Itti, L. (2010). State-of-the-art in visual attention modeling. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*.

Borji, A. and Itti, L. (2012a). Exploiting local and global patch rarities for saliency detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Borji, A. and Itti, L. (2012b). Salient object detection: A benchmark. In *European Conference on Computer Vision (ECCV)*.

Borji, A., Tavakoli, H. R., Sihite, D. N., and Itti, L. (2013). Analysis of scores, datasets, and models in visual saliency modeling. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal.

Bruce, N. D. B. and Tsotsos, J. K. (2009). Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):1–24.

Bundesen, C. and Habekost, T. (2005). Attention. In Lamberts, K. and Goldstone, R., editors, *Handbook of Cognition*. London: Sage Publications.

Carrasco, M. (2011). Visual attention: The past 25 years. *Vision Research*, 51:1484–1525.

Clemente, L. A., Davison, A. J., Reid, I. D., Neira, J., and Tardos, J. D. (2007). Mapping large loops with a single hand-held camera. In *Proceedings of Robotics: Science and Systems (RSS)*.

Connor, C. E., Egeth, H. E., and Yantis, S. (2004). Visual attention: Bottom-up versus top-down. *Current Biology*, 14.

Corbetta, M. and Shulman, G. L. (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews*, 3(3):201–215.

Draper, B. and Lionelle, A. (2003). Evaluation of selective attention under similarity transforms. In *Proceedings of the International Workshop on Attention and Performance in Computer Vision (WAPCV)*, pages 31–38.

Dubuc, B. (2014). The brain from top to bottom: An interactive website about the human brain and behaviour. http://thebrain.mcgill.ca/ (last accessed: Feb. 2014).

Engelhard, N., Endres, F., Hess, J., Sturm, J., and Burgard, W. (2011). Real-time 3D visual SLAM with a hand-held camera. In *Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*.

Ettinger, U. and Klein, C. (2014). Eye movements. In M., R. and C., M., editors, *Neuroeconomics*.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2).

Frintrop, S. (2006). *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, volume 3899 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer.

Frintrop, S. (2007). Visual robot localization and mapping based on attentional landmarks. In *Proceedings of the German Conference on Artificial Intelligence (KI)*, Lecture Notes in Computer Science (LNCS), pages 456–459. Springer.

Frintrop, S. (2008). The high repeatability of salient regions. In *Proceedings of the Workshop "Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environments" on European Conference on Computer Vision (ECCV)*.

Frintrop, S. (2010). General object tracking with a component-based target descriptor. In *Proceedings of the IEEE International Conference Robotics and Automation (ICRA)*.

Frintrop, S. (2011a). Computational visual attention. In Salah, A. A. and Gevers, T., editors, *Computer Analysis of Human Behavior*, Advances in Pattern Recognition. Springer.

Frintrop, S. (2011b). Towards attentive robots. *PALADYN, Journal of Behavioral Robotics*, 2(2).

Frintrop, S., Backer, G., and Rome, E. (2005). Goal-directed search with a top-down modulated computational attention system. In *Proceedings of the Annual meeting of the German Association for Pattern Recognition (DAGM)*, Lecture Notes in Computer Science (LNCS). Springer.

Frintrop, S. and Cremers, A. B. (2007). Top-down attention supports visual loop closing. In *Proceedings of the European Conference on Mobile Robotics (ECMR)*.

Frintrop, S. and Cremers, A. B. (2010). Visual landmark generation and redetection with a single feature per frame. In *Proceedings of the IEEE International Conference Robotics and Automation (ICRA)*.

Frintrop, S., García, G. M., and Cremers, A. B. (2014). A cognitive approach for object discovery. In *International Conference on Pattern Recognition (ICPR)*.

Frintrop, S. and Jensfelt, P. (2008a). Active gaze control for attentional visual SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Frintrop, S. and Jensfelt, P. (2008b). Attentional landmarks and active gaze control for visual SLAM. *IEEE Transactions on Robotics, Special Issue on Visual SLAM*, 24(5).

Frintrop, S., Jensfelt, P., and Christensen, H. I. (2006a). Attentional Landmark Selection for Visual SLAM. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

Frintrop, S., Jensfelt, P., and Christensen, H. I. (2006b). Pay attention when selecting features. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*.

Frintrop, S., Jensfelt, P., and Christensen, H. I. (2007). Simultaneous robot localization and mapping based on a visual attention system. In Paletta, L. and Rome, E., editors, *Attention in Cognitive Systems*, volume 4840 of *Lecture Notes on Artificial Intelligence (LNAI)*. Springer-Verlag.

Frintrop, S. and Kessel, M. (2008). Cognitive data association for visual person tracking. In *Proceedings of the IEEE Workshop on Human Detection from Mobile Platforms (HDMP) at the International Conference on Robotics and Automation (ICRA)*.

Frintrop, S. and Kessel, M. (2009). Most salient region tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Frintrop, S., Königs, A., Hoeller, F., and Schulz, D. (2009). Visual person tracking using a cognitive observation model. In *Proceedings of Workshop on People Detection and Tracking at the IEEE International Conference Robotics and Automation (ICRA)*.

Frintrop, S., Königs, A., Hoeller, F., and Schulz, D. (2010a). A component-based approach to visual person tracking from a mobile platform. *International Journal of Social Robotics*, 2(1):53–62.

Frintrop, S., Rome, E., and Christensen, H. I. (2010b). Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception*, 7(1).

Friston, K. (2012). The history of the future of the Bayesian brain. *Neuroimage*.

Gao, D., Han, S., and Vasconcelos, N. (2009). Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(6).

Gao, D. and Vasconcelos, N. (2007). Bottom-up saliency is a discriminant process. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

García, G. M. and Frintrop, S. (2013). A computational framework for attentional 3D object detection. In *Proceedings of the Annual Conference of the Cognitive Science Society*.

García, G. M., Frintrop, S., and Cremers, A. B. (2013). Attention-based detection of unknown objects in a situated vision framework. *German Journal of Artificial Intelligence*.

García, G. M., Klein, D. A., Stückler, J., Frintrop, S., and Cremers, A. B. (2012). Adaptive Multi-cue 3D Tracking of Arbitrary Objects. In *Proceedings of the Joint Pattern Recognition Symposium of the German Association for Pattern Recognition (DAGM) and the Austrian Association for Pattern Recognition (OAGM) (DAGM-OAGM)*.

Gegenfurtner, K. R. (2003). Cortical mechanisms of colour vision. *Nature Reviews Neuroscience*, 4:563–572.

Goferman, S., Zelnik-Manor, L., and Tal, A. (2011). Context-Aware Saliency Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Goodale, M. and Milner, A. (1992). Separate visual pathways for perception and action. *Trends Neuroscience*, 15(1).

Grill-Spector, K. (2003). The neural basis of object perception. *Current opinion in neurobiology*.

Grundmann, M., Kwatra, V., Han, M., and Essa, I. (2010). Discontinuous seam-carving for video retargeting. *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hamker, F. H. (1999). The role of feedback connections in task-driven visual search. In D. Heinke, G. W. Humphreys, A. O., editor, *Connectionist Models in Cognitive Neuroscience, Proceedings of the Neural Computation and Psychology Workshop (NCPW)*, pages 252–261. Springer Verlag.

Hamker, F. H. (2004). Modeling attention: From computational neuroscience to computer vision. In Paletta, L., Tsotsos, J. K., Rome, E., and Humphreys, G. W., editors, *Proceedings of the International Workshop on Attention and Performance in Computational Vision (WAPCV)*, pages 59–66.

Herbst, E., Henry, P., Ren, X., and Fox, D. (2011). Toward object discovery and modeling via 3-d scene comparison. In *International Conference on Robotics and Automation (ICRA)*.

Horbert, E., Martín García, G., Frintrop, S., and Leibe, B. (2014). Sequence Level Salient Object Proposals for Generic Object Detection in Video. Technical Report AIB-2014-06, RWTH Aachen (Conference paper under review).

Hou, X., Harel, J., and Koch, C. (2012). Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Hou, X. and Zhang, L. (2007). Saliency detection: a spectral residual approach. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hou, X. and Zhang, L. (2008). Dynamic visual attention: Searching for coding length increments. In *Advances in Neural Information Processing Systems*.

Hubel, D. and Wiesel, T. (1959). Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591.

Hyman, I. (2012). Remembering the father of cognitive psychology. Association for psychological Science: Observer.

Isard, M. and Blake, A. (1998). Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision (IJCV)*, 29(1):5–28.

Itti, L. (2003). Modeling primate visual attention. In Feng, J., editor, *Computational Neuroscience: A Comprehensive Approach*, pages 635–655. CRC Press.

Itti, L. and Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295–1306.

Itti, L. and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506.

Itti, L. and Koch, C. (2001a). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Itti, L. and Koch, C. (2001b). Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(11):1254–1259.

Itti, L., Rees, G., and Tsotsos, J., editors (2005). *Neurobiology of Attention*. Elsevier Academic Press.

Johnson-Roberson, M., Bohg, J., Björkman, M., and Kragic, D. (2010). Attention-based active 3D point cloud segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Jones, J. and Palmer, L. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258.

Kadir, T. and Brady, M. (2001). Saliency, scale and image description. *International Journal of Computer Vision (IJCV)*, 45(2):83–105.

Kadir, T., Zisserman, A., and Brady, M. (2004). An affine invariant salient region detector. In *Proceedings of European Conference of Computer Vision (ECCV)*.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (1996). *Essentials of Neural Science and Behavior*. McGraw-Hill/Appleton & Lange.

Karpathy, A., Miller, S., and Fei-Fei, L. (2013). Object Discovery in 3D Scenes via Shape Analysis. In *International Conference on Robotics and Automation (ICRA)*.

Katz, D., Kazemi, M., Bagnell, J. A., and Stentz, A. (2013). Clearing a pile of unknown objects using interactive perception. In *International Conference on Robotics and Automation (ICRA)*.

Kirkland, K. and Gerstein, G. (1999). A feedback model of attention and context dependence in visual cortical networks. *Journal of Computational Neuroscience*, 7.

Klein, D. and Cremers, A. (2011). Boosting scalable gradient features for adaptive real-time tracking. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Klein, D. A. and Cremers, A. B. (2013). Discriminable points that stick out of their environment. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*.

Klein, D. A. and Frintrop, S. (2011). Center-surround divergence of feature statistics for salient object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Klein, D. A. and Frintrop, S. (2012). Salient Pattern Detection using W_2 on Multivariate Normal Distributions. In *Proceedings of the Joint Pattern Recognition Symposium of the German Association for Pattern Recognition (DAGM) and the Austrian Association for Pattern Recognition (OAGM) (DAGM-OAGM)*.

Klein, D. A., Schulz, D., Frintrop, S., and Cremers, A. B. (2010). Adaptive real-time tracking for arbitrary objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Knill, D. C. and Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences*, 27(12).

Koch, C. and Ullman, S. (1985). Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219–227.

Kootstra, G., Nederveen, A., and de Boer, B. (2008). Paying attention to symmetry. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Krauskopf, J., Williams, D., and Heeley, D. (1982). Cardinal directions of color space. *Elsevier*.

Krüger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., Rodríguez-Sánchez, A. J., and Wiskott, L. (2013). Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Kuffler, S. W. (1953). Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology*.

Liao, Q., Leibo, J., Mroueh, Y., and Poggio, T. (2013). Can a biologically-plausible hierarchy effectively replace face detection, alignment, and recognition pipelines? arXiv:1311.4082v1.

Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., and Shum, H.-Y. (2009). Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Livingstone, M. S. and Hubel, D. H. (1987). Psychophysical evidence for separate channels for the perception of form, color, movement, and depth. *Journal of Neuroscience*, 7(11):3416–3468.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110.

Ma, Y.-F. and Zhang, H.-J. (2003). Contrast-based image attention analysis by using fuzzy growing. In *ACM International Conference on Multimedia*.

Maki, A., Nordlund, P., and Eklundh, J.-O. (2000). Attentional scene segmentation: Integrating depth and motion. *Computer Vision and Image Understanding (CVIU)*, 78(3):351–373.

Manén, S., Guillaumin, M., and Van Gool, L. (2013). Prime Object Proposals with Randomized Prim's Algorithm. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Marchesotti, L., Cifarelli, C., and Csurka, G. (2009). A framework for visual saliency detection with applications to image thumbnailing. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Michalke, T., Fritsch, J., and Goerick, C. (2008). Enhancing robustness of a saliency-based attention system for driver assistance. In *Proceedings of the International Conference on Computer Vision Systems (ICVS)*.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 27(10).

Milanese, R., Wechsler, H., Gil, S., Bost, J., and Pun, T. (1994). Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 781–785.

Milner, A. and Goodale, M. (2008). Two visual streams revisited. *Neuropsychologica*, 46(1):774–785.

Navalpakkam, V. and Itti, L. (2006). An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Navalpakkam, V., Rebesco, J., and Itti, L. (2004). Modeling the influence of knowledge of the target and distractors on visual search. *Journal of Vision*, 4(8):690.

Neisser, U. (1967). *Cognitive Psychology*. Appleton-Century-Crofts, New York.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, ISMAR, pages 127–136. IEEE Computer Society.

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

O'Reilly, R. C. and Munakata, Y. (2000). *Computational Exploration in Cognitive Neuroscience. Understanding the Mind by Simulating the Brain*. The MIT Press.

Pashler, H. (1997). *The Psychology of Attention*. MIT Press.

Peirce, C. S. Commens – Virtual Center for Peirce Studies at the University of Helsinki, http://www.helsinki.fi/science/commens/.

Perazzi, F., Krahenbuhl, P., Pritch, Y., and Hornung, A. (2012). Saliency filters: Contrast based filtering for salient region detection. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 733–740.

Pérez, P., Hue, C., Vermaak, J., and Gangnet, M. (2002). Color-based probabilistic tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Pérez, P., Vermaak, J., and Blake, A. (2004). Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3).

Porikli, F. (2005). Integral histogram: A fast way to extract histograms in Cartesian spaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Posner, M. and Cohen, Y. (1984). Components of visual orienting. In Bouma, H. and Bouwhuis, D., editors, *Attention and Performance X*, pages 531–556. London: Erlbaum.

Pylyshyn, Z. W. (2001). Visual indexes, preconceptual objects, and situated vision. *Cognition*, 80(1-2):127–158.

Rensink, R. A. (2000). The dynamic representation of scenes. *Visual Cognition*, 7:17–42.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025.

Rodieck, R. W. (1965). Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*.

Rotenstein, A., Andreopoulos, A., Fazl, E., Jacob, D., Robinson, M., Shubina, K., Zhu, Y., and Tsotsos, J. (2007). Towards the dream of intelligent, visually-guided wheelchairs. In *Proceedings of the International Conference on Technology and Aging*.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: an efficient alternative to sift or surf. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Rutishauser, U., Walther, D., Koch, C., and Perona, P. (2004). Is bottom-up attention useful for object recognition? In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schauerte, B. and Stiefelhagen, R. (2012). Quaternion-based spectral saliency detection for eye fixation prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Schillaci, G., Bodiroža, S., and Hafner, V. V. (2012). Evaluating the effect of saliency detection and attention manipulation in human-robot interaction. *International Journal of Social Robotics*.

Scholl, B. J. (2001). Objects and attention: the state of the art. *Cognition*, 80:1–46.

Sejnowski, T., Koch, C., and Churchland, P. (1988). Computational neuroscience. *Science*.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(3).

Siagian, C. and Itti, L. (2007). Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(2):300–312.

Siagian, C. and Itti, L. (2009). Biologically inspired mobile robot vision localization. *IEEE Transaction on Robotics*, 25(4):861–873.

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Spelke, E. S. (1990). Principles of object perception. *Cognitive Science*, 14.

Sun, X., Yao, H., and Ji, R. (2012). What are we looking for: Towards statistical modeling of saccadic eye movements and visual saliency. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Theeuwes, J. (2010). Top-down and bottom-up control of visual selection. *Acta Psychologica*, 135:77–99.

Tipper, S. P., Weaver, B., Jerreat, L. M., and Burak, A. L. (1994). Object-based and environment-based inhibition of return of visual attention. *Journal of Experimental Psychology*.

Treisman, A. M. and Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, 12:97–136.

Tseng, P., Carmi, R., Cameron, I. G. M., Munoz, D., and Itti, L. (2009). Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of Vision*, 9(7:4):1–16.

Tsotsos, J. K. (1990). Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13(3):423–445.

Tsotsos, J. K. (2011). *A Computational Perspective on Visual Attention*. The MIT Press.

Tsotsos, J. K., Verghese, G., Stevenson, S., Black, M., Metaxas, D., Culhane, S., Dickinson, S., Jenkin, M., Jepson, A., Milios, E., Nuflo, F., Ye, Y., and Mann, R. (1998). PLAYBOT: A visually-guided robot to assist physically disabled children in play. *Image and Vision Computing 16, Special Issue on Vision for the Disabled*, pages 275–292.

Tuytelaars, T., Lampert, C. H., Blaschko, M. B., and Buntine, W. (2010). Unsupervised object discovery: A comparison. *International Journal on Computer Vision (IJCV)*, 88:284–302.

Ungerleider, L. and Mishkin, M. (1982). Two cortical visual systems. In Ingle, D., Goodale, M., and Mansfield, R., editors, *Analysis of visual behavior*, pages 549–586. MIT Press.

Vernon, D. (2006). The space of cognitive vision. In *Cognitive Vision Systems*, volume 3948 of *Lecture Notes in Computer Science*. Springer.

Vijayakumar, S., Conradt, J., Shibata, T., and Schaal, S. (2001). Overt visual attention for a humanoid robot. In *Proceedings of the International Conference on Intelligence in Robotics and Autonomous Systems (IROS)*, pages 2332–2337.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154.

von Hofsten, C. and Spelke, E. (1985). Object perception and object-directed reaching in infancy. *Journal of Experimental Psychology*, 144(2).

Wagemans, J., Elder, J. H., Kubovy, M., e. Palmer, S., Peterson, M. A., Singh, M., and von der Heydt, R. (2012). A century of gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization. *Psychological Bulletin*.

Walther, D. and Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*.

Walther, D., Rutishauser, U., Koch, C., and Perona, P. (2005). Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding (CVIU)*, 100(1-2):41–63.

Wertheimer, M. (1922). Untersuchungen zur Lehre von der Gestalt, I: Prinzipielle Bemerkungen [Investigations in Gestalt theory: I. The general theoretical situation]. *Psychologische Forschung*, pages 47–58.

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238.

Wolfe, J. M. and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, 5:1–7.

Zeki, S. (1993). *A Vision of the Brain*. Blackwell Scientific.

Zhang, L., Tong, M. H., Marks, T. K., Shan, H., and Cottrell, G. W. (2008). SUN: A Bayesian framework for Saliency Using Natural Statistics. *Journal of Vision*, 8(32).

Zhang, X., Zhaoping, L., Zhou, T., and Fang, F. (2012). Neural activities in V1 create a bottom-up saliency map. *Neuron*, 73:183–192.

Zhu, L., Klein, D. A., Frintrop, S., Cao, Z., and Cremers, A. B. (2013). Multi-Scale Region-Based Saliency Detection Using W_2 Distance on N-Dimensional Normal Distributions. In *International Conference on Image Processing (ICIP)*.

Part II

Publications

Publication [1]

Simone Frintrop, Germán Martín García, and Armin B. Cremers. A cognitive approach for object discovery. *International Conference on Pattern Recognition (ICPR) (accepted)*, Stockholm, Sweden, 2014.

A Cognitive Approach for Object Discovery

Simone Frintrop, Germán Martín García and Armin B. Cremers
Institute of Computer Science III,
Rheinische Friedrich-Wilhelms-Universität Bonn
53117 Bonn, Germany
Email: {frintrop,martin,abc}@iai.uni-bonn.de

***Abstract*—Object discovery is the task of detecting unknown objects in images. The task is of large interest in many fields of machine vision, ranging from the automatic analysis of web images to interpreting data of a mobile robot or a driver assistant system. Here, we present a new approach for object discovery, based on findings of the human visual system. Proto-objects are detected with a segmentation module, generating perceptually coherent image regions. In parallel, a saliency system detects regions of interest in images and serves to select segments, depending on their saliency. We obtain very good results on a database of salient objects and on real-world office scenes.**

I. Introduction

One essential task in many machine vision applications is to automatically and quickly detect objects in the environment. This topic is of interest for many applications, for example automatically processing web images (thumbnailing, resizing, etc.), analyzing video data from devices such as Google Glass, or finding and manipulating objects with an autonomous robot. In contrast to object recognition or classification, the types of objects are not known in advance, there is no training phase, and the system starts without any pre-knowledge. Thus, the system addresses the question "what is an object?".[1] Object discovery is a challenging task for machine vision and belongs to the open problems in the field. The reason is the 'chicken-and-egg property' of the problem: how to search for an object before knowing how it looks like?

While difficult for machines, detecting objects is effortlessly, even unconsciously, done by humans. Thus, it is worth investigating how the human visual system achieves this task. We investigated the findings of psychology and neurobiology on object perception (cf. Sec. III) and developed a biologically inspired strategy that finds objects in a two step approach (cf. Fig. 1): first the image is segmented into perceptually coherent parts, called proto-objects; second, a saliency map is computed and proto-objects are selected depending on their saliency. The result are *object hypotheses* or *object proposals*.

Our contributions in this paper are twofold. First, we propose an improved saliency system that outperforms 7 state-of-the-art saliency models. Second, we propose a new approach for object discovery that is based on concepts from human perception and is applicable to web images as well as to real-world video data.

II. Related work

While object recognition is a well established field, object discovery still involves many challenges. Especially the discovery of objects in 2D images or videos, in which no depth information is available, is difficult. However, several people have investigated this problem and suggested promising approaches; a survey can be found in [30]. Many methods base on the fact that objects are consistent over several images while background is not, and identify regions across images that are visually similar [4], [22]. A related idea is to regard a sequence over time and detect changes, since it is likely that these changes correspond to objects [14]. In the approach of Manén et al. [21], the image is segmented into superpixels [9] and then, connected superpixels are grouped randomly by sampling partial spanning trees that have high sums of edge weights. Other approaches apply machine-learning techniques to learn which aspects of an image might correspond to an object [3]. This idea bases on the fusion of several feature channels, similar as in the field of salient object detection [5]. These approaches are often designed for web images and use often assumptions such as objects are large and central in an image (photographer bias). Instead, we propose here a generally applicable approach that works for single 2D images as well as for real-world image sequences.

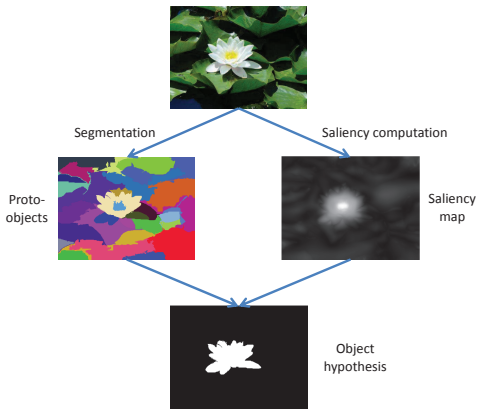


Fig. 1. Simplified overview of our object discovery approach for web images: Saliency selects the relevant proto-objects to form object hypotheses.

Recently, especially with the upcoming RGB-D sensors, several groups have investigated object discovery in 3D data. Karpathy et al. find objects on the 3D meshes obtained from RGB-D data [19]. Johnson-Roberson et al. do object segmentation on full point clouds [17]. The segmentation is seeded at salient points in the image that are mapped to the full point cloud. In [26], 3D object models are built by matching scans from partial views from which they subtract

[1] When referring to objects, we follow a definition from psychology: Objects are "manipulable units with internal coherence and external boundaries" [31].

points that correspond to planar surfaces: floor, walls, etc. In [10], objects were detected in RGB-D data by observing a scene over time and incrementally updating 3D object models. Generally, such 3D approaches have the advantage that they can exploit depth information which is a very helpful feature for object discovery. In this paper, we focus instead on 2D approaches for object discovery in which no depth information is available.

III. Human Object Perception

Object perception is deeply rooted in the human visual system which enables a fast and effortless detection of objects. Even objects of completely unknown appearance are easily recognized as objects, even by young infants [28]. It is not yet completely understood how object perception works in the human brain, but many findings are well known. We will concentrate here on the findings which are important for our framework of computational object discovery.

Physiologically, object detection and recognition take place in the *ventral stream* of the human visual system. This stream is also called *what pathway* since it is strongly involved in color and form processing and is responsible for deciding *what* is visible in a scene. This is opposed to the *dorsal* or *where pathway* that processes mainly motion and depth cues and is responsible for object localization [13]. The ventral visual pathway starts its processing as early as the retina, goes on through the LGN, V1, V2, and V4, until it ends in the inferotemporal cortex (IT), responsible for object recognition.

Many cells in these visual areas have a center-surround structure: they respond excitatorily to light at the center of their receptive field[2] and inhibitorily to light at the surround or vice versa. This means, they have the strongest response if the center is bright and the surround dark (ON-OFF cells) or vice versa (OFF-ON cells). Cells are divided into three types, organized in three channels: the luminance channel, the red-green channel, and the blue-yellow channel [12]. These channels lead from the retina to higher brain areas.

Cells exist with concentric receptive fields and with elongated ones. It has been shown that the concentric fields are modeled best with a two-dimensional Difference-of-Gaussian (DoG) function [25], while the elongated fields are modeled best with Gabor filters [18]. Both types of filters are frequently used in computer vision, because the blob and edge detection that they perform is equally important there as in human vision.

Coming back to object detection, there is evidence that the individuation of objects, which addresses the question of what is an object, takes place before object recognition [23]. The decision of which parts of the visual scene belong to objects results from perceptual organization rules, especially from segmentation processes that bundle parts of the visual input. Such segmentation mechanisms are believed to exist on all levels of the visual system [27] and the bundling is based on concepts such as similarity, proximity, and other processes described already early by the Gestalt principles. A recent review about the history of the Gestalt laws as well as new findings can be found in [32].

[2]The receptive field of a cell is the collection of other cells that influences the output of the cell.

The result of these segmentation processes are so called "proto-objects" [24]. They describe the local scene structure of a spatially limited region and might correspond to objects, but they might also be object parts or collections of several objects. Rensink [24] describes them as "volatile structures of limited spatial and temporal coherence", meaning that they are regenerated constantly and not stored in visual memory. Later on, proto-objects are combined by focused attention to form coherent objects. This is an important step, since it enables to decide which segments an object consists of.

IV. Computational Object Discovery

Formally, object discovery means we are interested in an algorithm that can answer the question of whether a given pixel set corresponds to an object or not. But even if we had a method to answer this question reliably, the problem would be complex: an image of $w \times h = n$ pixels consists of 2^n possible subsets that could potentially form an object (due to partial occlusions, object parts do not necessarily have to be connected). Tsotsos has proven that the related problem of unbounded visual search, that means search for an object whose features are unknown, is NP-hard [29]. And even when restricting the problem to a rectangular bounding box, the problem is still demanding: $O(n \cdot w \cdot h)$ subwindows have to be tested for their objectness, since at each pixel, subwindows of all possible sizes have to be tested. Depending on how computationally expensive the objectness measure itself is, this can easily take several seconds or even minutes which makes the approach inapplicable for real-time applications.

To deal with the complexity of the object discovery problem, we follow the strategy that nature developed and find objects in a two step approach: first the image is segmented into perceptually coherent parts (proto-objects [24]); second, a saliency map is computed and segments are selected depending on their saliency. Thus, the saliency system is responsible for prioritizing the data processing by providing reasonable regions of interest.

For generating proto-objects, we use the segmentation approach of Felzenzwalb and Huttenlocher [9] (cf. Fig. 1, left). This is a graph-based segmentation method that is based on two important Gestalt principles: the similarity and proximity of pixels. The method creates, as the authors state, "perceptually important regions". The second step addresses the question of which segments belong together to form objects. According to Rensink [24], we let attention select the relevant proto-objects. This is done by computing a saliency map that highlights regions of potential interest: the brighter a pixel in the saliency map, the more salient this region is and the larger the probability to contain perceptually relevant data. While in human vision, bottom-up as well as top-down cues play an important role for attention, top-down knowledge is not always available, and in absence of a task, bottom-up saliency is often the best that can be used. Therefore, we use here a pure bottom-up saliency map to select proto-objects, but if top-down information is available, a top-down map can equally well be used.

To compute the saliency map, we use the CoDi saliency system [20] since it is real-time capable, computes precise saliency maps, and works for web images as well as real-

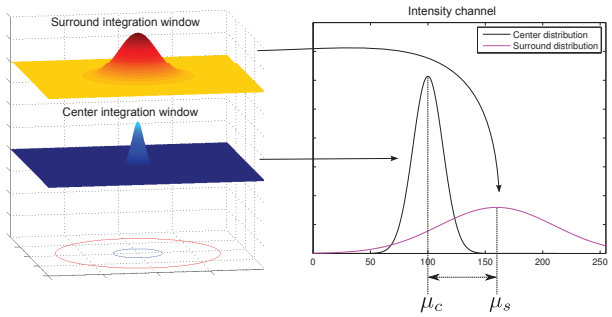


Fig. 2. Visualization of the center-surround computations in the original CoDi saliency system [20] and in our adapted version. Center and surround regions in the image (red and blue ellipses) are weighted with Gaussian windows (left) and feature distributions are determined (right: intensity distribution). The contrast was originally computed with the W_2 metric; here we use the Manhattan distance on the mean values of the distributions (right). The figure shows how this approach corresponds to a Difference of Gaussian approach since each mean value corresponds to the weighted mean of values in the corresponding ellipse.

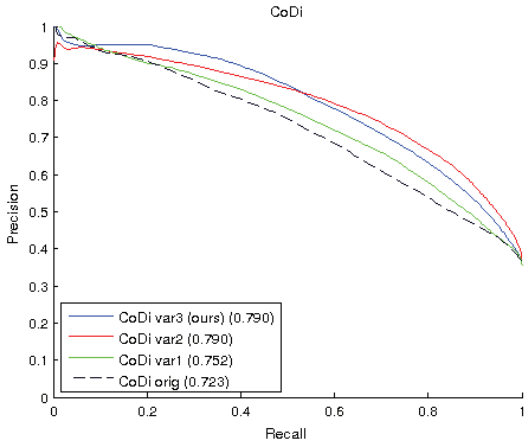


Fig. 3. Comparison of the original version of the CoDi-Saliency system [20] with 3 improvements that we suggested (see text for details). AUC values in parentheses. Evaluation done as described in Sec. V-A.

world images[3]. The CoDi system has shown to outperform many other saliency methods in [20] and source code is openly available[4]. The idea of the CoDi-Saliency is to compute center-surround contrast by comparing normal distributions that represent the feature statistics in the corresponding image regions. Distributions are compared with the W_2-distance (Wasserstein metric based on the Euclidean norm). This concept is visualized in Fig. 2, left. This center-surround measure is embedded into a scale-space structure to enable the detection of objects of different sizes. The computations are performed for intensity and color features, where the latter operates on an opponent-color space with one red-green and one blue-yellow axis. These dimensions correspond to the opponent color channels of the human visual system (cf. Sec. III).

We made several changes on the CoDi system to improve performance. The effect of each of the changes is visualized in Fig. 3 and Fig. 4. First, we adapted the size of the integration window for the center and the surround distribution from $\sigma_c = 1$ versus $\sigma_s = 10$ to $\sigma_c = 1$ versus $\sigma_s = 5$. The latter fits better to human perception [7] and it achieved better performance also in our experiments. We call CoDi with this improvement **variant 1**. Second, we changed the Difference of Gaussian pyramid to a Gaussian pyramid (**variant 2**, includes improvements of variant 1). This makes sense because the DoG operation computes contrasts, which is anyway done by the center-surround operation that is applied to each layer later on. So, it is reasonable to restrict the contrast computation to one place and operate directly on the Gaussian pyramid. This change had the largest visible effect from our improvements since it produces much preciser saliency maps (cf. Fig. 4).

The third change (**variant 3**, includes improvements of variants 1 and 2) affects the computation of the center-surround difference itself. The original CoDi system computes the W_2-distance of normal distributions. However, we found that using instead the much simpler Manhattan distance achieves basically the same results with less computational effort and results in cleaner saliency maps. Interestingly, the Manhattan distance which compares only the mean values of the normal distributions and ignores the variance, corresponds to a Difference of Gaussian approach which is the traditional way to simulate human ganglion and simple cells which are responsible for contrast detection in the human visual system [7]. The reason is the following: the normal distributions computed in CoDi are maximum-likelihood estimates of the center or the surround region, weighted by a Gaussian integration window. Thus, the mean of the normal distribution of a center region centered at pixel position (x,y), is defined as

$$\hat{\mu}_c(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} w(x-i, y-j) F(x-i, y-j), \quad (1)$$

for a $k \times k$ Gaussian window centered at (x,y) with variance ${\sigma_c}^2$ and resulting weights w; F contains the values of the corresponding feature channel, e.g., intensity or 2D color values. The mean of the surround region $\hat{\mu}_s$ is obtained in the same way with a σ_s that is larger than σ_c (as mentioned above, we used $\sigma_c = 1$ versus $\sigma_s = 5$). $\hat{\mu}_c$ and $\hat{\mu}_s$ are either single values (intensity), or two-dimensional vectors (color). Thus, by simply subtracting $\hat{\mu}_c$ from $\hat{\mu}_s$ or vice versa, we obtain the traditional Difference-of-Gaussian method. Since this can be done exactly in the same framework as the distribution-based version, it enables a direct comparison of the methods. This idea is visualized in Fig. 2.

While the AUC value did not change when switching from W_2 to Manhattan distance (cf. Fig. 3), the system is faster and obtained cleaner saliency maps (there are less bright borders around objects, cf. Fig. 4, right). The latter aspect resulted in considerably better performance when combining the saliency maps with segmentation. We call this variant 3 of CoDi **"Simple CoDi"**, since it is simpler and faster to compute while producing cleaner saliency maps than the original CoDi system.

Selecting proto-objects based on saliency is then done by combining all segments in which at least $k\%$ of the pixels are

[3]Many other recent approaches for saliency computation are only suitable for web images, since they make several assumptions on images, such as objects are large and central in an image and do not intersect with the image borders

[4]http://www.iai.uni-bonn.de/∼kleind/

Fig. 4. From left to right: Original image, saliency maps of the original version of the CoDi-Saliency system [20], of CoDi variant 1, of CoDi variant 2, and of CoDi variant 3 ("Simple CoDi") (see text for details). We used "Simple CoDi" in this work.

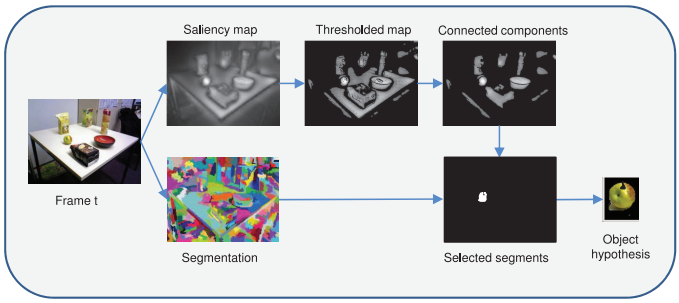


Fig. 5. Object discovery in real-world images.

above a saliency threshold t (we used $k = 25$ and $t = 112$). From these selected segments, all connected components form an object hypothesis (see Fig. 1).

While the described approach works very well on many web images, even without using assumptions about the location of objects (e.g. center-bias), real-world applications are more challenging in many aspects. When interpreting data from an autonomous mobile robot or a mobile device like Google Glass, images are, on the one hand, usually of lower quality due to illumination changes, motion blur, and cheaper cameras, but on the other hand much more complex in content because they contain more objects and clutter. To deal with several objects, we have to determine which proto-objects belong to which object hypothesis. Therefore, we have extended our approach for object discovery as follows. Here again, the saliency map is computed with our "Simple CoDi" system. Then, adaptive thresholding[5] (OpenCV method) thresholds the saliency map with help of a local Gaussian kernel, and connected components are found in the resulting map and ranked by average saliency. Finally, the overlap of each proto-object with these salient components is determined and all proto-objects that are covered by at least $k\%$ of a salient component are chosen to belong to the current object candidate. Thus, each salient component results in an object hypothesis and the precise boundaries are obtained by the segmentation process. Fig. 5 visualizes the process.

V. Experiments and Results

Our experiments are divided into three parts: first, we evaluate the improvements on the CoDi-saliency system. Second, we show the performance of the proposed object discovery approach on a database of salient objects. Finally, we show that the approach is also applicable to challenging real-world settings with many objects and clutter.

A. Saliency evaluation

We have compared our new adaption of the CoDi-saliency system with 6 other saliency systems: HSaliency [34], Yang 2013 [35], AC 2010 [2], HZ [15], AIM [6], and the SaliencyToolbox (ST) [33] which is a reimplementation of the Itti-system [16]. They have been chosen due to their popularity and frequency of citations [6], [15] or due to their recency and very good results on similar tasks [2], [34], [35], and due to the availability of source code.

We have evaluated the results on images from the *coffee machine sequence* which was also used in [11]. The sequence has 600 frames and shows a complex office scene. Each frame contains between 20 and 50 objects. Object ground truth was annotated on every 30-th frame. We chose this setting for evaluation instead of the commonly used benchmark datasets with web images, because we want to test the ability of the systems to deal with challenging real-world scenes that contain many objects. The images were evaluated according to the procedure proposed in [1]: thresholding the saliency maps with an increasing $k \in [0, 255]$ results in binarized maps. Then, each of these maps is matched against the ground truth to obtain precision and recall.

The results of the comparison are displayed in Fig. 6, some of the saliency maps are displayed in Fig. 7. It can be seen that the "simple CoDi" saliency system clearly outperforms all other systems in terms of precision and recall. Furthermore, the system is with 0.098 sec. on an 320×240 image (Intel Core i3-2330M, 4 x 2,2 GHz, 32bit, 4GB RAM) close to real-time on non-optimized code. Parallelization could further improve the speed of the system.

B. Object discovery on web images

In this section, we evaluate our object discovery approach on the MSRA-1000 database of salient objects [1]. The images contain objects that were marked as salient by 2 out of 3 users. Fig. 8 shows several examples from our approach for object discovery, and Fig. 9 shows how the new approach outperforms the CoDi-saliency method without segmentation. It can be seen that the curve drops considerably later when the recall values grow.

C. Object discovery on real-world scenes

Finally, we have applied the object discovery approach to real-world images obtained from the office sequence mentioned before. Some example images are shown in Fig. 11: on the left, a simple table-top scene to illustrate the idea (not used

[5] In most recent work, we obtained even better results with region growing instead of adaptive thresholding. Please check our newest publications at http://www.iai.uni-bonn.de/∼frintrop

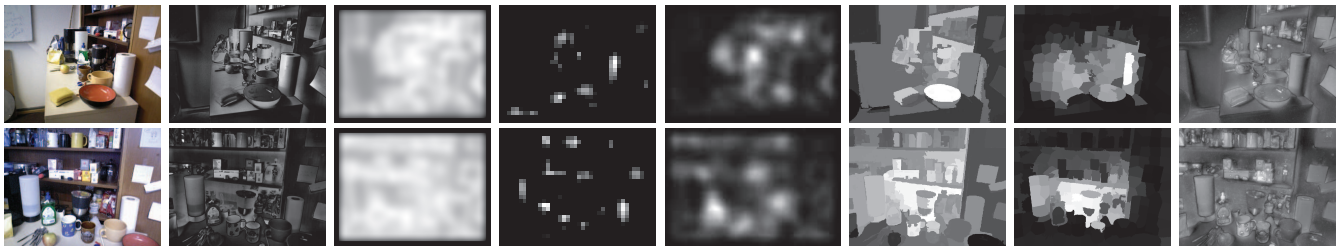


Fig. 7. Saliency maps from AC [2], AIM [6], SaliencyToolbox [33], HZ [15], HSaliency [34], Yang [35], and our "Simple CoDi" saliency system

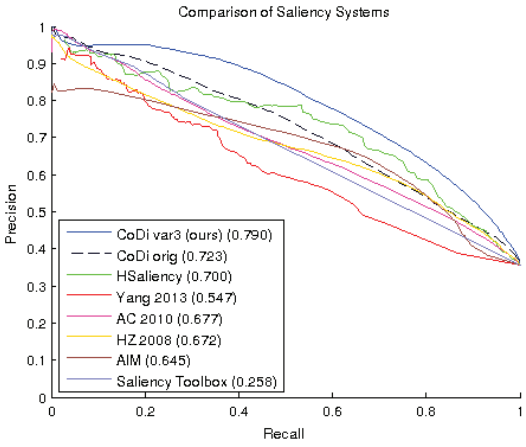


Fig. 6. Comparing our "Simple CoDi" saliency system to 7 state-of-the-art methods: CoDi orig [20], HSaliency [34], Yang [35], AC [2], HZ [15], AIM [6], and the SaliencyToolbox [33]. AUC values in parentheses.

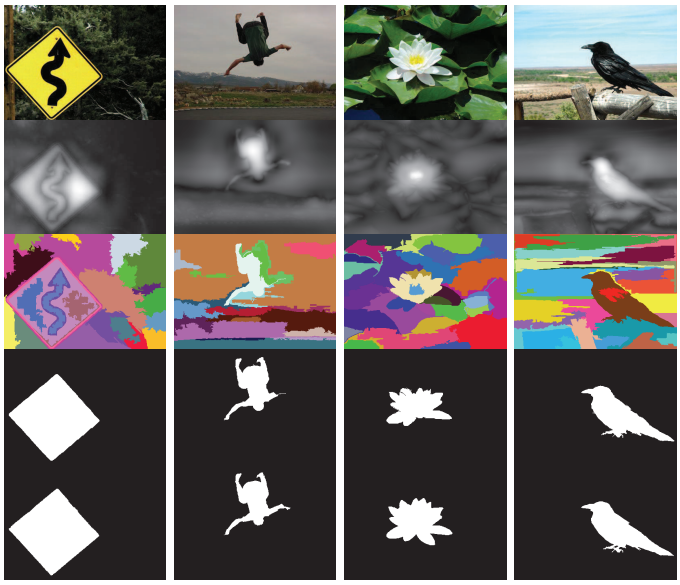


Fig. 8. Several examples of our object discovery. From top to bottom: original images, saliency maps, segmentations, object hypotheses, ground truth.

for the quantitative analysis), in the middle and on the right two examples for the office database (used for quantitative analysis).

We compute the recall, i.e., the percentage of objects which are found by our approach, and the precision, i.e., the percentage of valid object hypotheses that really represent an object[6]. For this, we consider a match as valid if the Pascal measure is satisfied (intersection-over-union > 0.5) [8]. We compare our method with two other approaches: the "objectness" measure of Alexe et al. [3], and the object discovery method of Manén and colleagues [21]. Since our approach assigns a saliency value to each detected proposal and the two other methods have a ranking for their proposals, we have a fair way of comparing the best N object candidates of all three approaches. This is often of advantage for real-time systems that have to prioritize processing capacities. Therefore, we sort the detected objects by their quality and evaluate the performance of the systems depending of the number of object hypotheses per image that are considered. Since the objectness measure returns bounding boxes instead of precise regions, we represent the ground truth also by boxes for their approach and evaluate our measure once with pixel-precise regions (green curve) and once with boxes (red curve) to enable a fair comparison.

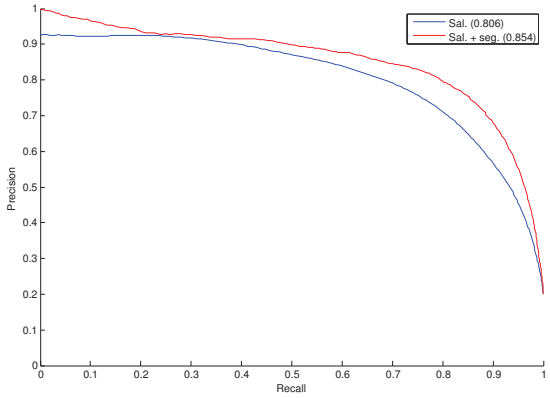


Fig. 9. Object discovery on the MSRA database. Blue curve (Sal.): CoDi-Saliency [20]; red curve (Sal. + Seg.): new combination of saliency and segmentation. AUC values in parentheses.

The results of the quantitative evaluation are shown in Fig. 10. It shows that our method outperforms the objectness measure clearly. Although it is also visible that the approach still misses many objects (there is no current method that can detect all objects in such challenging scenes), it can also be seen that the detected object hypotheses have a good quality and are good candidates as input for object recognition modules or for manipulation by a mobile robot. In the future, we plan to track proposals over time to improve the quality of the approach.

[6]Note that recall and precision measure different qualities here than in Sec. V-A

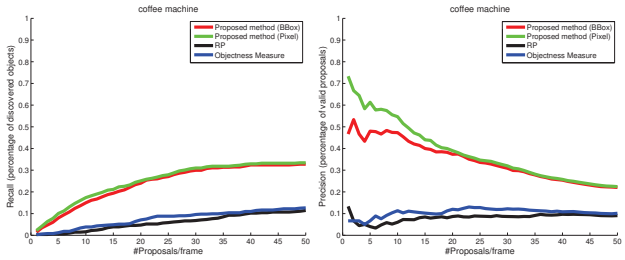


Fig. 10. Comparison of our object discovery method (once with pixel-precise regions and ground truth (green curve), once with bounding boxes (red curve) with the objectness measure from [3] (blue curve). Left: the percentage of discovered objects per frame (recall), right: the percentage of valid proposals (precision). Performance is plotted depending on the number of object proposals that were considered (best N proposals per frame).

Fig. 11. Top: some examples of our object discovery method on real-world office scenes. Each colored contour shows one detected object hypothesis. Bottom: separately displayed object hypotheses of the above images.

VI. Conclusion

We have presented a cognitive approach for object discovery that is based on several findings from human object perception. Perceptually coherent regions are detected with a segmentation method and saliency serves to select and combine segments to form object hypotheses. We have shown that the approach is able to detect objects in web images, which is useful for applications such as thumbnailing or automatic resizing, as well as to operate on real-world data as a mobile robot or a head-mounted camera would obtain. In future work, we will add Gestalt principles such as symmetry or convexity to evaluate whether the obtained object hypotheses are valid.

Acknowledgment

The authors would like to thank DFG for financing this research and Thomas Werner for his help with the experiments.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *Proc. CVPR*, 2009.

[2] R. Achanta and S. Süsstrunk. Saliency Detection using Maximum Symmetric Surround. In *Proc. of ICIP*, 2010.

[3] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the Objectness of Image Windows. *Trans. on PAMI*, 34(11):2189–2202, 2012.

[4] H. Arora, N. Loeff, D. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. In *Proc. of CVPR*, 2007.

[5] Ali Borji, Hamed R. Tavakoli, Dicky N. Sihite, and Laurent Itti. Analysis of scores, datasets, and models in visual saliency modeling. In *ICCV*, 2013.

[6] N. D. B. Bruce and J. K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *J. of Vision*, 9(3):1–24, 2009.

[7] C. Enroth-Cugell and J.G. Robson. The Contrast Sensitivity of Retinal Ganglion Cells of the Cat. *J. Physiol.*, 1966.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[10] G. Martín García, S. Frintrop, and A. B. Cremers. Attention-based detection of unknown objects in a situated vision framework. *German Journal of Artificial Intelligenz, Springer*, 2013.

[11] Germán Martín García and Simone Frintrop. A computational framework for attentional 3D object detection. In *Proc. of the Annual Conf. of the Cognitive Science Society*, 2013.

[12] K. R. Gegenfurtner. Cortical mechanisms of colour vision. *Nature Reviews Neuroscience*, 4:563–572, 2003.

[13] M.A. Goodale and A.D. Milner. Separate visual pathways for perception and action. *Trends Neuroscience*, 15(1), 1992.

[14] E. Herbst, P. Henry, X. Ren, and D. Fox. Toward object discovery and modeling via 3-d scene comparison. In *ICRA*, 2011.

[15] X. Hou and L. Zhang. Dynamic visual attention: Searching for coding length increments. In *Advances in Neural Information Processing Systems*, 2008.

[16] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[17] M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic. Attention-based active 3D point cloud segmentation. In *IROS*, 2010.

[18] J.P. Jones and L.A. Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.*, 58(6):1233–1258, 1987.

[19] A. Karpathy, S. Miller, and L. Fei-Fei. Object discovery in 3D scenes via shape analysis. In *ICRA*, 2013.

[20] D. A. Klein and S. Frintrop. Salient Pattern Detection using W_2 on Multivariate Normal Distributions. In *Proc. of (DAGM-OAGM)*, 2012.

[21] S. Manén, M. Guillaumin, and L. Van Gool. Prime Object Proposals with Randomized Prim's Algorithm. In *ICCV*, 2013.

[22] N. Payet and S. Todorovic. From a set of shapes to object discovery. In *Proc. of ECCV*, 2010.

[23] Z. W. Pylyshyn. Visual indexes, preconceptual objects, and situated vision. *Cognition*, 80(1-2):127–158, June 2001.

[24] R. A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7:17–42, 2000.

[25] R. W. Rodieck. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 1965.

[26] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard. Unsupervised learning of 3D object models from partial views. In *ICRA*, 2009.

[27] B. J. Scholl. Objects and attention: the state of the art. *Cognition*, 80:1–46, 2001.

[28] E. S. Spelke. Principles of object perception. *Cog. Science*, 14, 1990.

[29] J. K. Tsotsos. Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13(3):423–445, 1990.

[30] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *Int. j. of computer vision*, 88(2), 2010.

[31] C. von Hofsten and E.S. Spelke. Object perception and object-directed reaching in infancy. *Journal of Experimental Psychology*, 144(2), 1985.

[32] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. A Century of Gestalt Psychology in Visual Perception: I. perceptual Grouping and Figure-Ground Organization. *Psychological Bulletin*, 2012.

[33] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 2006.

[34] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical Saliency Detection. In *Proc. of CVPR*, 2013.

[35] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency Detection via Graph-based Manifold Ranking. In *Proc. of CVPR*, 2013.

Publication [2]

Germán Martín García and Simone Frintrop. A computational framework for attentional 3D object detection. In *Proceedings of the Annual Conference of the Cognitive Science Society*, Berlin, Germany, 2013.

A Computational Framework for Attentional 3D Object Detection

Germán Martín García and **Simone Frintrop**
{martin,frintrop}@iai.uni-bonn.de
Institute of Computer Science III, Universität Bonn, 53117 Bonn, Germany

Abstract

We present a computational framework for the detection of unknown objects in a 3D environment. It is based on a visual attention system that detects proto-objects which are improved by iterative segmentation steps. At the same time a 3D scene model is built from measurements of a depth camera. The detected proto-objects are projected into the 3D scene, resulting in 3D object models which are incrementally updated. Finally, environment- and object-based inhibition of return enables to withdraw the attention from one object and switch to the next. We show that the system works well in cluttered natural scenes and can find and segment objects without prior knowledge.

INTRODUCTION

Object detection is one of the tasks which are easy to solve for humans but hard for machines. Especially unsupervised object detection, i.e., finding all objects in a scene without previous learning, is largely unsolved in machine vision.[1] However, a system that is able to localize unknown objects in unknown environments is tremendously useful for robotics. For example, a future robot that shall assist in a household must be able to operate autonomously in a new house and is permanently faced with new, unknown objects. Since humans are able to solve such tasks easily, a promising approach for technical systems is to mimic the human visual system.[2]

In humans as in machines, one of the challenges is to deal with the huge amount of perceptual input. Despite the parallelity of the brain, its capacity is not sufficient to deal with all sensory data in detail and a selection has to take place. Neisser (1967) was the first who proposed a two-stage processing of perception that solves this task: first, a pre-attentive process selects regions of interest in parallel, and, second, an attentive process investigates these regions sequentially in more detail. This view has since then widely spread and many psychological theories and models build upon this dichotomy (e.g. Treisman & Gelade, 1980; Wolfe, 1994). Rensink (2000) has further developed this idea with his coherence theory of attention. It states that the pre-attentive processing determines structures, which he calls proto-objects, that describe the local scene structure of a spatially limited region. After that, focused attention selects a small number of proto-objects which form a coherence field representing a specific object.

Here, we present a computational framework that follows Rensink's idea of proto-objects as pre-processing step for object detection. Our approach generates proto-objects with a bottom-up visual attention system (Klein & Frintrop, 2012) and improves their shape by iterative segmentation steps. In contrast to other attention models, we operate on 3D data from a depth camera and are thus able to obtain 3D object models in space, which are incrementally updated by integrating new perceptual data.

In computational systems based on bottom-up visual attention, the focus of attention is directed to the most salient region in the scene. In order to scan the whole scene, this requires a way to withdraw attention from that region and switch to the next. In human vision, this is performed by inhibition of return mechanisms (IOR) that inhibit the currently attended region (Tipper et al., 1994).

In most computational systems, IOR is implemented by zeroing values in the saliency map (Itti et al., 1998). This is sufficient in static images, but when acting in a 3D world, the correspondence between spatial locations and image regions is required. This affects also the IOR mechanism, since when the perspective of the observer changes or objects are moving, inhibition has to move with them, preventing attention to re-visit the objects directly. This motivates the use of a 3D map that grounds the perceptions in space and enables to maintain a coherent IOR representation over space and time. Corresponding to human vision (Tipper et al., 1994), our IOR mechanism is both object- and environment-based.

The contributions of this paper are threefold. First, instead of operating on 2D images, we perform attention-based object detection on 3D data; this enables us to situate the attention system in a 3D environment, resulting in a coherent representation of objects over time. Secondly, it allows for performing not only an environment-based but also an object-based inhibition of return mechanism that operates in space and time. Finally, the use of salient blobs instead of only fixation points for initializing the segmentation process lets us bound the amount of perceptual data to be processed.

Related Work

Many computational attention systems have been built during the last two decades, first for the purpose of mimicking and understanding the human visual system (survey in Heinke & Humphreys, 2004), and second to improve technical systems in terms of speed and quality (survey in Frintrop et al., 2010). The general structure of attention systems is based on psychological models such as the Feature Integration Theory (Treisman & Gelade, 1980) and states that features are computed in parallel before they are fused to a saliency map.

One component of attention systems is the inhibition of return mechanism. While IOR is simple on static images, image sequences introduce the challenge of establishing correspon-

[1] The winner of the latest Semantic Robot Vision Challenge (http://www.semantic-robot-vision-challenge.org) was only able to detect 13 out of 20 objects (Meger et al., 2010), although in this challenge, the target objects were known in advance.

[2] However, note that our intention is to obtain an improved technical system rather than to mimic the HVS as closely as possible.

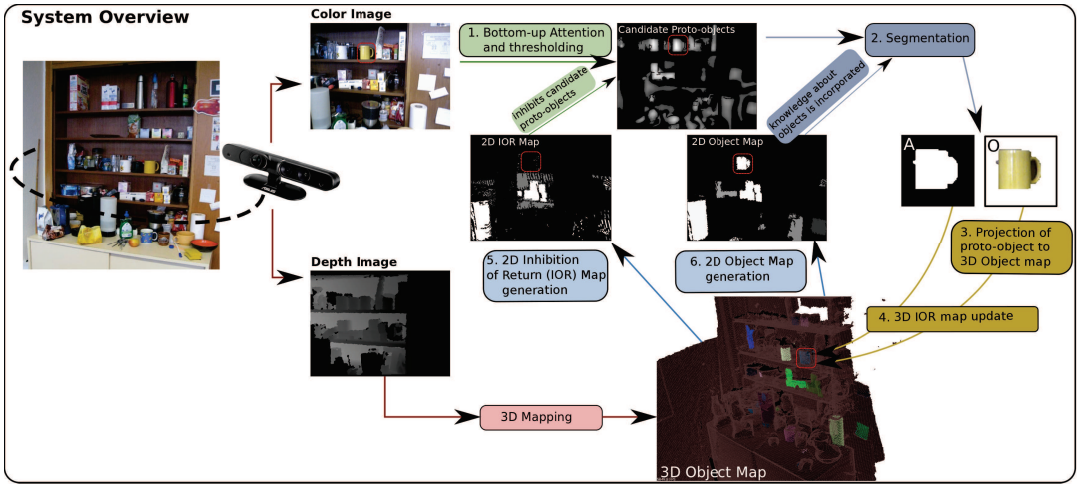


Figure 1: System Overview. The RGB-D camera provides color and depth streams that are processed to obtain proto-objects and a 3D representation of the scene. Here, one proto-object is fixated (1), segmented (2), and projected to the 3D scene (3). The inhibition (5) did not yet take place.

dences between objects over time. In this context, Backer et al. (2001) perform object-centered IOR. However, their approach operates on simple artificially rendered scenes instead of real world data and on 2D images instead of 3D data as we do. Additionally, we combine object-centered and environment-centered IOR to enable both types of inhibition.

Walther and Koch (2006) use an attention system to obtain saliency maps and generate proto-objects inside this map by thresholding. Unsupervised object detection was also tackled by Kootstra and Kragic (2011) who produce saliency maps with a symmetry-based attention system. They use the most salient points as hypothetical centers of objects; these are then provided as seeds to the segmentation process. The figural goodness of the segmentations is evaluated by Gestalt principles. In a robotics context, Meger et al. (2010) search for objects with the mobile robot "Curious George". The robot used a peripheral vision system to identify object candidates with help of a visual attention module. Then, close-up views of these candidates were recorded with a foveal vision system and investigated by a recognition module to identify the object.

General Structure

A general overview of the system is depicted in Figure 1. We acquire data with a depth camera that provides color as well as depth information, and is moved around the scene to obtain different viewpoints. The color and the depth information are investigated in two separate processing streams. The color stream determines proto-objects with help of a bottom-up visual attention system (Fig. 1, top), while the depth stream generates a 3D map of the scene (Fig. 1, bottom). The two streams are combined by projecting the proto-objects into the 3D scene. This results in 3D object models that are incrementally updated when new camera frames are available.

The system operates in two behaviors: the *saccade* behavior and the *fixate* behavior. When the system starts, it first finds the most salient proto-object (1. in Fig. 1), which is then attended for several frames (fixate behavior), allowing other modules to improve the shape of the attended proto-object by segmentation (2.) and project it to the 3D scene (3.). After fixating an object for a while, the saccade behavior takes over to determine the next focus of attention. This is enabled by object-based and environment-based inhibition of return mechanisms (4.), that inhibit the region of the segmented object O and the surrounding region A. To maintain a coherent inhibition of return representation, even when moving the camera, the inhibition values are stored within the 3D map data. From its 3D representation, the data can be projected to produce a 2D IOR map (5.), that is used for inhibiting proto-objects in the saliency map. When the attended object is inhibited, a saccade to the next salient proto-object is generated.

Proto-Object Detection

We perform object detection in two steps: first, we detect proto-objects in each frame with a visual attention system and second, the extend of the proto-objects is improved by a segmentation step.

Attention System: Generation of Proto-Objects

The first step of object detection is the generation of proto-objects with a visual attention system that mimics the pre-attentive processing stage of the human visual system. Such systems usually investigate several feature channels such as color and orientation in parallel and finally fuse the resulting conspicuities in a single saliency map (Frintrop et al., 2010). The peaks in the saliency map can be interpreted as proto-objects (e.g. Walther & Koch, 2006). While in human attention, top-down factors also play an important role, such information is not always available in robotics. Therefore, we compute here only the bottom-up attention.

Figure 2: Top left to bottom right: original RGB image; its corresponding saliency map SM; saliency map after adaptive thresholding SM'; the SM'' map after the final thresholding.

In this work, we use the CoDi system to compute saliency maps (Klein & Frintrop, 2012). The structure follows the standard architecture of Itti et al. (1998), consisting of intensity, color, and orientation feature channels which belong to the most important features in the human visual system (Wolfe & Horowitz, 2004). In contrast to other saliency systems, the center-surround contrast is computed with respect to feature distributions; these are approximated by Normal distributions and their distance is quickly computed by the W_2-distance (Wasserstein metric based on the Euclidean norm).

To allow the detection of arbitrarily sized salient regions, we perform the computations on 8 different scales. The color channel consists of a red-green and a blue-yellow channel, following the opponent-process theory of human color vision (Hurvich & Jameson, 1957). The orientation channel computes center surround differences of Gabor filters of four different orientations: 0°, 45°, 90°, 135°. The saliency map SM is the result of fusing the color and orientation channels.

To generate the image blobs that correspond to proto-objects, two thresholding operations are performed: first an adaptive thresholding using a Gaussian kernel[3]

$$SM'(x,y) = \begin{cases} SM(x,y) & : SM(x,y) > T(x,y) \\ 0 & : \text{otherwise} \end{cases} \quad (1)$$

where $T(x,y)$ is the weighted mean of the neighborhood of (x,y). Finally, a binary thresholding is performed on SM' at a percentage of the global maximum saliency value MAX:

$$SM''(x,y) = \begin{cases} SM'(x,y) & : SM'(x,y) > 0.3 \times MAX \\ 0 & : \text{otherwise} \end{cases} \quad (2)$$

Fig. 2 shows the saliency map SM and the thresholded maps SM' and SM'' for an example image. On SM'' we find the connected components (proto-objects) and compute their average saliency $\overline{sal}$. This method provides us with salient blobs instead of only fixation points which determines the center of fixation as well as the size of the region to use for further investigation. Too small or too big blobs are discarded. If information for the inhibition of objects is already available in terms of a 2D IOR map I (see below), it is used to inhibit already visited regions. This is done by computing the overlap o between each blob and I. Finally, the proto-object with the highest value $\overline{sal} * (1-o)$ is attended.

Thus, the computational attention system fulfills its two main purposes: first, it directs attention to a region of interest and, second, it bounds the amount of perceptual data to be processed afterwards while ignoring the rest.

Improving Proto-Objects by Segmentation

After finding proto-objects, we improve their shape by a segmentation step that bundles parts of the image data. This has a similar effect as grouping mechanisms in human perception that facilitate figure-ground segregation (Wagemans et al., 2012). Such segmentation steps are likely to exist at all levels of human visual processing (Scholl, 2001).

Here, we use the approved GrabCut segmentation (Rother et al., 2004) that was originally proposed for segmenting objects in images with help of user interaction. It takes a rectangle as input, as well as an initialization of pixels with their likelihoods of being object or background. Segmentation is based on the color similarity of neighboring pixels, thus regarding two of the most important factors of perceptual grouping (similarity and proximity). GrabCut performs foreground/background segmentation by iteratively minimizing an energy function. The energy function measures how different each pixel is from the foreground/background model to which it is assigned, as well as from its direct neighbors. It penalizes pixels different from the foreground model to be labeled as foreground as well as labeling pixels as foreground when all its neighbors are background.

The rectangle required for initialization is determined automatically with help of the proto-objects and the information about already detected objects. The pixels of the currently attended proto-object are merged with the information of this object from previous frames (if available). This information can be gathered from the 3D scene representation raycasted to a 2D object map that will be explained later on (cf. Fig. 1). Now, the smallest rectangle r containing all merged pixels is determined (cf. Fig. 4, top), as well as a rectangle r', obtained by expanding r's dimensions by 10%.

For initializing segmentation, GrabCut requires four possible pixel likelihood values: FG (foreground), BG (background), PR_FG (probably foreground) and PR_BG (probably background). These are obtained by defining three intervals between 0 and the saliency maximum max in R:

$$L(x,y) = \begin{cases} FG & : SM''(x,y) \in [v_3, max], (x,y) \in R \\ PR\_FG & : SM''(x,y) \in [v_1, v_3], (x,y) \in R \\ PR\_BG & : SM''(x,y) \in [0, v_1], (x,y) \in R \\ BG & : (x,y) \in R' \setminus R, \end{cases} \quad (3)$$

where R and R' are the sets of pixels contained in rectangles r

[3] We use the adaptiveThreshold function of the OpenCV library: http://opencv.org/

Figure 3: Top: a book as example object. Middle: initialization of GrabCut, the grayscale values correspond to the four possible likelihoods FG (white), PR_FG (light gray), PR_BG (dark gray), and BG (black). Bottom: the segmentation result.

and r' respectively, and $v_i = i \cdot \frac{max}{4}$ defines each of the interval limits. The likelihoods are corrected by incorporating the information about the current and all other objects. This is done by setting the pixels that correspond to the current object in the 2D object map as PR_FG, and the ones corresponding to other objects as BG. An example of the initialization values is displayed in Fig. 3. Five iterations of GrabCut produce a binary object mask O for the attended blob.

Creating a 3D Scene Map

While the color image was used to detect proto-objects, the depth data is used to build a 3D map of the scene. This is done with the KinectFusion algorithm[4] (Newcombe et al., 2011), which builds a 3D map of the environment by integrating multiple range scans from a moving depth camera such as Kinect. It performs two processes in parallel, namely, tracking of the pose of the camera, and registration of the depth scans into a complete scene representation. The result is a 3D scene map consisting of voxels (cf. Fig. 5, right).

To represent the scene at time k, a global truncated signed distance function (TSDF) $S_k(p) \rightarrow [F_k(p), W_k(p)]$ is computed by integrating the depth measurements, where $p \in \mathbb{R}^3$ is a point in space, $F_k(p)$ the TSDF value and $W_k(p)$ a weight. The function is discretized in a voxel grid; its zero crossings are points that lie on surfaces. Thus, from the voxel grid, a point cloud can be rendered by choosing the voxels containing zero TSDF values.

Extended 3D Scene Map

Our system stores all object information in a 3D structure. It is an extended version of the voxel grid defined in the previous section. For convenience, we will refer to the new voxel grid as $S_k[c]$, where voxel $c = (x, y, z)$, $x, y, z \in [1..Vol]$ and Vol is the number of cells into which the grid is discretized. We extend the S_k function to

$$S_k[c] \rightarrow \{F_k[c], W_k[c], L_k[c], LW_k[c], I_k[c], IW_k[c]\}, \quad (4)$$

where $F_k[c]$ and $W_k[c]$ are the values defined before, $L_k[c], LW_k[c]$ are variables that contain object label information, and $I_k[c], IW_k[c]$ are IOR related and will be explained later on. The 3D information from the voxel grid can at any time be projected to produce a 2D image containing IOR or object label information (details follow).[5]

Generating 3D Object Models

Now, the 3D object models are created and updated using the binary object mask O from the segmentation stage. Let us denote the function that maps pixels in the image to voxels in the grid as map : $p \in \mathbb{Z}^2, T \in \mathbb{R}^4, D \in \mathbb{Z}^{m \times n} \rightarrow c \in \mathbb{Z}^3$, where p is a pixel, T the camera pose, and D a depth image with dimensions $m \times n$. The pixels in the object mask are mapped to their corresponding voxels in the grid:

$$\text{map}(O, T_{g,k}, D_k) \rightarrow O' = \{c : c \in \mathbb{Z}^3\}, \quad (5)$$

where g is the global frame of reference.

Now it has to be decided which label to assign to the voxels in O'. There are two mechanisms corresponding to the fixate and saccade behaviors of the system. During the fixate behavior, the label of the currently attended object is used. When the saccade behavior selects a new focus of attention, it performs as follows. On the set of voxels O' corresponding to the new proto-object, we extract the current labels > 0: $Lab = \{L_k[c] : L_k[c] > 0, c \in O'\}$. We find the most frequently occurring label l in Lab. If less than 5% of the voxels are labeled, we assign l a new value corresponding to a newly detected object. The value of l is now used to update the voxels contained in O'. This simple scheme lets us integrate the overlapping segmentations of different views of the same objects in the 3D map.

To be flexible against wrong segmentations or overlapping objects, weights are assigned to the labels. Every time the same label is assigned to a voxel, its label weight LW_k is incremented. If a voxel is updated with a different label, the weight is decremented. Eventually it could reach 0, resulting in an unlabeled voxel. This mechanism lets us incrementally build the object representations with a certain tolerance to failure; furthermore, by thresholding the label weight we can specify the degree of confidence in our object representations that we want for rendering the labeled point cloud. In our experiments, we used $LW_k = 5$, meaning that a voxel has to be assigned to a specific object at least 5 times to be considered for this object.

3D IOR Map

After fixating an object for several frames, the object must be inhibited to enable the next saccade. To allow a coherent IOR over time, we store the inhibition values within the 3D voxel grid: $I_k[c]$ is a binary flag denoting whether that voxel shall be inhibited and $IW_k[c]$ is a weight that determines how long the effect shall take place. Having IOR information in 3D coordinates lets us generate 2D IOR maps I_k from the required camera poses throughout the sequence.

[4] We use the open source implementation available in the Point Cloud Library (http://pointclouds.org/)

[5] In (Newcombe et al., 2011), the $TSDF$ function is raycasted, given a camera pose, to generate a depth map prediction. Using this method in our extended $TSDF$ function means we can generate 2D IOR or object label maps for every new pose of the camera.

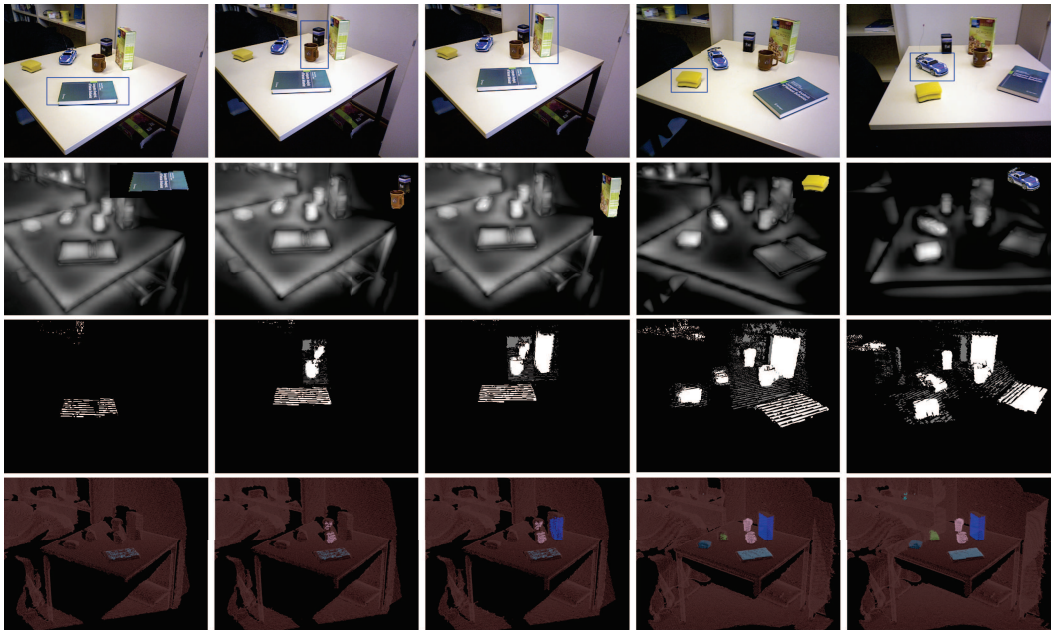


Figure 4: Table Top sequence at different points in time (columns). From top to bottom: (i) image of the scene with currently attended object (blue rectangle); (ii) the saliency map and the segmented part from the currently attended object; (iii) inhibition of return maps; white: object-based IOR, gray: environment-based IOR; (iv) the 3D scene map including detected objects

According to human vision, we use two types of IOR mechanisms: *environment-based* and *object-based* IOR (Tipper et al., 1994). The latter comes intuitively from the segmented object mask O. The environment-based IOR is initialized by the regions close to the object but not on the object, i.e., from a so called attended mask $A = R' \setminus O$. The two masks are mapped as in the previous section to obtain their respective voxel sets O' and A'. For every voxel c in O' and A', its weight $IW_k[c]$ is incremented. When it reaches a certain threshold, the IOR flag $I_k[c]$ is activated. The weight of all not considered voxels is decremented. If a weight eventually reaches 0, the IOR flag is reset to 0 as well.

Evaluation

To evaluate our system we recorded two video sequences in an office environment with an RGB-D camera that provides depth as well as color information. The first sequence shows a setting of objects on a table top (cf. Fig. 4). The complexity of this setting corresponds to the complexity of scenes in current state of the art benchmarks and papers on unsupervised object detection in machine vision (cf. Meger et al., 2010; Kootstra & Kragic, 2011). However, the real world can be much more complex. Therefore, we recorded a second sequence, that shows a very cluttered setting (Fig. 5). Both settings were recorded turning the camera so that the scene was observed from different viewpoints (cf. Fig. 1).[6]

Fig. 4 illustrates several steps of our approach at different time points. First, the book was attended (fixate behavior). After fixating it for several frames, the region is inhibited (3rd row) and the attention switches to the next proto-object (saccade behavior). This proto-object consists of two real objects (cup and tea box) since these objects are overlapping from this point of view and have similar saliency. The procedure continues, until all objects on the table have been detected.

For the second sequence, we present for space reasons only the resulting 3D map with detected objects (Fig. 5, right). Here, the approach finds 19 objects after 438 frames ($\sim$13 sec). More objects could be found by longer observing the sequence, but some would be missed, e.g., due to high similarity to the background, and no current computer vision system would be able to find all objects without pre-knowledge in such a complex setting. Note that several of the "objects" still have proto-object characteristics, meaning that they show parts of objects (handle of dishwashing brush (6), bottom of coffee machine (18)) or clusters of objects (tea boxes (11)). Such semantic ambiguities could only be resolved by a recognition system that investigates the attended regions in more detail, or by a robot that interacts with objects and decides on objectness depending on the connectivity of object parts.

To evaluate our system quantitatively, we measure how precisely the detected objects were segmented. For this, the points in the 3D map corresponding to objects were manually labeled to serve as ground truth. We generally denote the ground truth of each object as G, and the 3D points of the object detected by our system as S. We measure the precision p and recall r of the detected objects with respect to the ground truth as $p = (S \cap G)/S$, and $r = (S \cap G)/G$. The values are shown in Tab. 1 and Fig. 5. It can be seen that the

[6] Videos of the complete sequences as well as the resulting 3D representations can be found at http://vimeo.com/cogbonn/

object	1	2	3	4	5	6	7	8	9	10
precision	93	69	92	99	62	52	90	60	100	99
recall	40	43	28	40	61	28	36	36	21	37
object	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	
precision	23	90	83	98	91	99	100	89	100	
recall	47	40	35	39	31	30	8	1	3	

Figure 5: Coffee Machine sequence. Left: color image. Right: 3D scene map with detected objects (numbers denote labels). Bottom: precision/recall values in %

object	Book	Cup	Cereals Box	Car	Sponge	Pot
precision	99	55	98	99	97	94
recall	64	62	53	54	56	9

Table 1: Table Top sequence: precision/recall values in % (cf. Fig. 4).

precision values are mostly very good (more than 90% for 17 out of 25 objects), that means that only few voxels were accidentally assigned to an object. A bad value usually indicates that a cluster of objects was detected and compared with separate objects in the ground truth (e.g. objects 5 and 11). The recall values are lower, meaning that often not all of the voxels that belong to an object were detected. In the future, this can be improved by additional post-processing steps based on grouping mechanisms for figure-ground segregation.

Conclusion

We have presented a flexible framework for the detection of unknown objects in a 3D scene. Unlike other approaches, the system uses depth values additionally to a color image of a scene and is thus able to generate 3D object models that are incrementally updated when new information is available. All perceptual data is spatially grounded and thus consistent over different viewpoints. The results show that the algorithm is able to detect many objects in scenes with high clutter, without using any prior knowledge about the type of objects.

Applying attention mechanisms in space and time introduces new challenges, for example the question of how and when to switch attention between salient regions. We introduced an environment- and object-based inhibition of return mechanism that addresses this problem by using the information from the 3D environment and object models for inhibition.

References

Backer, G., Mertsching, B., & Bollmann, M. (2001). Data- and model-driven gaze control for an active-vision system. *IEEE Trans. on PAMI*, *23(12)*.

Frintrop, S., Rome, E., & Christensen, H. I. (2010). Computational visual attention systems and their cognitive foundations: A survey. *ACM Trans. on Applied Perception*, *7*(1).

Heinke, D., & Humphreys, G. W. (2004). Computational models of visual selective attention. A review. In *Connectionist models in psychology.* Psychology Press.

Hurvich, L., & Jameson, D. (1957). An opponent-process theory of color vision. *Psychological review*, *64*(6).

Itti, L., Koch, C., & Niebur, E. (1998, Nov). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on PAMI*, *20*(11).

Klein, D. A., & Frintrop, S. (2012). Salient pattern detection using W2 on multivariate normal distributions. In *Proc. of DAGM-OAGM.* Springer.

Kootstra, G., & Kragic, D. (2011). Fast and bottom-up object detection, segmentation, and evaluation using Gestalt principles. In *IEEE Int'l Conf. on Robotics and Automation.*

Meger, D., Muja, M., Helmer, S., Gupta, A., Gamroth, C., Hoffman, T., et al. (2010). Curious george: An integrated visual search platform. In *Canadian conference on computer and robot vision.*

Neisser, U. (1967). *Cognitive psychology*. New York: Appleton-Century-Crofts.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., et al. (2011). KinectFusion: Real-time dense surface mapping and tracking. In *Proc. of IEEE Int'l Symposium on Mixed and Augmented Reality.*

Rensink, R. A. (2000). The dynamic representation of scenes. *Visual Cognition*, *7*, 17-42.

Rother, C., Kolmogorov, V., & Blake, A. (2004). GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, *23*, 309-314.

Scholl, B. J. (2001). Objects and attention: the state of the art. *Cognition*, *80*, 1-46.

Tipper, S. P., Weaver, B., Jerreat, L. M., & Burak, A. L. (1994). Object-based and environment-based inhibition of return of visual attention. *J. of Experimental Psychology*.

Treisman, A. M., & Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, *12*, 97-136.

Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., et al. (2012). A century of Gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization. *Psychological Bulletin*.

Walther, D., & Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*, *19*(9), 1395 - 1407.

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, *1*(2).

Wolfe, J. M., & Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, *5*, 1-7.

Publication [3]

Germán Martín García, Simone Frintrop, and Armin B. Cremers. Attention-based detection of unknown objects in a situated vision framework. *German Journal of Artificial Intelligence, Springer*, 27, 2013.

Noname manuscript No.
(will be inserted by the editor)

Attention-based Detection of Unknown Objects in a Situated Vision Framework

Germán Martín García · Simone Frintrop · Armin B. Cremers

Received: date / Accepted: date

Abstract We present an attention-based approach for the detection of unknown objects in a 3D environment. The ability of addressing individual objects in the environment without having previous knowledge about their properties or their identity is one important requirement of the Situated Vision theory. Based on saliency maps, our attention system determines the regions where objects are likely to be found; these are the proto-objects whose extent is refined by a 2D segmentation step. At the same time a 3D scene model is built from measurements of a depth camera. The detected objects are projected into the 3D scene, resulting in 3D object models which are incrementally updated. We show the validity of our approach in an RGB-D sequence recorded in an office environment.

Keywords object detection · computational attention systems · bottom-up

1 Introduction

Object detection belongs to the key competences of an autonomous agent that acts in an unknown environment and is supposed to fulfill service tasks for humans. Grasping, manipulating, or using objects in any way involves the ability to first detect and localize the objects in the scene. In order to be flexible and independent of human supervision, it is important that the agent is able to detect objects without previously knowing what types of objects might occur or how they might look like. This is a chicken and egg problem. How to detect objects without knowing how they look like? To tackle this task, we use the Situated Vision theory in its different senses from different disciplines as a theoretical background.

In the artificial intelligence and robotics community, a *situated* agent is an agent that is embedded in its environment and aware of the situation that it is acting in. Especially in dynamic and changing environment, this awareness is essential to cope with the complexity of the world and to adapt to new challenges. Thus, an industrial robotic arm performing the task of assembling cars would not be situated, whereas a service robot at a home environment expected to complete domestic tasks would. In this context, Schlemmer [11] gives his view of the Situated Vision theory: vision should be treated as a function that is situated not only in the environment, but also in a broader cognitive architecture, to which it serves for a specific task.

In the area of psychology, Pylyshyn postulates that a theory of Situated Vision needs to establish direct connections between elements in the visual field and visual representations in the brain [8]. This requires a mechanism that visually individuates the elements in the environment before their properties or categories are known. Furthermore, Pylyshyn assumes that visual representations of objects are constructed incrementally by continuously adding new information to the current representation.

In the present work, we attempt to bring together both views of the Situated Vision paradigm. We developed a visual system that is able to identify individual objects in the environment regardless of their properties and without previous knowledge of them; furthermore, it is situated in its environment and uses all information available up to that moment. The system creates 3D object representations that could be used by other

Germán Martín García (Γ) · Simone Frintrop ·
Armin B. Cremers
Institute of Computer Science III, Universität Bonn, 53117 Bonn, Germany
E-mail: martin@iai.uni-bonn.de

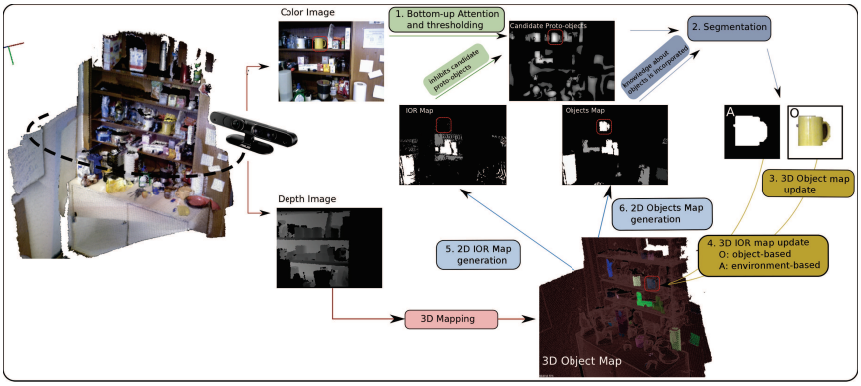


Fig. 1: System Overview. The RGB-D camera provides color and depth streams that are processed to obtain proto-objects and a 3D representation of the scene. Here, one proto-object is fixated (1), segmented (2), and projected to the 3D scene (3). The inhibition (5) did not yet take place.

Fig. 2: Top left: original RGB image. Top right: its corresponding saliency map SM. Bottom left: saliency map after adaptive thresholding SM'. Bottom right: map SM'' after the final thresholding operation.

cognitive mechanisms to inspect their properties and categories, such as object classifiers[1].

2 System Description

A general overview of the system is depicted in Figure 1. We acquire data with a depth camera that provides color as well as depth information, and is moved around the scene to obtain different viewpoints. The color and the depth information are investigated in two separate processing streams. The color stream determines proto-objects with help of a bottom-up visual attention system (Fig. 1, top), while the depth stream generates a 3D map of the scene (Fig. 1, bottom). The two streams are combined by projecting the proto-objects into the 3D scene. This results in 3D object models that are incrementally updated when new camera frames are available.

The system operates in two behaviors: the *saccade* behavior and the *fixate* behavior. When the system starts, it first finds the most salient proto-object (1. in Fig. 1), which is then attended for several frames (fixate behavior), allowing other modules to improve the shape of the attended proto-object by segmentation (2.) and project it to the 3D scene (3.). After fixating an object for a while, the saccade behavior takes over to determine the next focus of attention. This is enabled by object-based and environment-based inhibition of return mechanisms (4.), that inhibit the region of the segmented object O and the surrounding region A. To maintain a coherent inhibition of return representation, even when moving the camera, the inhibition values are stored within the 3D map data. From its 3D representation, the data can be projected to produce a 2D IOR map (5.), that is used for inhibiting proto-objects in the saliency map. When the attended object is inhibited, a saccade to the next salient proto-object is generated.

3 Attention Mechanism

The entry point of our system is the attention module. It follows the idea of Rensink [9] that a pre-attentive processing stage determines structures, which he calls proto-objects, that describe the local scene structure of a spatially limited region. Focused attention selects a small number of proto-objects which form a coherence field representing a specific object. In a similar way to [14], our implementation determines proto-objects by thresholding a saliency map and selecting the surviving structures, or blobs. Using blobs as the focus of attention, as opposed to focal points, has the benefit that the image region to be further processed —e.g. for segmentation— is bounded. Thus, the computational attention system fulfills its two main purposes: first, it directs attention to a region of interest and, second, it bounds the amount of perceptual data to be processed afterwards while ignoring the rest.

We perform object detection in two steps: first, we detect proto-objects in each frame with a visual attention system and second, the extend of the proto-objects is improved by a segmentation step.

[1] This work is part of DFG DACH project FR 2598/5-1 called *Situated Vision to Perceive Object Shape and Affordances*, in cooperation with TU Wien, RTWH Aachen and IDIAP

3.1 Attention System: Generation of Proto-Objects

The first step of object detection is the generation of proto-objects with a visual attention system. Such systems usually investigate several feature channels such as color and orientation in parallel and finally fuse the resulting conspicuities in a single saliency map [2]. The peaks in the saliency map can be interpreted as proto-objects [14].

We make use of the approach of Klein and Frintrop [4] to compute saliency maps. The main idea is to represent the distribution of feature statistics in a center and in a surround area around a pixel by Gaussians and compare them by the W_2-distance (Wasserstein metric based on the Euclidean norm). This method allows a quick computation of saliency maps also for large sizes of the center-surround filter which enables the detection of large proto-objects in a scene. To allow the detection of arbitrarily sized salient regions, we perform the computations on 8 different scales. The feature channels we use to determine the distributions are intensity, color and orientation which belong to the most important feature channels in the human visual system [15]. The saliency map SM is the result of fusing the three channels.

To generate the image blobs that correspond to proto-objects, two thresholding operations are performed: first an adaptive thresholding using a Gaussian kernel[2]

$$SM'(x,y) = \begin{cases} SM(x,y) : SM(x,y) > T(x,y) \\ 0 \quad\quad\quad\quad : \text{otherwise} \end{cases} \tag{1}$$

where $T(x,y)$ is the weighted mean of the neighborhood of (x,y). Finally, a binary thresholding is performed on SM' at a percentage of the global maximum saliency value MAX:

$$SM''(x,y) = \begin{cases} SM'(x,y) : SM'(x,y) > 0.3 \times MAX \\ 0 \quad\quad\quad\quad\quad\quad : \text{otherwise} \end{cases} \tag{2}$$

Fig. 2 shows the saliency map SM and the thresholded maps SM' and SM'' for an example image. On SM'' we find the connected components (proto-objects) and compute their average saliency $\overline{sal}$. Too small or too big blobs are discarded. If information for the inhibition of objects is already available in terms of a 2D IOR map I (see below), it is used to inhibit already visited regions. This is done by computing the overlap o between each blob and I. Finally, the proto-object with the highest value $\overline{sal} * (1-o)$ is attended.

Thus, the computational attention system fulfills its two main purposes: first, it directs attention to a region of interest and, second, it bounds the amount of perceptual data to be processed afterwards while ignoring the rest.

[2] We use the adaptiveThreshold function of the OpenCV library: http://opencv.org/

3.2 Improving Proto-Objects by Segmentation

After finding proto-objects, we improve their shape by a segmentation step with the well-known GrabCut algorithm [10] that was originally proposed for segmenting objects in images with help of user interaction. It takes a rectangle as input, as well as an initialization of pixels with their likelihoods of being object or background, and iteratively minimizes an energy functional that evaluates how well the labeled pixels fit to the foreground/background models, as well as how smooth transitions are from similar neighboring pixels.

The rectangle required for initialization is determined automatically with help of the proto-objects and the information about already detected objects. The pixels of the currently attended proto-object are merged with the information of this object from previous frames (if available). This information can be gathered from the 3D scene representation raycasted to a 2D object map that will be explained later on (cf. Fig. 1). Now, the smallest rectangle r containing all merged pixels is determined (cf. Fig. 4, top), as well as a rectangle r', obtained by expanding r's dimensions by 10%.

The rectangles r and r' are used to determine pixel likelihoods that GrabCut requires for initializing the segmentation. There are four possible initialization values: FG (foreground), BG (background), PR_FG (probably foreground) and PR_BG (probably background). These are obtained by defining three intervals between 0 and the saliency maximum max in R:

$$L(x,y) = \begin{cases} FG \quad\quad : SM''(x,y) \in [v_3, max], (x,y) \in R \\ PR\_FG : SM''(x,y) \in [v_1, v_3], (x,y) \in R \\ PR\_BG : SM''(x,y) \in [0, v_1], (x,y) \in R \\ BG \quad\quad : (x,y) \in R' \setminus R, \end{cases} \tag{3}$$

where R and R' are the sets of pixels contained in rectangles r and r' respectively, and $v_i = i \cdot \frac{max}{4}$ defines each of the interval limits. The likelihoods are corrected by incorporating the information about the current and all other objects. This is done by setting the pixels that correspond to the current object in the 2D object map as PR_FG, and the ones corresponding to other objects as BG. An example of the initialization values is displayed in Fig. 3. Five iterations of GrabCut produce a binary object mask O for the attended blob.

Fig. 3: Top: a book as example object. Middle: initialization of GrabCut, the grayscale values correspond to the four possible likelihoods FG (white), PR_FG (light gray), PR_BG (dark gray), and BG (black). Bottom: the segmentation result.

4 Creating a 3D Scene Map

While the color image was used to detect proto-objects, the depth data is used to build a 3D map of the scene. This is done with the KinectFusion algorithm[3] [7], which builds a 3D map of the environment by integrating multiple range scans from a moving depth camera such as Kinect. It performs two processes in parallel, namely, tracking of the pose of the camera, and registration of the depth scans into a complete scene representation. The result is a 3D scene map consisting of voxels.

To represent the scene at time k, a global truncated signed distance function (TSDF) $S_k(p) \rightarrow [F_k(p), W_k(p)]$ is computed by integrating the depth measurements, where $p \in \mathbb{R}^3$ is a point in space, $F_k(p)$ the TSDF value and $W_k(p)$ a weight. The function is discretized in a voxel grid; its zero crossings are points that lie on surfaces. Thus, from the voxel grid, a point cloud can be rendered by choosing the voxels containing zero TSDF values.

5 Extended 3D Scene Map

Our system stores all object information in a 3D structure. It is an extended version of the voxel grid defined in the previous section. For convenience, we will refer to the new voxel grid as $S_k[c]$, where voxel $c = (x, y, z)$, $x, y, z \in [1..Vol]$ and Vol is the number of cells into which the grid is discretized. We extend the S_k function to

$$S_k[c] \rightarrow \{F_k[c], W_k[c], L_k[c], LW_k[c], I_k[c], IW_k[c]\}, \quad (4)$$

where $F_k[c]$ and $W_k[c]$ are the values defined before, $L_k[c], LW_k[c]$ are variables that contain object label information, and $I_k[c], IW_k[c]$ are IOR related and will be explained later on. The 3D information from the voxel grid can at any time be raycasted to produce a 2D image containing IOR or object label information.[4]

[3] We use the open source implementation available in the Point Cloud Library (http://pointclouds.org/)

[4] In [7], the $TSDF$ function is raycasted, given a camera pose, to generate a depth map prediction. Using this method in our extended $TSDF$ function means we can generate 2D IOR or object label maps for every new pose of the camera.

5.1 Generating 3D Object Models

Now, the 3D object models are created and updated using the binary object mask O from the segmentation stage. Let us denote the function that maps pixels in the image to voxels in the grid as map : $p \in \mathbb{Z}^2, T \in \mathbb{R}^4, D \in \mathbb{Z}^{m \times n} \rightarrow c \in \mathbb{Z}^3$, where p is a pixel, T the camera pose, and D a depth image with dimensions $m \times n$. The pixels in the object mask are mapped to their corresponding voxels in the grid:

$$\text{map}(O, T_{g,k}, D_k) \rightarrow O' = \{c : c \in \mathbb{Z}^3\}, \quad (5)$$

where g is the global frame of reference.

Now it has to be decided which label to assign to the voxels in O'. There are two mechanisms corresponding to the fixate and saccade behaviors of the system. During the fixate behavior, the label of the currently attended object is used. When the saccade behavior selects a new focus of attention, it performs as follows. On the set of voxels O' corresponding to the new proto-object, we extract the current labels > 0: $Lab = \{L_k[c] : L_k[c] > 0, c \in O'\}$. We find the most frequently occurring label l in Lab. If less than 5% of the voxels are labeled, we assign l a new value corresponding to a newly detected object. The value of l is now used to update the voxels contained in O'. This simple scheme lets us integrate the overlapping segmentations of different views of the same objects in the 3D map.

To be flexible against wrong segmentations or overlapping objects, weights are assigned to the labels. Every time the same label is assigned to a voxel, its label weight LW_k is incremented. If a voxel is updated with a different label, the weight is decremented. Eventually it could reach 0, resulting in an unlabeled voxel. This mechanism lets us incrementally build the object representations with a certain tolerance to failure; furthermore, by thresholding the label weight we can specify the degree of confidence in our object representations that we want for rendering the labeled point cloud.

5.2 3D IOR Map

After fixating an object for several frames, the object must be inhibited to enable the next saccade. To allow a coherent IOR over time, we store the inhibition values within the 3D voxel grid: $I_k[c]$ is a binary flag denoting whether that voxel shall be inhibited and $IW_k[c]$ is a weight that determines how long the effect shall take place. Having IOR information in 3D coordinates lets

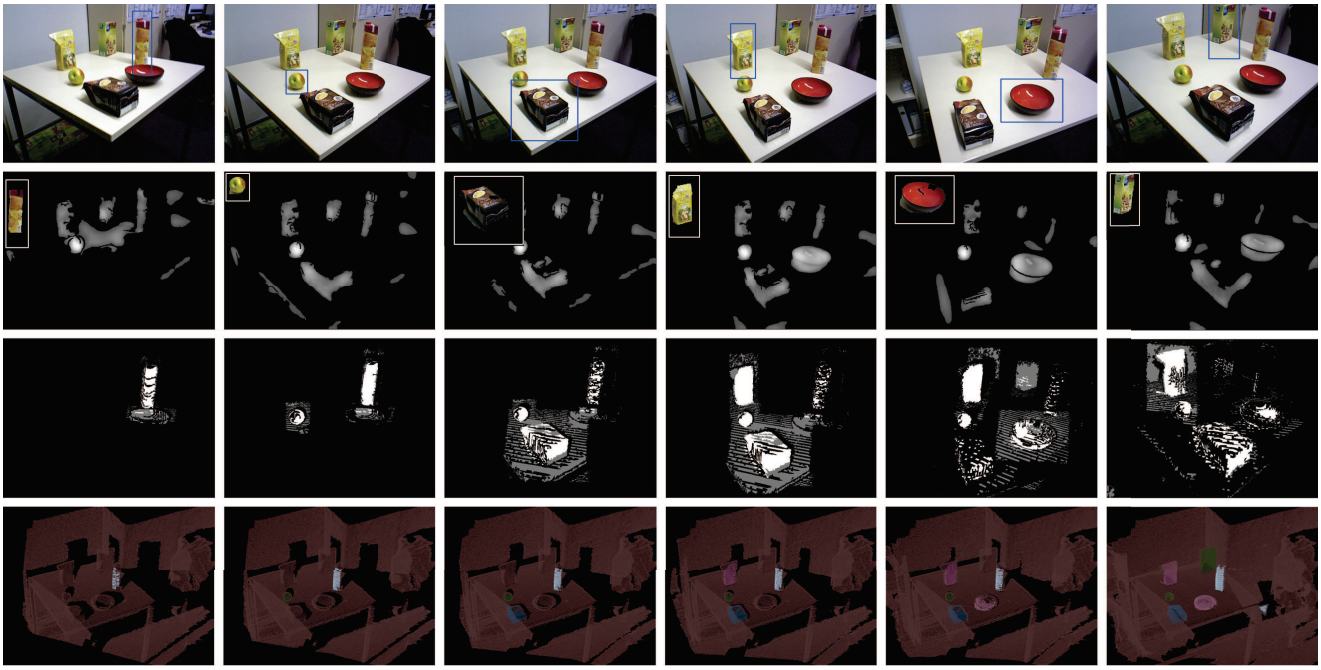


Fig. 4: Table Top sequence at different points in time (columns). From top to bottom: (i) image of the scene with currently attended object (blue rectangle); (ii) the candidate proto-objects and, in the top left corner, the segmentation of the currently attended object; (iii) inhibition of return maps; white: object-based IOR, gray: environment-based IOR; (iv) the 3D scene map including detected objects.

us generate 2D IOR maps I_k from the required camera poses throughout the sequence.

According to human vision, we use two types of IOR mechanisms: *environment-based* and *object-based* IOR. The latter comes intuitively from the segmented object mask O. The environment-based IOR is initialized by the regions close to the object but not on the object, i.e., from a so called attended mask $A = R' \setminus O$. The two masks are mapped as in the previous section to obtain their respective voxel sets O' and A'. For every voxel c in O' and A', its weight $IW_k[c]$ is incremented. When it reaches a certain threshold, the IOR flag $I_k[c]$ is activated. The weight of all not considered voxels is decremented. If a weight eventually reaches 0, the IOR flag is reset to 0 as well.

6 Evaluation

To evaluate the performance of our system we recorded a video sequence in an office environment with an RGB-D camera that provides depth as well as color information. The sequence shows a setting of objects on a table top (cf. Fig. 4). The complexity of this setting corresponds to the complexity of scenes in current state of the art benchmarks and papers on unsupervised object detection in machine vision [6,5]. We have recorded scenes with clutter, however, due to space limitations we can not show the results here.

Fig. 4 illustrates, in each column, several steps of our approach at different time points in time. The first object to be fixated is the orange juice pack. When the attention system goes to pick the next salient blob (displayed in row two), the apple is selected; the blobs that correspond to the previously fixated object are now inhibited. In the last row, the evolution of the object representations can be seen: each detected object has a different color in the map. The same procedure goes on until the end of the sequence where all the objects have been successfully detected.

To evaluate our system quantitatively, we measure how precisely the detected objects were segmented. For this, objects were manually labeled in the final 3D map to serve as ground truth. Now, for each detected object, we measure precision and recall of the points of the detected object with respect to the ground truth. The values are shown in Tab. 1. It can be seen that all the precision values are above 90%, that means that only few voxels were accidentally assigned to an object. The recall values are lower, meaning that often not all of the voxels that belong to an object were detected. In the future, this can be improved by additional post-processing steps based on grouping mechanisms for figure-ground segregation.

object	Juice	Apple	Coff.1	Coff.2	Bowl	Box
prec.	97	99	93	97	95	98
recall	30	40	47	48	53	61

Table 1: Table Top sequence results: precision/recall values in % in the same order of appearance of Fig.4.

7 Conclusion

We have presented a flexible framework for the detection of unknown objects in a 3D scene. Unlike other approaches, the system uses depth values additionally to a color image of a scene and is thus able to generate 3D object models that are incrementally updated when new information is available. The results show that the algorithm is able to detect many objects in scenes with high clutter, without using any prior knowledge about the type of objects.

References

1. Frintrop, S., Rome, E., Christensen, H.I.: Computational visual attention systems and their cognitive foundations: A survey. ACM Trans. on Applied Perception **7**(1) (2010)
2. Klein, D.A., Frintrop, S.: Salient pattern detection using W2 on multivariate normal distributions. In: Proc. of DAGM-OAGM. Springer (2012)
3. Kootstra, G., Kragic, D.: Fast and bottom-up object detection, segmentation, and evaluation using Gestalt principles. In: IEEE Int'l Conf. on Robotics and Automation (2011)
4. Meger, D., Muja, M., Helmer, S., Gupta, A., Gamroth, C., Hoffman, T., Baumann, M., Southey, T., Fazli, P., Wohlkinger, W., Viswanathan, P., Little, J.J., Lowe, D.G., Orwell, J.: Curious george: An integrated visual search platform. In: Canadian Conference on Computer and Robot Vision (2010)
5. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: Proc. of IEEE Int'l Symposium on Mixed and Augmented Reality, ISMAR '11 (2011)
6. Pylyshyn, Z.W.: Visual indexes, preconceptual objects, and situated vision. Cognition **80**(1-2), 127–158 (2001)
7. Rensink, R.A.: Seeing, sensing and scrutinizing. Vision Research **40**, 1469–1487 (2000)
8. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**, 309–314 (2004)
9. Schlemmer, M.: Getting past passive vision - on the use of an ontology for situated perception in robots. Ph.D. thesis, Faculty of Electrical Engineering and Information Technology, Vienna University of Technology (2009)
10. Walther, D., Koch, C.: Modeling attention to salient proto-objects. Neural Networks **19**(9), 1395 – 1407 (2006)
11. Wolfe, J.M., Horowitz, T.S.: What attributes guide the deployment of visual attention and how do they do it? Nature Reviews Neuroscience **5**, 1–7 (2004)

Germán Martín García received his Dipl-Ing. degree in Computer Engineering in the Universidad Autónoma de Madrid in 2008. In 2012 received a MSc. Degree in Computer Science in the University of Bonn, where he is currently enrolled as a PhD. student. His interest is in the field of visual attention and unknown object detection.

Simone Frintrop is a senior researcher at the Computer Science department at the University of Bonn and is currently heading the Cognitive Vision Group. She received a doctoral degree from the university of Bonn in 2005. Her research interests include computational visual attention, cognitive computer vision, and robot vision.

Armin B. Cremers is Professor of Computer Science at the University of Bonn. He holds a doctoral degree in mathematics from the University of Karlruhe. His former faculty appointments were in Karlsruhe, Los Angeles and Dortmund. Since 2002 he is Director of the Bonn-Aachen International Center for Information Technology (B-IT) in Bonn. His research areas are software engineering, information systems, and artificial intelligence. Currently, Prof. Cremers also holds a Visiting Professorship at the Institute of Pattern Recognition and Artificial Intelligence of HUST in Wuhan.

Publication [4]

Simone Frintrop. Towards attentive robots. *PALADYN Journal of Behavioral Robotics, Springer*, 2(2), 2011.

Paladyn. Journal of Behavioral Robotics manuscript No.
(will be inserted by the editor)

Towards Attentive Robots

Simone Frintrop[1]

Received: date / Accepted: date

Abstract This paper introduces *Attentive Robots*: robots that attend to the parts of their sensory input that are currently of most potential interest. The concept of selecting the most promising parts is adopted from human perception where selective attention allocates the brain resources to the most interesting parts of the sensory input. We give an overview of current approaches to integrate computational attention into robotic systems, with a focus on biologically-inspired visual attention methods. Example applications range from localization with salient landmarks over object manipulation to the design of social robots. A brief outlook gives an impression of how future ways to obtain attentive robots might look like.

Keywords

1 Introduction

Imagine you bought a new home robot, Dobby, at some point in the future. Dobby is supposed to do most of the housework while you are at work or are meeting with friends. It shall receive and unpack the groceries that come from the supermarket, do the laundry, and tidy up the mess that the kids made when playing in the living room. At every moment, Dobby has to process a large amount of sensory input and the possibilities of what to do first easily become overwhelming. Since robots have limited processing power as well as physical limitations such as a limited number of sensors, arms, etc., a selection mechanism that determines where to concentrate the resources is of high interest. In humans, the mechanism that determines which part of the sensory input is currently most promising is called selective attention [Pashler, 1997]. Accordingly, we call robots that attend to the most promising part of their sensor data "Attentive Robots" (cf. Fig. 1).

The term "attention" is used in many contexts and many definitions exist. It is a term of common language (William James: "Everyone knows what attention is..."

S. Frintrop
Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität,
53117 Bonn, Germany.
E-mail: frintrop@iai.uni-bonn.de

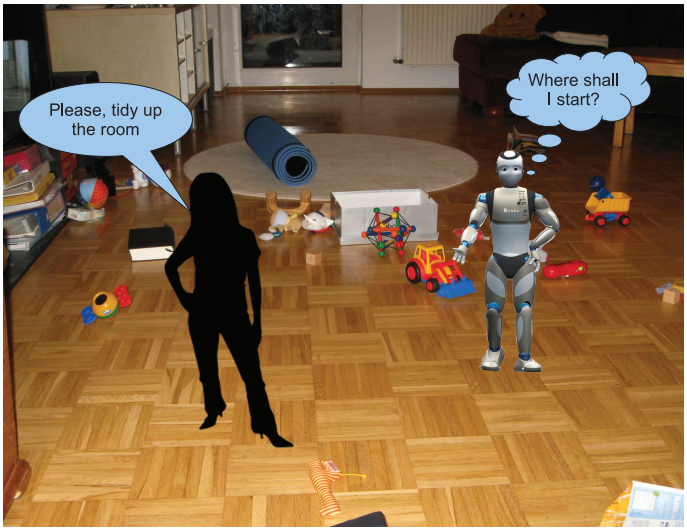
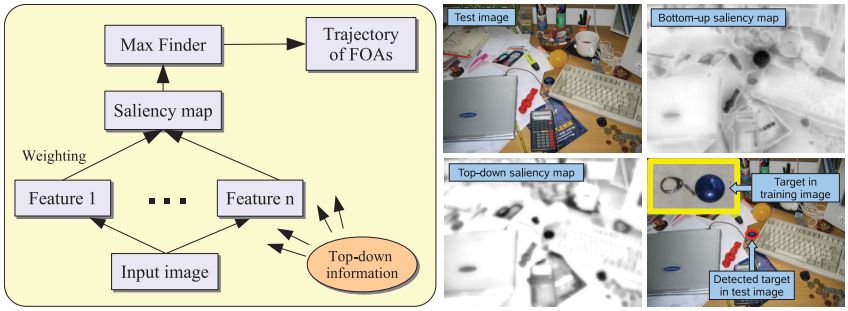


Fig. 1 The scene visualizes the concept of an attentive robot: to tidy up the room, the robot has to investigate the scene and therefore attend to the objects on the floor, one object at a time. An attention module endows it with the capability to focus on regions of most potential interest. This enables efficient processing and prioritizes the robot's actions.

[James, 1890]), it is an active research area in psychophysics since many decades, and it is frequently used in machine vision and robotics to refer to mechanisms that focus further processing on regions of interest. The latter perspective of attention is very broad, in principle, any pre-processing method of sensor data could be called attentional since it focuses further processing on parts of the data. We believe that closely mimicking the human system has the advantage that it results in human-like behavior, which is beneficial for systems that should interact with humans in a natural and intuitive manner. Therefore, in this article, we focus on methods that are based on concepts of human perception. Following this direction, one of the best fitting definitions of attention comes from Wikipedia: "Attention is the cognitive process of selectively concentrating on one aspect of the environment while ignoring other things" [1].

While the concept of attention exists for all senses, most research focuses on the visual part of attention. This is true both for human visual attention, due to the fact that vision is the most important sense in humans, and for computational attention system. Thus, with a few exceptions the approaches mentioned in this article focus on analyzing visual data.

During the last decade, attentional modules for autonomous robots have significantly gained in popularity. The reasons are two-fold. First, adequate computational resources are now available to compute the focus of attention in real-time [Frintrop et al., 2007, Xu et al., 2009] and the methods are robust enough to deal with real-world conditions. Second, basic techniques such as localization and collision avoidance have reached a quite mature level and interest has moved on to higher level tasks and challenges. The more complex a system becomes, the more urgent is the need for optimizing the visual processing. The high level of interest



(a) (b)

Fig. 2 (a) General structure of most visual attention systems. (b) Example of saliency computation with the attention system VOCUS: bottom-up exploration (top-right) and top-down search for the target "key fob" (bottom).

in such capabilities has led to a large number of EU projects on Cognitive Systems during the last decade. Many of the robots developed in these projects have an attentional module, e.g., in the projects MACS, PACO-PLUS, RobotCub, and GRASP.

In this paper, we will give an overview of the current state of the art in computational attention systems for autonomous robots. While being far from an exhaustive overview, we aim to give the reader an impression of what attention systems can do for cognitive robots and which directions already exist. Finally, we briefly discuss possible future ways to lead us closer to the dream of attentive robots.

2 The Basic Structure of Computational Attention Systems

Most biologically inspired attention systems have a similar structure, which is depicted in Fig. 2 (a). This structure is originally adapted from psychological theories like the Feature Integration Theory [Treisman and Gelade, 1980] and the Guided Search model [Wolfe, 1994]. The main idea is to compute several feature channels such as intensity, color, orientation or motion, in parallel and to fuse their conspicuities in a saliency map. This map is a gray-level image with pixel brightness proportional to the saliency (cf. Fig. 2 (b), top right). This approach is adopted from the parallel processing of different features in the human brain; some brain areas are mainly involved in processing color while others concentrate on motion processing and so on [Palmer, 1999]. If top-down information is available, e.g., prior knowledge on the context, the task, the searched object, etc., it can be used to influence the processing. An example of top-down attention is shown in Fig. 2 (b), bottom row. Here, knowledge about the target object "key fob" is fed into the attention system as a feature descriptor, resulting in a top-down saliency map (details in [Frintrop, 2006]).

The feature computations are usually based on contrast computations with *center-surround filters*. Such filters are inspired by cells in the human visual system (e.g. ganglion cells in the retina) that compute the contrast of a center and

a surround region [Palmer, 1999]. Computationally, they are usually modeled by Difference-of-Gaussian or Gabor filters. The feature channels most frequently implemented in computational attention systems are intensity, color, orientation, and motion.

One of the most important capabilities of attention systems is their ability to detect regions that differ from the rest of the image, a property that makes an object "salient". That means, the saliency of an object depends on the context. A red ball on grass is salient, while it is not salient among other red balls. Therefore, attention systems usually weight the feature maps according to the uniqueness of the feature. Feature maps with much activation obtain a low weight while those with few strong activation peaks obtain a high weight (details in [Frintrop, 2006]).

After obtaining a saliency map, the maxima in this map denote the regions that are investigated by the focus of attention (FOA) in the order of decreasing saliency. This trajectory of FOAs imitates human eye movements. Output of a computational attention system is either the saliency map itself or a trajectory of focused regions.

While most attention systems share this general structure, there are different ways of implementing the details. One of the best known computational models is the iNVT from the group around Itti [Itti et al., 1998]. In our group, we have developed the VOCUS model [Frintrop, 2006, 2011], that has adopted and extended several ideas from the iNVT. It is real-time capable and has a top-down mode to search for objects. Tsotsos and his group have developed the selective tuning model. A full description of the model and an overview of attention theories are available in his recent book [Tsotsos, 2011]. During the last years, some approaches came up that use information-theoretic concepts to determine visual saliency [Bruce and Tsotsos, 2009, Gao et al., 2009]. A survey on the cognitive foundations and state of the art of computational attention systems can be found in [Frintrop et al., 2010], an introduction to the topic for students and people new to the field is available in [Frintrop, 2011].

As mentioned in the introduction, this paper focuses on approaches that are based on concepts of human perception and share the above structure. However, it is worth noting that numerous approaches exist that compute saliency in ways that are less or not at all biologically-motivated. For example, Hou and Zhang [2007] compute the spectral residual of an image in the frequency domain and Gould et al. [2007] and Liu et al. [2009] learn an optimal feature combination with machine learning techniques.

3 Attentive Robots: The State of the Art

A future attentive robot is supposed to use attention for many tasks, on different levels of abstraction. If Dobby shall tidy up the room, it must focus on objects at unusual places and has to know where each object belongs. If it shall bring you the salt shaker, it should focus on the cupboard where the shaker is usually stored and should concentrate on features fitting to the appearance of the shaker. If you give Dobby an order, it has to interpret your gestures, facial expressions, and voice, e.g., it should follow your gaze and your pointing finger.

The tasks of the robot that involve visual attention might be classified roughly into three categories. The first, most low-level category, uses attention to detect

salient landmarks that can be used for localization and scene recognition (sec. 3.1). The second, mid-level category considers attention as a front-end for object recognition (sec. 3.2). In the third, highest-level category, attention is used in a human-like way to guide the actions of an autonomous system like a robot, i.e., to guide object manipulation or human-robot interaction (sec. 3.3).

3.1 Salient Landmarks

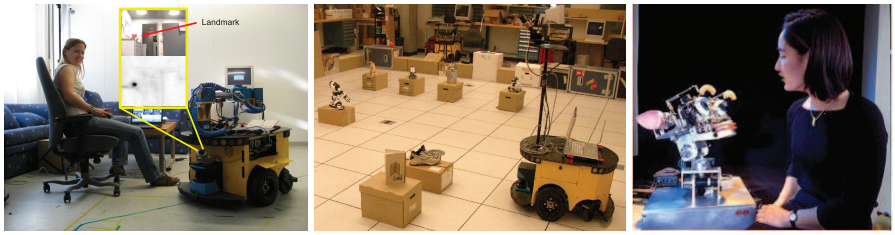
A basic capability of autonomous mobile robots is to localize themselves in their environment. Based on a known map of the surrounding, the robot has to determine its position in this map by interpreting its sensor data. When based on visual data, this is done by detecting visual landmarks with a known position. A visual landmark can be anything that the robot can see: a blob on the wall, a corner of an object, the edge of a door, or the door itself. One of the primary requirements of a visual landmark is that it should be redetectable under changing illumination conditions and from new viewpoints. It should also be possible to compute it quickly and to store it without much effort. Therefore complex object descriptions are seldom used. Salient landmarks are excellent candidates for localization since they have a high uniqueness. This makes them easy to redetect and diminishes the risk of confusing them with other landmarks. This also enables a landmark detection algorithm to concentrate on a sparse set of landmarks which reduces computation complexity. We have shown that the repeatability of salient regions in different scenes is significantly higher than the repeatability of standard detectors [Frintrop, 2008].

An early project that used salient landmarks for **localization** was the ARK project [Nickerson et al., 1998]. It relied on hand-coded maps, including the locations of known static obstacles as well as the locations of natural visual landmarks. Siagian and Itti [2009] presented an approach for **scene classification** and global localization based on salient landmarks. Additionally to the landmarks, the authors use the "gist" of the scene, a feature vector which captures the appearance of the scene, to obtain a coarse localization hypothesis.

In the above examples, a map of the environment is initially known. A more difficult task is **simultaneous localization and mapping (SLAM)** in which a robot has to build a map and localize itself inside it at the same time. We investigated the combination of visual attention and SLAM in [Frintrop and Jensfelt, 2008]. Salient regions are detected with the attention system VOCUS, tracked over several frames to obtain a 3D position of the landmarks, and matched to database entries of all previously seen landmarks. This enables the robot to detect if it closed a loop (see Fig. 3 (a)). Active camera control facilitated the redetection of landmarks.

3.2 Supporting Object Detection and Recognition

In addition to navigation, object detection and recognition are important tasks for autonomous robots, especially for manipulating objects. The terms object detection, object localization, object recognition, and classification are closely related and often used interchangeably. Let us therefore clarify our understanding of the



(a) Dumbo: Attentive visual SLAM (b) Curious George: Attentive object detection (c) Kismet: An Attentive, Social Robot

Fig. 3 Three application scenarios for visual attention systems: (a) Simultaneous localization and mapping (SLAM): robot Dumbo corrects its position estimate by redetecting a landmark which it has seen before. Landmark detection is done with the attention system VOCUS. The yellow rectangle shows the view of the robot: an image with a landmark and the corresponding saliency map (Fig. from http://www.iai.uni-bonn.de/∼frintrop/research.html) (b) Curious George: attention regions are detected in a peripheral camera image and investigated in detail by a foveal camera (Fig. from Forssén et al. [2008]). (c) Kismet is a social robot that interacts with people. Its gaze is controlled by a visual attention system (Fig. from [Breazeal, 2000] © Sam Ogden).

terms. Object detection or localization tackles the problem of localizing objects in images, e.g., by providing a bounding box around the object. Usually, the object is comparably small in the scene which makes the task challenging. The object to find might be a specific object (my favorite cup) or, as in the PASCAL VOC object detection challenge [Everingham et al., 2010], any instance of a certain class (any cup). In psychological literature on visual perception, the task to find an object is usually called visual search. A candidate to solve the visual search problem is top-down tuned visual attention [Frintrop, 2006]. Localizing an object often involves recognizing it, but may also be restricted to providing location candidates that are classified in a second step. The detection of any instance of an object class in cluttered images is still largely unsolved. In the latest PASCAL VOC 2010 challenge [Everingham et al., 2010], the best methods for object detection achieved only an average precision between 13% (potted plants) and 58.4% (aeroplanes). Sometimes, object detection refers also to the task to find anything in the scene that is an object (also called general object detection). Bottom-up attention is a perfect candidate for this kind of task since it does not require any prior knowledge on the objects.

Object classification deals with finding all instances of a certain class in a scene, e.g. faces. It is usually applied to pre-segmented objects or it uses a sliding windows approach, in which subregions of the images are successively investigated by the classifier. The term recognition is mostly used for the recognition of instances but is sometimes also used as synonym for classification.

Visual attention methods are of special interest for all tasks in which the object is comparably small in the image, as in object detection and localization or in classification on non pre-segmented images. These tasks become considerably easier if an attentional mechanism first focuses the processing on regions of potential interest. Thus is because of two reasons. First, this reduces the search space and results in reduction in computational complexity. Second, most recognition and

classification methods work best if the object occupies a dominant portion of the image.

Several approaches have been proposed to use visual attention as preprocessing step for **classification** or **object detection**. Miau et al. [2001] present a biologically motivated approach that combines an attentional front-end with the biologically motivated object recognition system HMAX. The experiments are restricted to recognize simple artificial objects like circles or rectangles. Alternatively, the authors have used a support vector machine to detect pedestrians in natural images. Walther [2006] combine their Saliency Toolbox, a Matlab implementation of the iNVT, with an object recognizer based on SIFT features and show that the recognition results are improved by the attentional front-end. Vogel and de Freitas [2008] combine the iNVT with a classifier to perform gaze planning in complex scenes. In the above mentioned approaches, the attentional part is separated from the object recognition; both systems work independently. In human perception, these processes are strongly intertwined. Accordingly, Walther and Koch [2007] suggest a unifying framework for object recognition and attention. It is based on the HMAX model and modulates the activity by spatial and feature modulation functions which suppress or enhance locations or features due to spatial attention.

While the above approaches are not applied in a robotics context, some groups have recently integrated attentive object detection on real robots. Two approaches that determine regions of interest with visual attention in a peripheral vision system, focus on these regions with a foveal vision system, and investigates these high-resolution images with an object recognition method are presented in [Gould et al., 2007] and [Meger et al., 2008]. The robot in the latter approach, curious George, placed first in the robot league of the Semantic Robot Vision Challenge (SRVC)[1] both in 2007 and 2008, and first in the software league for 2009 (see also Fig. 3 (b)).

All of these systems rely only on bottom-up information and therefore on the assumption that the objects of interest are sufficiently salient by themselves. For some object classes like traffic signs or toys, which are intentionally designed salient, this works quite well; for other applications, top-down information is needed to enable the system to focus on the desired objects. A combination of a top-down modulated computational attention system with a classifier is presented by Mitri et al. [2005]. Here, the attention system VOCUS generates object hypotheses which are verified or falsified by a classifier. For the application of ball detection in the robot soccer scenario RoboCup, the amount of false detections is reduced significantly. Recently, Xu et al. [2010] have used visual bottom-up and top-down attention to detect objects with the Autonomous City Explorer (ACE) robot.

Some groups have used attentive object detection to support **object manipulation** on robots or robot arms. One of the earliest works on this topic was presented by Bollmann et al. [1999]: a Pioneer1 robot used the active vision system NAVIS to play at dominoes. The group around Tsotsos is working on a smart wheelchair to support disabled children [Tsotsos et al., 1998, Rotenstein et al., 2007]. The wheelchair has a display as easily accessible user interface which shows pictures of places and toys. Once a task like "go to table, point to toy" is selected, the system drives to the selected location and searches for the specified toy, us-

[1] http://www.semantic-robot-vision-challenge.org/

ing mechanisms based on a visual attention system. Rasolzadeh et al. [2010] use bottom-up and top-down attention to control a KUKA arm for detecting, recognizing, and grasping objects on a table. In [Björkman and Kragic, 2010] and [Johnson-Roberson et al., 2010] the FOAs from the same attention system were used as seeds for 3D segmentation of objects from stereo data.

3.3 Guiding Robot Action

A robot which has to act in a complex world faces the same problems as a human: it has to decide what to do next. Such decisions include where to go (drive), what to look at, what to grasp, and who to interact with. Thus, even if computational power would allow it to find all correspondences, to recognize all objects in an image, and process everything of interest, it would still be necessary to filter out the relevant information to determine the next action [Mehta et al., 2000, Loach et al., 2008]. This decision is based first, on the current sensor input and second, on the internal state, for example the current tasks and goals.

A field in which the decision about the next action is intrinsically based on visual data is **active vision**, i.e., the problem of where to look next [Bajcsy, 1985]. It deals with controlling "the geometric parameters of the sensory apparatus ... in order to improve the quality of the perceptual results" [Aloimonos et al., 1988]. Thus, it directs the camera to regions of potential interest as the human visual system directs the gaze, the head, and even the body of a person. Since visual attention triggers this control in humans, it is also an intuitive candidate for the active vision problem on machines. In Sec. 4, we discuss the relation of visual attention and the active vision problem in more detail, let us here focus on approaches that have used visual attention to perform active vision control.

One of the first active vision systems that integrated visual attention was presented by Clark and Ferrier [1988]. They describe how to steer a binocular robotic head with visual attention and perform simple experiments to fixate and track the most salient region in artificial scenes composed of geometric shapes. Vijayakumar et al. [2001] present an attention system which is used to guide the gaze of a humanoid robot. The authors consider only one feature, visual flow, which enables the system to attend to moving objects. To simulate the different resolutions of the human eye, two cameras per eye are used: one wide-angle camera for peripheral vision and one narrow-angle camera for foveal vision. In more recent work, the humanoid robot iCub bases its decisions to move eyes and neck on visual and acoustic saliency maps [Ruesch et al., 2008]. Additionally, all the object manipulation approaches of the previous section include active vision to focus on the detected objects.

In the future, we want to interact with robots as naturally and intuitively as possible. Studies in the field of **human-robot interaction** have shown that humans treat robots like people [Nass and Moon, 2000, Fong et al., 2003]. The more human-like the robot acts, the easier the communication with a human. An essential part for purposefully interacting with humans is to generate a joint focus of attention. A computational attention system similar to the human one can help a robot to focus on the same region as a human. According to this, Breazeal [1999] introduced the social robot Kismet that interacts with humans in a natural and intuitive way. Its gaze is controlled by a visual attention system (see Fig. 3 (c)).

For humans, following pointing gestures of other humans is an important ability to jointly focus their attention on objects of interest. Approaches to endow robots with a similar capability were proposed by Heidemann et al. [2004] and by Schauerte et al. [2010]. They analyze the direction of a pointing finger and fuse this top-down information with the bottom-up saliency of objects. A robot that learns visual scene exploration by imitating human gaze shifts is presented by Belardinelli [2008]. Nagai [2009] developed an action learning model based on spatial and temporal continuity of bottom-up features.

Finally, Muhl et al. [2007] presented an interesting sociological study in which the interaction of a human with a robot simulation is investigated. A robot face on a screen attends to objects, shown by a human, with help of a visual attention system. If the robot was artificially diverted and directed its gaze away from the object, humans tried to reobtain the robots attention by waving hands, making noise, or approaching to the robot. This shows that people established a communicative space with the robot and accepted it as a social partner.

4 Discussion and Outlook

This paper gives an overview of the state of the art in the field of computational visual attention for mobile robots. Several fields are related to the computation of attention and we will briefly discuss the similarities and differences to some of them. First, the computation of visual saliency clearly has some similarities to the computation of interest points or regions. Both approaches compute a local contrast within some feature dimension, some use even the same methods, e.g., Difference of Gaussians [Lowe, 2004]. The main difference is that standard interest points are local methods that are only influenced by a small local neighborhood, while salient regions are defined by the context. They "stick out of the scene" and thus, the whole scene or at least a large neighborhood influences the saliency of a region. Both methods are usually computed on several scales, but interest points use smaller scales than visual saliency, leading to smaller regions that influence the point and usually to a large amount of points per image (usually several hundreds or even thousands). This is reasonable and useful for tasks such as object recognition or image registration but less so for controlling the camera. Salient regions on the other hand are usually computed on larger scales to consider context information. Additionally, the uniqueness of features is computed that takes into consideration the global (or large local) surround of the region and is usually implemented as a non-linear weighting on top of the center-surround feature computations [Itti et al., 1998, Frintrop, 2006]. This method favors regions that occur seldom in the scene, an essential aspect of visual saliency. Additionally, classical interest points are usually restricted to one feature dimension (e.g. intensity or color contrast), while visual attention systems integrate the results from several feature channels. Finally, a strength of visual attention systems is that top-down information can be integrated easily into the system.

As mentioned in Sec. 3, there is also a strong relation of visual attention to the active vision problem. To distinguish the two, it is worth clarifying that visual attention is a method that can be applied to different problems while the active vision problem is a problem that looks for a method to solve it. Visual attention claims to focus the processing resources to regions of most potential interest. That

makes it a perfect candidate to solve the active vision problem. It is however neither the only method that can be used to solve this problem, nor is the active vision problem the only problem that can be solved with visual attention. While the first point is obvious – there are dozens of methods that tackle the active vision problem which are not related to visual attention – the second point is less clear. What can be done with attention except directing the camera? Well, human selective attention is well known to be separated into covert and overt attention. Overt attention corresponds to controlling eye movements and is therefore directly related to the active vision problem. Covert attention stands for processing parts of the sensory input without looking at them with the fovea. While covert attention usually precedes eye movements, this is not always the case. For example, Johansson et al. [2001] show that simple manipulation tasks can be done without overt attention. Equally, it makes perfect sense for a robot to process some parts of the sensory input directly without steering the camera explicitly into this direction or zooming in. Here, one can also take advantage of the fact that robot sensory input is different from the human one: while the human eye produces data that has high resolution in the center and low resolution in the periphery, most cameras can capture high resolution images in the entire field of view. These images are often artificially sub-sampled to reduce the amount of data that needs to be processed. This makes it possible to perform object recognition and many other tasks directly on the input data, without controlling the camera. Active vision can be left to occasions in which this data is not sufficient, e.g. if new viewpoints of an object have to be gathered.

Let us now discuss how far we are from attentive robots such as Dobby and which parts are still missing. In the field of attention systems themselves there are still several open issues. Among these are questions like "which are the optimal features for a robot?", "how are these features integrated best?", and "how do bottom-up and top-down cues interact?". While the bottom-up part is already quite well investigated and many good solutions exist, less is known about top-down attention and existing approaches are limited to some aspects. Up to now, the prior knowledge that has been used as top-down information has mainly concentrated on two aspects of this area. First, people have used object information to search for simple objects, e.g., highlighting red regions to find fire extinguishers, [Frintrop, 2006, Navalpakkam and Itti, 2006]. Second, context information about the scene has been investigated to guide the gaze, e.g., people are likely to be on the street level of an image rather than on the sky area [Torralba et al., 2006]. However, many other cues and memories influence human perception and should also be used for attentive robots. Thus, more sophisticated knowledge about objects, people, and the situation, knowledge about typical locations of objects, as well as action cues from human interactors will strongly support the selection of regions of most potential interest.

Additionally, while this review and most existing research focuses on visual attention, other sensors can be an important source of useful information that should be exploited. Some work in this direction is our previous work on saliency detection in laser data [Frintrop et al., 2005] and the combination of visual and acoustic saliency cues for the humanoid robot iCub [Ruesch et al., 2008]. It is also important to consider that robots and humans differ considerably and that concepts that are optimized for the human brain are not necessarily optimal for a machine. While most current approaches directly transfer concepts, an important

direction of future research is to investigate how systems have to be adapted to best fit the robots' embodiment and environment.

Finally, it should be mentioned that current systems use attention mechanisms for clearly specified tasks such as landmark detection or object manipulation. While good results have been obtained in these areas, it is still a long way to obtain an attentive robot such as Dobby. Among the parts that are still missing is certainly a close interaction between different modules. In computer vision, recent work has shown that tasks such as object detection, segmentation, tracking, and categorization profit strongly from each other if the modules collaborate and share information [Leibe et al., 2008, Ess et al., 2010]. Similarly, future attentive robots will strongly profit from interacting modules. Context information and prior knowledge from other modules can enable an attentive robot to obtain better, more useful regions of interest. On the other hand, the computation of attention regions will also improve the performance of other modules since more processing resources can be provided to essential parts of the sensory input.

References

1. Definition of attention. http://en.wikipedia.org/wiki/Attention, June 2011.

Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *International Journal of Computer Vision (IJCV)*, 1(4):333–356, 1988.

R. Bajcsy. Active perception vs. passive perception. In *Proc. IEEE Workshop on Computer Vision: Representation and Control*, Bellaire MI, 1985.

A. Belardinelli. *Salience features selection: Deriving a model from human evidence*. PhD thesis, Sapienza Universita di Roma, Rome, Italy, 2008.

M. Björkman and D. Kragic. Active 3D scene segmentation and detection of unknown objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, 2010.

M. Bollmann, R. Hoischen, M. Jesikiewicz, C. Justkowski, and B. Mertsching. Playing domino: A case study for an active vision system. In H.I. Christensen, editor, *Computer Vision Systems*, pages 392–411. Springer, 1999.

C. Breazeal. A context-dependent attention system for a social robot. In *Proc. of the Int'l Joint Conference on Artifical Intelligence (IJCAI 99)*, pages 1146–1151, Stockholm, Sweden, 1999.

C. Breazeal. *Sociable Machines: Expressive Social Exchange Between Humans and Robots*. PhD thesis, Department of Electrical Engineering and Computer Science. MIT, 2000.

N. D. B. Bruce and J. K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):1–24, 2009.

J. J. Clark and N. J. Ferrier. Modal control of an attentive vision system. In *Proc. of the 2nd International Conference on Computer Vision*, Tampa, Florida, US, Dec 1988.

A. Ess, K. Schindler, B. Leibe, and L. Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *International Journal of Robotics Research*, 29(14):1707–1725, 2010.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/, 2010.

T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166, 2003.

P.-E. Forssén, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe. Informed visual search: Combining attention and object recognition. In *International Conference on Robotics and Automation*, 2008.

S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, volume 3899 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer, Berlin/Heidelberg, 2006.

S. Frintrop. The high repeatability of salient regions. In *Proc. of ECCV workshop "Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environments"*, 2008.

S. Frintrop. Computational visual attention. In A. A. Salah and T. Gevers, editors, *Computer Analysis of Human Behavior (to appear)*, Advances in Pattern Recognition. Springer, 2011.

S. Frintrop and P. Jensfelt. Attentional landmarks and active gaze control for visual SLAM. *IEEE Trans. on Robotics, Special Issue on Visual SLAM*, 24(5), Oct 2008.

S. Frintrop, E. Rome, A. Nüchter, and H. Surmann. A bimodal laser-based attention system. *J. of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance in Computer Vision*, 100(1-2):124–151, Oct-Nov 2005.

S. Frintrop, M. Klodt, and E. Rome. A real-time visual attention system using integral images. In *Proc. of the 5th Int'l Conf. on Computer Vision Systems (ICVS)*, Bielefeld, Germany, March 2007.

S. Frintrop, E. Rome, and H. I. Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Trans. on Applied Perception*, 7(1), 2010.

D. Gao, S. Han, and N. Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Trans. on PAMI*, 31(6), 2009.

S. Gould, J. Arfvidsson, A. Kaehler, B. Sapp, M. Messner, G. Bradski, P. Baumstarck, S. Chung, and A. Y. Ng. Peripheral-foveal vision for real-time object recognition and tracking in video. In *Proc. of the 20th Int. Joint Conference on Artifical intelligence (IJCAI)*, 2007.

G. Heidemann, R. Rae, H. Bekel, I. Bax, and H. Ritter. Integrating context-free and context-dependent attentional mechanisms for gestural object reference. *Machine Vision and Applications*, 16(1):64–73, 2004.

X. Hou and L. Zhang. Saliency detection: a spectral residual approach. In *Proc. of CVPR*, 2007.

L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

W. James. *The Principles of Psychology*. Dover Publications, New York, 1890.

R. Johansson, G. Westling, A. Backstrom, and J. Flanagan. Eye-hand coordination in object manipulation. *The Journal of Neuroscience*, 21(17):6917–6932, 2001.

M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic. Attention based active 3D point cloud segmentation. In *Proc. of the 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, October 2010.

B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int. J. of Computer Vision, Special Issue on Learning for Recognition and Recognition for Learning*, 77(1-3):259–289, 2008.

T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

D. Loach, A. Frischen, N. Bruce, and J. K. Tsotsos. An attentional mechanism for selecting appropriate actions afforded by graspable objects. *Psychological Science*, 19(12), 2008.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. of Computer Vision (IJCV)*, 60(2):91–110, 2004.

D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, D. G. Lowe, and B. Dow. Curious george: An attentive semantic robot. *Journal Robotics and Autonomous Systems*, 56(6), 2008.

A. D. Mehta, I. Ulbert, and C. E. Schroeder. Intermodal selective attention in monkeys. I: Distribution and timing of effects across visual areas. *Cerebral Cortex*, 10(4), 2000.

F. Miau, C. Papageorgiou, and L. Itti. Neuromorphic algorithms for computer vision and attention. In *Proc. SPIE 46 Annual Int'l Symposium on Optical Science and Technology*, volume 4479, pages 12–23, Nov 2001.

S. Mitri, S. Frintrop, K. Pervölz, H. Surmann, and A. Nüchter. Robust object detection at regions of interest with an application in ball recognition. In *IEEE Proc. of the Int'l Conf. on Robotics and Automation (ICRA '05)*, 2005.

C. Muhl, Y. Nagai, and G. Sagerer. On constructing a communicative space in HRI. In *Proc. of the 30th German Conference on Artificial Intelligence (KI 2007)*. Springer, 2007.

Y. Nagai. From bottom-up visual attention to robot action learning. In *IEEE 8th Int'l Conf. on Development and Learning*, 2009.

C. Nass and Y. Moon. Machines and mindlessness: Social responses to computers. *Journal of Social Issues*, 56(1):81–103, 2000.

V. Navalpakkam and L. Itti. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

S. B. Nickerson, P. Jasiobedzki, D. Wilkes, M. Jenkin, E. Milios, J. K. Tsotsos, A. Jepson, and O. N. Bains. The ARK project: Autonomous mobile robots for known industrial environments. *Robotics and Autonomous Systems*, 25(1-2): 83–104, 1998.

S. E. Palmer. *Vision Science: Photons to Phenomenology*. The MIT Press, Cambridge, MA, 1999.

H. Pashler. *The Psychology of Attention*. MIT Press, Cambridge, MA, 1997.

B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic. An active vision system for detecting, fixating and manipulating objects in real world. *International Journal of Robotics Research*, 29(2-3), 2010.

A. Rotenstein, A. Andreopoulos, E. Fazl, D. Jacob, M. Robinson, K. Shubina, Y. Zhu, and J.K. Tsotsos. Towards the dream of intelligent, visually-guided wheelchairs. In *Proc. 2nd Int'l Conf. on Technology and Aging*, 2007.

J. Ruesch, M. Lopes, A. Bernardino, J. Hörnstein, J. Santos-Victor, and R. Pfeifer. Multimodal saliency-based bottom-up attention: A framework for the humanoid robot icub. In *Proc. of Int'l Conf. on Robotics and Automation (ICRA)*, 2008.

B. Schauerte, J. Richarz, and G. A. Fink. Saliency-based identification and recognition of pointed-at objects. In *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

C. Siagian and L. Itti. Biologically inspired mobile robot vision localization. *IEEE Transaction on Robotics*, 25(4):861–873, July 2009.

A. Torralba, A. Oliva, M. Castelhano, and J. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features on object search. *Psychological Review*, 113(4), 2006.

A. M. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

J. K. Tsotsos. *A Computational Perspective on Visual Attention*. The MIT Press, 2011.

J. K. Tsotsos, G. Verghese, S. Stevenson, M. Black, D. Metaxas, S. Culhane, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflo, Y. Ye, and R. Mann. PLAYBOT: A visually-guided robot to assist physically disabled children in play. *Image and Vision Computing 16, Special Issue on Vision for the Disabled*, pages 275–292, April 1998.

S. Vijayakumar, J. Conradt, T. Shibata, and S. Schaal. Overt visual attention for a humanoid robot. In *Proc. International Conference on Intelligence in Robotics and Autonomous Systems (IROS 2001)*, pages 2332–2337, Hawaii, 2001.

J. Vogel and N. de Freitas. Target-directed attention: Sequential decision-making for gaze planning. In *Proc. of ICRA*, 2008.

D. Walther. *Interactions of visual attention and object recognition: computational modeling, algorithms, and psychophysics*. PhD thesis, California Institute of Technology, Pasadena, CA, 2006.

D. Walther and C. Koch. Attention in hierarchical models of object recognition. *Computational Neuroscience: Theoretical insights into brain funciton, Progress in Brain research*, 165:57–78, 2007.

J. M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238, 1994.

T. Xu, T. Pototschnig, K. Kühnlenz, and M. Buss. A high-speed multi-GPU implementation of bottom-up attention using CUDA. In *Proc. of the International Conference on Robotics and Automation, (ICRA)*, 2009.

T. Xu, T. Zhang, K. Kühnlenz, and M. Buss. Attentional object detection of an active multi-vocal vision system. *Int. J. of Humanoid Robotics*, 7(2), 2010.

Publication [5]

Dominik A. Klein and Simone Frintrop. Center-surround divergence of feature statistics for salient object detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011.

Center-surround Divergence of Feature Statistics for Salient Object Detection

Dominik A. Klein and Simone Frintrop
Rheinische Friedrich-Wilhelms Universität Bonn
Institute of Computer Science III, Römerstr. 164, 53117 Bonn
{kleind,frintrop}@iai.uni-bonn.de

Abstract

In this paper, we introduce a new method to detect salient objects in images. The approach is based on the standard structure of cognitive visual attention models, but realizes the computation of saliency in each feature dimension in an information-theoretic way. The method allows a consistent computation of all feature channels and a well-founded fusion of these channels to a saliency map. Our framework enables the computation of arbitrarily scaled features and local center-surround pairs in an efficient manner. We show that our approach outperforms eight state-of-the-art saliency detectors in terms of precision and recall.

1. Introduction

Salient objects have the quality to visually stand out from their surroundings and are likely to attract human attention. A key property that makes an object salient is the visual difference to the background. A polar bear is salient on dark rocks, but almost invisible in snow. The detection of visual saliency is of high interest in many computer vision applications, ranging from general object detection in web images [3], over image thumbnailing [17], to computing a joint focus of attention in human robot interaction [20].

Visual saliency and, more general, visual attention have been widely investigated in neurobiology and psychophysics [18] and many computational models have been built based on such findings [22, 12, 6]. A survey on biologically-inspired attention systems can be found in [7]. Recently, several saliency approaches came up that are based on computational and mathematical ideas and usually less biologically motivated. These approaches range from the computation of entropy [13, 10], over determining features that best discriminate between a target and a null hypothesis [8], to learning the optimal feature combination with machine learning techniques [15, 3].

In this work, we present a new approach to compute visual saliency that combines the general structure of psychological attention models [21, 25] with a sound mathematical foundation, and additionally enables an efficient computational implementation. We define the saliency of an image region in an information-theoretic way by means of the Kullback-Leibler-Divergence (KLD). For a center and a surround region, we estimate the distributions of visual feature occurrences. Then, the KLD between these distributions expresses how much more capacity one can expect to require when events following the center distribution are coded according to the surround distribution. In other words, KLD measures how much the feature statistics in the center diverge from those in the surround.

This formulation of saliency has two advantages. First, it allows a consistent computation for all feature channels, in contrast to approaches that apply different feature extraction methods for each channel [15, 3]. Second and more important, it allows a well-founded fusion of feature channels. While absolute values of such channels quantify miscellaneous properties that are not necessarily unifiable in a straight-forward way, KLD abstracts them to a common entity. Additionally, we incorporate an efficient scale-space computation of center-surround pairs of arbitrary sizes.

We evaluate our approach on a standard benchmark database of salient objects [1] and compare the results with eight state of the art saliency detectors. It shows that our approach outperforms all other methods in terms of precision and recall. Our method shows its strength especially for small objects, for which good precision values are usually more difficult to obtain.

2. State of the Art

The concept of visual saliency comes from human perception and correlates with the ability of a region to attract attention [18]. While human attention can be attracted by bottom-up, data-driven as well as by top-down, knowledge-driven factors, saliency is associated with bottom-up attention that automatically attracts the human gaze.

Bottom-up attention has been widely studied in cognitive fields. A basis for many computational attention models are the Feature Integration Theory (FIT) [21] and the Guided Search model [25]. The FIT has introduced the structure

that still serves as basis for many computational attention systems: several feature channels (e.g. color or orientation), each divided into several feature types (e.g. red, yellow, horizontal, vertical), are investigated in parallel. Finally, the conspicuities are collected in a *master map of attention*. In later works, this map has been called *saliency map*.

Many computational models have been built according to this structure [12, 6, 24], among them one of the most popular systems, the iNVT of Itti et al. [12]. While these systems have obtained good results in simulating human eye movements and in applications ranging from object recognition to robotics [7], one problem is that the fusion of feature channels with per se not comparable properties is usually somewhat arbitrary.

During the last decade, several approaches came up to model saliency with computational and mathematical methods that are mostly less biologically motivated. Kadir and Brady have introduced entropy-based saliency [13]. More recently, Hou and Zhang have computed the incremental coding length to measure the perspective entropy gain [10]. Entropy-based methods generally capture image regions with a lot of structure, which corresponds often but not always to salient regions. A problem occurs if the absence of structure makes an item salient, such as a person wearing white clothes in the jungle (cf. last row of Fig. 4).

Ma and Zhang have proposed a contrast-based method that uses fuzzy growing to extract regions from their saliency map [16]. Achanta et al. have introduced a simple approach that determines the difference of pixels to the average color and intensity value of the image [1, 2]. While their system has problems to detect saliencies for several classical pop-out experiments (cf. Sec. 4.1), it is fast and simple to implement.

Some groups have investigated alternative ways to compute saliency by applying different computer vision methods to obtain feature channels, which are finally fused by machine learning techniques. Liu et al. combined multi-scale contrast, center-surround histograms, and color spatial-distributions with conditional random fields [15]. Alexe et al. combined multi-scale saliency, color contrast, edge density, and superpixels in a Bayesian framework [3].

Information theory also has entered the field of saliency detection. Itti and Baldi have computed temporal saliency based on a Baysian notion of surprise [11]. Gao et al. have presented a decision-theoretic approach based on mutual information [8] and Chen has computed the co-saliency of two objects in different images with the KLD [5].

Bruce and Tsotsos have presented an interesting approach that computes the self-information of image regions with respect to their surround [4]. There are some parallels of this work to our approach. The differences are that while they base their feature detection on ICA coefficients that are learned from a large variety of images, we have specifically designed scalable feature detectors to represent the distributions in different feature channels. This enables us to compute features on any scale in a computationally feasible way and disengages us from the need of a training set. Furthermore, we compute the KLD instead of the self-information, apply local instead of global surround regions, and compute the saliencies on several scales.

Figure 1. Center-surround filter based on the Kullback-Leibler Divergence (KLD).

3. The Saliency Model

The main structure of our saliency system, called BITS (Bonn Information-Theoretic Saliency model), is based on the general layout of psychological attention models like the ones in [21, 25]: several feature channels are investigated in parallel and the conspicuities are fused to a single saliency map. The feature channels intensity, color, and orientation have been chosen since they belong to the basic features of the human attention system [26].[1]

The saliency computation itself is rather computationally than biologically motivated and consists of two steps. First, basic features analyze the occurrence of certain intensities, colors, and orientations on different scales. In a second step, the center-surround contrast is determined in an information-theoretic way. Two distributions of visual feature occurrences are determined for a center and a surround region and the Kullback-Leibler Divergence determines the difference between these distributions (cf. Fig. 1). An overview of the system is depicted in Fig. 2.

3.1. Basic Feature Cues

For our visual saliency system, we model the basic features of color, orientation, and intensity. From an input image in the HSL color space, integral layers are built in order to quickly compute pyramid representations from our scalable basic features. These integral layers enable the calculation of summed and averaged values of arbitrary sized rectangular regions in constant time [23].

The intensity feature is the average of the lightness layer within a rectangle of a certain scale. The color feature is also the average of a rectangular region, but a little trickier to compute in order to account for the saturation of the occurring colors. Hue and saturation that represent polar co-

[1] Another important feature is motion, but because here we concentrate on saliency detection in web images, motion is not required.

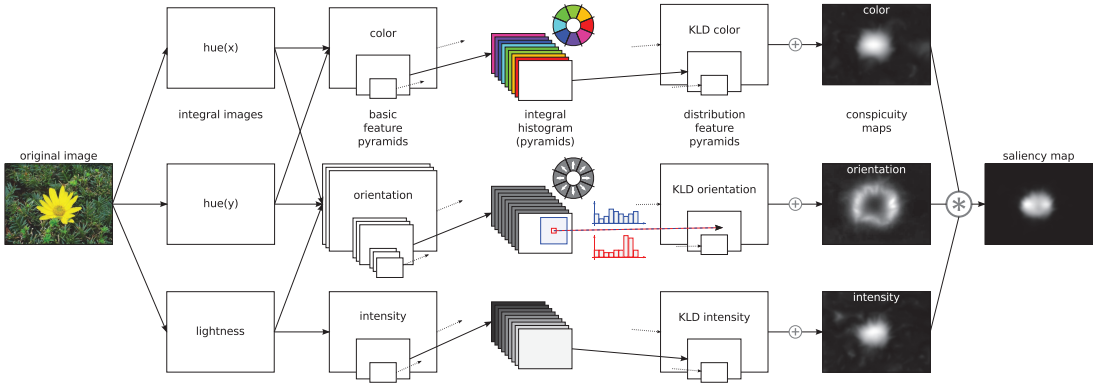


Figure 2. Schematic overview of our saliency system BITS.

ordinates in the HSL color space are converted into Cartesian coordinates, referred to as hue(x)- and hue(y)-layers (cf. Fig. 2). From those, average colors can be computed via integral layers, before the representation is transformed back into hue/saturation. The orientation feature computes partial derivatives from region-wise averages to determine the gradient direction on a given scale as in [14]. We apply the orientation feature separately for lightness-, hue(x)- and hue(y)-layer, because orientation should be observable from intensity as well as color contrasts. We compute pyramids of eight different feature scales in steps of factor $\sqrt{2}$. For this, we do not need to scale the image data, but the feature size, which is an advantage in terms of speed. The feature sampling rate depends on their scales, larger features are more coarsely sampled than smaller ones.

3.2. Center-Surround Distribution Feature based on Information Theory

Information theory is an area of statistics that is used to analyze signals and their transmission over channels. One of the main concepts is the notion of entropy, which quantifies the expected value of information that a signal of a given coding scheme contains. A coding scheme equates to a probability distribution over the occurrence of certain messages. The less predictable the occurrence of a message is, the higher the entropy. For instance the entropy of a uniform distribution is highest, while if one can predict the next message for sure, the entropy is zero.

As mentioned above, the difference of a region to its surroundings is essential to obtain visual saliency. One can convey this principle of difference to information theory by using the Kullback-Leibler Divergence,

$$\mathcal{D}_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x. \qquad (1)$$

KLD is a measure between two probability distributions P and Q, that meters the expected value how much longer a message must be to express events from P based on Q. The more P differs from Q, the higher the KLD.

For each pyramid layer of basic feature results, our contrast feature is computed. This feature is based on KLD and integrates a local center-surround mechanism to rate the conspicuity of a region (cf. Fig. 1). Here, the information-theoretic notion of a message is the parameter value of a visual feature. Therefore, we need to estimate local distributions of basic feature results: we split every basic feature layer into an integral histogram [19]. An integral histogram consists of layers that count the summed number of values top and left from a pixel that fall into a certain histogram bin. Here, sums of relative distances of the values on the corresponding basic feature map to centered values of the neighboring bins are counted. Thus, one can obtain bilinearly interpolated histograms for rectangular regions. For the periodic color and orientation feature, we build radial histograms with an additional center bin. In case of color, the saturation determines how much a feature sample counts for the center or a radial bin. For orientation and gradient magnitude we proceed correspondingly. This allows to contrast the absence or occurrence of orientation and color in the center with the surround distribution. Utilizing these integral histograms, we calculate the discrete KLD feature

$$D_{\mathrm{KL}}(C\|S) = \sum_{i=1}^{\mathrm{b}} C(i) \log \frac{C(i)}{S(i)}, \qquad (2)$$

in constant time, where b denotes the number of bins (here: 13), and C and S are distributions of center and surround regions with size ratios of 0.2 and 0.3. Increasing the number of bins did not substantially increase the quality of the system. The KLD feature maps are scale normalized corresponding to the ratio of the feature's surround region to the image size. Then they are rescaled and added per pixel on highest resolution to form one conspicuity map per channel. Fusion of conspicuity maps into a single saliency map is done by per element multiplication. This results in a fusion exponential proportional to geometric mean, but we omit calculation of the n^{th} root.

4. Experiments and Results

We evaluated our saliency method on two kinds of data: psychological patterns (Sec. 4.1) and a database of salient objects [1] (Sec. 4.2). On both data sets, we compared our approach with eight state-of-the-art saliency models: the iNVT by Itti et al. [12], the Saliency Toolbox (ST) [24], two systems of Hou and Zhang (HZ07,HZ08) [9, 10], the AIM model of Bruce and Tsotsos [4], the system of Ma and Zhang (MZ) [16], and two versions of Achanta et al. (AC09,AC10) [1, 2]. For iNVT, ST, HZ08, AIM, AC09 and AC10 we used the code from the authors' web pages. For HZ07 and MZ we used the saliency maps provided online[2].

4.1. Psychological Patterns

Detecting outliers in "pop-out" images is an essential step for a saliency model, since the results clearly show the strengths and limitations of an approach. We designed intensity and color patterns with a gray background and items with the same intensity contrasts to the background. This allows to make sure that saliency really results from an item-item contrast and not from an item-background contrast.

Fig. 4 shows the results on these patterns for all saliency methods with available source code. Saliency maps that have their most salient region on the outlier are marked with a green bounding box, others with a red one. Except for our model, none of the models was able to detect all outliers. Some results can be explained by the system design: Achanta cannot detect orientation pop-outs, since it is a purely color/intensity based approach. AIM and AC09 cannot detect local pop-outs (row 4), since they use a global instead of local surround. The result of Hou (last row) shows that purely using entropy to compute saliency is not always sufficient: the uniform square on a textured background is not considered salient, since it shows low entropy compared to the high entropy of the background. On the other hand, the non-salient region in the result of AIM is due to the filter size and could be avoided by a scale-space extension.

4.2. Salient Object Database

Additionally, we performed quantitative experiments on the image set that was used in [1, 2]. It is a database of 1000 images, which is a subset of the MSRA salient object database [15]. The latter contains objects that were marked as salient by 2 out of 3 users. For the 1000 image subset, binary maps are available that show accurate contours of the salient objects. Fig. 5 shows some images of this database and the corresponding saliency maps for the saliency methods with available source code.

The saliency maps were evaluated according to [1]. A binary map was obtained from the saliency map by varying a threshold on the intensity values $[0, 255]$. Each of these 256 maps was compared to the ground truth binary map from the database and precision and recall were computed. This resulted in the precision-recall curves shown in Fig. 3, top. It should be noted that some of the methods (e.g., iNVT, ST, AIM) are designed rather for simulating human eye movements than for the detection of salient objects in web images. Therefore, these results should be regarded with caution. We have included them for completeness.

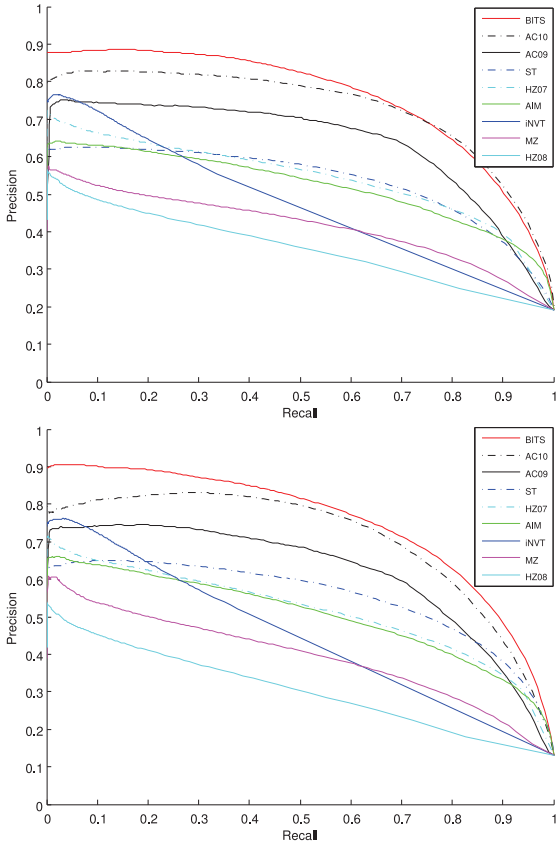


Figure 3. Precision-recall curves for the saliency maps of our system BITS and 8 other saliency detectors on the dataset of 1000 images from [1] (top) and of a subset of small objects (max. 20% of image) (bottom). See text for details.

As already pointed out in [15], obtaining high precision-recall values for images with large objects is not too difficult: if an object occupies 80% of the image, an algorithm that selects the whole image obtains 80% precision with 100% recall. Thus, it is more challenging to obtain high precision-recall curves for small objects. To test this, we determined a subset of the database containing small objects, similarly as in [15]. We selected 549 images with objects occupying at most 20% of the image area. The resulting precision-recall curves are shown in Fig. 3, bottom. Here it shows more clearly that our approach outperforms the other methods.

[2] http://ivrg.epfl.ch/supplementary_material/RK_CVPR09

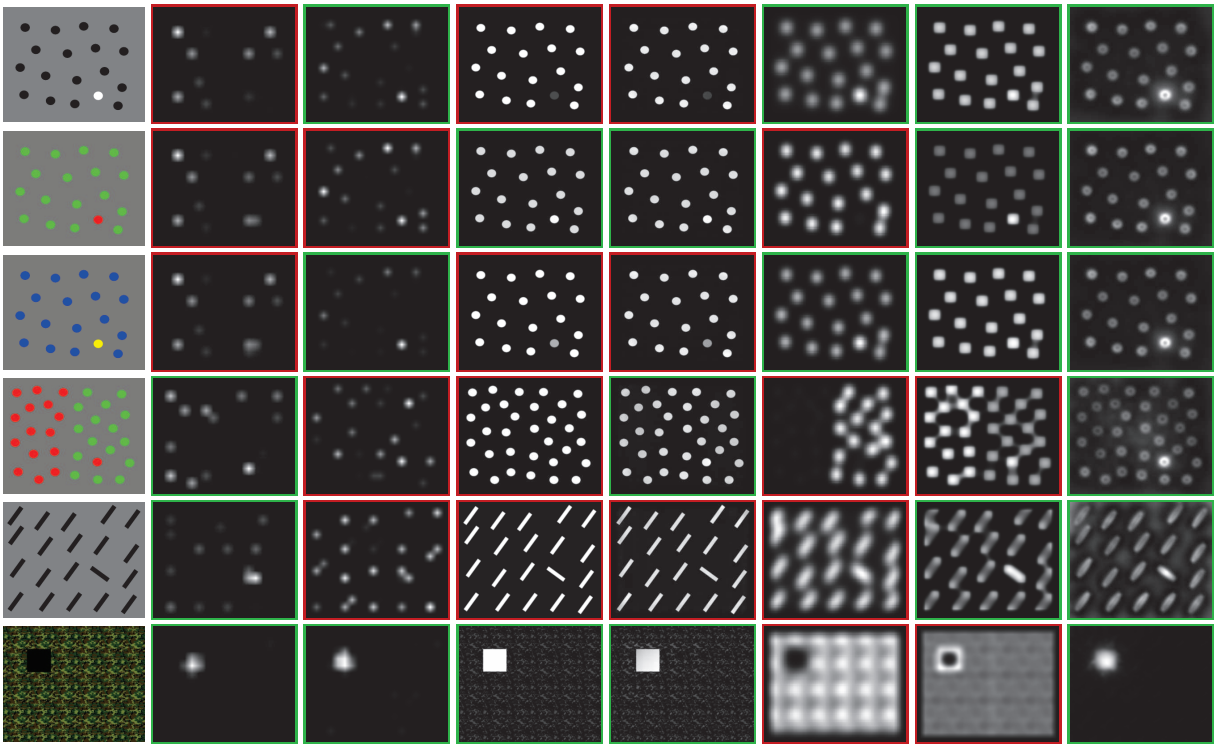


Figure 4. Comparison of saliency maps on psychological patterns. Saliency methods from left to right: iNVT [12], ST [24], AC09 [1], AC10, [2], HZ08 [10], AIM [4], our approach BITS. Green bounding boxes: outlier detected; red boxes: failure.

5. Conclusion

We presented a new approach to compute visual saliency in an information-theoretic way. By means of the Kullback-Leibler Divergence, we determine the contrast of the center and the surround distribution of features for the dimensions intensity, color, and orientation. This enables a well-founded fusion of channels based on a common entity. We have shown that the new approach outperforms eight other saliency computation methods, especially for small objects.

Since information-theoretic approaches are based on feature distributions, the computation is intrinsically more computationally expensive than the classical area-based center-surround filters. To obtain a system that is applicable in reasonable time, calculations are often restricted. For example, AIM uses a center patch with a fixed size and one global surround distribution that covers the complete image.

However, since the detection of salient structures relies essentially on center-surround pairs of different sizes, it is important to integrate scalable feature computations in a computationally still feasible way. Especially for applications on large image databases or on mobile robots, real-time performance is an essential requirement. With our integral image based framework, we found a good compromise between accuracy and speed. With less than 0.5 sec (320×240 pixel image, 2.66 GHz quad-core PC using double precision computations) the system is close to real-time performance. Since the code is not yet optimized, we are confident to obtain real-time performance easily by standard optimizations and/or more extensive parallelization.

Systems as the proposed one always include many parameters and design choices. The parameters used here have shown to be reasonable for the detection of salient objects in web images. We tested the approach also on other images, and it shows to be quite stable and not strongly dependent on parameter choices. Our combinations of center-surround pairs enable the detection of a wide range of sizes of salient regions. Nevertheless, for other applications such as modeling human eye movements, the parameters might have to be adapted to yield optimal performance.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, 2009.

[2] R. Achanta and S. Süsstrunk. Saliency detection using maximum symmetric surround. In *Proc. of Int'l Conf. on Image Processing (ICIP)*, 2010.

[3] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proc. CVPR*, 2010.

[4] N. Bruce and J. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):1–24, 2009.

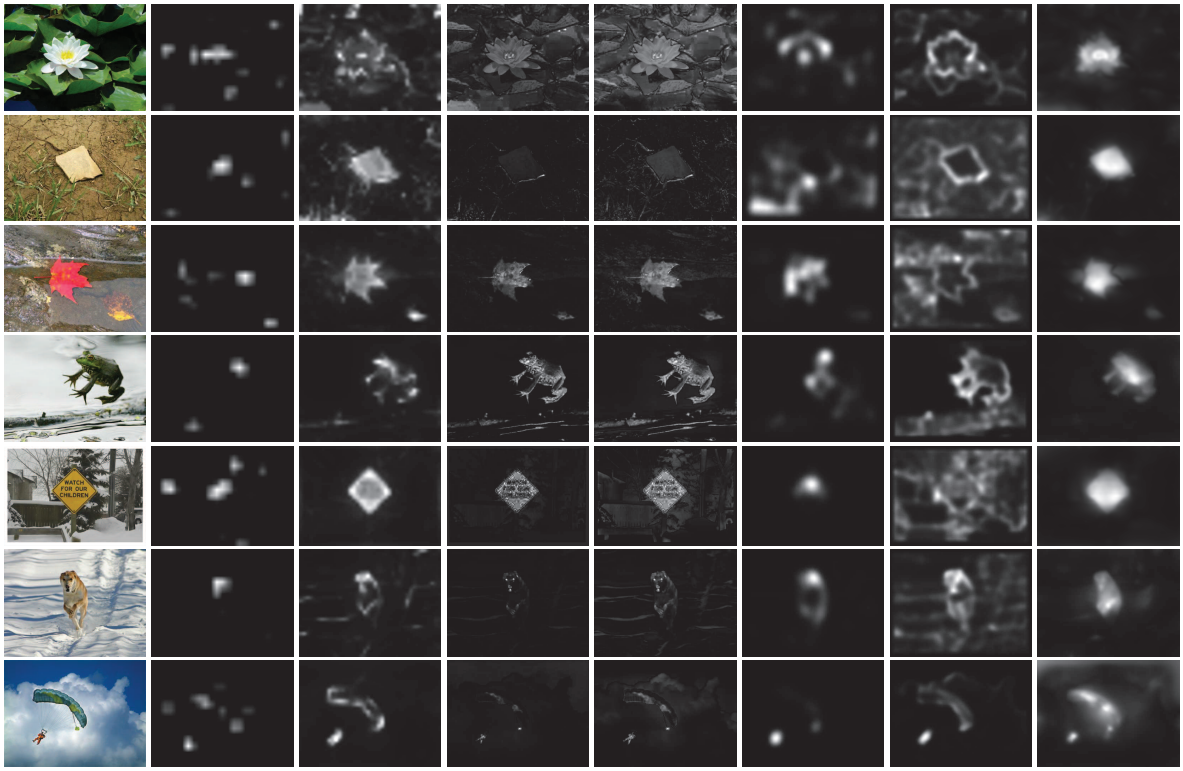


Figure 5. Comparison of saliency maps on natural images from the MSRA dataset [15]. Saliency methods from left to right: iNVT [12], ST [24], AC09 [1], AC10, [2], HZ08 [10], AIM [4], our approach BITS.

[5] H.-T. Chen. Preattentive co-saliency detection. In *Proc. of ICIP*, 2010.

[6] S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, volume 3899 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer, 2006.

[7] S. Frintrop, E. Rome, and H. I. Christensen. Computational visual attention systems and their cognitive foundation: A survey. *ACM Trans. on Applied Perception*, 7(1), 2010.

[8] D. Gao and N. Vasconcelos. Bottom-up saliency is a discriminant process. In *Proc. of ICCV*, 2007.

[9] X. Hou and L. Zhang. Saliency detection: a spectral residual approach. In *Proc. of CVPR*, 2007.

[10] X. Hou and L. Zhang. Dynamic visual attention: Searching for coding length increments. In *Advances in Neural Information Processing Systems*, 2008.

[11] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295–1306, 2009.

[12] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on PAMI*, 20(11), 1998.

[13] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proc. of ECCV*, 2004.

[14] D. Klein and A. Cremers. Boosting scalable gradient features for adaptive real-time tracking. In *Proc. of ICRA*, 2011.

[15] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Trans. on PAMI*, 2009.

[16] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Int'l Conf. on Multimedia*, 2003.

[17] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *Proc. of ICCV*, 2009.

[18] H. Pashler. *The Psychology of Attention*. MIT Press, Cambridge, MA, 1997.

[19] F. Porikli. Integral histogram: A fast way to extract histograms in Cartesian spaces. In *Proc. CVPR*, 2005.

[20] B. Schauerte and G. A. Fink. Focusing computational visual attention in multi-modal human-robot interaction. In *Proc. of Int'l Conf. on Multimodal Interfaces (ICMI)*, 2010.

[21] A. M. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[22] J. K. Tsotsos. Towards a computational model of visual attention. In *Early Vision and Beyond*. MIT Press, 1995.

[23] P. Viola and M. Jones. Robust real-time object detection. *Int. Journal of Computer Vision*, 57(2):137–154, 2002.

[24] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 2006.

[25] J. M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2), 1994.

[26] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, 5:1–7, 2004.

Publication [6]

Simone Frintrop. Computational visual attention. In Albert A. Salah and Theo Gevers, editors, *Computer Analysis of Human Behavior*, Advances in Pattern Recognition. Springer, 2011.

Computational Visual Attention

Simone Frintrop

Visual attention is one of the key mechanisms of perception that enables humans to efficiently select the visual data of most potential interest. Machines face similar challenges as humans: they have to deal with a large amount of input data and have to select the most promising parts. In this chapter, we explain the underlying biological and psychophysical grounding of visual attention, show how these mechanisms can be implemented computationally, and discuss why and under what conditions machines, especially robots, profit from such a concept.

1 What Is Attention? And Do We Need Attentive Machines?

Attention is one of the key mechanisms of human perception that enables us to act efficiently in a complex world. Imagine you visit Cologne for the first time, you stroll through the streets and look around curiously. You look at the large Cologne Cathedral and at some street performers. After a while, you remember that you have to catch your train back home soon and you start actively to look for signs to the station. You have no eye for the street performers any more. But when you enter the station, you hear a fire alarm and see that people are running out of the station. Immediately you forget your waiting train and join them on their way out.

This scenario shows the complexity of human perception. Plenty of information is perceived at each instant, much more than can be processed in detail by the human brain. The ability to extract the relevant pieces of the sensory input at an early processing stage is crucial for efficient acting. Thereby, it depends on the context which part of the sensory input is relevant. When having a goal like catching a train, the signs are relevant, without an explicit goal, salient things like the street performers attract the attention. Some things or events are so salient that they even override

Simone Frintrop
Rheinische Friedrich-Wilhelms Universität Bonn, Institute of Computer Science III, Römerstrasse 164, 53117 Bonn; e-mail: frintrop@iai.uni-bonn.de

your goals, such as the fire alarm. The mechanism to direct the processing resources to the potentially most relevant part of the sensory input is called *selective attention*. One of the most famous definitions of selective attention is from William James, a pioneering psychologist, who stated in 1890: "Everyone knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought" [11]. While the concept of attention exists for all senses, here we will concentrate on visual attention and thus on the processing of images and videos.

While it is obvious that attention is a useful concept for humans, why is it of interest for machines and which kinds of machines profit most from such a concept? To answer these questions, let us tackle two goals of attention separately. The first goal is to handle the complexity of the perceptual input. Since many visual processing tasks concerned with the recognition of arbitrary objects are NP-hard [23], an efficient solution is often not achievable. Problems arise for example if arbitrary objects of arbitrary sizes and extends shall be recognized, i.e. everything from the fly on the wall to the building in the background. The typical approach to detect objects in images is the sliding-window paradigm in which a classifier is trained to detect an object in a subregion of the image and is repeatedly applied to differently sized test windows. A mechanism to prioritize the image regions for further processing is of large interest, especially if large image databases shall be investigated or if real-time processing is desired, e.g. on autonomous mobile robots.

The second goal of attention is to support action decisions. This task is especially important for autonomous robots that act in a complex, possibly unknown environment. Even if equipped with unlimited computational power, robots still underlie similar physical constraints as humans: at one point in time, they can only navigate to one location, zoom in on one or a few regions, and grasp one or a few objects. Thus, a mechanism that selects the relevant parts of the sensory input and decides what to do next is essential. Since robots usually operate in the same environments as humans, it is reasonable to imitate the human attention system to fulfill these tasks. Furthermore, in domains as human-robot interaction, it is helpful to generate a joint focus of attention between man and machine to make sure that both communicate about the same object[1]. Having similar mechanisms for both human and robot facilitates this task.

As a conclusion, we can state that the more general a system shall be and the more complex and undefined the input data are, the more urgent the need for a prioritizing attention system that preselects the data of most potential interest.

This chapter aims to provide you with everything you must know to build a computational attention system[2]. It starts with an introduction to human perception (sec. 2). This section gives you an insight to the important mechanisms in the brain that are involved in visual attention and thus provides the background knowledge that is required when working in the field of computational attention. If you are mainly interested in how to build a computational system, you might skip this

[1] The social aspect of human attention is described in chapter 8, section 5.4.1

[2] Parts of this chapter have been published before in [4].

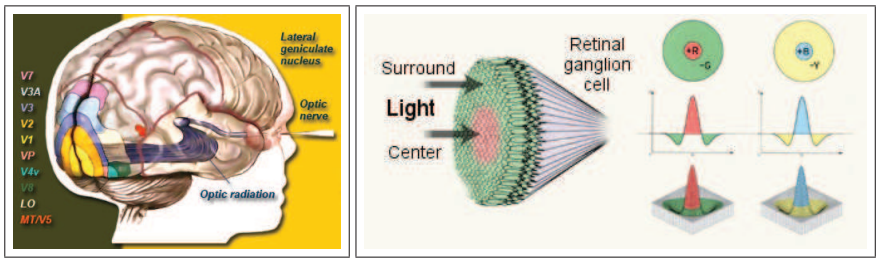


Fig. 1 Left: The human visual system (Fig. adapted from http://www.brain-maps.com/visual-fields.html). Right: The receptive field of a ganglion cell with center and surround and its simulation with Difference-of-Gaussian filters (Fig. adapted from [15]).

section and directly jump to sec. 3. This section explains how to build a bottom-up system of visual attention and how to extend such a system to perform visual search for objects. After that, we discuss different ways to evaluate attention systems (sec. 4) and mention two applications of such systems in robotic contexts (sec. 5). At the end of the chapter you find some useful links to Open Source code, freely accessible databases, and further readings on the topic.

2 Human Visual Attention

In this section, we will introduce some of the cognitive foundations of human visual attention. We start with the involved brain mechanisms, continue with several psychological concepts and evaluation methods, and finally present two influential psychological models.

2.1 The Human Visual System

Let us first regard some of the basic concepts of the human visual system. While being far from an exhaustive explanation, we focus on describing parts that are necessary to understand the visual processing involved in selective attention. The most important visual areas are illustrated in Fig. 1, left.

2.1.1 Eye, Retina, and Ganglion Cells

The light that enters the eye through the *pupil* passes through the *lens*, and reaches the *retina* at the back of the eye. The retina is a light-sensitive surface and is densely covered with over 100 million photoreceptor cells, *rods* and *cones*. The rods are more numerous and more sensitive to light than the cones but they are not sensitive

to color. The cones provide the eye's color sensitivity: among the cones, there are three different types of color reception: long-wavelength cones (L-cones) which are sensitive primarily to the red portion of the visible spectrum, middle-wavelength cones (M-cones) sensitive to green, and short-wavelength cones (S-cones) sensitive to blue. In the center of the retina is the *fovea*, a rod-free area with very thin, densely packed cones. It is the center of the eye's sharpest vision. Because of this arrangement of cells, we perceive the small region currently fixated in a high resolution and the whole surrounding only diffuse and coarse. This mechanism makes eye movements an essential part of perception, since they enable high resolution vision subsequently for different regions of a scene.

The photoreceptors transmit information to the so called *ganglion cells*, which combine the trichromatic input by subtraction and addition to determine color and luminance opponency. The receptive field of a ganglion cell, i.e. the region the cell obtains input from, is circular and separated into two areas: a center and a surround (cf. Fig. 1, right). There are two types of cells: *on-center cells* which are stimulated by light at the center and inhibited by light at the surround, and *off-center cells* with the opposite characteristic. Thus, on-center cells are well suited to detect bright regions on a dark background and off-center cells vice versa. Additional to the luminance contrast, there are also cells that are sensitive to red-green and to blue-yellow contrasts. The center-surround concept of visual cells can be modeled computationally with Difference-of-Gaussian filters (cf. Fig. 1, right) and is the basic mechanism for contrast detection in computational attention systems.

2.1.2 From the Optic Chiasm to V1

The visual information leaves the eye via the optic nerve and runs to the *optic chiasm*. From here, two pathways go to each brain hemisphere: the smaller one goes to the *superior colliculus (SC)*, which is e.g. involved in the control of eye movements. The more important pathway goes to the *Lateral Geniculate Nucleus (LGN)* and from there to higher brain areas. The LGN consists of six main layers composed of cells that have center-surround receptive fields similar to those of retinal ganglion cells but larger and with a stronger surround. From the LGN, the visual information is transmitted to the *primary visual cortex (V1)* at the back of the brain.

V1 is the largest and among the best-investigated cortical areas in primates. It has the same spatial layout as the retina itself. But although spatial relationships are preserved, the densest part of the retina, the fovea, takes up a much smaller percentage of the visual field (1%) than its representation in the primary visual cortex (25%). The cells in V1 can be classified into three types: *simple cells, complex cells*, and *hypercomplex cells*. As the ganglion cells, the simple cells have an excitatory and an inhibitory region. Most of the simple cells have an elongated structure and, therefore, are orientation sensitive. Complex cells take input from many simple cells. They have larger receptive fields than the simple cells and some are sensitive to moving lines or edges. Hypercomplex cells, in turn, receive the signals from com-

plex cells as input. These neurons are capable of detecting lines of a certain length or lines that end in a particular area.

2.1.3 Beyond V1: the Extrastriate Cortex and the Visual Pathways

From the primary visual cortex, a large collection of neurons sends information to higher brain areas. These areas are collectively called *extrastriate cortex*, in opposite to the striped architecture of V1. The areas belonging to the extrastriate cortex are V2, V3, V4, the infero-temporal cortex (IT), the middle temporal area (MT or V5) and the posterior-parietal cortex (PP).[3]

On the extrastriate areas, much less is known than on V1. One of the most important findings of the last decades was that the processing of the visual information is not serial but highly parallel. While not completely segregated, each area has a prevalence of processing certain features such as color, form (shape), or motion. Several pathways lead to different areas in the extrastriate cortex. The statements on the number of existing pathways differ: the most common belief is that there are three main pathways, one color, one form, and one motion pathway which is also responsible for depth processing [12].

The color and form pathways go through V1, V2, and V4 and end finally in IT, the area where the recognition of objects takes place. In other words, IT is concerned with the question of "what" is in a scene. Therefore, the color and form pathway together are called the *what pathway*. It is also called *ventral stream* because of its location on the ventral part of the body. The motion-depth pathway goes through V1, V2, V3, MT, and the parieto occipale area (PO) and ends finally in PP, responsible for the processing of motion and depth. Since this area is mainly concerned with the question of "where" something is in a scene, this pathway is also called *where pathway*. Another name is *dorsal stream* because it is considered to lie dorsally.

Finally, it is worth to mention that although the processing of the visual information was described above in a feed-forward manner, it is usually bi-directional. Top-down connections from higher brain areas influence the processing and go down as far as LGN. Also lateral connections combine the different areas, for example, there are connections between V4 and MT, showing that the "what" and "where" pathway are not completely separated.

2.1.4 Neurobiological Correlates of Visual Attention

The mechanisms of selective attention in the human brain still belong to the open problems in the field of research on perception. Perhaps the most prominent outcome of neuro-physiological findings on visual attention is that there is no single brain area guiding the attention, but neural correlates of visual selection appear to be reflected in nearly all brain areas associated with visual processing. Attentional

[3] The notation V1 to V5 comes from the former belief that the visual processing would be serial.

mechanisms are carried out by a network of anatomical areas. Important areas of this network are the posterior parietal cortex (PP), the superior colliculus (SC), the Lateral IntraParietal area (LIP), the Frontal Eye Field (FEF) and the pulvinar.

Brain areas involved in guiding eye movements are the FEF and the SC. There is also evidence that a kind of saliency map exists in the brain, but the opinions where it is located diverge. Some researchers locate it in the FEF, others at the LIP, the SC, at V1 or V4 (see [4] for references). Further research will be necessary to determine the tasks and interplay of the brain areas involved in the process of visual attention.

2.2 Psychological Concepts of Attention

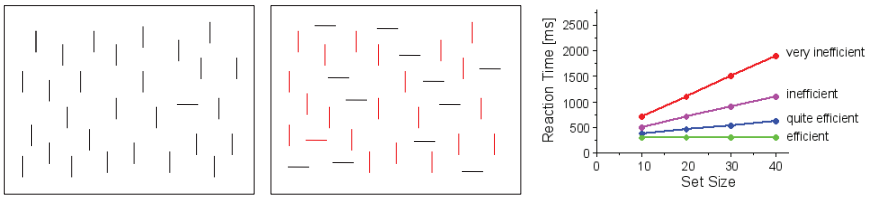
Certain concepts and expressions are frequently used when investigating human visual attention and shall be introduced here.

Usually, directing the focus of attention to a region of interest is associated with eye movements (*overt attention*). However, it is also possible to attend to peripheral locations of interest without moving the eyes, a phenomenon which is called *covert attention*. The allocation of attention is guided by two principles: *bottom-up and top-down factors*. Bottom-up attention (or *saliency*) is derived solely from the perceptual data. Main indicators for visual bottom-up saliency are a strong contrast of a region to its surround and the uniqueness of this region. Thus, a clown in the parliament is salient, whereas it is not particularly salient among other clowns (however, a whole group of clowns in the parliament is certainly salient!). The bottom-up influence is not voluntary suppressible: a highly salient region captures your attention regardless of the task, an effect called *attentional capture*. This effect might save your life, e.g. if an emergency bell or a fire captures your attention.

On the other hand, top-down attention is driven by cognitive factors such as preknowledge, context, expectations, and current goals. In human viewing behaviour, top-down cues always play a major role. Not only looking for the train station signs in the introductory example is an example of top-down attention, also more subtle influences like looking at food when being hungry. In psychophysics, top-down influences are often investigated by so called *cueing experiments*, in which a cue directs the attention to a target. A cue might be an arrow that points into the direction of the target, a picture of the target, or a sentence (“search for the red circle”).

One of the best investigated aspect of top-down attention is *visual search*. The task is exactly what the name implies: given a target and an image, find an instance of the target in the image. Visual search is omnipresent in every-day life: finding a friend in a crowd or your keys in the livingroom are examples.

In psychophysical experiments, the efficiency of visual search is measured by the *reaction time* (RT) that a subject needs to detect a target among a certain number of *distractors* (the elements that differ from the target) or by the search accuracy. To measure the RT, a subject has to report a detail of the target or has to press one button if the target was detected and another if it is not present in the scene. The RT is represented as a function of set size (the number of elements in the display).



(a) Feature search (b) Conjunction search (c) Continuum of search slopes

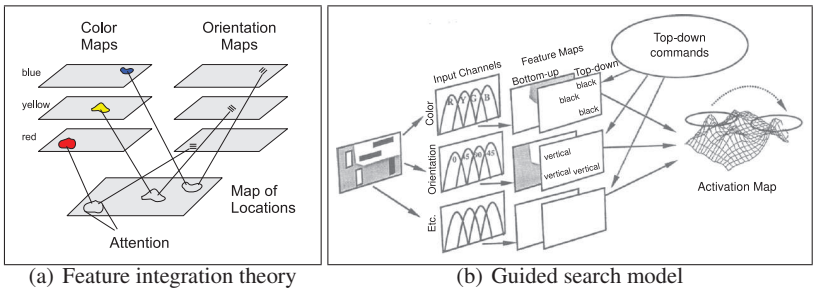
Fig. 2 (a) Feature search: the target (horizontal line) differs from the distractors (vertical lines) by a unique visual feature (pop-out effect). (b) Conjunction search: the target (red, horizontal line) differs from the distractors (red, vertical and black, horizontal lines) by a conjunction of features. (c) The reaction time (RT) of a visual search task is a function of set size. The efficiency is measured by the intercept and slopes of the functions (Fig. redrawn from [27]).

The search efficiency is determined by the slopes and the intercepts of these RT $\times$ set size functions (cf. Fig. 2 (c)). The searches vary in their efficiency: the smaller the slope of the function and the lower the value on the y-axis, the more efficient the search. Two extremes are serial and parallel search. In serial search, the reaction time increases with the number of distractors, whereas in parallel search, the slope is near zero. But note that the space of search slope functions is a continuum.

Feature searches take place in settings in which the target is distinguished from the distractors by a single basic feature (such as color or orientation)(cf. Fig. 2, (a)). In *conjunction searches* on the other hand, the target differs by more than one feature (see Fig. 2 (b)). While feature searches are usually fast and conjunction searches slower, this is not generally the case. Also a feature search might be slow if the difference between target and distractors is small (e.g. a small deviation in orientation). Generally, it can be said that search becomes harder as the target-distractor similarity increases and easier as distractor-distractor similarity increases. The most efficient search takes place for so called "pop-out" experiments that denote settings in which a single element immediately captures the attention of the observer. You understand easily what this means by looking at Fig. 2 (a). Other methods to investigate visual search is by measuring accuracy or eye movements. References for further readings on this topic can be found in [6].

One purpose of such experiments is to study the *basic features* of human perception, that means the features that are early and pre-attentively processed in the human brain and guide visual search. Undoubted basic features are color, motion, orientation and size (including length and spatial frequency) [28]. Some other features are guessed to be basic but there is limited data or dissenting opinions.

An interesting effect in visual search tasks are *search asymmetries*, that means the effect that a search for stimulus 'A' among distractors 'B' produces different results than a search for 'B' among 'A's. An example is that finding a tilted line among vertical distractors is easier than vice versa. An explanation is proposed by [22]: the authors claim that it is easier to find deviations among canonical stimuli than vice versa. Given that vertical is a canonical stimulus, the tilted line is a deviation and may be detected quickly.



(a) Feature integration theory (b) Guided search model

Fig. 3 Left: Model of the *Feature Integration Theory (FIT)* (Fig. redrawn from [20]) Right: The *Guided Search model* of Wolfe (Fig. adapted from [26] ©1994 Psychonomic Society).

2.3 Important Psychological Attention Models

In the field of psychology, there exists a wide variety of theories and models on visual attention. Their objective is to explain and better understand human perception. Here, we introduce two approaches which have been most influential for computational attention systems.

The *Feature Integration Theory (FIT)* of Treisman claims that "different features are registered early, automatically and in parallel across the visual field, while objects are identified separately and only at a later stage, which requires focused attention" [21]. Information from the resulting *feature maps* — topographical maps that highlight conspicuities according to the respective feature — is collected in a *master map of location*. Scanning serially through this map focuses the attention on the selected scene regions and provides this data for higher perception tasks (cf. Fig. 3 (a)). The theory was first introduced in 1980 but it was steadily modified and adapted to current research findings.

Beside FIT, the *Guided Search Model* of Wolfe is among the most influential work for computational visual attention systems [26]. The basic goal of the model is to explain and predict the results of visual search experiments. Mimicking the convention of numbered software upgrades, Wolfe has denoted successive versions of his model as Guided Search 1.0 to Guided Search 4.0. The best elaborated description of the model is available for Guided Search 2.0 [26]. The architecture of the model is depicted in Figure 3 (b). It shares many concepts with the FIT, but is more detailed in several aspects which are necessary for computer implementations. An interesting point is that in addition to bottom-up saliency, the model also considers the influence of top-down information by selecting the feature type which distinguishes the target best from its distractors.

3 Computational Attention Systems

Computational attention systems model the principles of human selective attention and aim to select the part of the sensory input data that is most promising for further investigation. Here, we concentrate on visual attention systems that are inspired by concepts of the human visual system but are designed with an engineering objective, that means their purpose is to improve vision systems in technical applications.[4]

3.1 General structure

Most computational attention systems have a similar structure, which is depicted in Fig. 4. This structure is originally adapted from psychological theories like the Feature Integration Theory and the Guided Search model (cf. Sec. 2.3). The main idea is to compute several features in parallel and to fuse their conspicuities in a saliency map. If top-down information is available, this can be used to influence the processing at various levels of the models. A saliency map is usually a gray-level image in which the brightness of a pixel is proportional to its saliency. The maxima in the saliency map denote the regions that are investigated by the focus of attention (FOA) in the order of decreasing saliency. This trajectory of FOAs shall resemble human eye movements. Output of a computational attention system is either the saliency map or a trajectory of focused regions.

While most attention systems share this general structure, there are different ways to implement the details. One of the best known computational attention systems is the iNVT from Itti and colleagues [10]. The VOCUS model [4] has adopted and extended several of their ideas. It is real-time capable and has a top-down mode to search for objects (visual search). Itti and Baldi presented an approach that is able to detect temporally salient regions, called *surprise theory* [8]. Bruce and Tsotsos compute saliency by determining the self-information of image regions with respect to their surround [1]. The types of top-down information that can influence an attention model are numerous and only a few have been realized in computational system. For example, the VOCUS model uses pre-knowledge about a target to weight the feature maps and perform visual search. Torralba et al. use context information about the scene to guide the gaze, e.g., to search for people on the street level of an image rather than on the sky area [19]. More abstract types of top-down cues, such as emotions and motivations, have to our knowledge not yet been integrated into computational attention systems.

In this chapter, we follow the description of the VOCUS model as representative of one of the classic approaches to compute saliency.[5] We start with introducing the

[4] In this chapter, we assume that the reader has basic knowledge on image processing, otherwise you find a short explanation of the basic concepts in the appendix of [4].

[5] While the description here is essentially the same as in [4], some improvements have been made in the meantime that are included here. Differences of VOCUS to the iNVT can be found in [4].

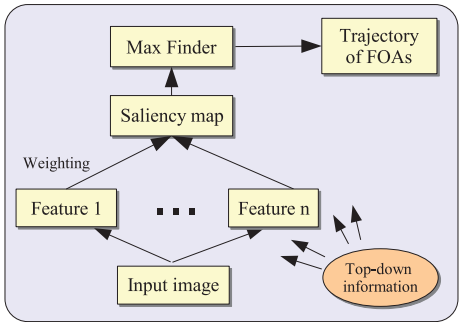


Fig. 4 General structure of most visual attention systems. Several features are computed in parallel and fused to a single saliency map. The maxima in the saliency map are the foci of attention (FOAs). Output is a trajectory of FOAs, ordered by decreasing saliency. Top-down cues may influence the processing on different levels.

bottom-up part (Sec. 3.2), followed by a description of the top-down visual search part (Sec. 3.3).

3.2 Bottom-up saliency

Bottom-up saliency is usually a combination from different feature channels. The most frequently used features in visual attention systems are intensity, color, and orientation. When image sequences are processed, also motion and flicker are important. The main concept to compute saliency are contrast computations that determine the difference between a center region and a surrounding region with respect to a certain feature. These contrasts are usually computed by *center-surround filters*. Such filters are inspired by cells in the human visual system, as the ganglion cells and the simple and complex cells introduced in Sec. 2.1. Cells with circular receptive fields are best modeled by Difference-of-Gaussian filters (cf. Fig. 1, right) while cells with elongated receptive fields are best modeled by Gabor functions. In practice, the circular regions are usually approximated by rectangles.

To enable the detection of regions of different extends, the center as well as the surround vary in size. Instead of directly adapting the filter sizes, the computations are usually performed on the layers of an image pyramid.

The structure of the bottom-up part of the attention system VOCUS is shown in Fig. 5. Let us regard the computation of the intensity feature in more detail now to understand the concept and then extend the ideas to the other feature channels.

3.2.1 Intensity Channel

Given a color input image I, this image is first converted to an image I_{Lab} in the Lab (or CIELAB) color space. This space has the dimension 'L' for lightness and 'a' and 'b' for the color-opponent dimensions (cf. Fig. 5, bottom right); it is perceptually

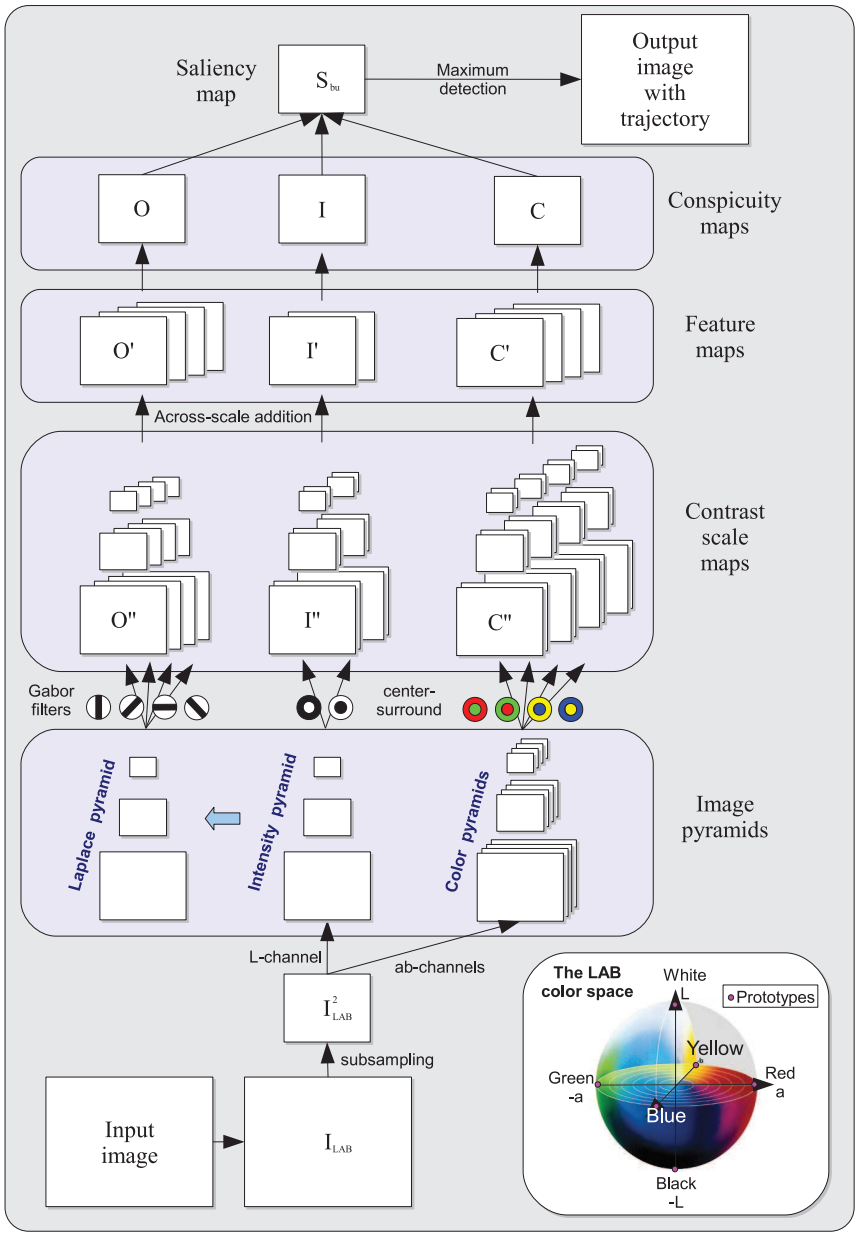


Fig. 5 The bottom-up saliency computation of the attention system VOCUS.

(a) Input image (b) Gaussian pyramid, s_0 to s_4

Fig. 6 The image which serves as demonstration example throughout this chapter (a) and the derived Gaussian image pyramid (b).

uniform, which means that a change of a certain amount in a color value is perceived as a change of about the same amount in human visual perception.

From I_{Lab}, a Gaussian pyramid is determined by successively smoothing the image with a Gaussian filter and subsampling it with a factor of 2 along each coordinate direction (see Fig. 6). In VOCUS, we use a 5×5 Gaussian kernel. The level of the pyramid determines the area that the center-surround filter covers: on high levels of the pyramid (fine resolution), small salient regions are detected while on low levels (coarse resolution), large regions obtain the highest response. In VOCUS, 5 pyramid levels (scales) are computed: I^s_{Lab}, $s \in \{0,..,4\}$. Level I^1_{Lab} is only an intermediate step used for noise reduction, all computations take place on levels 2 – 4.[6]

The intensity computations can be performed directly on the images I^s_L that originate from the 'L' channel of the LAB image. According to the human system, we determine two feature types for intensity: the on-center difference responding strongly to bright regions on a dark background, and the off-center difference vice versa. Note that it is important to treat both types separately and to not fuse them in a single map since otherwise it is not possible to detect bright-dark pop-outs, such as in Fig. 12. This yields 12 intensity scale maps $I''_{i,s,\sigma}$ with $i \in \{_{(\text{on}),(\text{off})}\}, s \in \{2,3,4\}, \sigma \in \{3,7\}$. A pixel (x,y) in one of the on-center scale maps is thus computed as:

$$\begin{aligned} I''_{on,s,\sigma}(x,y) &= center(I^s_L,x,y) - surround_\sigma(I^s_L,x,y) \\ &= I^s_L(x,y) - \frac{1}{(2\sigma+1)^2-1}\left(\sum_{i=-\sigma}^{\sigma}\sum_{j=-\sigma}^{\sigma} I^s_L(x+i,y+j) - I^s_L(x,y)\right) \end{aligned} \tag{1}$$

The off-center maps $I''_{\text{off},s,\sigma}(x,y)$ are computed equivalently by $surround - center$. The straight-forward computation of the surround value is quite costly, especially for large surrounds. To compute the surround value efficiently, it is convenient to use *integral images* [24].

[6] The number of levels that is reasonable depends on the image size as well as on the size of the objects you want to detect. Larger images and a wide variety of possible object sizes require deeper pyramids. The presented approach usually works well for images of up to 400 pixels in width and height in which the objects are comparatively small as in the example images of this chapter.

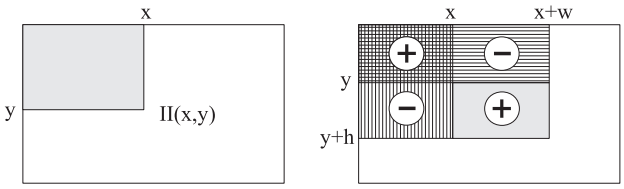


Fig. 7 Left: The integral image contains at $II(x,y)$ the sum of the pixel values in the shaded region. Right: the computation of the average value in the shaded region is based on four operations on the four depicted rectangles according to eq. 5.

The advantage of an integral image (or summed area table) is that when it is once created, the sum and mean of the pixel values of a rectangle of arbitrary size can be computed in constant time. An integral image II is an intermediate representation for the image and contains for a pixel position (x,y) the sum of all gray scale pixel values of image I above and left of (x,y), inclusive:

$$II(x,y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} I(x',y'). \tag{2}$$

The process is visualized in Fig. 7, left. The integral image can be computed recursively in one pass over the image with help of the cumulative sum s:

$$s(x,y) = s(x,y-1) + I(x,y) \tag{3}$$
$$II(x,y) = II(x-1,y) + s(x,y) \tag{4}$$

with $s(x,-1) = 0$ and $II(-1,y) = 0$. This intermediate representation allows to compute the sum of the pixel values in a rectangle F using four references (see Fig. 7 (right)):

$$\begin{aligned} F(x,y,h,w) = {} & II(x+w-1,y+h-1) - II(x-1,y+h-1) \\ & -II(x+w-1,y-1) + II(x-1,y-1). \end{aligned} \tag{5}$$

The '-1' elements in the equation are required to obtain a rectangle that includes (x,y). By replacing the computation of the surround in (1) with the integral computation in (5) we obtain:

$$I''_{on,s,\sigma}(x,y) = I^s_L(x,y) - \frac{F(x-\sigma,y-\sigma,2\sigma+1,2\sigma+1) - I^s_L(x,y)}{(2\sigma+1)^2 - 1} \tag{6}$$

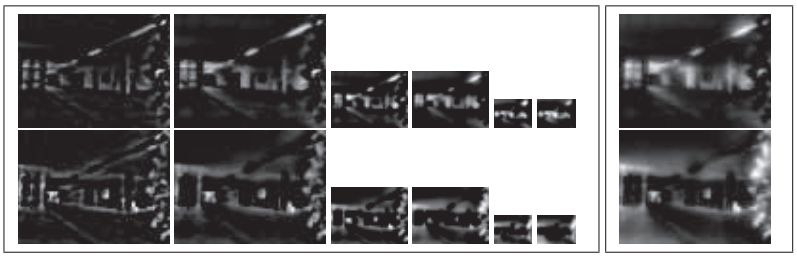


Fig. 8 Left: the 12 intensity scale maps $I''_{i,s,\sigma}$. First row: the *on-maps*. Second row: the *off-maps*. Right: the two intensity feature maps $I'_{(\text{on})}$ and $I'_{(\text{off})}$ resulting from the sum of the corresponding six scale maps on the left.

To enable this computation, one integral image has to be computed for each of the three pyramid levels $I^s_L, s \in \{2,3,4\}$. This pays off since then each surround can be determined by three simple operations. The intensity scale maps I'' are depicted in Fig. 8, left.

The six maps for each center-surround variation are summed up by *across-scale addition*: first, all maps are resized to scale 2 whereby resizing scale i to scale $i-1$ is done by bilinear interpolation. After resizing, the maps are added up pixel by pixel. This yields the intensity feature maps I':

$$I'_i = \bigoplus_{s,\sigma} I''_{i,s,\sigma}, \tag{7}$$

with $i \in \{(\text{on}),(\text{off})\}, s \in \{2,3,4\}, \sigma \in \{3,7\}$, and $\bigoplus$ denoting the across-scale addition. The two intensity feature maps are shown in Fig. 8, right.

3.2.2 Color Channel

The color computations are performed on the two-dimensional color layer I_{ab} of the Lab image that is spanned by the axes 'a' and 'b'. Besides its resemblance to human visual perception, the Lab color space fits particularly well as basis for an attentional color channel since the four main colors red, green, blue and yellow are at the end of the axes 'a' and 'b'. Each of the 6 ends of the axes that confine the color space serves as one prototype color, resulting in two intensity prototypes for white and black and four color prototypes for red, green, blue, and yellow.

For each color prototype, a color prototype image is computed on each of the pyramid levels 2 – 4. In these maps, each pixel represents the Euclidean distance to the prototype:

$$C^s_\gamma(x,y) = V_{max} - ||I^s_{ab}(x,y) - P_\gamma||, \qquad \gamma \in \{\text{R,G,B,Y}\}, \tag{8}$$

Fig. 9 Top: the color prototype images of scale s_2 for red, green, blue, yellow. Bottom: the corresponding color feature C'_γ maps which result after applying center-surround filters.

where V_{max} is the maximal pixel value and the prototypes P_γ are the ends of the 'a' and 'b' axes (thus, in an 8-bit image, we have $V_{max} = 255$ and $P_{\mathrm{R}} = (255,127), P_{\mathrm{G}} = (0,127), P_{\mathrm{B}} = (127,0), P_{\mathrm{Y}} = (127,255)$). The color prototype maps show to which degree a color is represented in an image, i.e., the maps in the pyramid P_{R} show the "redness" of the image regions: the brightest values are at red regions and the darkest values at green regions (since green has the largest distance to red in the color space). Analogical to the intensity channel, it is also important here to separate red-green and blue-yellow in different maps to enable red-green and blue-yellow pop-outs. The four color prototype images I^2_γ are displayed in Fig. 9, top.

On these pyramids, the color contrast is computed by on-center differences yielding $4*3*2 = 24$ color scale maps:

$$C''_{\gamma,s,\sigma} = center(C^s_\gamma, x, y) - surround_\sigma(C^s_\gamma, x, y), \tag{9}$$

with $\gamma \in \{\mathrm{R,G,B,Y}\}, s \in \{2,3,4\}$, and $\sigma \in \{3,7\}$. According to the intensity channel, the center is a pixel in the corresponding color prototype map, and the surround is computed according to eq. 6. The off-center-on-surround difference is not needed, because these values are represented in the opponent color pyramid. The maps of each color are rescaled to the scale 2 and summed up into 4 color feature maps C'_γ:

$$C'_\gamma = \bigoplus_{s,\sigma} C''_{\gamma,s,\sigma}. \tag{10}$$

Fig. 9, bottom shows the color feature maps for the example image.

3.2.3 Orientation Channel

The orientation maps are computed from *oriented pyramids*. An oriented pyramid contains one pyramid for each represented orientation (cf. Fig.10, left). Each of these pyramids highlights edges with this specific orientation. To obtain the oriented pyramid, first a Laplacian Pyramid is obtained from the Gaussian pyramid I^s_L by subtracting adjacent levels of the pyramid. The orientations are computed by *Gabor filters* which respond most to bar-like features according to a specified orien-

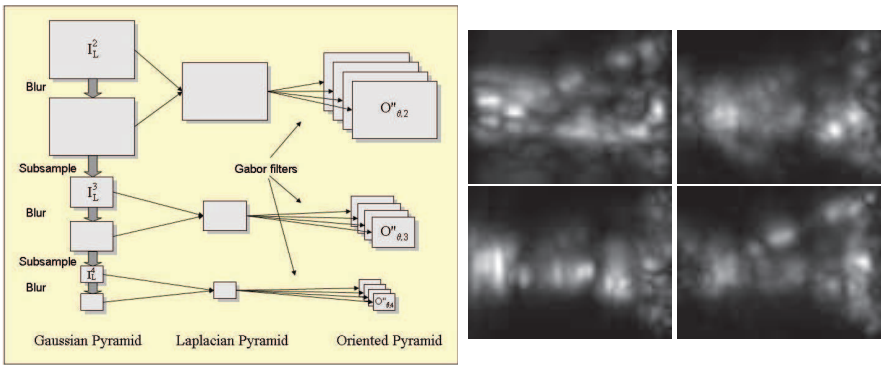


Fig. 10 Left: to obtain an oriented pyramid, a Gaussian pyramid is computed from the input image, then a Laplacian pyramid is obtained from the Gaussian pyramid by subtracting two adjacent levels and, finally, Gabor filters of 4 orientations are applied to each level of the Laplacian pyramid. Right: The four orientation feature maps $O'_{0^\circ}, O'_{45^\circ}, O'_{90^\circ}, O'_{135^\circ}$ for the example image.

tation. Gabor filters, which are the product of a symmetric Gaussian with an oriented sinusoid, simulate the receptive field structure of orientation-selective cells in the primary visual cortex (cf. 2.1). Thus, the Gabor filters replace the center-surround filters of the other channels.

Four different orientations are computed yielding $4 \times 3 = 12$ orientation scale maps $O''_{\theta,s}$, with the orientations $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ and scales $s \in \{2,3,4\}$. The orientation scale maps $O''_{\theta,s}$ are summed up by across-scale addition for each orientation, yielding four orientation feature maps O'_θ, one for each orientation:

$$O'_\theta = \bigoplus_s O''_{\theta,s}, \tag{11}$$

The orientation feature maps for the example image are depicted in Fig. 10, right.

3.2.4 Motion Channel

If image sequences are used as input for the attention system, motion is an important additional feature. It can be computed easily by determining the optical flow field. Here, we use a method based on total variation regularization that determines a dense optical flow field and is capable to operate in real-time [29]. If the horizontal u and the vertical v component of the optical flow are visualized as images, the center-surround filters can be applied to these images directly. By applying on- as well as off-center filters to both images, we achieve four motion maps for each scale s which we call $M''_{\vartheta,s}$, with $\vartheta = \{\text{right, left, up, down}\}$. After accross-scale addition we obtain four motion feature maps

Fig. 11 The motion feature maps M' for a scene in which a ball rolls from right to left through the image. From left to right: example frame, motion maps $M'_{\text{right}}, M'_{\text{left}}, M'_{\text{up}}, M'_{\text{down}}$.

$$M'_{\vartheta} = \bigoplus_{s} M''_{\vartheta,s}. \tag{12}$$

An example for a sequence in which a ball rolls from right to left through the image is displayed in Fig. 11. In videos, motion itself is not necessarily salient, but the contrast of the motion in the current frame to the motion (or absence of motion) in previous frames. Itti and Baldi describe in their surprise theory how such temporal saliency can be integrated into a computational attention system [8].

3.2.5 The Uniqueness Weight

Up to now, we have computed local contrasts for each of the feature channels. While contrast is an important aspect of salient regions, they additionally have an important property: they are rare in the image, in the best case unique. A red ball on grass is very salient, while it is much less salient among other red balls. That means, we need a measure for the uniqueness of a feature in the image. Then, we can strengthen maps with rare features and diminish the influence of maps with omnipresent features.

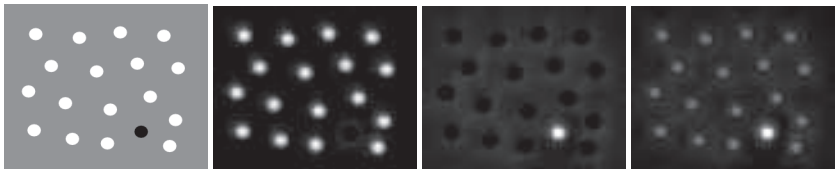
A simple method to determine the uniqueness of a feature is to count the number of local maxima m in a feature map X. Then, X is divided by the square root of m:

$$W(\mathrm{X}) = \mathrm{X}/\sqrt{m}, \tag{13}$$

In practice, it is useful to only consider maxima in a pre-specified range from the global maximum (in VOCUS, the threshold is 50% of the global maximum of the map). Fig. 12 shows how the uniqueness weight enables the detection of pop-outs. Other solutions to determine the uniqueness are described in [10, 9].

3.2.6 Normalization

Before the feature maps can be fused, they have to be normalized. This is necessary since some channels have more maps than others. Let us first understand why this step is not trivial. The easiest solution would be to normalize all maps to a fixed range. This method goes along with a problem: normalizing maps to a fixed range removes important information about the magnitude of the maps. Assume that one



(a) Input image (b) $I'_{(\text{on})}$ (c) $I'_{(\text{off})}$ (d) $I = \sum W(I')$

Fig. 12 The effect of the uniqueness weight function W (eq. 13). The off-center intensity feature map $I'_{(\text{off})}$ has a higher weight than the on-center intensity feature map $I'_{(\text{on})}$, because it contains only one strong peak. So this map has a higher influence and the region of the black dot pops out in the conspicuity map I.

intensity and one orientation map belonging to an image with high intensity but low orientation contrasts are to be fused into one saliency map. The intensity map will contain very bright regions, but the orientation map will show only some moderately bright regions. Normalizing both maps to a fixed range forces the values of the orientation maps to the same range as the intensity values, ignoring that orientation is not an important feature in this case.

A similar problem occurs when dividing each map by the number of maps in this channel: imagine an image with equally strong intensity and color blobs. A color map would be divided by 4, an intensity map only by 2. Thus, although all blobs have the same strength, the intensity blobs would obtain a higher saliency value.

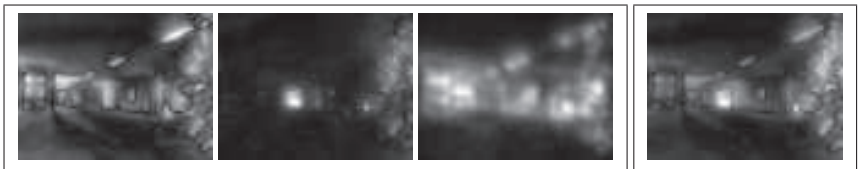
Instead, we propose the following normalization technique: To fuse the maps $\mathbf{X} = \{X_1, .., X_k\}$, determine the maximum value M of all $X_i \in \mathbf{X}$ and normalize each map to the range [0..M]. Normalization of map X_i to the range [0..M] will be denoted as $N_{[0..M]}(X_i)$ in the following.

3.2.7 The Conspicuity Maps

The next step in the saliency computation is the generation of the *conspicuity maps*. The term conspicuity is usually used to denote feature specific saliency. To obtain the maps, all feature maps of one feature dimension are weighted by the uniqueness weight W, normalized, and combined into one conspicuity map, yielding map I for intensity, and C for color, O for orientation, and M for motion:

$$\begin{aligned}
I &= \sum_i N_{[0..M_i]}(W(I'_i)), & M_i &= maxvalue_i(I'_i), & i &\in \{\text{on,off}\},\\
C &= \sum_\gamma N_{[0..M_\gamma]}(W(C'_\gamma)), & M_\gamma &= maxvalue_\gamma(C'_\gamma), & \gamma &\in \{\text{R,G,B,Y}\},\\
O &= \sum_\theta N_{[0..M_\theta]}(W(O'_\theta)), & M_\theta &= maxvalue_\theta(O'_\theta), & \theta &\in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\},\\
M &= \sum_\vartheta N_{[0..M_\vartheta]}(W(M'_\vartheta)), & M_\vartheta &= maxvalue_\vartheta(C'_\vartheta), & \vartheta &\in \{\text{right, left, up, down}\},
\end{aligned} \tag{14}$$

where W is the uniqueness weight, N the normalization and *maxvalue* the function that determines the maximal value from several feature maps. The conspicuity maps I, C, and O are illustrated in Fig. 13 (a) - (c).[7]



(a) Consp. map I (b) Consp. map C (c) Consp. map O (d) Saliency map S

Fig. 13 The three conspicuity maps for intensity, color, and orientation, and the saliency map.

3.2.8 The Saliency Map and Focus Selection

Finally, the conspicuity maps are weighted and normalized again, and summed up to the bottom-up saliency map S:

$$S_{\text{bu}} = \sum_{X_i} N_{[0..M_C]}(W(X_i)), \quad M_C = maxvalue(I,C,O,M), \quad X_i \in \{I,C,O,M\}. \quad (15)$$

The saliency map for our (static) example is illustrated in Fig. 13 (d). While it is sometimes sufficient to compute the saliency map and provide it as output, if is often required to determine a trajectory of image locations which resembles eye movements. To obtain such a trajectory from the saliency map, it is common practice to determine the local maxima in the saliency map, ordered by decreasing saliency. These maxima are usually called *Focus of Attention (FOA)*. Here, we first discuss the standard, biologically motivated approach to find FOAs, then we introduce a simple, computationally convenient solution.

The standard approach to detect FOAs in the saliency map is via a *Winner-Take-All Network (WTA)* (cf. Fig. 14) [13]. A WTA is a neural network that localizes the most salient point x_i in the saliency map. Thus, it represents a neural maximum finder. Each pixel in the saliency map gives input to a node in the input layer. Local competitions take place between neighboring units and the more active unit transmits the activity to the next layer. Thus, the activity of the maximum will reach the top of the network after $k = log_m(n)$ time steps if there are n input units and local comparisons take place between m units. However, since it is not the value of the maximum that is of interest but the location of the maximum, a second pyramid out of auxiliary units is attached to the network. It has a reversed flow of information

[7] Since input is a static image, the motion channel is empty and omitted here.

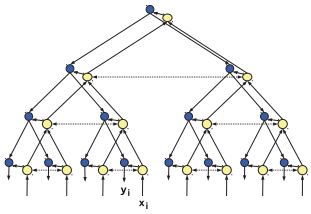


Fig. 14 A Winner-Take-All network (WTA) is a neural maximum finder that detects the most salient point x_i in the saliency map. Fig. redrawn from [13].

and "marks" the path of the most active unit. An auxiliary unit is activated if it receives excitation from its main unit as well as from the auxiliary unit at the next higher layer. The auxiliary unit y_i, corresponding to the most salient point x_i, will be activated after at most $2log_m(n)$ time steps. On a parallel architecture with locally connected units, such as the brain, this is a fast method to determine the maximum. It is also a useful approach on a parallel computer architecture, such as a graphics processing unit (GPU). However, if implemented on a serial machine, it is more convenient to simply scan the saliency map sequentially and determine the most salient value. This is the solution chosen for VOCUS.

When the most salient point has been found, the surrounding salient region is determined by *seeded region growing*. This method starts with a seed, here the most salient point, and recursively finds all neighbors with similar values within a certain range. In VOCUS, we accept all values that differ at most 25% from the value of the seed. We call the selected region *most salient region (MSR)*. Some MSRs are shown in Fig. 18. For visualization, the MSR is often approximated by an ellipse (cf. Fig. 22).

To allow the FOA to switch to the next salient region with a WTA, a mechanism called *inhibition of return (IOR)* is used. It inhibits all units corresponding to the MSR by setting their value to 0. Then, the WTA activates the next salient region. If it is desired that the FOA may return to a location after a while, as it is the case in human perception, the inhibition is only active for a predefined time and diminishes after that. If no WTA is used, it is more convenient to directly determine all local maxima in the saliency map that exceed a certain threshold (in VOCUS, 50% of the global maximum), sort them by saliency value, and then switch the focus from one to the next. This also prevents border effects that result from inhibition when the focus returns to the borders of an inhibited region.

3.3 Visual Search with Top-down Cues

While bottom-up saliency is an important part of visual attention, top-down cues are even more important in many applications. Bottom-up saliency is useful if no pre-knowledge is available, but the exploitation of available pre-knowledge naturally increases the performance of every system, both biological and technical. One of the best investigated aspects of top-down knowledge is visual search. In visual search,

a target shall be located in the image, e.g. a cup, a key-fob, or a book. Here, we describe the visual search mode of the VOCUS model. Learning the appearance of the target from a training image and searching for the target in a test image are both directly integrated into the previously described model. Top-down and bottom-up cues interact to achieve a joint focus of attention.

An overview of the complete algorithm for visual search is shown in Fig. 15.

Learning mode (input: training image and region of interest (ROI)):
- compute bottom-up saliency map S_{bu}
- determine most salient region (MSR) in ROI of S_{bu}
- for each feature and conspicuity map X_i
  - compute target descriptor value v_i

Search mode (input: test image and target descriptor v):
- compute bottom-up saliency map S_{bu}
- compute top-down saliency map S_{td}:
  - compute excitation map $E = \sum_i (v_i * X_i)$ $\quad \forall i : v_i > 1$
  - compute inhibition map $I = \sum_i ((1/v_i) * X_i)$ $\quad \forall i : v_i < 1$
  - compute top-down saliency map $S_{\text{td}} = E - I$
- compute saliency map $S = t * S_{\text{td}} + (1-t) * S_{\text{bu}}$ with $t \in [0..1]$
- determine most salient region(s) in S

Fig. 15 The algorithm for visual search

3.3.1 Learning Mode

"Learning" in our application means to determine the object properties of a specified target from one or several training images. In learning mode, the system is provided with a region of interest (ROI) containing the target object and learns which features distinguish the target best from the remainder of the image. For each feature, a value is determined that specifies to what amount the feature distinguishes the target from its background. This yields a target descriptor **v** which is used in search mode to weight the feature maps according to the search task (cf. Fig. 16).

The input to the system in learning mode is a training image and a region of interest (ROI). The ROI is a rectangle which is usually determined manually by the user but might also be the output of a classifier that specifies the target. Inside the ROI, the *most salient region (MSR)* is determined by first computing the bottom-up saliency map and, second, determining the most salient region within the ROI. This method enables the system to determine automatically what is important in a specified region and to ignore the background. Additionally, it makes the system stable since usually the same MSR is computed, regardless of the exact coordinates of the rectangle. So the system is independent of variations the user makes when determining the rectangle manually and it is not necessary to mark the target exactly.

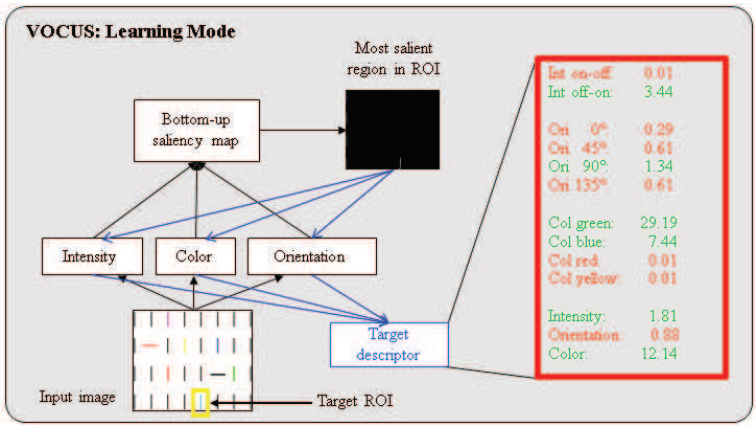


Fig. 16 In learning mode, VOCUS determines the *most salient region (MSR)* within the *region of interest (ROI)* (yellow rectangle). A target descriptor **v** is determined by the ratio of MSR vs. background for each feature and conspicuity map. Values $v_i > 1$ (green) are target relevant and used in search mode for excitation, values $v_i < 1$ (red) are used for inhibition.

Next, a *target descriptor* **v** is computed. It has one entry for each feature and each conspicuity map X_i. The values v_i indicate how important a map is for detecting the target and are computed as the ratio of the mean target saliency and the mean background saliency:

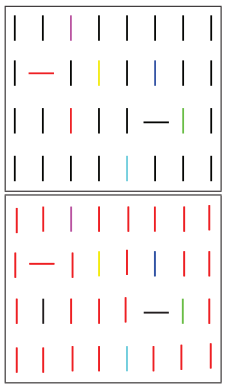
$$v_i = m_{i,(MSR)} / m_{i,(X_i - MSR)}, \qquad i \in \{1, ..., 13\}, \tag{16}$$

where $m_{i,(MSR)}$ denotes the mean intensity value of the pixels in the MSR in map X_i, showing how strong this map contributes to the saliency of the region of interest, and $m_{i,(X_i - MSR)}$ is the mean of the remainder of the image in map X_i, showing how strong the feature is present in the surroundings.

Fig. 16 shows the target descriptor for a simple example. Values larger than 1 (green) are features that are relevant for the target while features smaller than 1 (red) are more present in the background and are used for inhibition.

Learning features of the target is important for visual search but if these features also occur in the environment they might be of not much use. For example, if a red target is placed among red distractors it is not reasonable to consider color for visual search, although red might be the strongest feature of the target. In VOCUS, not only the target's features but also the features of the background are considered and used for inhibition. This method is supported by psychophysical experiments, showing that both excitation and inhibition of features are important in visual search. Fig. 17 shows the effect of background information on the target descriptor.

Note that it is important that target objects are learned in their typical environment since otherwise their appearance with respect to the background cannot be represented adequately. Fig. 18 shows some typical training images and the regions that the system determined to represent the target.



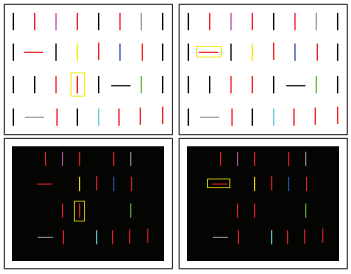
Feature	target vector **v** (top)	target vector **v** (bottom)
intensity on/off	0.01	0.01
intensity off/on	9.13	13.17
orientation 0°	20.64	**29.84**
orientation 45°	1.65	1.96
orientation 90°	0.31	0.31
orientation 135°	1.65	1.96
color green	0.00	0.00
color blue	0.00	0.01
color red	**47.60**	10.29
color yellow	**36.25**	9.43
conspicuity I	4.83	6.12
conspicuity O	7.90	**11.31**
conspicuity C	**17.06**	2.44

Fig. 17 Effect of background information on the target vector. Left: the same target (red horizontal bar, 2nd in 2nd row) in different environments: all vertical bars are black (top) resp. red (bottom). Right: the target vectors (most important values printed in bold face). In the upper image, red is the most important feature. In the lower image, surrounded by red distractors, red is no longer the prime feature to detect the bar but orientation is (image from [4]).

Fig. 18 Top: some training images with targets (name plate, fire extinguisher, key fob). Bottom: The part of the image that was marked for learning (region of interest (ROI)) and the contour of the region that was extracted for learning (most salient region (MSR)) (images from [4]).

3.3.2 Several Training Images

Learning weights from one single training image yields good results if the target object occurs in all test images in a similar way, i.e., the background color is similar and the object always occurs in a similar orientation. These conditions often occur if the objects are fixed elements of the environment. For example, name plates or fire extinguishers are within the same building usually placed on the same kind of wall, so the background has always a similar color and intensity. Furthermore, since the object is fixed, its orientation does not vary and thus it makes sense to learn that fire extinguishers usually have a vertical orientation.



Feature	weights for red bar v,b	h,b	v,d	h,d	average
int on/off	0.00	0.01	**8.34**	**9.71**	0.14
int off/on	**14.08**	**10.56**	0.01	0.04	0.42
ori 0°	1.53	**21.43**	0.49	**10.52**	3.61
ori 45°	2.66	1.89	1.99	2.10	2.14
ori 90°	**6.62**	0.36	**5.82**	0.32	1.45
ori 135°	2.66	1.89	1.99	2.10	2.14
col green	0.00	0.00	0.00	0.00	0.00
col blue	0.00	0.00	0.01	0.01	0.00
col red	**18.87**	**17.01**	**24.13**	**24.56**	**20.88**
col yellow	**16.95**	**14.87**	**21.21**	**21.66**	**18.45**
consp I	7.45	5.56	3.93	4.59	5.23
consp O	4.34	7.99	2.87	5.25	4.78
consp C	4.58	4.08	5.74	5.84	5.00

Fig. 19 Influence of averaging the target descriptor from several training images. Left: four training examples to learn red bars of horizontal and vertical orientation and on different backgrounds. The target is marked by the yellow rectangle. Right: The learned target descriptors. Column 2–5: the weights for a single training image (v=vertical,h=horizontal,b=bright background,d=dark background). The highest values are highlighted in bold face. Column 6: average vector. Color is the only stable feature (example from [4]).

To automatically determine which object properties are general and to make the system robust against illumination and viewpoint changes, the target descriptor $\mathbf{v}$ can be computed from several training images by computing the average descriptor from n training images with the geometric mean:

$$v_i = \sqrt[n]{\prod_{j=1}^{n} v_{ij}}, \qquad i \in \{1,..,13\} \tag{17}$$

where v_{ij} is the i-th feature in the j-th training image. If one feature is present in the target region of some training images but absent in others, the average values will be close to 1 leading to only a low activation in the top-down map. Fig. 19 shows the effect of averaging target descriptors on the example of searching for red bars in different environments.

In practice, best results are usually obtained by only two training images. In complicated image sets, up to 4 training images can be useful (see experiments in [4]). Since not each training image is equally useful, it can be preferable to select the training images automatically from a set of training images. An algorithm for this issue is described in [4].

3.3.3 Search Mode

In search mode, we search for a target by means of the previously learned target descriptor. The values are used to excite or inhibit the feature and conspicuity maps

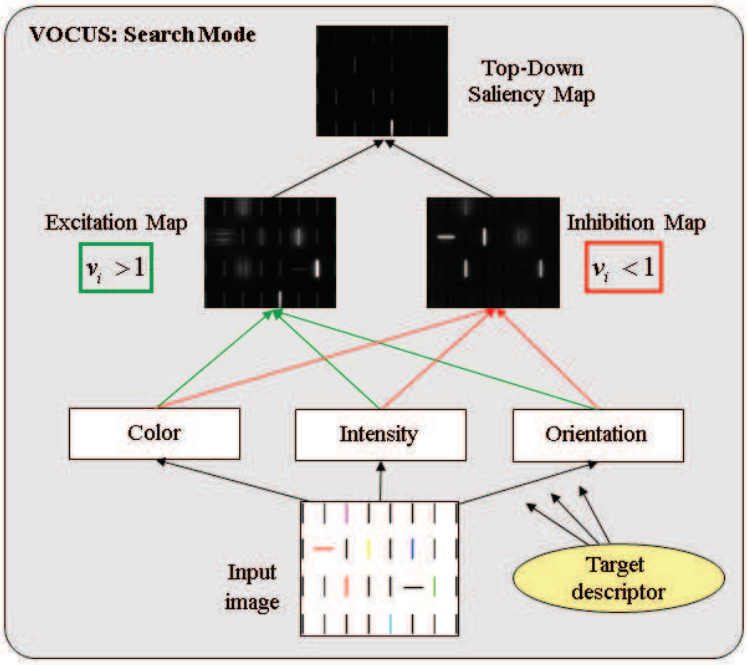


Fig. 20 Computation of the top-down saliency map S_{td} that results from an excitation map E and an inhibition map I. These maps result from the weighted sum of the feature and conspicuity maps, using the learned target descriptor.

according to the search task. The weighted maps contribute to a top-down saliency map highlighting regions that are salient with respect to the target and inhibiting others. Fig. 20 illustrates this procedure.

The excitation map E is the weighted sum of all feature and conspicuity maps X_i that are important for the target, namely the maps with weights greater than 1:

$$E = \sum_{i:\, v_i > 1} (v_i * X_i). \tag{18}$$

The inhibition map I collects the maps in which the corresponding feature is less present in the target region than in the remainder of the image, namely the maps with weights smaller than 1:[8]

$$I = \sum_{i:\, v_i < 1} ((1/v_i) * X_i). \tag{19}$$

[8] Entries with value 1 are ignored since they indicate that the mean saliency of the target region is exactly the same as the mean saliency of the surrounding; such a feature is completely useless for detecting the target. However, in practice this usually does not occur unless a feature is not present at all, e.g., color is not present in a gray-scale image and the color weights are set to 1.

The excitation and inhibition map are not normalized to the same range since we want to preserve the differences among the maps.

The top-down map is obtained by subtracting the inhibition map from the excitation map:

$$S_{\text{td}} = E - I. \tag{20}$$

After subtraction, negative values are clipped to 0. Fig. 20 shows that both, excitation and inhibition are important to find a target: when searching for the cyan vertical bar, the excitation map shows bright values for the cyan bar but the brightest region for the green bar. However, green contains also yellow which is inhibited for a cyan target. Thus in the resulting top-down map, only the cyan bar is salient.

If the task is pure visual search for a target, the top-down saliency map can be directly used to determine the focus of attention.[9] This is done equivalently to sec. 3.2.8. However, if bottom-up cues shall be regarded additionally, the bottom-up and the top-down saliency map have to be fused. This will be discussed in the next section.

3.3.4 Bottom-up and Top-down Cues Compete for Attention

In human perception, bottom-up and top-down cues compete for attention all of the time. Depending on how engrossed in a task you are, the influences of bottom-up and top-down vary. The introductory city-visiting example illustrates this: without a clear task, the salient street performers attract your gaze. When you start to actively look for the train station, your top-down attention is focusing on street signs. Finally, the fire alarm is salient enough to override the task and captures your attention.

Consequently, it is important for a technical system to know what the overall tasks are, which one the most important task is at the moment, and how important it is. Depending on such pre-knowledge, the influence of bottom-up and top-down factors might be determined. After obtaining such a factor, the bottom-up and top-down saliency map are weighted accordingly and finally fused to a global saliency map S. To make the maps comparable, S_{td} is normalized in advance to the same range as S_{bu}:

$$S = (1-t) * S_{\text{bu}} + t * N_{[0..M_S]} S_{\text{td}}, \qquad M_S = maxvalue(S_{\text{bu}}). \tag{21}$$

Here, $t \in [0..1]$ is the top-down factor that determines the amount of top-down influence. Determining t is not trivial. Probably the best solution is to learn it while performing some tasks on a real system, but this is beyond the scope of this article. Note that a simple solution for a technical system is to not fuse bottom-up and top-down saliency but to process them independently. Bottom-up salient regions might be fed to an object recognition module that recognizes the objects, builds a semantic map of the environment with object annotations, and successively improves the

[9] Note that in human perception, bottom-up cues always play a role and thus should be considered if similarity to human perception is desired.

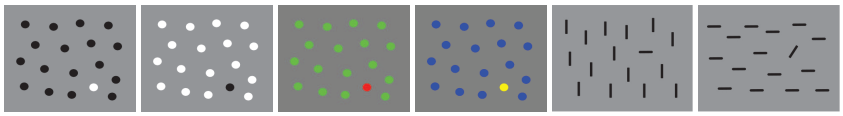


Fig. 21 Typical pop-out images. Attention systems should be able to detect the outliers.

background knowledge of the system, while top-down cues can be used to solve the current task by searching for desired objects.

4 Evaluation of computational attention systems

The evaluation of computational attention systems can be done from a psychophysical perspective, e.g. by comparing their results with human perception, or from a technical perspective, e.g. by measuring the success in an application.

When considering bottom-up systems of attention, the first step is to determine whether the system is able to detect pop-outs in the dimension of the implemented features. These tests are important to ensure the basic capabilities of the systems and are suitable to reveal their strengths and limitations. Thus, a system with the standard features intensity, color, and orientation should be able to detect popouts as the ones in Fig. 21. Hereby, the saliency of the target depends on the similarity to the distractors, the more it differs, the higher the saliency. Thus, a target that differs only slightly from the distractors might not be detected with the first fixation. This is in accordance with the psychophysical findings that the more similar target and distractors are, the slower the visual search (cf. Sec. 2.2)

The evaluation on artificial patterns is only the first step, testing on natural images is important too. Here, it is usually less clear which region shall be salient, top-down influences play a larger role and saliency depends stronger on the context and of preknowledge of the observer. A possibility for evaluation is to compare the output of the system with human eye movement data (see also Sec. 7 and chapter 11, Sec. 3.2.2). Note that a computational attention system can only roughly approximate such eye movement trajectories since the top-down cues that influence human perception are hardly possible to model in such a general scenario and thus the systems usually operate in bottom-up mode. It is however possible to compare different attention system based on such data.

An alternative that occurred recently in the computer vision community is the evaluation on image databases with salient objects, manually labeled by different users [14]. Note however that the database in [14] contains many close-up views of objects that cover a large portion of the image, a case for which the human attention system is not designed. In contrast, the task of human attention is to direct the gaze to a small region in a complex scene which is afterwards investigated in detail. Thus, a system as the one described here is designed to operate on scene images rather than on close-up views of objects and might have to be adapted accordingly to work

on the above database. A similar approach for evaluation was used by Elazary and Itti, who used 24 836 pictures of natural scenes from the LabelMe database, in which objects were manually marked and labeled by a large population of users. They found that the hot spots in the saliency map predict the locations of objects significantly above chance [3].

From a technical point-of-view it is not necessarily important that a computational attention system operates similar to human perception, as long as the outcome is useful for an application. Two applications in which attention system are applied are mentioned in sec. 5. But even in these cases, a system should be able to detect outliers as in Fig. 21 since this belongs to the basic capabilities of visual attention systems.

The evaluation of top-down systems is easier. Here, the task is clearly specified and it can be determined easily if a target was detected or not. Note, that a top-down attention system is no object recognizer, that means it cannot decide whether an object is present in an image or not. It can simply determine locations that are likely to obtain the target, usually in form of a trajectory of locations. Thus, instead of determining a detection rate, it is more reasonable to determine the *hit number*, i.e. the number of the focus that is on the target. A hit number of 1 is best and means that the first focus of attention was on the target. An example of the evaluation of visual search with VOCUS is displayed in Fig. 22.

Target	# test im.	av. hit number [%]
Fire extinguisher	46	1.09
Key fob	30	1.23

Fig. 22 Left: Average hit number of VOCUS for two targets on a set of test images. The target descriptors were computed from two training images each (examples of training images cf. Fig. 18). Right: Two example test images with foci of attention (red ellipses) (example from [4]).

5 Applications in computer vision and robotics

In the introduction, we have pointed out the importance of attentional selection for tasks that deal with large amounts of image data. Especially in the field of autonomous mobile robots, the concept of visual attention has increasingly gained interest during the last decade. A large number of EU projects on cognitive robotics has been launched, e.g. the projects MACS, CogVis, POP, and SEARISE. In many of these projects, visual attention has been used as perception module.

We will concentrate here on two applications of visual attention systems. A broader overview can be found in [6]. The first application that we will introduce is visual robot localization. Here, a robot has to determine its position in the world by

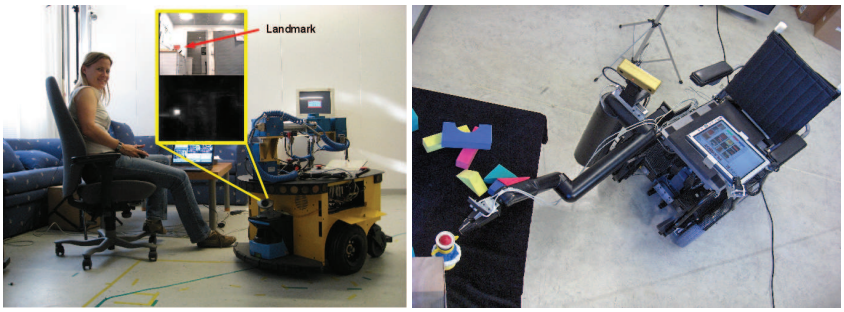
interpreting its sensor data. When a camera is used as sensor, this is usually done by detecting visual landmarks in the environment and computing the robot position based on the position estimation of the landmarks. An important property of landmarks is the redetectability in frames that are taken from different viewpoints. Using salient regions as landmarks is a natural way of exploiting that salient regions are "special" in an environment and, thus, easy to redetect. An example of a typical salient landmark is a fire extinguisher. As part of the EU project NEUROBOTICS, we have used salient visual landmarks for simultaneous localization and mapping (SLAM) [5]. This task is more difficult than pure localization since the robot initially does not know anything about its environment and has to build a map and localize itself inside the map at the same time. We have detected salient regions with VOCUS, tracked them over several frames to determine the most stable ones and to determine their 3D position, and stored them as landmarks in a database. At every time step, currently seen salient regions are compared with landmarks from the database to enable the robot to detect that it has returned to a previously visited location (loop closing). This is an especially important step in SLAM to correct accumulated position errors. A picture of the process is displayed in Fig. 23, left.

Another application is the PlayBot project, lead by Prof. John K. Tsotsos from York university, Canada [18].[10] Goal of the project is to develop a smart wheelchair to support disabled children. The wheelchair has a display as easily accessible user interface which shows pictures of places and toys. Once a task like "go to table, point to toy" is selected, the system drives to the selected location and searches for the specified toy, using mechanisms based on visual attention (see Fig. 23, (b)).

6 Summary

Computational attention systems are inspired by human perception and aim to detect the most promising regions in images. While computational attention systems already do a good job in bottom-up saliency computation, many open questions remain in the field of top-down attention. All kinds of background knowledge about the context, the current situation, the layout of the scene, and the specification of the current task influence the visual processing in humans and should therefore also be integrated into a technical system. The more technical systems advance, the more urgent the need for preprocessing modules such as attention systems that prioritize the data and enable efficient processing with limited resources. Especially in the field of autonomous robots such a mechanism is important to facilitate the decision which actions to perform next.

[10] More on http://web.me.com/john.tsotsos/Applications/Playbot.html



(a) Visual SLAM (b) Object manipulation in PlayBot

Fig. 23 Two application scenarios for visual attention systems: (a) Attentional landmarks for visual SLAM (simultaneous localization and mapping) at the Royal Institute of Technology (KTH) in Stockholm: robot Dumbo corrects its position estimate by redetecting a salient landmark based on the attention system VOCUS. The yellow rectangle shows the currently seen frame with a landmark (top) and the corresponding saliency map (bottom) [5] (Fig. from http://www.iai.uni-bonn.de/∼frintrop/research.html). (b) PlayBot: a visually guided robotic wheelchair for disabled children. The selective tuning model of visual attention supports the detection of objects of interest (Fig. fromhttp://www.cse.yorku.ca/∼playbot).

7 Open Source code, databases, and further reading

Open Source code:

- The iLab Neuromorphic Vision C++ Toolkit (iNVT, pronounced “invent”) from the group of Laurent Itti is probably the best known and most distributed Open Source code for computational attention systems [10]. It includes the surprise model for temporal saliency [8] and is available at http://ilab.usc.edu/toolkit/.
- The SaliencyToolbox from Dirk B. Walther [25] is a more compact reimplementation of iNVT in Matlab: http://www.saliencytoolbox.net/
- The original VOCUS source code is not freely available, but a reimplementation of the bottom-up part (in C++) can be found http://sourceforge.net/projects/-openvolksbot/
- The AIM model (Attention based on Information Maximation) is an attention system based on information theory. It determines the self-information of a center region with respect to a global surround [1]. Matlab code is available at: http://www-sop.inria.fr/members/Neil.Bruce
- For implementing an own attention system, it is convenient to use the Open Source Computer Vision Library OpenCV that contains many basic techniques, from displaying images over computing pyramids to converting images to other color spaces: http://sourceforge.net/projects/opencvlibrary.

Databases:

Several databases are available for testing and evaluating visual attention system:

- Image databases of popout search arrays and natural images can be found on the websites of the iLab: http://ilab.usc.edu/imgdbs/
- Eye tracking data from 20 test persons on 120 still images can be found on: http://www-sop.inria.fr/members/Neil.Bruce/
- Eye-tracking data from human volunteers watching complex video stimuli are available from the CRCNS (Collaborative Research in Computational Neuroscience) data sharing website: http://crcns.org/data-sets/eye
- The MSRA Salient Object Database contains 25000 images with manually labeled salient objects: http://research.microsoft.com/en-us/um/people/jiansun/ - SalientObject/salient_object.htm. For a subset of 1000 images, binary maps of the salient objects are available as ground truth: http://ivrg.epfl.ch/supplementary_-material/RK_CVPR09

Further reading:

More about the human visual system can be found in the books of Palmer [16] or Kandel et al. [12]. The psychology of attention and details on many psychological attention models are described in a book by Pashler [17] and in the chapter "Attention" by Bundesen & Habekost in the Handbook of Cognition [2]. A description of the social aspects of attention can be found later in this book in chapter 8, section 5.4.1. Wolfe has written a comprehensive article that contains everything you ever wanted to know about visual search [27]. One of the first computational models of visual attention was introduced by Koch and Ullman in 1985 with a detailed description of the winner-take-all approach [13]. The basic paper that describes the widely used computational attention model by the group of Laurent Itti in a comprehensive manner is [10]. Recently, several groups have used information-theoretic approaches to determine visual saliency [8, 1, 7]. The latter also tackle the aspect of top-down saliency for object recognition by determining salient features that best distinguish a visual class from other classes [7]. Top-down information in the form of knowledge about the scene and its visual layout was used by Torralba et al. to guide visual attention to relevant parts of an image [19]. A survey on computational attention systems that aims to bridge the gap between the research on human and computational visual attention can be found in [6].

Research papers on computational attention appear on conferences and in journals of many different areas, e.g. cognitive perception, computer vision, and cognitive robotics. Important journals for cognitive aspects of attention are "Attention, Perception, and Psychophysics" and the "Journal of Vision". In the technical fields, much work can be found on workshops on cognitive systems that usually take place as satellites of big conferences, such as the "International Symposium on Attention in Cognitive Systems" at IJCAI 2011. Journal articles appear e.g. in "Computer Vision and Image Understanding" and in the "IEEE Transactions on Pattern Analysis

and Machine Intelligence", or, if related to robotics, in the "IEEE Transactions on Robotics" and the "Robotics and Autonomous Systems".

8 Questions

The following questions shall help you to think more deeply about certain important aspects of attention systems, leading hopefully to a better understanding of the abilities and limitations of such approaches.

- Which objects of the following list are likely to be detected with a bottom-up attention system and which are not: a traffic sign, a glass, a large object among small ones, an apple on the table, an apple in a box full of apples?
- You notice that the attention system detects very small salient regions on your test images. How could you adapt the attention system to detect larger objects as well? What could you do if you do not have access to the source code and you can only adapt the input image itself?
- Why is the arithmetic mean not an adequate alternative for eq. 17? Tip: consider two training images with $v_i = 0.5$ and $v_i = 2$ respectively, for feature map i. Which value would you expect and what do you get by arithmetic/geometric mean?
- What happens if you search for a target object with the top-down attention system in an image where the target is not present?
- How does an attention system differ from a standard interest point detector such as the Difference of Gaussian detector or the Harris corner detector? How does a top-down attention system differ from an object recognition module?

9 Glossary

- Bottom-up attention: one of the factors that guide human attention (the other is top-down attention). Bottom-up attention is purely data-driven and guides the gaze to salient regions in a scene. Indicators that attract bottom-up attention are strong contrasts and the uniqueness of a region.
- Center-surround filters: the main concept in visual attention systems to detect contrasts. They are inspired by on-center and off-center cells of the human visual system.
- Saliency: The quality of a region to stand out relative to its surround.
- Top-down attention: one of the factors that guide human attention (the other is bottom-up attention). Top-down attention is driven by cognitive factors such as pre-knowledge, context, expectations, motivations, and current goals. One of the best investigated areas of top-down attention is visual search.

- Visual search: the task to find an item in a scene. It is one of the best investigated parts of top-down attention. Visual search experiments are used frequently in cognitive sciences to investigate the human visual system.

References

1. N. D. B. Bruce and J. K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):1–24, 2009.
2. C. Bundesen and T. Habekost. Attention. In K. Lamberts and R. Goldstone, editors, *Handbook of Cognition*. London: Sage Publications, 2005.
3. L. Elazary and L. Itti. Interesting objects are visually salient. *J. of Vision*, 8(3:3), 2008.
4. S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, volume 3899 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer Berlin/Heidelberg, 2006.
5. S. Frintrop and P. Jensfelt. Attentional landmarks and active gaze control for visual SLAM. *IEEE Trans. on Robotics, Special Issue on Visual SLAM*, 24(5), Oct 2008.
6. S. Frintrop, E. Rome, and H. I. Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Trans. on Applied Perception*, 7(1), 2010.
7. D. Gao, S. Han, and N. Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 31(6), 2009.
8. L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295–1306, 2009.
9. L. Itti and C. Koch. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169, Jan 2001.
10. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
11. W. James. *The Principles of Psychology*. Dover Publications, New York, 1890.
12. E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Essentials of Neural Science and Behavior*. McGraw-Hill/Appleton & Lange, 1996.
13. C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219–227, 1985.
14. T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.
15. M. Douma, curator. Color Vision and Art. Retrieved Nov 2010 from http://webexhibits.org/colorart/ganglion.html, 2008.
16. S. E. Palmer. *Vision Science: Photons to Phenomenology*. MIT Press, Cambridge, MA, 1999.
17. H. Pashler. *The Psychology of Attention*. MIT Press, Cambridge, MA, 1997.
18. A. Rotenstein, A. Andreopoulos, E. Fazl, D. Jacob, M. Robinson, K. Shubina, Y. Zhu, and J. Tsotsos. Towards the dream of intelligent, visually-guided wheelchairs. In *Proc. 2nd Int'l Conf. on Technology and Aging*, Toronto, Canada, June 2007.
19. A. Torralba, A. Oliva, M. Castelhano, and J. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features on object search. *Psychological Review*, 113(4), 2006.
20. A. Treisman. Preattentive processing in vision. *Computer vision, graphics, and image procession*, 31:156–177, 1985.
21. A. M. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.
22. A. M. Treisman and S. Gormican. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review*, 95(1):15–48, 1988.

23. J. K. Tsotsos. A 'complexity level' analysis of vision. In *Proc. of International Conference on Computer Vision: Human and Machine Vision Workshop*, London, England, June 1987.
24. P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, May 2004.
25. D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 2006.
26. J. M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238, 1994.
27. J. M. Wolfe. Visual search. In H. Pashler, editor, *Attention*, pages 13–74. Hove, U.K.: Psychology Press, 1998.
28. J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, 5:1–7, 2004.
29. C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime $TV-L^1$ optical flow. In *Proc. of the Annual meeting of the German Assoc. for Pattern Recognition (DAGM)*, 2007.

Publication [7]

Simone Frintrop, Erich Rome, and Henrik I. Christensen. Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception*, 7(1), 2010.

Computational Visual Attention Systems and their Cognitive Foundations: A Survey

SIMONE FRINTROP
Rheinische Friedrich-Wilhelms-Universität
ERICH ROME
Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)
and
HENRIK I. CHRISTENSEN
Georgia Tech

Based on concepts of the human visual system, computational visual attention systems aim to detect regions of interest in images. Psychologists, neurobiologists, and computer scientists have investigated visual attention thoroughly during the last decades and profited considerably from each other. However, the interdisciplinarity of the topic holds not only benefits but also difficulties: concepts of other fields are usually hard to access due to differences in vocabulary and lack of knowledge of the relevant literature. This paper aims to bridge this gap and bring together concepts and ideas from the different research areas. It provides an extensive survey of the grounding psychological and biological research on visual attention as well as the current state of the art of computational systems. Furthermore, it presents a broad range of applications of computational attention systems in fields like computer vision, cognitive systems and mobile robotics. We conclude with a discussion on the limitations and open questions in the field.

Categories and Subject Descriptors: A.1 [**Introductory and Survey**]: ; I.2.10 [**Vision and Scene Understanding**]: ; I.4 [**Image Processing and Computer Vision**]: ; I.6.5 [**Model Development**]: ; I.2.9 [**Robotics**]:

General Terms: Algorithms, Design

Additional Key Words and Phrases: visual attention, saliency, regions of interest, biologically motivated computer vision, robot vision

1. INTRODUCTION

Every stage director is aware of the concepts of human selective attention and knows how to exploit them to manipulate his audience: A sudden spotlight illuminating a person in the dark, a motionless character starting to move suddenly, a voice from a

Authors' addresses: S. Frintrop, Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität, Römerstr. 164, 53117 Bonn, Germany, email: frintrop@iai.uni-bonn.de
E. Rome, Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), Schloss Birlinghoven, 53757 Sankt Augustin, Germany, email: erich.rome@iais.fraunhofer.de
H.I. Christensen, Georgia Tech, College of Computing, 85 Fifth Street, Atlanta, GA, 30308, USA, email: hic@cc.gatech.edu

character hidden in the audience, these effects not only keep our interest alive, they also guide our gaze, telling where the current action takes place. The mechanism in the brain that determines which part of the multitude of sensory data is currently of most interest is called *selective attention*. This concept exists for each of our senses; for example, the cocktail party effect is well-known in the field of auditory attention. Although a room may be full of different voices and sounds, it is possible to voluntarily concentrate on the voice of a certain person [Cherry 1953]. Visual attention is sometimes compared with a spotlight in a dark room. The fovea – the center of the retina – has the highest resolution in the eye. Thus, directing the gaze to a certain region complies with directing a spotlight to a certain part of a dark room [Shulman et al. 1979]. By moving the spotlight around, one can obtain an impression of the contents of the room, while analogously, by scanning a scene with quick eye movements, one can obtain a detailed impression of it.

Evolution has favored the concepts of selective attention because of the human need to deal with a high amount of sensory input at each moment. This amount of data is in general too high to be completely processed in detail and the possible actions at one and the same time are restricted; the brain has to prioritize. The same problem is faced by many modern technical systems. Computer vision systems have to deal with thousands, sometimes millions of pixel values from each frame and the computational complexity of many problems related to the interpretation of image data is very high [Tsotsos 1987]. The task becomes especially difficult if a system has to operate in real-time. Application areas in which real-time performance is essential are cognitive systems and mobile robotics since the systems have to react to their environment instantaneously.

For mobile autonomous robots, focusing on the relevant data is even more important than for pure vision systems. Many modules have to share resources on a robot. Usually, different modules share a visual sensor and each module has its own requirements. An obstacle avoidance module requires access to peripheral data to generate a motion flow, whereas a recognition module requires high resolution central data. Such a module might profit from zooming to the object, other modules might require gaze shifts. These resource conflicts depend on a selection mechanism which controls and prioritizes possible actions. Furthermore, cameras are often used in combination with other sensors, and modules concerned with tasks like navigation and manipulation of objects require additional computation power. And in contrast to early robotic systems applied in simple industrial conveyor belt tasks, current systems are supposed to drive and act autonomously in complex, previously unknown environments with challenges such as changing illuminations and people that walk around. Thus, for humans as well as for robots, limited resources require a selection mechanism which prioritizes the sensory input from "very important" to "not useful right now".

In order to cope with these requirements, people have investigated how the concepts of human selective attention can be exploited for computational systems. For many years, these investigations have been of mainly theoretical interest since the computational demands were too high for practical applications. Only during the last 5-10 years, the computational power enabled implementations of computational attention system that are useful in practical applications, causing an increasing in-

terest in such mechanisms in fields like computer vision, cognitive systems and mobile robotics. Example applications include object recognition, robot localization or human-robot interaction.

In this paper, we provide a survey of computational visual attention systems and their applications. The article is intended to bridge the gap between communities. For researchers from engineering sciences interested in computational attention systems, it provides the necessary psychophysical and neuro-scientific background knowledge about human visual attention. For psychologists and neuro-biologists, it explains the techniques applied to build computational attention systems. And for all researchers concerned with visual attention, it provides an overview of the current state of the art and of applications in computer vision and robotics.

This work focuses on systems which are both biologically motivated and serve a technical purpose. Such systems aim to improve computational vision systems in speed and/or quality of detection and recognition. Other computational attention systems focus on the objective to basically simulate and understand the concepts of human visual attention. A brief overview is given in section 2.3, but for a more thorough exposition the authors point the interested reader to the following review papers. A review of computational attention systems with a psychological objective can be found in [Heinke and Humphreys 2004], and a survey on computational attention models significantly inspired from neurobiology and psychophysics is presented by Rothenstein and Tsotsos [2006a]. Finally, a broad review on psychological attention models in general is found in [Bundesen and Habekost 2005].

Since the term "attention" is not clearly defined, it is sometimes used in other contexts. In the broadest sense, any pre-processing method might be called attentional, because it focuses subsequent processing to parts of the data which seem to be relevant. For example, Viola and Jones [2004] present an object recognition technique which they call "attentional cascade", since it starts processing at a coarse level and intensifies processing only at interesting regions. In this paper, we focus on approaches which are motivated by human visual attention (see sec. 2.2.1 for a definition).

The structure of the paper is as follows. In section 2, we introduce the concepts of human visual attention and present the psychological theories and models which have been most influential for computational attention systems. Section 3 describes the general structure and characteristics of computational attention systems and provides an overview over the state of the art in this field. Applications of visual attention systems in computer vision and robotics are described in section 4. A discussion on the limitations and open questions in the field concludes the paper.

2. HUMAN VISUAL ATTENTION

This section introduces background knowledge on human visual attention that researchers should have when dealing with computational visual attention. We start by briefly sketching the human visual system in sec. 2.1. After that, section 2.2 introduces the concepts of visual attention. Finally, we present in sec. 2.3 the most important psychological theories and models of visual attention which form the basis for most current computational systems.

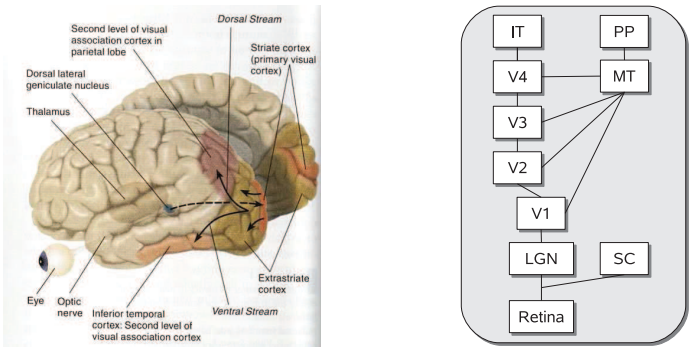


Fig. 1. Left: Visual areas and pathways in the human brain (Fig. from http://philosophy.hku.hk/courses/cogsci/ncc.php). Right: some of the known connections of visual areas in the cortex (Fig. adapted from [Palmer 1999]).

2.1 The Human Visual System

Here, we start with providing a very rough overview of the human visual system (cf. Fig. 1). Further literature on this topic can be found in [Palmer 1999; Kandel et al. 1996] and [Zeki 1993].

The light that arrives at the eye is projected onto the retina and then the visual information is transmitted via the optic nerve to the optic chiasm. From there, two pathways go to each brain hemisphere: the collicular pathway leading to the Superior Colliculus (SC) and, more important, the retino-geniculate pathway, which transmits about 90% of the visual information and leads to the Lateral Geniculate Nucleus (LGN). From the LGN, the information is transferred to the primary visual cortex (V1). Up to here, the processing stream is also called *primary visual pathway*. Many simple feature computations take part during this pathway. Already in the retina, there are cells responding to color contrasts and orientations. Up through the pathway, cells become more complex and combine results obtained from many previous cell outputs.

From V1, the information is transmitted to the "higher" brain areas V2 – V4, infero temporal cortex (IT), the middle temporal area (MT or V5) and the posterior parietal cortex (PP). Although there are still many open questions concerning V1 [Olshausen and Field 2005; 2006], even less is known on the extrastriate areas. One of the most important findings during the last decades was that the processing of the visual information is not serial but highly parallel. Many authors have claimed that the extrastriate areas are functionally separated [Kandel et al. 1996; Zeki 1993; Livingstone and Hubel 1987; Palmer 1999]. Some of the areas process mainly color, some form, and some motion.

The processing leads to mainly two different locations in the brain: First, the color and form processing leads to IT, the area where the recognition of objects takes place. Since IT is concerned with the question of "what" is in a scene, this pathway is called the *what pathway*. Other names are the *P pathway* or *ventral stream* because of its location on the ventral part of the body. Second, the motion and depth processing leads to PP. Since this area is mainly concerned with the

question of "where" something is in a scene, this pathway is also called *where pathway*. Other names are the *M pathway* or *dorsal stream* because it lies dorsally.

Newer findings propose that there is much less segregation of feature computations. It is for example indicated that luminance and color are not separated but there is a continuum of cells, varying from cells that respond only to luminance, to a few cells that do not respond to luminance at all [Gegenfurtner 2003]. Additionally, the form processing is not clearly segregated from color processing since most cells that respond to oriented edges respond also to color contrasts.

2.2 Visual Attention

In this section, we discuss several concepts of visual attention. More detailed information can be found in some books on this topic, e.g. [Pashler 1997; Styles 1997; Johnson and Proctor 2003]. Here, we start with a definition of visual attention, and introduce the concepts of covert and overt attention, the units of attention, bottom-up saliency and top-down guidance. Then, we elaborate on visual search, its efficiency, pop-out effects, and search asymmetries. Finally, we discuss the neurobiological correlates of attention.

2.2.1 *What is Visual Attention?.* The concept of selective attention refers to a fact already mentioned by [Aristotle]: "it is impossible to perceive two objects coinstantaneously in the same sensory act". Although we usually have the impression to retain a rich representation of our visual world and that large changes to our environment will attract our attention, various experiments reveal that our ability to detect changes is usually highly overestimated. Only a small region of the scene is analyzed in detail at each moment: the region that is currently attended. This is usually but not always the same region that is fixated by the eyes. That other regions than the attended one are usually largely ignored is shown, for example, in experiments on *change blindness* [Simons and Levin 1997; Rensink et al. 1997]. In these experiments, a significant change in a scene remains unnoticed, that means the observer is "blind" for this change.

The reason why people are nevertheless effective in every-day life is that they are usually able to automatically attend to regions of interest in their surrounding and to scan a scene by rapidly changing the focus of attention. The order in which a scene is investigated is determined by the mechanisms of *selective attention*. A definition is given for example in [Corbetta 1990]: "Attention defines the mental ability to select stimuli, responses, memories, or thoughts that are behaviorally relevant among the many others that are behaviorally irrelevant". Although the term attention is also often used to refer to other psychological phenomena (e.g., the ability to remain alert for long periods of time), in this work, attention refers exclusively to perceptual selectivity.

2.2.2 *Covert versus Overt Attention.* Usually, directing the focus of attention to a region of interest is associated with eye movements (*overt attention*). However, this is only half of the story. We are also able to attend to peripheral locations of interest without moving our eyes, a phenomenon which is called *covert attention*. This phenomenon was already described in the 19th century by von Helmholtz [1896]: "I found myself able to choose in advance which part of the dark field off to the side of the constantly fixated pinhole I wanted to perceive by indirect vision"

(English translation from M. Mackeben in [Nakayama and Mackeben 1989]). This mechanism should be well known to each of us when we detect peripheral motion or suddenly spot our name in a list.

There is evidence that simple manipulation tasks can be performed without overt attention [Johansson et al. 2001]. On the other hand, there are cases in which an eye movement is not preceeded by covert attention: Findlay and Gilchrist [2001] found that in tasks like reading and complex object search, *saccades* (quick, simultaneous movements of both eyes in the same direction [Cassin and Solomon 1990]) were made with such frequency that covert attention could not have scanned the scene first. Even though, covert attention and saccadic eye movements usually work together: the focus of attention is directed to a region of interest followed by a saccade that fixates the region and enables the perception at a higher resolution. That covert and overt attention are not independent was shown by Deubel and Schneider [1996]: it is not possible to attend to one location while directing the eyes to a different one.

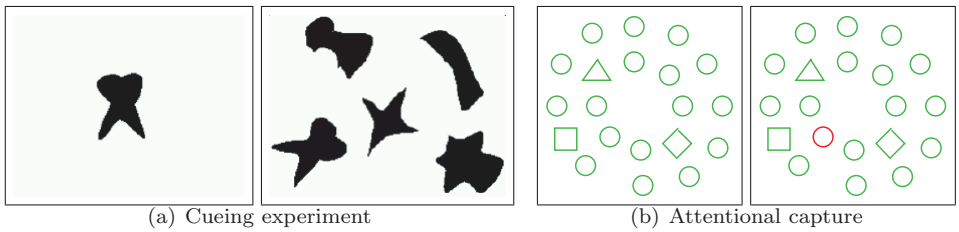
2.2.3 *The unit of attention.* During the last decades, there has been a long debate about the units of attention, that means about the target our attentional focus is directed to. Do we attend to *spatial locations*, to *features*, or to *objects*?

The majority of studies, both from psychophysics and from neurobiology, is about *space-based attention* (also referred to as *location-based attention*) [Posner 1980; Eriksen and St. James 1986; Yantis et al. 2002; Bisley and Goldberg 2003]. However, there is also strong evidence for *feature-based attention* [Treisman and Gelade 1980; Giesbrecht et al. 2003; Liu et al. 2003] and for *object-based attention* [Duncan 1984; Driver and Baylis 1998; Scholl 2001; Ben-Shahar et al. 2007; Einhäuser et al. 2008]. Today, most researchers believe that these theories are not mutually exclusive but that visual attention can be deployed to each of these candidate units [Vecera and Farah 1994; Fink et al. 1997; Yantis and Serences 2003]. A broad introduction and overview over the different approaches and studies can be found in [Yantis 2000].

Finally, it is worth mentioning that there is often not only a single unit of attention. Humans are able to attend simultaneously to multiple regions of interest, usually between 4 and 5 regions. This has been shown in psychological [Pylyshyn and Storm 1988; Pylyshyn 2003; Awh and Pashler 2000] as well as neurobiological experiments [McMains and Somers 2004].

2.2.4 *Bottom-up versus Top-down Attention.* There are two major categories of factors that drive attention: *bottom-up factors* and *top-down factors* [Desimone and Duncan 1995]. Bottom-up factors are derived solely from the visual scene [Nothdurft 2005]. Regions of interest that attract our attention in a bottom-up way are called *salient* and the responsible feature for this reaction must be sufficiently discriminative with respect to surrounding features. Beside *bottom-up attention*, this attentional mechanism is also called *exogenous*, *automatic*, *reflexive*, or *peripherally cued* [Egeth and Yantis 1997].

On the other hand, *top-down attention* is driven by cognitive factors such as knowledge, expectations and current goals [Corbetta and Shulman 2002]. Other terms for top-down attention are *endogenous* [Posner 1980], *voluntary* [Jonides 1981], or *centrally cued* attention. There are many intuitive examples of this pro-



(a) Cueing experiment (b) Attentional capture

Fig. 2. (a) Cueing experiment: a cue (left) is presented for 200 ms. Then, human subjects have to search for the cued shape in a search array (right) (Fig. reprinted with permission from [Vickery et al. 2005] © 2005 The Association for Research in Vision and Ophthalmology (ARVO)).
(b) Attentional capture: in both displays, human subjects had to search for the diamond. Although they knew that color was unimportant in this search task, the red circle in the right display slowed down the search about 65 ms (885 vs 950 ms) [Theeuwes 2004]. That means, the color pop-out "captures" the attention independent of the task (Fig. adapted from [Theeuwes 2004]).

cess. Car drivers are more likely to see the petrol stations in a street and cyclists notice cycle tracks. If you are looking for a yellow highlighter on your desk, yellow regions will attract the gaze more readily than other regions.

Yarbus [1967] has already early shown that eye movements depend on the current task: for the same scene ("an unexpected visitor" which shows a room with a family and a person entering the room), subjects got different instructions such as "estimate the material circumstances of the family", "what are the ages of the people", or simply to freely examine the scene. Eye movements differed considerably for each of these cases. Visual context, such as the *gist* (semantic category) or the spatial layout of objects, also influence visual attention in a top-down manner. For example, Chun and Jiang [1998] have shown that targets appearing in learned configurations were detected more quickly.

In psychophysics, top-down influences are often investigated by so called *cueing experiments*. In these experiments, a "cue" directs the attention to the target. Cues may have different characteristics: they may indicate *where* the target will be, for example by a central arrow that points into the direction of the target [Posner 1980; Styles 1997], or *what* the target will be, for example the cue is a (similar or exact) picture of the target or a word (or sentence) that describes the target ("search for the black, vertical line") [Vickery et al. 2005; Wolfe et al. 2004] (cf. Fig. 2 (a)).

The performance in detecting a target is typically better in trials in which the target is present at the cued location than in trials in which the target appears at an uncued location; this was called the *Posner cueing paradigm* [Posner 1980]. A cue speeds up the search if it matches the target exactly and slows down the search if it is invalid. Deviations from the exact match slow down search speed, although they lead to faster speed compared with a neutral cue or a semantic cue [Vickery et al. 2005; Wolfe et al. 2004]. Recent physiological evidence from monkey experiments support these findings: neurons give enhanced responses when a stimulus in their receptive field matches a feature of the target [Bichot et al. 2005].

Evidence from neuro-physiological studies indicates that two independent but interacting brain areas are associated with the two attentional mechanisms [Corbetta and Shulman 2002]. During normal human perception, both mechanisms interact.

As per Theeuwes [2004], the bottom-up influence is not voluntary suppressible: a highly salient region "captures" the focus of attention regardless of the task. For example, if there is an emergency bell, you will probably stop reading this article, regardless of how engrossed in the text you were. This effect is called *attentional capture* (cf. Fig. 2 (b)). Neural evidence from monkey experiments support these findings: Ogawa and Komatsu [2004] show that even if monkeys searched for a target of one dimension (shape or color), singletons (pop-out elements) from the other dimension (color or shape) induced high activation in some neurons. However, although attentional capture is definitely a strong effect which occurs frequently, there is also evidence that in some cases the bottom-up effects can be overridden completely [Bacon and Egeth 1994]. These difficulties are discussed in more detail in [Connor et al. 2004]; a review of different studies on attentional capture can be found in [Rauschenberger 2003].

Bottom-up attention mechanisms have been more thoroughly investigated than top-down mechanisms. One reason is that data-driven stimuli are easier to control than cognitive factors such as knowledge and expectations. Even less is known on the interaction between the two processes.

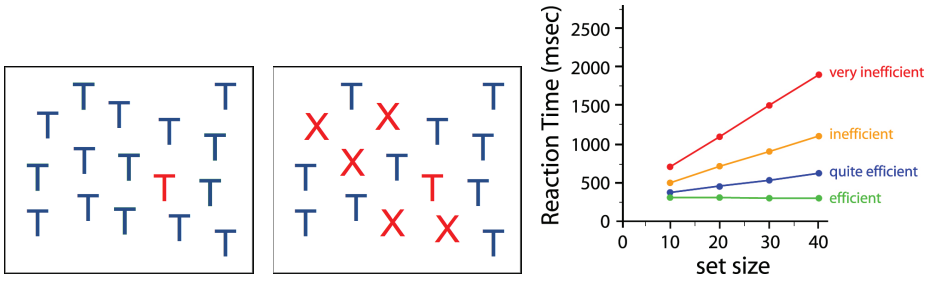
2.2.5 *Visual Search and Pop-out Effect.* An important tool in research on visual attention is *visual search* [Neisser 1967; Styles 1997; Wolfe 1998a]. The general question of visual search is: given a target and a test image, is there an instance of the target in the test image? We perform visual search all the time in every-day life. For example, finding a friend in a crowd is such a visual search task. Tsotsos has proven that the problem of *unbounded visual search* is so complex that it in practice is unsolvable in acceptable time[1] [Tsotsos 1987; 1990]. In contrast, *bounded visual search* (the target is explicitly known in advance) can be performed in linear time. Also, psychological experiments on visual search with known targets report that the search time complexity is linear and not exponential, thus the computational nature of the problem strongly suggests that attentional top-down influences play an important role during the search.

In psychophysical experiments, the *efficiency* of visual search is measured by the *reaction time* (also *response time*) (RT) that a subject needs to detect a target among a certain number of distractors (the elements that differ from the target) or by the *search accuracy*.

To measure the RT, a subject has to report a detail of the target or has to press one button if the target was detected and another if it is not present in the scene. The RT is represented as a function of *set size* (the number of elements in the display). The search efficiency is determined by the slopes and the intercepts of these RT $\times$ set size functions (cf. Fig. 3 (c)).

The searches vary in their efficiency: the smaller the slope of the function and the lower the value on the y-axis, the more efficient the search. Two extremes hereby are *serial* and *parallel* search. In serial search, the reaction time increases with the number of distractors, whereas in parallel search, the slope is near zero, i.e., there is

[1]The problem is *NP-complete*, i.e., it belongs to the hardest problems in computer science. No polynomial algorithm is known for this class of problems and they are expected to require exponential time in the worst case [Garey and Johnson 1979].



(a) Feature search (b) Conjunction search (c) The continuum of search slopes

Fig. 3. (a) Feature search: the target (red T) differs from the distractors (blue T's) by a unique visual feature (pop-out effect). (b) Conjunction search: the target (red T) differs from the distractors (red X's and blue T's) by a conjunction of features. (c) The reaction time (RT) of a visual search task is a function of set size. The efficiency is measured by the intercept and slopes of the functions (Fig. adapted from [Wolfe 1998a]).

no significant variation in reaction time if the number of distractors grows; here, a target is found immediately without the need to perform several shifts of attention. Experiments by Wolfe [1998b] indicate that the studies of visual search should not be classified into the distinct groups "parallel" and "serial" since the increase in reaction time is a continuum. He suggests instead to describe them as *"efficient"* versus *"inefficient"*. This allows one to use expressions like "more efficient than", "quite efficient" or "very inefficient" (cf. Fig. 3 (c)).

The concept of efficient search has been discovered a long time ago. Already in the 11th century, Ibn Al-Haytham (English translation: [Sabra 1989]) found that "some of the particular properties of which the forms of visible objects are composed appear at the moment when sight glances at the object, while others appear only after scrutiny and contemplation". This effect is nowadays referred to as *pop-out effect*, according to the subjective impression that the target leaps out of the display to grab attention (cf. Fig. 3 (a)). Scenes with pop-outs are sometimes also referred to as *odd-man-out* scenes. Efficient search is often but not always accompanied by pop-out [Wolfe 1994]. Usually, pop-out effects only occur when the distractors are homogeneous, for example, the target is red and the distractors are green. Instead, if the distractors are green and yellow, search is efficient but there is no pop-out effect.

In *conjunction search tasks* (also *conjunctive search*), in which the target is defined by several features, the search is usually less efficient (cf. Fig. 3 (b)). However, the steepness of the slope depends on the experiment; there are also search tasks in which conjunction search is quite efficient [Wolfe 1998a; 1998b].

While experimentally simple to perform, RT measures are not sufficient to answer all questions concerning visual search. It documents only the completion of search and not the search process itself. Thus, neither spatial information (where is the subject looking during search and how many saccades are performed) nor temporal information (how long is each part fixated) can be measured. According to Zelinsky and Sheinberg [1997], measuring eye movements is more suitable to provide such information.

Another method to determine search efficiency is by measuring accuracy. A search stimulus is presented only briefly and followed by a mask that terminates the search. The time between the onset of the stimulus and that of the mask is called *stimulus onset asynchrony (SOA)*. The SOA is varied and accuracy is plotted as a function of SOA [Wolfe 1998a]. Easy search tasks can be performed efficiently even with short SOAs, whereas harder search tasks require longer SOAs. A single-stage Signal Detection Theory (SDT) model can predict these accuracy results in terms of the probability of correctly detecting the presence or absence of the target [Verghese 2001; Cameron et al. 2004] (cf. sec. 2.3.3).

Finally, it is worth mentioning the *eccentricity effect*: the physical layout of the retina, with high resolution in the center and low resolution in the periphery, makes targets at peripheral locations more difficult to detect. Both reaction times and errors increase with increasing distance from the center [Carrasco et al. 1995].

There has been a multitude of experiments on visual search and many settings have been designed to discover which features enable efficient search and which do not. Some interesting examples are the search for numbers among letters, for mirrored letters among normal ones, for the silhouette of a "dead" elephant (legs to the top) among normal elephants [Wolfe 2001a], and for the face of another race among faces of the same race as the test subject [Levin 1996].

One purpose of these experiments is to study the *basic features* (also *primitive features* or *attributes*) of human perception, that means the features which are early and pre-attentively processed in the human brain and guide visual search. Testing the efficiency of visual search helps to investigate this since efficient search is said to take place if the target is defined by a single basic feature and the distractors are homogeneous [Treisman and Gormican 1988]. Thus, finding out that a red blob pops out among green ones indicates that color is a basic feature. Opinions on what are basic features are still controversial. Some features are doubtless basic, others are guessed to be basic but there is limited data or dissenting opinions. A listing of the current opinion is presented by Wolfe and Horowitz [2004]. According to them, undoubted basic features are color, motion, orientation and size (including length and spatial frequency). The role of luminance (intensity) is still unclear. In some studies luminance behaves like colors, whereas in others it acts more independently [Wolfe 1998a]. Probable basic features are luminance onset (flicker), luminance polarity, Vernier offset (a small lateral break in a line), stereoscopic depth and tilt, pictorial depth cues, shape, line termination, closure, topological status and curvature. Features which are possibly basic, but have even less confidence, are lighting direction (shading), glossiness (luster), expansion, number and aspect ratio. Features which are unconvincing but still possible are novelty, letter identity, and alphanumeric category. Finally, features which are probably not basic are intersection, optic flow, color change, three-dimensional volumes, faces, your name and semantic categories as "animal" or "scary". While this listing does not claim to be exhaustive, it gives a good overview about the current state of research.

An interesting effect in visual search tasks are *search asymmetries*, that means the effect that a search for stimulus 'A' among distractors 'B' produces different results from a search for 'B' among 'A's. An example is that finding a tilted line among vertical distractors is easier than vice versa (cf. Fig. 4). An explanation is

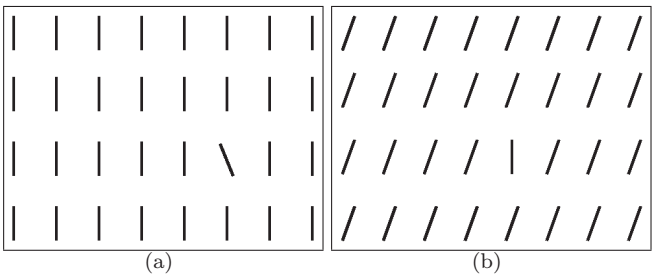


Fig. 4. Search asymmetries: it is easier to detect a tilted line among vertical distractors (a) than vice versa (b)

proposed by Treisman and Gormican [1988]: the authors claim that it is easier to find deviations among canonical stimuli than vice versa. Given that vertical is a canonical stimulus, the tilted line is a deviation and may be detected fast. Therefore, by investigating search asymmetries it is possible to determine the canonical stimuli of visual processing which might be identical to feature detectors. For example, Treisman suggests that for color the canonical stimuli are red, green, blue, and yellow; for orientation, they are vertical, horizontal, and left and right diagonal, and for luminance there exist separate detectors for darker and lighter contrasts [Treisman 1993]. Especially when building a computational model of visual attention this is of significant interest: if it is clear what feature detectors exist in the human brain, it might be adequate to focus on the computation of these features. However, one should be careful to accept evidence about search asymmetries. Findings by Rosenholtz [2001] indicate that the asymmetries in many of the studies are due to built-in design asymmetries instead of to an underlying asymmetry in the search mechanism. A comprehensive overview about search asymmetries is provided by Wolfe [2001a], more papers can be found in the same issue of *Perception & Psychophysics, 63 (3), 2001.*

2.2.6 *Neurobiological Correlates of Visual Attention.* The mechanisms of selective attention in the human brain still belong to the open problems in the field of research on perception. Perhaps the most prominent outcome of neuro-physiological findings on visual attention is that there is no single brain area guiding the attention, but neural correlates of visual selection appear to be reflected in nearly all brain areas associated with visual processing [Maunsell 1995]. Additionally, new findings indicate that many brain areas share the processing of information from different senses and there is growing evidence that large parts of the cortex are multisensory [Ghazanfar and Schroeder 2006].

Attentional mechanisms are carried out by a network of anatomical areas [Corbetta and Shulman 2002]. Important areas of this network are the posterior parietal cortex (PP), the superior colliculus (SC), the Lateral IntraParietal area (LIP), the Frontal Eye Field (FEF) and the pulvinar. Regarding the question which area fulfills which task, the opinions diverge. We review several findings here.

Posner and Petersen [1990] describe three major functions concerning attention: first, orienting of attention, second, target detection, and third, alertness. They

claim that the first function, the orienting of attention to a salient stimulus, is carried out by the interaction of three areas: the PP, the SC, and the pulvinar. The PP is responsible for disengaging the focus of attention from its present location (inhibition of return), the SC shifts the attention to a new location, and the pulvinar is specialized in reading out the data from the indexed location. Posner and Petersen call this combination of systems the *posterior attention system*. The second attentional function, the detection of a target, is carried out by what the authors call the *anterior attention system*. They claim that the anterior cingulate gyrus in the frontal part of the brain is involved in this task. Finally, they state that the alertness to high priority signals is dependent on activity in the norepinephrine system (NE) arising in the locus coeruleus.

Brain areas involved in guiding eye movements are the FEF and the SC. Furthermore, Bichot [2001] claims that the FEF is the place where a kind of saliency map is located which derives information from bottom-up as well as from top-down influences. Other groups locate the saliency map at different areas, e.g., at LIP [Gottlieb et al. 1998], at SC [Findlay and Walker 1999], at V1 [Li 2005], or at V4 [Mazer and Gallant 2003].

There has been evidence that the source of top-down biasing signals may derive from a network of areas in parietal and frontal cortex. According to Kastner and Ungerleider [2001], these areas include the superior parietal lobule (SPL), the FEF and the supplementary eye field (SEF), and, less consistently, areas in the inferior parietal lobule (IPL), the lateral prefrontal cortex in the region of the middle frontal gyrus (MFG), and the anterior cingulate cortex. Corbetta and Shulman [2002] find transient responses to a cue in the occipital lobe (fusiform and MT+) and more sustained responses in the dorsal posterior parietal cortex along the intraparietal sulcus (IPs) and in the frontal cortex at or near the putative human homologue of the FEFs. According to Ogawa and Komatsu [2004], the interaction of bottom-up and top-down cues takes place in V4.

To sum up, at the current time it is known that there is not a single brain area that controls attention but a network of areas. Several areas have been verified to be involved in attentional processes but the accurate task and behavior of each area as well as the interplay among them still remain open questions.

2.3 Psychophysical Theories and Models of Attention

In the field of psychology, there exists a wide variety of theories and models on visual attention. Their objective is to explain and better understand human perception. Here, we introduce the theories and models which have been most influential for computational attention systems. More on psychological attention models can be found in the review of Bundesen and Habekost [2005].

2.3.1 *Feature Integration Theory.* The Feature Integration Theory (FIT) of Treisman has been one of the most influential theories in the field of visual attention. The theory was first introduced in 1980 [Treisman and Gelade] but it was steadily modified and adapted to current research findings. One has to be careful when referring to FIT, since some of the older findings concerning a dichotomy between serial and parallel search are no longer believed to be valid (cf. sec. 2.2.5). An overview of the theory is found in [Treisman 1993].

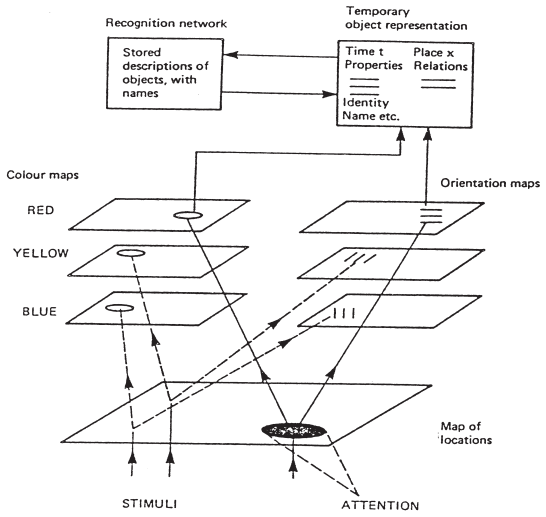


Fig. 5. Model of the *Feature Integration Theory (FIT)* (Fig. reprinted with permission from [Treisman and Gormican 1988] © 1988 American Psychological Association (APA)).

The theory claims that "different features are registered early, automatically and in parallel across the visual field, while objects are identified separately and only at a later stage, which requires focused attention" [Treisman and Gelade 1980]. Information from the resulting *feature maps* — topographical maps that highlight conspicuities according to the respective feature — is collected in a *master map of location*. This map specifies *where* in the display things are, but not *what* they are. Scanning serially through this map focuses the attention on the selected scene regions and provides this data for higher perception tasks (cf. Fig. 5).

Treisman mentions that the search for a target is easier the more features differentiate the target from the distractors. If the target has no unique features but differs from the distractors only in how its features are combined, the search is more difficult and often requires focused attention (conjunctive search). This usually results in longer search times. However, if the features of the target are known in advance, conjunction search can sometimes be accomplished rapidly. She proposes that this is done by inhibiting the feature maps which code non-target features.

Additionally, Treisman introduced so called *object files* as "temporary episodic representations of objects". An object file "collects the sensory information that has so far been received about the object. This information can be matched to stored descriptions to identify or classify the object" [Kahneman and Treisman 1992].

2.3.2 *Guided Search Model.* Beside FIT, the Guided Search Model of Wolfe is among the most influential work for computational visual attention systems. Originally, the model was created as an answer to some criticism on early versions

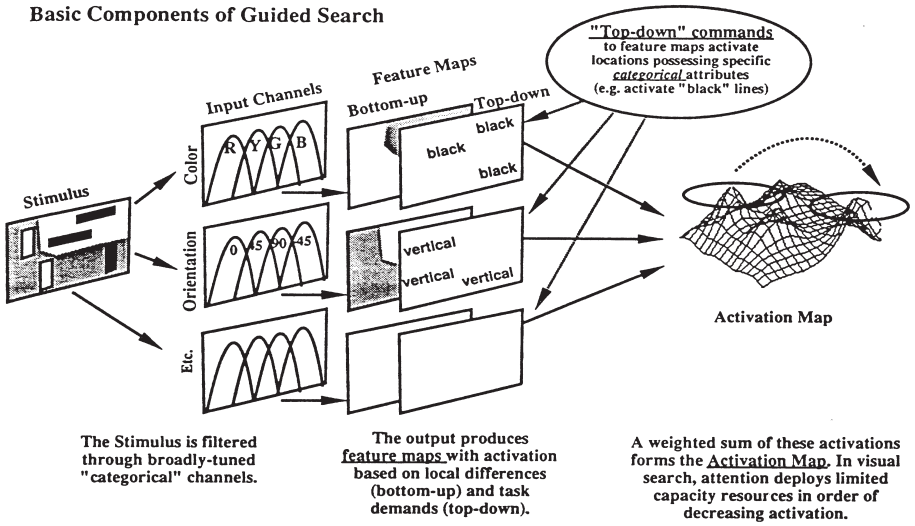


Fig. 6. The *Guided Search model* of Wolfe (Fig. reprinted with permission from [Wolfe 1994] ©1994 Psychonomic Society).

of the FIT. During the years, a competition arose between Treisman's and Wolfe's work, resulting in continuously improved versions of the models.

The basic goal of the model is to explain and predict the results of visual search experiments. There has been also a computer simulation of the model [Cave and Wolfe 1990; Wolfe 1994]. As Treisman's work, the model has been continuously developed further over the years. Mimicking the convention of numbered software upgrades, Wolfe has denoted successive versions of his model as Guided Search 1.0 [Wolfe et al. 1989], Guided Search 2.0 [Wolfe 1994], Guided Search 3.0 [Wolfe and Gancarz 1996], and Guided Search 4.0 [Wolfe 2001b; 2007]. Here, we focus on Guided Search 2.0 since this is the best elaborated description of the model. Versions 3.0 and 4.0 contain changes which are of minor importance here, for example, in 3.0 eye movements are included into the model and in 4.0 the implementation of memory for previously visited items and locations is improved.

The architecture of the model is depicted in Figure 6. It shares many concepts with the FIT, but is more detailed in several aspects which are necessary for computer implementations. An interesting point is that in addition to bottom-up saliency, the model also considers the influence of top-down information by selecting the feature type which distinguishes the target best from its distractors.

2.3.3 *Other theories and models.* Beside these approaches, there is a wide variety of psychophysical models on visual attention. Eriksen and St. James [1986] have introduced the *zoom lens model*. In this model, the spatial extent of the attentional focus can be manipulated by precueing. In this model, the scene is investigated by a spotlight with varying size. Many attention models fall into the category of connectionist models, that means models based on neural networks. They are composed of a large number of processing units connected by inhibitory or excitatory

links. Examples are the *dynamic routing circuit* [Olshausen et al. 1993], and the models MORSEL [Mozer 1987], SLAM (SeLective Attention Model) [Phaf et al. 1990], SERR (SEarch via Recursive Rejection) [Humphreys and Müller 1993], and SAIM (Selective Attention for Identification Model) [Heinke and Humphreys 2003].

A formal mathematical model is presented by Logan [1996]: the CODE Theory of Visual Attention (CTVA). It integrates the COntour DEtector (CODE) theory for perceptual grouping [van Oeffelen and Vos 1982] with the Theory of Visual Attention (TVA) [Bundesen 1990]. The theory is based on a *race model* of selection. In these models, a scene is processed in parallel and the element that first finishes processing is selected (the winner of the race). That means, a target is processed faster than the distractors in a scene. Newer work concerning CTVA can be found in [Bundesen 1998].

Another type of psychological models is based on the *signal detection theory (SDT)*, a method to measure the search accuracy by quantifying the ability to distinguish between signal and noise [Green and Swets 1966; Abdi 2007]. The distractors in a search task are considered to be noise and the target is signal plus noise. In a SDT experiment, one or several search displays are presented briefly and masked afterwards. In yes/no designs, one display is presented and the observer has to decide whether the target was present or not; in an M-AFC (alternative forced-choice) design, M displays are shown and the observer has to identify the display containing the target. The order of presentation is varied randomly in different trials. Performance is measured by determining how well the target can be distinguished from the distractors and the SDT model is used to calculate the performance degradation with increasing set size. SDT models which have been used to predict human performance for detection and localization of targets have been presented in [Palmer et al. 1993; Verghese 2001; Eckstein et al. 2000].

An interesting theoretical model has been introduced by Rensink [2000]. His *triadic architecture* consists of three parts: first, a low-level vision system produces *proto-objects* rapidly and in parallel. Second, a limited-capacity attentional system forms these structures into stable object representations. Finally, a non-attentional system provides setting information, for example, on the *gist* — the abstract meaning of a scene, e.g., beach scene, city scene, etc. — and on the *layout* — the spatial arrangement of the objects in a scene. This information influences the attentional system, for example, by restricting the search for a person on the sand region of a beach scene and ignoring the sky region.

3. COMPUTATIONAL ATTENTION SYSTEMS

In computer vision and robotics, there is increasing interest in a selection mechanism which determines the most relevant parts within the large amount of visual data. Visual attention is such a selection mechanism and therefore, many computational attention systems have been built during the last three decades (mainly during the last 5-10 years). The systems which are considered here have in common that they built on the psychological and neurobiological concepts and theories which have been presented in the previous section. In contrast to the models described in sec. 2.3, we focus here on computational systems with an engineering objective. The objective of these systems is less in understanding human perception but more

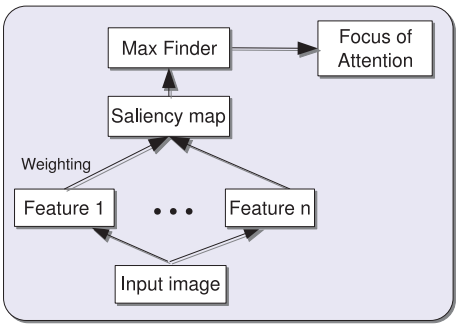


Fig. 7. General structure of most bottom-up attention systems.

in improving existing vision systems. Usually, they are able to cope not only with synthetical images but also with natural scenes. The systems vary in detail, but most of them have a similar structure.

We start with a description of the general structure of typical computational attention systems (sec. 3.1). Then, we continue with a more detailed investigation of the characteristics of different approaches. Connectionist versus filter models are distinguished (sec. 3.2), the choice of different feature channels is discussed (sec. 3.3), and the integration of top-down cues is introduced (sec. 3.4). Finally, we provide a chronological overview of important computational attention systems (sec. 3.5).

3.1 General structure

Most computational attention systems have a very similar structure which is depicted in Figure 7. This structure is originally adapted from psychological theories like the feature integration theory [Treisman and Gormican 1988] and the Guided Search model [Wolfe 1994]. The main idea is to compute several features in parallel and to fuse their saliencies in a representation which is usually called *saliency map*. Detailed information on how to implement such a system is presented for example in [Itti et al. 1998] or [Frintrop 2005]. The necessary background knowledge on computer vision methods is summed up in the appendix of [Frintrop 2005]. An overview of the techniques follows.

In filter-based models (cf. Sec. 3.2), usually the first step is to compute one or several image pyramids from the input image, to enable the computation of features on different scales [Itti et al. 1998]. This saves computation time since it avoids explicitly applying large filters to the image. The following computations are performed on several of the layers of the pyramid, usually ignoring the first, finest layers to reduce the influence of noise. An alternative approach is to use integral images for a fast computation of features on different scales [Frintrop et al. 2007].

An interesting approach is to exchange this standard uniform sampling scheme with a more biologically plausible space-variant sampling, according to the space-variant arrangement of photoreceptors in the retina. However, Vincent et al. [2007] have found that this causes feature coding unreliability and that there is "only a very weak relation between target eccentricity and discrimination performance".

Interesting in this context would be a replacement of the normal camera with a retina-like sensor to achieve space-variant sampling [Sandini and Metta 2002].

Next, several features are computed in parallel, and feature-dependent saliencies are computed for each feature channel. The information for different features is collected in *maps*. These might be represented as gray-scale images, in which the brightness of a pixel is proportional to its saliency (cf. Fig. 8), or as collections of nodes of an artificial neural network.

Commonly used features are intensity, color, and orientation; a detailed investigation of the choice of features is presented in sec. 3.3. Usually, the computation of these feature dimensions is subdivided into the computation of several *feature types*, for example, for the feature dimension color the feature types red, green, blue, and yellow may be computed. The feature types are usually displayed in *feature maps* and summed up to feature dependent saliency maps which are often called *conspicuity maps*, a term first used by Milanese [1993]. The conspicuity maps are finally fused to a single *saliency map* [Koch and Ullman 1985], a term that is widely used and corresponds to Treisman's master map of location.

The feature maps collect the local within-map contrast. This is usually computed by *center-surround mechanisms*, also called *center-surround differences* [Marr 1982]. This operation compares the average value of a center region to the average value of a surrounding region, inspired from the ganglion cells in the visual receptive fields of the human visual system [Palmer 1999]. In most implementations, the feature detectors are based on rectangular regions, which makes them less accurate than a circular filter but much easier to implement and faster to compute.

A very important aspect of attentional systems, maybe even the most important one, is the way different maps are fused, i.e., how the between-map interaction takes place. How is it accomplished that the important information is not lost in the large collection of maps? How is it achieved that the red ball on green grass pops out, although this saliency only shows up strongly in one of the maps, namely the red-green map? It is not yet completely clear how this task is solved in the brain nor is an optimal solution known how to solve this problem in a computational system. Usually, a weighting function, we call it *uniqueness weight* [Frintrop 2005], is applied to each map before summing up the maps. This weighting function determines the uniqueness of features: if there is only a single bright region in a map, its uniqueness weight is high, if there are several equally bright regions, it is lower. One simple solution to compute this is to determine the number of local maxima m in each map and divide each pixel by the square root of m [Frintrop 2005]. Other solutions are presented for example in [Itti et al. 1998; Itti and Koch 2001b; Harel et al. 2007]. An evaluation of different weighting approaches has, to our knowledge, not yet been done. However, even if it is not clear what the optimal weighting looks like, all these approaches are able to reproduce the human pop-out effect and detect outliers in images from psychophysical experiments such as the one in Figure 3(a). An example of applying such a weighting function to real-world images is shown in Figure 8. Note, that this weighting by uniqueness covers only the bottom-up aspect of visual attention. In human visual attention almost always top-down effects participate and guide our attention according to the current situation. These effects will be discussed in sec. 3.4.

Before the weighted maps are summed up, they are usually normalized. This is done to weed out the differences between a priori not comparable modalities with different extraction mechanisms. Additionally, it prevents the higher weighting of channels that have more feature maps than others. Most straightforward is to normalize all maps to a fixed range [Itti et al. 1998]. This results in problems if one channel is more important than another since information about the magnitude of the maps is removed. A method which keeps this information is to determine the maximum M of all maps which shall be summed up and normalize each map to the range $[0..M]$ [Frintrop et al. 2005]. An alternative that scales each conspicuity map with respect to a long-term estimate of its maximum is presented in [Ouerhani et al. 2006].

After weighing and normalizing, the maps are summed up to the saliency map. This saliency map might already be regarded as an output of the system since it shows the saliency for each region of a scene. But usually, the output of the system is a trajectory of image regions – mimicking human saccades – which starts with the highest saliency value. The selected image regions are local maxima in the saliency map. They might be determined by a *winner-take-all (WTA)* network which was introduced by Koch and Ullman [1985]. It shows how the selection of a maximum is implementable by neural networks, that means by single units which are only locally connected. This approach is strongly biologically motivated and shows how such a mechanism might work in the human brain. A simpler, more technically motivated alternative to the WTA with the same result is to straightforwardly determine the pixel with the largest intensity value in the image. This method requires fewer operations to compute the most salient region, but note that the WTA might be a good solution if implemented on a parallel architecture like a GPU.

Since the focus of attention (FOA) is usually not on a single point but on a region (we call it *MSR (most salient region)*), the next step is to determine this region. The simplest approach is to determine a fixed-sized circular region around the most salient point [Itti et al. 1998]. More sophisticated approaches integrate segmentation approaches on feature [Walther 2006] or saliency maps [Frintrop 2005] to determine a irregularly shaped attention region.

After the FOA has been computed, some systems determine a *feature vector* which describes how much each feature contributes to the region. Usually, also the local or global surrounding of the region is considered [Navalpakkam et al. 2005; Frintrop et al. 2005]. The vector can be used to match the region to previously seen regions, e.g., to search for similar regions in a top-down guided visual search task [Frintrop et al. 2005] or to track a region over subsequent frames [Frintrop and Kessel 2009]. Such a feature vector resembles the psychological concept of *object files* as temporary episodic representations of objects, which were introduced by Treisman (cf. sec. 2.3.1).

To obtain a trajectory of image regions which mimics a human search trajectory, most common is a method called *inhibition of return (IOR)*. It refers to the observation that in human vision, the speed and accuracy with which a target is detected is impaired after the target was attended. It was first described by Posner and Cohen [1984] and prevents that the FOA stays at the most salient region. In computational systems, IOR is implemented by inhibiting (reseting) the sur-

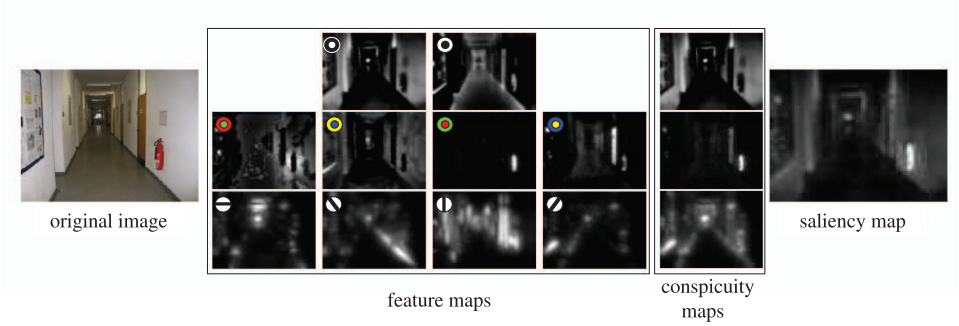


Fig. 8. Feature, conspicuity and saliency map(s) for an example image computed with the attention system VOCUS [Frintrop 2005]. 1st row: intensity maps (on-off and off-on). 2nd row: color maps (green, blue, red, yellow). 3rd row: orientation maps (0°, 45°, 90°, 135°). The feature map 'red' is weighted highest since the red fire extinguisher is unique in the scene. This results in a strong peak in the conspicuity color map and finally in a strong saliency in the saliency map.

rounding region in the saliency map. The surrounding region can be a fixed region around the FOA (spatial inhibition) or the MSR (feature-based inhibition), or a combination as in [Aziz and Mertsching 2007]. Interesting in this context is that Horowitz and Wolfe [2003] discovered that human visual search has no complete memory, i.e., not all items in a search display are marked after they have been considered. That means, IOR works probably only for a few items at a time. A possible implementation inhibits each distractor for a short time, dependent on a probabilistic value. In [Wolfe 2007], this results on average in about three inhibited items at a time. An alternative which is simple to implement and obtains good results is to determine all peaks in the saliency map, sort them by their saliency values, and direct the FOA attention subsequently to each salient region [Frintrop and Cremers 2007]. IOR is not necessary in this approach. We found that this method yielded better results than the IOR method since it avoids "border effects" in which the FOA returns to the border of the inhibited region. More difficult is IOR in dynamic scenes since not only the currently focused region must be tracked over time but also every inhibited region [Backer et al. 2001].

The structure described so far was purely bottom-up. Including prior knowledge and target information to the system in a top-down manner is described in sec. 3.4.

3.2 Connectionist versus Filter Models

A basic difference between models concerns the underlying structure which is either based on neural networks (connectionist models) or on a collection of gray-scale maps (filter models). Usually, the connectionist models claim to be more biologically plausible than the filter models since they have single units corresponding to neurons in the human brain, but it has to be noted that they are still a high abstraction from the processes in the brain. Examples of connectionist systems of visual attention are presented for instance in [Olshausen et al. 1993; Postma 1994; Tsotsos et al. 1995; Baluja and Pomerleau 1997; Cave 1999]. Many psychophysical models fall into this category, too, for example [Mozer 1987; Phaf et al. 1990;

Humphreys and Müller 1993; Heinke and Humphreys 2003]. An advantage of connectionist models is that they are — at least theoretically — able to show a different behavior for each neuron whereas in filter models usually each pixel in a map is treated equally. In practice, treating each unit differently is usually too costly and so a group of units shows the same behavior.

Advantages of filter models are that they can profit from approved techniques in computer vision and that they are especially well suited for the application to real-world images. Examples of linear filter systems of visual attention are presented for instance in [Milanese 1993; Itti et al. 1998; Backer et al. 2001; Sun and Fisher 2003; Heidemann et al. 2004; Hamker 2005; Frintrop 2005].

3.3 The Choice of Features

Many computational attention systems focus on the computation of mainly three features: intensity, color, and orientation [Itti et al. 1998; Draper and Lionelle 2005; Sun and Fisher 2003; Ramström and Christensen 2004]. Reasons for this choice are that these features belong to the basic features proposed in psychological and biological work [Treisman 1993; Palmer 1999; Wolfe 1994; Wolfe and Horowitz 2004] and that they are relatively easy to compute. A special case of color computation is the separate computation of skin color [Rae 2000; Heidemann et al. 2004; Lee et al. 2003]. This is often useful if faces or hand gestures have to be detected. Other features that are considered are for example curvature [Milanese 1993], spatial resolution [Hamker 2005], optical flow [Tsotsos et al. 1995; Vijayakumar et al. 2001], flicker [Itti et al. 2003], or corners [Fraundorfer and Bischof 2003; Heidemann et al. 2004; Ouerhani et al. 2005]. Several systems compute also more complex features that use approved techniques of computer vision to extract image information. Examples for such features are entropy [Kadir and Brady 2001; Heidemann et al. 2004], Shannon's self-information measure [Bruce and Tsotsos 2005b], ellipses [Lee et al. 2003], eccentricity [Backer et al. 2001], or symmetry [Backer et al. 2001; Heidemann et al. 2004; Lee et al. 2003].

A very important feature in human perception is motion. Some systems that consider motion as a feature are presented in [Maki et al. 2000; Ouerhani 2003; Itti et al. 2003; Rae 2000]. These approaches implement a simple kind of motion detection: usually, two subsequent images in a video stream are subtracted and the difference codes the feature conspicuity. Note that these approaches require a static camera and are not applicable on a mobile system as a robot. A sophisticated approach concerning motion was proposed by Tsotsos et al. [2005]. This approach considers the direction of movements, and processes motion on several levels similar to the processing in the brain regions V1, MT, and MST. In the above approaches, motion and static features are combined in a competitive scheme: they all contribute to a saliency map and the strongest cue wins. Bur et al. [2007] propose instead a motion priority scheme in which motion is prioritized by suppressing the static features in presence of motion.

Another important but rarely considered aspect in human perception is depth. From the psychological literature it is not clear whether depth is simply a feature or something else; definitely, it has some unusual properties distinguishing it from other features: if one of the dimensions in a conjunctive search is depth, a second feature can be searched in parallel [Nakayama and Silverman 1986], a property that

does not exist for the other features. Computing depth for an attention system is usually solved with stereo vision [Maki et al. 2000; Bruce and Tsotsos 2005a; Björkman and Eklundh 2007]. Another approach is to use special sensors to obtain depth data, for example 3D laser scanners, which provide dense and precise depth information and may provide additionally reflection data [Frintrop et al. 2005], or 3D cameras [Ouerhani and Hügli 2000].

Finally, it may be noted that although considering more features usually results in more accurate and biologically plausible detection results, it also reduces the processing speed since the parallel models are usually implemented sequentially. Therefore, a trade-off has to be found between accuracy and speed. Using three to four feature channels seems to be a useful compromise for most systems.

3.4 Top-down Cues

As outlined in section 2.2.4, top-down cues play an important role in human perception. For a computational attention system, they are equally important: most systems shall not only detect bottom-up salient regions but there are goals to achieve and targets to detect. Despite the well-known significance of top-down cues, most systems consider only bottom-up computations.

In human perception, there exist different kinds of top-down influences. They have in common that they represent information on the world or the state of the subject (or system). This includes aspects like current tasks and prior knowledge about the target, the scene or the objects that might occur in the environment, but also emotions, desires, and motivations. In the following, we discuss these different kinds of top-down information.

Emotions, desires, and motivations are hard to conceptualize and are not realized in any computer system we know about. Wells and Matthews [1994] provide a review from a psychological perspective about attention and emotion; Fragopanagos and Taylor [2006] present a neuro-biological model about the interplay of attention and emotions in the human brain. The interaction of attention, emotions, motivations, and goals is discussed by Balkenius [2000], but in his computer simulation these aspects are not considered.

Top-down information that refers to knowledge of the outer world, that means of the background scene or of the objects that might occur, is considered in several systems. In these approaches, for example, all objects of a database that might occur in a scene are investigated in advance and their most discriminative regions are determined, i.e., the regions that distinguish an object best from all others in the database [Fritz et al. 2004; Pessoa and Exel 1999]. Another approach is to regard context information, that means searching for a person in a street scene is restricted to the street region; the sky region is ignored. The contextual information is obtained from past search experiences in similar environments [Oliva et al. 2003; Torralba 2003b]. Another kind of context which can be integrated into attention models is the gist, i.e., the semantic category of the scene such as “office scene” or “forest” [Oliva 2005]. The gist is known to guide eye movements [Torralba 2003a] and is usually computed as a vector of contextual features. In visual attention systems, the gist may be computed directly from the feature channels [Siagian and Itti 2009].

One important kind of top-down information is the prior knowledge about a target that is used to perform visual search. Systems regarding this kind of top-down information use knowledge of the target to influence the computation of the most salient region. This knowledge is usually learned in a preceding training phase but might in simpler approaches also be provided manually by the user.

In existing systems, the target information influences the processing at different stages: the simplest solution computes the bottom-up saliency map and investigates the target similarity of the most salient regions [Rao et al. 2002; Lee et al. 2003]. Only the most salient targets in a scene can be found with this approach. More elaborated is the tuning of the conspicuity maps [Milanese et al. 1994; Hamker 2005], but biologically most plausible and also most useful from an engineering perspective is the approach to already bias the feature types [Tsotsos et al. 1995; Frintrop et al. 2005; Navalpakkam and Itti 2006a]. This is supported by findings of Navalpakkam and Itti [2006b]: not only the information about the feature dimensions influence top-down search but also information about feature types.

Different methods exist for influencing the maps with the target information. Some approaches inhibit the target-irrelevant regions [Tsotsos et al. 1995; Choi et al. 2004], whereas others prefer to excite target-relevant regions [Hamker 2005]. Newer findings suggest that inhibition and excitation both play an important rule [Navalpakkam et al. 2004]; this is realized in [Navalpakkam et al. 2005] and [Frintrop et al. 2005]. Navalpakkam and Itti [2006a] present an interesting approach in which not only knowledge about a target but also about distractors influences the search. Vincent et al. [2007] learn the optimal feature map weights with multiple linear regression.

If human behavior shall be imitated, the bottom-up and the top-down saliency have to be fused to obtain a single focus of attention. Note however that in a computational system, it is also possible to deal with both maps in parallel and use the bottom-up and the top-down information for different purposes. The decision whether to fuse the maps or not has to be done depending on the application. If the maps shall be fused, one difficulty is how to combine the weighting for uniqueness (bottom-up) and the weighting for target-relevance (top-down). One possibility is to multiply the bottom-up maps with the top-down feature weights after applying the uniqueness weight [Hamker 2005; Navalpakkam et al. 2005]. A problem with this approach is that it is difficult to find non-salient objects, since the bottom-up computations assign a very low saliency to the target region. One approach to overcome this problem is to separate bottom-up and top-down computations and to finally fuse them again as done by Frintrop et al. [2005]. Here, the contribution of bottom-up and top-down cues is adjusted by a parameter t which has to be set according to the system state: in exploration mode there is a high bottom-up contribution, in search mode the parameter shall be set proportionally to the search priority. Rasolzadeh et al. [2009] have adopted this idea and present an extension in which t can vary over time depending on the energy of bottom-up and top-down saliency maps. Xu et al. [2009] propose an approach that switches automatically between bottom-up and top-down behavior depending on the two internal robot states 'observing' and 'operating'.

The evaluation of top-down attention systems will be discussed in sec. 3.6.

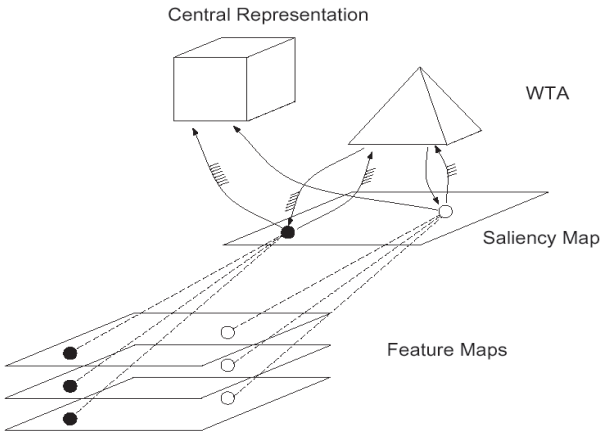


Fig. 9. The Koch-Ullman model. Different features are computed in parallel and their conspicuities are represented in several *feature maps*. A central *saliency map* combines the saliencies of the features and a *winner take all network (WTA)* determines the most salient location. This region is routed to the *central representation* where complex processing takes place (Fig. reprinted with permission from [Koch and Ullman 1985] © Springer Science and Business Media).

3.5 Important Attention Systems in Chronological Order

In this section, we will present some of the most important attention systems in a chronological order and mention their particularities.

The first computational architecture of visual attention was introduced by **Koch and Ullman [1985]** which was inspired by the Feature Integration Theory. When it was first published, the model was not yet implemented, but it provided the algorithmic reasoning serving as a foundation for later implementations and for many current computational attention systems. An important contribution of their work is the WTA network (see Fig. 9).

One of the first implementations of an attention system was presented by **Clark and Ferrier [1988]**. Based on the Koch-Ullman model, it contains feature maps which are weighted and summed up to a saliency map. The feature computations are performed by filter operations, realized by a special purpose image processing system, so the system belongs to the class of filter-based models.

Another early filter-based attention model was introduced by **Milanese [1993]**. In a derivative, Milanese et al. [1994] include top-down information from an object recognition system realized by *distributed associative memories (DAMs)*. By first introducing concepts like conspicuity maps and feature computations based on center-surround mechanisms (called "conspicuity operator"), the system has set benchmarks for several techniques which are used in computational attention models until today.

One of the oldest attention models which is widely known and still developed further is **Tsotsos'** *selective tuning (ST) model* of visual attention [Tsotsos 1990; 1993; Tsotsos et al. 1995]. It is a *connectionist model* which consists of a pyramidal architecture with an *inhibitory beam* (see Fig. 10). It is also possible to consider target-specific top-down cues by either inhibiting all regions with features different

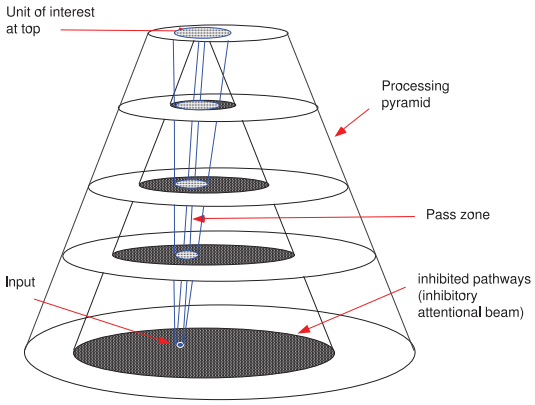


Fig. 10. The *inhibitory attentional beam* of Tsotsos et al. The selection process requires two traversals of the pyramid: first, the input traverses the pyramid in a feedforward manner (pass zone). Second, the hierarchy of WTA processes is activated in a top-down manner to localize the strongest item in each layer while pruning parts of the pyramid that do not contribute to the most salient item (inhibit zone) (Fig. kindly provided by John Tsotsos).

from the target features or regions of a specified location. The model has been implemented for several features, for example luminance, orientation, or color opponency [Tsotsos et al. 1995], motion [Tsotsos et al. 2005], and depth from stereo vision [Bruce and Tsotsos 2005a]. Originally, each version of the ST model processed only one feature dimension, but recently, it was extended to perform feature binding [Rothenstein and Tsotsos 2006b; Tsotsos et al. 2008].

An unusual adaptation of Tsotsos's model is provided by Ramström and Christensen [2002]: the distributed control of the attention system is performed by game theory concepts. The nodes of the pyramid are subject to trading on a market, the features are the goods, rare goods are expensive (the features are salient), and the outcome of the trading represents the saliency.

One of the currently best known attention systems is the *Neuromorphic Vision Toolkit (NVT)* (Fig. 11), a derivative of the Koch-Ullman model, that is steadily kept up to date by the group around **Itti** [Itti et al. 1998; Itti and Koch 2001a; Navalpakkam and Itti 2006a]. Their model as well as their implementation serve as a basis for many research groups; one reason for this is the good documentation and the online availability of the source code[2]. Itti et al. introduce image pyramids for the feature computations, which enables an efficient processing of real-world images. In its original version, the system concentrates on computing bottom-up attention. In newer work, Navalpakkam and Itti [2006a] introduce a derivative of the NVT which is able to deal with top-down cues to enable visual search. Interesting to mention is also that Itti and Baldi [2009] recently introduced a Bayesian model of surprise which aims to predict eye movements. For tasks like watching video games, they found better correspondences to eye movements for the surprise model than for their saliency model.

[2]http://ilab.usc.edu/

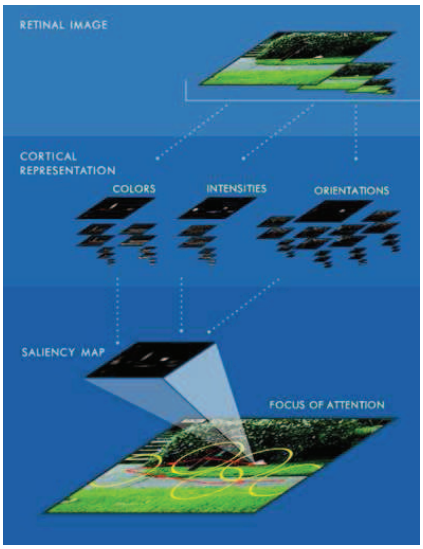


Fig. 11. Model of the *Neuromorphic Vision Toolkit (NVT)* by Itti et al. For each input image, image pyramids are computed to enable processing on different scales. Several feature channels investigate feature-dependent conspicuity independently. These are fused to a saliency map and a winner take all network determines the most salient location in this map (Fig. reprinted with permission from http://ilab.usc.edu/).

Since the NVT belongs to the best known and most distributed systems that exist, many groups tested it and suggested several improvements. For example, Draper and Lionelle [2005] came along with the system SAFE (selective attention as a front end) which shows several differences: e.g., it does not combine the feature maps across scales but keeps them, resulting in a pyramid of saliency maps. They show that this approach is more stable with respect to geometric transformations like translations, rotations, and reflections. Additionally, Frintrop [2005] suggested to separate the intensity feature computations into on-off and off-on computations instead of combining them in a single map and showed that certain pop-out effects are only detected by this separation. The same applies to the separation of red and green as well and blue and yellow.

The attention system of **Hamker** lays special emphasis on closely mimicking the neural processes in the human visual cortex [Hamker 2005; 2006]. In addition to bottom-up saliency which is similar to Itti's NVT, the system belongs to the few systems considering top-down influences. It is able to learn a target, that means it remembers the feature values of a presented stimulus. An interesting point is that Hamker's system is able to perform a very rough kind of object recognition by so called *match detection units*.

An approach to hierarchical object-based selection of regions of interest is presented by **Sun and Fisher [2003]**. Regions of interest are computed on different scales, first on a coarse scale and then, if the region is sufficiently interesting, it is investigated on a finer scale. This yields foci of attention of different extents.

Backer presented an interesting model of attention with two selection stages [Backer et al. 2001; Backer 2004]. The first stage resembles standard architectures like [Koch and Ullman 1985], but the result is not a single focus but a small number, usually 4, of salient locations. In the second selection stage, one of these locations is selected and yields a single focus of attention. The model investigates some of the more unregarded experimental data on multiple object tracking and object-based inhibition of return.

The system VOCUS of **Frintrop** has several aspects which make it well suitable for applications in computer vision and robotics. The top-down part enables an easy, user-friendly search for target objects [Frintrop 2005]. The system is largely robust to illumination and viewpoint changes and it is real-time capable (50 ms per frame for a 400×300 pixel image on a 2.8 GHz PC) [Frintrop et al. 2007].

3.6 The Evaluation of Computational Attention Systems

There are mainly two possibilities to evaluate computational attention systems. First, the obtained saliency maps can be compared with the results from psychophysical experiments to determine how well the systems simulate human behavior. Second, one can evaluate how well systems perform a certain task, how they compare to standard algorithms for these tasks, and how different systems compare to each other.

Several groups have compared the performance of bottom-up attention systems with human eye movements. These evaluations are not trivial since there is a high variability between scanpaths of different subjects and, in free-viewing tasks, there is usually no "best" scanpath. This variability may partly be explained by the fact that in human attention, always top-down cues like motivations, emotions, and pre-knowledge influence the processing. Easiest is the evaluation on simple, artificial scenes containing pop-outs, as the one in Figure 3. Here, it is clear what the most salient spot is and most computational systems perform well in finding these pop-outs immediately (cf. [Frintrop 2005]).

Several groups have also compared the correspondence of saliency models with eye movements for natural scenes. Parkhurst et al. [2002] reported a significant coherence of human eye movements with a computational saliency map, which was highest for the initial fixation. Especially high correspondence was found for fixations that followed stimulus onset. The correspondence was higher for artificial images like fractals than for natural images, probably because the top-down influence is lower for artificial scenes. Also Tatler et al. [2005] discovered that features like contrast, orientation energy, and chromaticity all differ between fixated and non-fixated locations. The consistency of fixated locations between participants was highest for the first few fixations. In [Tatler et al. 2006] they state that especially short saccades are dependent on the image features while long are less so. It may be also noted that the first fixations of subjects who have the task of viewing scenes on a monitor tend to be clustered around the middle of the screen. This is called the *central bias*. While a final explanation is still to be found, Tatler [2007] provides several results and an interesting discussion on this topic. Probably the broadest evaluation of bottom-up saliency was presented by Elazary and Itti [2008]. They used the *LabelMe* database which contained 24 836 photographs of natural scenes in which objects were manually marked and labeled by a large population

of users. They found that the hot spots in the saliency map predict the locations of objects significantly above chance.

Henderson et al. [2007] investigated the influence of visual saliency on fixated locations during active search. They compared predictions from the bottom-up saliency model of Itti and Koch with fixation sequences of humans and concluded that the evidence for the visual saliency hypothesis in active visual search is relatively weak. This is not surprising since obviously top-down cues are essential in active search. Attention systems able to perform active search, as for example [Navalpakkam et al. 2005] or [Frintrop 2005], are likely to achieve a larger correspondence in such settings. Other work comparing computational saliency with human visual attention is presented in [Ouerhani et al. 2004; Bruce and Tsotsos 2005b; Itti 2005; Peters et al. 2005; Peters and Itti 2008]. For example, Peters and Itti [2008] compared human eye movements with the prediction of a computational attention system in video games.

Several people have investigated how strongly the separate feature channels correspond to eye movements. Parkhurst et al. [2002] found that not one channel is generally superior to the others, but that the relative strength of each feature dimension depends on the image type: for fractal and home interior images, color was superior, for natural landscapes, buildings and city scenes, intensity was dominant. Color and intensity contributed in general more than orientation, but for buildings and city scenes, orientation was superior to color. Also Frey et al. [2008] found such a dependency of performance on different categories. While color had almost no influence on overt attention for some categories like faces, there is a high influence for images from other categories, e.g., *Rainforest*. This is especially interesting since there is evidence that it is the rainforest where the trichromatic color vision evolved [Sumner and Mollon 2000]. Furthermore, Frey et al. [2008] found that the saliency model they investigated (Itti's NVT) exhibits good prediction performance of eye movements in more than half of the investigated categories. Kootstra et al. [2008] found that symmetry is a better predictor for human eye movements than contrast. Tatler et al. [2005] and Baddeley and Tatler [2006] compared the visual characteristics on images at fixated and non-fixated locations with signal detection and information theoretic techniques. In [Tatler et al. 2005], they state that "contrast and edge information was more strongly discriminatory than luminance or chromaticity". In [Baddeley and Tatler 2006], they found that the mapping was dominated by high frequency edges and that low frequency edges and contrast on the other hand had an inhibitory effect. They claim that previous correlates between fixations and contrast were simply artefacts of their correlates with edges. Color was not investigated in these experiments. In active search tasks, Vincent et al. [2007] discovered that color made the largest contribution for the search performance while edges made no important contribution. Altogether, it seems like further research is necessary to determine which features are most relevant in which settings and tasks.

Evaluating computational top-down attention systems in visual search tasks is easier than evaluating bottom-up systems since a target is known and the detection rate for this target can be determined. As in human perception, the performance depends on the target and on the setting. Some results can be found in [Hamker

2005; Navalpakkam et al. 2005; Frintrop 2005; Vincent et al. 2007]. For example in [Frintrop 2005], a target object in natural environments was in most cases found with the first fixation, e.g., a fire extinguisher in a corridor or a key fob on a desk. Vincent et al. [2007] have found a relatively low fixation probability for real world targets in their approach, especially for difficult search targets like a wine glass. These approaches are difficult to compare since they operate on different data. So, it is hard to distinguish which differences come from the implementation of the system and which from the difference in data. In [Frintrop 2005], we present a comparison of VOCUS with the systems in [Hamker 2005] and [Navalpakkam et al. 2005], each on the same image data sets.

Another possibility to evaluate the quality of attentional systems is their use in applications. If the system performance is increased in either time or quality, it is not necessarily important to achieve exact correspondences to human eye movements. Several application domains of visual attention systems will be presented in the next section.

4. APPLICATIONS IN COMPUTER VISION AND ROBOTICS

Restricting the large amount of visual data to a manageable rate has been an omnipresent topic during the last years in research areas concerned with image data. Although machines became much faster and hardware cheaper, processing all information is still not possible and will either not be possible in the future. The reason is that the complexity of many problems is very high — as mentioned before, unbounded visual search is NP-complete — so finding a polynomial solution for such a problem is extremely unlikely.

Therefore, concepts like selective visual attention arouse much interest in computer vision and robotics. They provide an intuitive method to determine the most interesting regions of an image in a "natural", human-like way and are a promising approach to improve computational vision systems.

We organize the applications of computational attention systems roughly into three categories: in the first, low-level category, attentional regions are used as low-level features, so called *interest points* or *regions of interest (ROIs)* for tasks like image matching (sec. 4.1). The second, mid-level category considers attention as a front-end for high-level tasks as object recognition (sec. 4.2). In the third, highest-level category, attention is used in a human-like way to guide the action of an autonomous system like a robot, i.e., to guide object manipulation or human-robot interaction (sec. 4.3).

4.1 Attention as Salient Interest Point Detector

Detecting regions of interest is an important method in many computer vision tasks. Many methods exist to detect interest points or regions in images, an overview is provided by Tuytelaars and Mikolajczyk [2007]. An alternative to these approaches are attention regions. While common detectors usually work on gray-scale images, computational attention systems integrate several features and determine the overall saliency from many cues. Another difference is that attention systems focus on a few, highly discriminative features while common detectors often tend to find many similar regions. Depending on the application, the restriction to a few discriminative regions is favorable because it reduces computation complexity. We have shown

that the repeatability of regions in different scenes is significantly higher for salient regions than for regions detected by standard detectors [Frintrop 2008][3].

One application area of salient ROIs is **image segmentation**. Segmentation is the problem of grouping parts of an image together according to some measure of similarity. The automatic segmentation of images into regions usually deals with two major problems: first, setting the starting points for segmentation (seeds) and second, choosing the similarity criterion to segment regions. Ouerhani [2003] presents an approach that supports both aspects by visual attention: the saliency spots of the attention system serve as natural candidates for the seeds and the homogeneity criterion is adapted according to the features that discriminate a region from its surrounding. A comparison to other segmentation algorithms has, to our knowledge, not yet been done.

Another application area is **image and video compression**. The idea is to compress non-focused regions stronger than focused ones, based on the findings that there is correspondence between the regions focused by humans and those detected by computational attention systems. Ouerhani [2003] performs *focused image compression* with a visual attention system. A color image compression method adaptively determines the number of bits to be allocated for coding image regions according to their saliency. Regions with high saliency have a higher reconstruction quality than less salient regions. Itti [2004] uses his attention system to perform video compression by blurring every frame, increasingly with distance from salient locations.

A large field with many application areas is **image matching**, i.e., finding correspondences between two or more images which show the same scene or the same object. When searching for correspondences between two images, it is computationally too expensive to compare images on a pixel basis and variations in illumination and viewpoint make such a simple approach unsuitable. Instead, ROIs can be used to find such correspondences. This is necessary for tasks like stereo matching, building panoramas, place recognition, or robot localization.

To compare two ROIs, a *descriptor* is required. Attentional descriptors are vectors which determine the feature saliencies of the ROI and its surrounding (cf. sec. 3.1) [Navalpakkam et al. 2005; Frintrop et al. 2005]. Since matching with an attentional descriptor alone is usually not powerful enough, several groups have combined their attention regions with other detectors or descriptors. A common approach is the SIFT descriptor (scale invariant feature transform) which captures the gradient magnitude in the surrounding of a region [Lowe 2004]. It is very powerful also under image transformations. Walther [2006] and Siagian and Itti [2009] detect SIFT keypoints (intensity extrema in scale space and combined with a SIFT descriptor) inside the attention regions, i.e., the attentional regions determine a search area whereas the matching is based on the SIFT keypoints. Note however that this approach is sometimes problematic since attention regions favor homogeneous regions whereas corner features are usually detected at textured areas. Thus, the combination results often in very few features which makes matching difficult. In our work, we obtained better results by directly applying a SIFT descriptor to the attention regions [Frintrop and Jensfelt 2008].

[3]See also http://www.informatik.uni-bonn.de/∼frintrop/research/saliency.html

One application scenario in which image matching is used is **robot localization**. Based on a known map of the surrounding, the robot has to determine its position in this map by interpreting its sensor data. Standard approaches for such problems use range sensors such as laser scanners and there are good and stable solutions for such problems. However, in outdoor environments and open areas, the standard methods for localization are likely to fail. Instead, a promising approach is localization by detecting visual landmarks with a known position. Attentional mechanisms can facilitate the search of landmarks during operation by selecting interesting regions in the sensor data. An early project that followed this approach was the ARK project [Nickerson et al. 1998]. It relied on hand-coded maps, including the locations of known static obstacles as well as the locations of natural visual landmarks. Ouerhani et al. [2005] track salient spots over time and use them as landmarks for robot localization. The results must be considered preliminary since testing was done on the training sequence on a straight corridor without loops. **Scene classification** and global localization based on salient landmarks was presented in [Siagian and Itti 2009]. Additionally to the landmarks, the authors use the "gist" of the scene, a feature vector which captures the appearance of the scene, to obtain a coarse localization hypothesis.

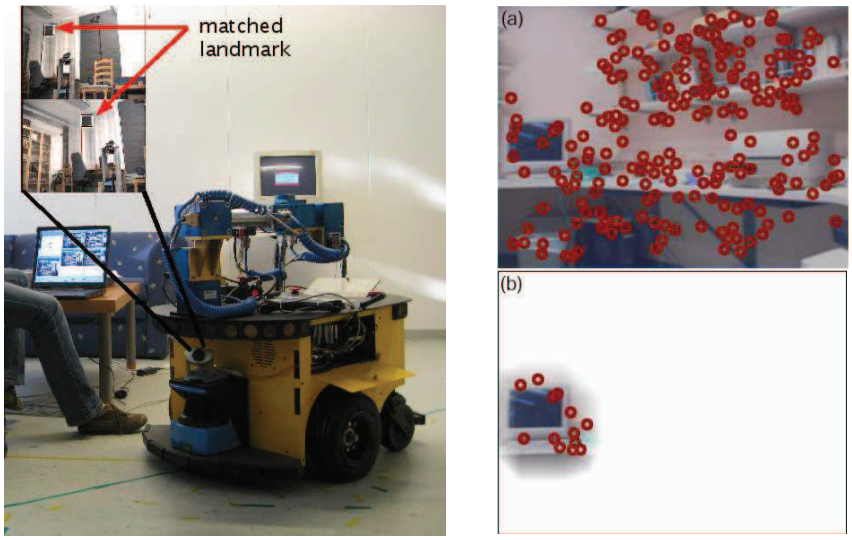
In the above examples, a map of the environment is initially known. Usually, it is obtained in a training phase. A more difficult task is **simultaneous localization and mapping (SLAM)** in which a robot initially does not know anything about its environment and has to build a map and localize itself inside the map at the same time. This topic was up to now rarely investigated in combination with visual attention. Frintrop et al. investigated the combination of visual attention and SLAM [Frintrop and Jensfelt 2008]. The salient regions are detected with the attention systems VOCUS, matched with a SIFT descriptor and tracked over several frames to obtain a 3D position of the landmarks. Finally, they are matched to database entries of the landmarks to detect if the robot closed a loop, i.e., returned to a previously visited area (see Fig. 12 (a)).

In addition to the presented application areas, image matching with attentional ROIs is sometimes also used for object recognition. This aspect will be described in the next section.

4.2 Attention as Front-end for Object Recognition

Probably the most suggestive application of an attention system is object recognition since the two-stage approach of a preprocessing attention system and a classifying recognizer mimics human perception [Neisser 1967]. Miau et al. [2001] present a biologically motivated approach that combines an attentional front-end with the biologically motivated object recognition system HMAX [Riesenhuber and Poggio 1999] which simulates processes in human cortex and has rather limited capabilities. It is restricted to recognize simple artificial objects like circles or rectangles. Miau et al. [2001] also replaced the HMAX system by a support vector machine to detect pedestrians in natural images. This approach is much more powerful with respect to the recognition rate but computationally expensive.

Salah et al. [2002] combine an attention system with neural networks and an observable Markov model for handwritten digit recognition and face recognition and Ouerhani [2003] presents an attention-based traffic sign recognition system. In



(a) Robot localization and mapping (b) Object recognition

Fig. 12. Two application scenarios for visual attention systems: (a) Robot localization and mapping: robot Dumbo corrects its position estimate by detecting a landmark which it has seen before. Landmark detection is done with the attention system VOCUS. The top-left corner shows the currently seen frame (top) and the frame from the database (bottom) with the matched landmark [Frintrop and Jensfelt 2008]. (b) Object recognition: top: SIFT keypoints are extracted for the whole image. Bottom: attentional regions of interest restrict the keypoints to regions which are likely to contain objects. This enables unsupervised learning in cluttered scenes (Fig. reprinted with permission from [Walther 2006]).

[Frintrop et al. 2004], we have combined an attention system with an AdaBoost-based object classifier [Viola and Jones 2004] which was trained for objects in laser scanner data. Walther [2006] combine an attention system with an object recognizer based on SIFT features [Lowe 2004] and show that the recognition results are improved by the attentional front-end (see Fig. 12 (b)).

All of these systems rely only on bottom-up information and therefore on the assumption that the objects of interest are sufficiently salient by themselves. Non-salient objects are not detected. For some object classes like traffic signs which are intentionally designed salient, this works quite well; for other applications, top-down information is needed to enable the system to focus on the desired objects. A combination of a top-down modulated computational attention system with a classifier is presented by Mitri et al. [2005]. Here, the attention system VOCUS generates object hypotheses which are verified or falsified by a classifier. For the application of ball detection in the robot soccer scenario ROBOCUP[4], the amount of false detections is reduced significantly.

In the above mentioned approaches, the attentional part is separated from the object recognition; both systems work independently. In human perception, these processes are strongly intertwined. A few groups have recently started to work

[4]http://www.robocup.org

on approaches in which both processes share resources. Hamker [2005] introduces *match detection units* that compare the encoded pattern with the target template. If these patterns are similar, an eye movement is initiated towards this region and the target is said to be detected. Currently, results have to be considered conceptual since recognition does not consider spatial configuration of features and recognizes only patterns that are presented with the same orientation as during learning. An interesting approach is presented by Walther and Koch [2007]. The authors suggest a unifying framework for object recognition and attention. It is based on the HMAX model for object recognition and modulates the activity by spatial and feature modulation functions which suppress or enhance locations or features due to spatial attention.

Another interesting approach is provided by Rybak et al. [1998]: although the attentional part of their system is rather limited (it uses only one feature (orientation) and no target-specific tuning of the feature computations), they present a sophisticated approach to investigate an image guided by prior knowledge. In a memorizing mode, a sequence of fixation points is determined and stored in two kinds of memories: the sensory memory ("what"-structure) stores the features of the fixations and the motor memory ("where"-structure) stores the relative shifts between the fixations. This information is used in search mode to guide the visual search and to compare the stored fixation patterns with the current image.

A different view on attention for object recognition present Fritz et al. [2004]: an information-theoretic saliency measure is used to determine discriminative regions of interest in objects. The saliency measure is computed by the conditional entropy of estimated posteriors of the local appearance patterns. That means, regions of an object are considered as salient if they discriminate the object well from other objects in an object data base. A similar approach pursue Pessoa and Exel [1999].

4.3 Attention Systems for Guiding Robot Action

A robot which has to act in a complex world faces the same problems as a human: it has to decide what to do next. Because of limited resources, usually only one task can be performed at a time: the robot can only manipulate one object, it can only follow one object with the camera, and it can only interact with one person at the same time (even if these capabilities could be slightly extended by additional hardware to a few parallel tasks, such extensions are very limited). Thus, even if computational power would allow us to find all correspondences, to recognize all objects in an image, and process everything of interest, it would still be necessary to filter out the relevant information to determine the next action. This decision is based first, on the current sensor input and second, on the internal state, for example the current tasks and goals.

A topic in which the decision about the next action is intrinsically based on visual data is **active vision**, i.e., the problem of where to look next. It deals with controlling "the geometric parameters of the sensory apparatus ... in order to improve the quality of the perceptual results" [Aloimonos et al. 1988]. Thus, it is the technical equivalent for overt attention: it directs the camera to regions of potential interest as the human visual system directs the gaze. Active vision is of special interest in robotics: it makes "vision processing more robust and more

closely tied to the activities that a robotic system may be engaged in" [Clark and Ferrier 1989].

One of the first approaches to realize an active vision system with the help of visual attention was presented by Clark and Ferrier [1988]. They describe how to steer a binocular robotic head with visual attention and perform simple experiments to fixate and track the most salient region in artificial scenes composed of geometric shapes. [Mertsching et al. 1999; Bollmann 1999] use the neural active vision system NAVIS once with a fixed stereo camera head and once on a mobile robot with a monocular camera head. Vijayakumar et al. [2001] present an attention system which is used to guide the gaze of a humanoid robot. The authors consider only one feature, visual flow, which enables the system to attend to moving objects. To simulate the different resolutions of the human eye, two cameras per eye are used: one wide-angle camera for peripheral vision and one narrow-angle camera for foveal vision. Dankers et al. [2007] introduced an architecture for reactive visual analysis of dynamic scenes as part of an active stereo vision system. Saliency is computed for each camera separately. Active gaze control for simultaneous robot localization and mapping was recently presented in [Frintrop and Jensfelt 2008]. The robot actively controls the camera by switching between the behaviors tracking, redetection and exploration. Thus, it obtains a better distribution of landmarks and facilitates the redetection of landmarks.

Many of the above examples include the **visual tracking** problem, i.e., the problem of consistently following a region or object over several frames. The problem becomes difficult if illumination changes, if the object is partially and/or temporary occluded and if not only the object or the camera but both of them are mobile. Walther et al. [2004] track objects in underwater videos by detecting them with a bottom-up attention system and tracking them with Kalman filters. Currently, we investigate general object tracking based on visual attention [Frintrop and Kessel 2009]. The appearance of an object is quickly learned from a single frame and the most salient part of the person is redetected with top-down directed attention in subsequent frames. An extension of this work deals with people tracking from a mobile platform, an important task for service robots [Frintrop et al. 2010].

Another area in which the visual input determines the next action is **object manipulation**. A robot that has to grasp and manipulate objects has to detect and probably also to recognize the object first. Attentional mechanisms can support these tasks. For example, Bollmann et al. [1999] present a robot that uses the active vision system NAVIS to play at dominoes. In [Rae 2000], a robot arm has to grasp an object a human has pointed at. The group around Tsotsos is working on a smart wheelchair to support disabled children. The wheelchair has a display as easily accessible user interface which shows pictures of places and toys. Once a task like "go to table, point to toy" is selected, the system drives to the selected location and searches for the specified toy, using mechanisms based on a visual attention system (see Fig. 13) [Tsotsos et al. 1998; Rotenstein et al. 2007].

In the field of **robot navigation**, the problem of **visual servoing** has become a well-established robot control technique which integrates vision in feedback control loops. The technique is mainly employed for controlling the robot's position. Clark and Ferrier [1992] describe how to realize a visual servo control system which

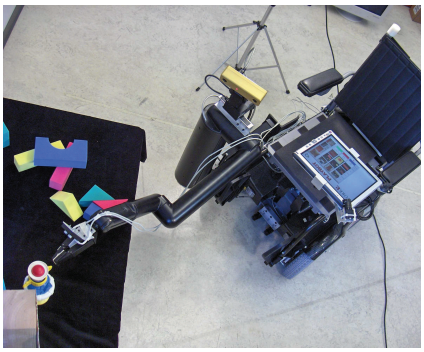


Fig. 13. PlayBot: a visually guided robotic wheelchair for disabled children. The selective tuning model of visual attention supports the detection of objects of interest (Fig. reprinted with permission from http://www.cse.yorku.ca/∼playbot).

implements attentive control of a binocular vision system. Results on simple artificial scenes in which the most salient region is fixated and tracked are shown in [Clark and Ferrier 1988]. In [Scheier and Egner 1997] a mobile robot uses an attention system to approach large objects. Since larger objects have a higher saliency, only the regions with the highest saliency have to be approached. In [Baluja and Pomerleau 1997], an attention system is used to support autonomous road following by highlighting relevant regions in a saliency map. Borji [2009] investigates the control of motor commands for an artificial agent in a navigation scenario by reinforcement learning. The current state of the system is derived from object and scene recognition at the focus of attention.

Finally, **human-robot interaction** is an intuitive application area for computational attention systems. If robots shall purposefully interact with humans, it is convenient if both attend to the same object or region of interest. A computational attention system similar to the human one can help a robot to focus on the same region as a human. Breazeal [1999] introduces a robot that shall actively look at people or toys. Although top-down information would be necessary to focus on a particular object relevant for a certain task, bottom-up information can be useful, too, if it is combined with other cues. For example, Heidemann et al. [2004] combine an attention system with a system that follows the direction of a pointing finger and can adjust to the selected region accordingly. This approach was used by Rae [2000] to guide a robot arm towards an object and grasp it. Belardinelli [2008] presents methods to let a robot learn visual scene exploration by imitating human gaze shifts. Nagai [2009] developed an action learning model based on spatial and temporal continuity of bottom-up features. Finally, an interesting sociological study in which the interaction of a human with a robot simulation is investigated is presented by Muhl et al. [2007]. Human subjects had to show an object to a robot face on a screen which attended to the object with help of a visual attention system. If the robot was artificially diverted and directed its gaze away from the object, humans tried to reobtain the robots attention by waving hands, making noise, or approaching to the robot. This shows that people established a communicative space with the robot and accepted it as a social partner.

5. DISCUSSION AND CONCLUSION

This paper gives a broad overview over computational visual attention systems and their cognitive foundations and aims to bridge the gap between different research areas. Visual attention is a highly interdisciplinary field and the disciplines investigate the area from different perspectives. Psychologists usually investigate human behavior on special tasks to understand the internal processes in the brain, resulting often in psychophysical theories or models. Neuro-biologists take a view directly into the brain with new techniques like functional Magnetic Resonance Imaging (fMRI). These methods visualize which brain areas are active under certain conditions. Computer scientists use the findings from psychology and biology to build improved technical systems.

During the last years, the different disciplines have profited considerably from each other. Psychologists refer to neuro-biological findings to improve their attention models and neuro-biologists consider psychological experiments to interpret their data. Additionally, more and more psychologists implement their models computationally or refer to computational models to verify if the behavior of the systems equals human perception. These findings help to improve the understanding of the mechanisms and can also lead to improved computational systems.

Of course, in all of the three areas presented in this paper, namely human attention, computational systems, and applications, there are still many open questions. Let us try to address some of them.

One important question is, what are the basic features of attention? Although intensively studied, this question is still not fully answered (see e.g. [Wolfe and Horowitz 2004]). Other research questions relate to how these features interact. The theory that peak salience computed from local feature contrast maxima in several feature dimensions determine human fixations has been questioned in some articles. For example, the correlations between local image statistics and the locations of human fixations have been investigated, leading to new hypotheses, for instance that high spatial frequency edges guide attention rather than contrast in other feature dimensions [Baddeley and Tatler 2006]. These new ideas require more investigation.

Other questions concern the nature of top-down cues and processes. Visual search in artificial search arrays has been well investigated and also studies on natural images have been done (e.g. [Peters et al. 2005]). For both, especially for the research on natural scenes, certainly open questions remain. A still largely unexplored area is the investigation of visual perception in dynamic scenes (but see e.g. [Peters and Itti 2008]) and, even more challenging, during interactions of humans in the real world (e.g. [Land 2006]). Additionally, top-down influences are not limited to target search. Other cues like prior knowledge, motivations, and emotions influence the visual system and are worth being investigated further. Interesting are also questions like "how much learning is involved in visual processing?", "how does context influence the search?" and "how much memory is involved in these mechanisms?". Some current findings on these topics can be found in [Kunar et al. 2008]. When going beyond visual attention, questions arise like "how does visual attention interact with other senses?" [Fritz et al. 2007], "which concepts of selective attention are shared in the brain among different senses?" [Ghazanfar and Schroeder 2006] and "how do visual attention and object recognition interact?".

For computational attention systems, similar questions remain, starting from "which are the optimal features?" and "how are these features integrated?" to "how do top-down cues influence the computation?" and "how do bottom-up and top-down cues interact?". However, we want to claim here that computational systems do not necessarily have to mimic biology perfectly to achieve similar performance. A camera differs from the eye and a computer is not the brain. Even parallel hardware like multi-processors or parallel computations on GPUs differ considerably from the architecture of neurons. Especially interesting is to find out which concepts of human perception make sense in computational systems and which have to be adapted accordingly.

Finally, concerning the applications of computational attention systems, a current challenge is to capacitate the systems to be used in the real world. That means, the systems have to be robust to noise, image transformations and illumination changes, and they have to be fast enough to process images at frame rate. Robustness to noise has been shown by Itti et al. [1998], invariance to 2D similarity transformations to a large extend is achieved by Draper and Lionelle [2005], and robustness of a top-down attention system to viewpoint changes and illumination variations has been shown by Frintrop [2005]. Recently, there have been approaches to extend to the concept of 2D saliency maps to 3D [Fleming et al. 2006; Schauerte et al. 2009]. The speed of the systems has prevented real-time applications for a long time. Parallelizations on several CPU's [Itti 2002], on dedicated hardware [Ouerhani 2003], or on a GPU [May et al. 2007; Xu et al. 2009] enable a significant speed-up. Also software solutions based on integral images have enabled real-time performance making the systems flexibly applicable without special hardware [Frintrop et al. 2007]. Interesting is also the investigation of how the concepts of attention apply to other sensors than cameras, e.g. laser scanners (a visual attention system based on laser scanner data is presented by Frintrop et al. [2005]). More research is necessary to find out how these concepts might be adapted to best fit the properties of different sensors and how the information from different sensors may be fused.

Computational attention has gained significantly in popularity over the last decade. First of all, adequate computational resources are now available to study attentional mechanisms with a high degree of fidelity. In addition, a large number of cognitive projects have been launched, particularly in Europe. Good examples include MACS, CogVis, POP, and SEARISE.[5] In most of these approaches, visual attention is included in the perception module and helps to deal with the complexity of the real world. Over the next few years, a number of embodied cognitive agents will be studied as part of new generation systems both in Europe and in the US. The European efforts are part of the emphasis on cognitive systems whereas the US efforts are part of the NSF Cyber Physical Systems program [Lee 2008]. As vision systems are integrated into complete systems, the need for optimization of the visual process in terms of overt and covert attention becomes more explicit. In addition the interplay between attention and tasking can be studied more explicitly. The more complex the systems and their tasks become, the more urgent the need for a pre-selecting attention system which determines in advance the regions of highest potential interest in the sensor data.

[5] http://cordis.europa.eu/ist/cognition/projects.htm#list

REFERENCES

Abdi, H. 2007. Signal detection theory (SDT). In *Encyclopedia of Measurement and Statistics*, N. Salkind, Ed. Tousand Oaks (CA): Sage.

Aloimonos, Y., Weiss, I., and Bandopadhay, A. 1988. Active vision. *International Journal of Computer Vision (IJCV) 1,* 4, 333–356.

Aristotle. *On Sense and the Sensible*. The Internet Classics Archive, 350 B.C.E., Translated by J. I. Beare.

Awh, E. and Pashler, H. 2000. Evidence for split attentional foci. *Journal of Experimental Psychology: Human Perception and Performance 26,* 2, 834–846.

Aziz, M. Z. and Mertsching, B. 2007. Pop-out and IOR in static scenes with region based visual attention. In *ICVS Workshop on Computational Attention and Applications (WCAA 2007)*. Applied Computer Science Group, Bielefeld University, Germany, Bielefeld, Germany.

Backer, G. 2004. Modellierung visueller Aufmerksamkeit im Computer-Sehen: Ein zweistufiges Selektionsmodell für ein Aktives Sehsystem. Ph.D. thesis, Universität Hamburg, Germany.

Backer, G., Mertsching, B., and Bollmann, M. 2001. Data- and model-driven gaze control for an active-vision system. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 23(12)*, 1415–1429.

Bacon, W. and Egeth, H. 1994. Overriding stimulus-driven attentional capture. *Perception & Psychophysics 55,* 5, 485–496.

Baddeley, R. J. and Tatler, B. W. 2006. High frequency edges (but not contrast) predict where we fixate: A bayesian system identification analysis. *Vision Research 46*, 2824–2833.

Balkenius, C. 2000. Attention, habituation and conditioning: Towards a computational model. *Cognitive Science Quarterly 1,* 2, 171–214.

Baluja, S. and Pomerleau, D. 1997. Expectation-based selective attention for visual monitoring and control of a robot vehicle. *Robotics and Autonomous Systems 22,* 3-4, 329–344.

Belardinelli, A. 2008. Salience features selection: Deriving a model from human evidence. Ph.D. thesis, Sapienza Universita di Roma, Rome, Italy.

Ben-Shahar, O., Scholl, B., and Zucker, S. 2007. Attention, segregation, and textons: Bridging the gap between object-based attention and texton-based segregation. *Vision Research 47,* 6, 173–178.

Bichot, N. P. 2001. Attention, eye movements, and neurons: Linking physiology and behavior. In *Vision and Attention*, M. Jenkin and L. R. Harris, Eds. Springer Verlag, Chapter 11.

Bichot, N. P., Rossi, A. F., and Desimone, R. 2005. Parallel and serial neural mechanisms for visual search in macaque area V4. *Science 308,* 5721 (April), 529 – 534.

Bisley, J. and Goldberg, M. 2003. Neuronal activity in the lateral intraparietal area and spatial attention. *Science 299,* 5603 (Jan.), 81–86.

Björkman, M. and Eklundh, J.-O. 2007. Vision in the real world: Finding, attending and recognizing objects. *Int'l Journal of Imaging Systems and Technology 16,* 2, 189–208.

Bollmann, M. 1999. Entwicklung einer Aufmerksamkeitssteuerung für ein aktives Sehsystem. Ph.D. thesis, Universität Hamburg, Germany.

Bollmann, M., Hoischen, R., Jesikiewicz, M., Justkowski, C., and Mertsching, B. 1999. Playing domino: A case study for an active vision system. In *Computer Vision Systems*, H. Christensen, Ed. Springer, 392–411.

Borji, A. 2009. Interactive learning of task-driven visual attention control. Ph.D. thesis, Institute for Research in Fundamental Sciences (IPM), School of Cognitive Sciences (SCS), Tehran, Iran.

Breazeal, C. 1999. A context-dependent attention system for a social robot. In *Proc. of the Int'l Joint Conference on Artifical Intelligence (IJCAI 99)*. Stockholm, Sweden, 1146–1151.

Bruce, N. D. B. and Tsotsos, J. K. 2005a. An attentional framework for stereo vision. In *Proc. of Canadian Conference on Computer and Robot Vision*.

Bruce, N. D. B. and Tsotsos, J. K. 2005b. Saliency based on information maximization. In *Proc. of Neural Information Processing Systems (NIPS)*. Vancouver, Canada.

Bundesen, C. 1990. A theory of visual attention. *Psychological Review 97*, 523–547.

Bundesen, C. 1998. A computational theory of visual attention. *Philosophical Transactions of the Royal Society of London, Series B 353*, 1271–1281.

Bundesen, C. and Habekost, T. 2005. Attention. In *Handbook of Cognition*, K. Lamberts and R. Goldstone, Eds. London: Sage Publications.

Bur, A., Wurtz, P., Müri, R., and Hügli, H. 2007. Motion integration in visual attention models for predicting simple dynamic scenes. In *Human Vision and Electronic Imaging XII. Proceedings of SPIE*, B. E. Rogowitz and S. J. Pappas, Thrasyvoulos N.and Daly, Eds. Vol. 6492.

Cameron, E., Tai, J., Eckstein, M., and Carrasco, M. 2004. Signal detection theory applied to three visual search tasks. *Spatial Vision 17,* 4-5 (Sept.). Springer.

Carrasco, M., Evert, D. L., Chang, I., and Katz, S. M. 1995. The eccentricity effect: Target eccentricity affects performance on conjunction searches. *Perception & Psychophysics 57,* 8, 1241–1261.

Cassin, B. and Solomon, S. 1990. *Dictionary of Eye Terminology.* Triad Publishing Company, Gainsville, Florida.

Cave, K. R. 1999. The FeatureGate model of visual selection. *Psychological Research 62*, 182–194.

Cave, K. R. and Wolfe, J. M. 1990. Modeling the role of parallel processing in visual search. *Cognitive Psychology 22,* 2, 225–271.

Cherry, E. C. 1953. Some experiments on the recognition of speech, with one and with two ears. *Journal of the Acoustical Society of America 25*, 975–979.

Choi, S.-B., Ban, S.-W., and Lee, M. 2004. Biologically motivated visual attention system using bottom-up saliency map and top-down inhibition. *Neural Information Processing-Letters and Reviews 2,* 1.

Chun, M. M. and Jiang, Y. 1998. Contextual cueing: Implicit learning and memory of visual context guides spatial attention. *Cognitive Psychology 36*, 28–71.

Clark, J. J. and Ferrier, N. J. 1988. Modal control of an attentive vision system. In *Proc. of the 2nd International Conference on Computer Vision*. Tampa, Florida, US.

Clark, J. J. and Ferrier, N. J. 1989. Control of visual attention in mobile robots. In *IEEE Conference on Robotics and Automation*. 826–831.

Clark, J. J. and Ferrier, N. J. 1992. Attentive visual servoing. In *An Introduction to Active Vision*, A. Blake and A. Yuille, Eds. MIT Press, Cambridge Massachusetts, Chapter 10.

Connor, C. E., Egeth, H. E., and Yantis, S. 2004. Visual attention: Bottom-up versus top-down. *Current Biology 14*.

Corbetta, M. 1990. Frontoparietal cortical networks for directing attention and the eye to visual locations: Identical, independent, or overlapping neural systems? *Proc. of the National Academy of Sciences of the United States of America 95*, 831–838.

Corbetta, M. and Shulman, G. L. 2002. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews 3,* 3, 201–215.

Dankers, A., Barnes, N., and Zelinsky, A. 2007. A reactive vision system: Active-dynamic saliency. In *Proc. of the 5th Int'l Conf. on Computer Vision Systems (ICVS 2007)*. Applied Computer Science Group, Bielefeld University, Bielefeld, Germany.

Desimone, R. and Duncan, J. 1995. Neural mechanisms of selective visual attention. *Annual Reviews of Neuroscience 18*, 193–222.

Deubel, H. and Schneider, W. X. 1996. Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research 36,* 12, 1827–1837.

Draper, B. A. and Lionelle, A. 2005. Evaluation of selective attention under similarity transformations. *Journal of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance 100,* 1-2, 152–171.

Driver, J. and Baylis, G. C. 1998. Attention and visual object segmentation. In *The Attentive Brain*, R. Parasuraman, Ed. MIT Press, Cambridge, MA, 299–326.

Duncan, J. 1984. Selective attention and the organization of visual information. *Journal of Experimental Psychology 113*, 501–517.

Eckstein, M., Thomas, J., Palmer, J., and Shimozaki, S. 2000. A signal detection model predicts the effects of set size on visual search accuracy for feature, conjunction, triple conjunction, and disjunction displays. *Perception & Psychophysics 62,* 3, 425–451.

Egeth, H. E. and Yantis, S. 1997. Visual attention: control, representation, and time course. *Annual Review of Psychology 48*, 269–297.

Einhäuser, W., Spain, M., and Perona, P. 2008. Objects predict fixations better than early saliency. *Journal of Vision 8,* 14, 1–26.

Elazary, L. and Itti, L. 2008. Interesting objects are visually salient. *Journal of Vision 8,* 3:3 (Mar), 1–15.

Eriksen, C. W. and St. James, J. D. 1986. Visual attention within and around the field of focal attention: A zoom lens model. *Perception and Psychophysics 40*, 225–240.

Findlay, J. M. and Gilchrist, I. D. 2001. Active vision perspective. In *Vision & Attention*, M. Jenkin and L. R. Harris, Eds. Springer Verlag, Chapter 5, 83–103.

Findlay, J. M. and Walker, R. 1999. A model of saccade generation based on parallel processing and competitive inhibition. *Behavioral and Brain Sciences 22*, 661–721.

Fink, G., Dolan, R., Halligan, P., Marshall, J., and Frith, C. 1997. Space-based and object-based visual attention: shared and specific neural domains. *Brain 120,* 11, 2013–2028.

Fleming, K. A., Peters II, R. A., and Bodenheimer, R. E. 2006. Image mapping and visual attention on a sensory ego-sphere. In *Conference on Intelligent Robots and Systems (IROS)*. Beijing, China, 241–246.

Fragopanagos, N. and Taylor, J. 2006. Modelling the interaction of attention and emotion. *Neurocomputing 69,* 16-18, 1977–1983.

Fraundorfer, F. and Bischof, H. 2003. Utilizing saliency operators for image matching. In *Proc. of the Int'l Workshop on Attention and Performance in Computer Vision (WAPCV)*. Graz, Austria, 17–24.

Frey, H.-P., Honey, C., and König, P. 2008. What's color got to do with it? the influence of color on visual attention in different categories. *Journal of Vision 8,* 14, 1–17.

Frintrop, S. 2005. VOCUS: a visual attention system for object detection and goal-directed search. Ph.D. thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Germany. Published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer Verlag.

Frintrop, S. 2008. The high repeatability of salient regions. In *Proc. of ECCV workshop "Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environments"*.

Frintrop, S., Backer, G., and Rome, E. 2005. Goal-directed search with a top-down modulated computational attention system. In *Proc. of the Annual meeting of the German Association for Pattern Recognition (DAGM)*. Lecture Notes in Computer Science (LNCS). Springer.

Frintrop, S. and Cremers, A. B. 2007. Top-down attention supports visual loop closing. In *Proc. of European Conference on Mobile Robotics (ECMR 2007)*. Freiburg, Germany.

Frintrop, S. and Jensfelt, P. 2008. Attentional landmarks and active gaze control for visual SLAM. *IEEE Trans. on Robotics, Special Issue on Visual SLAM 24,* 5 (Oct).

Frintrop, S. and Kessel, M. 2009. Most salient region tracking. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA'09)*. Kobe, Japan.

Frintrop, S., Klodt, M., and Rome, E. 2007. A real-time visual attention system using integral images. In *Proc. of the 5th Int'l Conf. on Computer Vision Systems (ICVS)*. Bielefeld, Germany.

Frintrop, S., Königs, A., Hoeller, F., and Schulz, D. 2010. A component-based approach to visual person tracking from a mobile platform. In *accepted for the International Journal of Social Robotics*.

Frintrop, S., Nüchter, A., Surmann, H., and Hertzberg, J. 2004. Saliency-based object recognition in 3D data. In *Proc. of the Int'l Conf. on Intelligent Robots and Systems (IROS)*. Conference: Sendai, Japan, 2167 – 2172.

Frintrop, S., Rome, E., Nüchter, A., and Surmann, H. 2005. A bimodal laser-based attention system. *J. of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance in Computer Vision 100,* 1-2 (Oct-Nov), 124–151.

Fritz, G., Seifert, C., and Paletta, L. 2004. Attentive object detection using an information theoretic saliency measure. In *Proc. of the 2nd Int'l Workshop on Attention and Performance in Computational Vision (WAPCV)*, L. Paletta, J. K. Tsotsos, E. Rome, and G. W. Humphreys, Eds. Conference: Prague, Czech Republic, 136–143.

Fritz, J. B., Elhilali, M., David, S. V., and Shamma, S. A. 2007. Auditory attention - focusing the searchlight on sound. *Current Opinion in Neurobiology 17*, 437–455.

Garey, M. and Johnson, D. S. 1979. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. Freeman, San Francisco.

Gegenfurtner, K. R. 2003. Cortical mechanisms of colour vision. *Nature Reviews Neuroscience 4*, 563–572.

Ghazanfar, A. and Schroeder, C. 2006. Is neocortex essentially multisensory? *Trands Cogn Sci 10*, 278–285.

Giesbrecht, B., Wodorff, M., Song, A., and Mangun, G. 2003. Neural mechanisms of top-down control during spatial and feature attention. *Neuroimage 19*, 496–512.

Gottlieb, J. P., Kusunoki, M., and Goldberg, M. E. 1998. The representation of visual salience in monkey parietal cortex. *Nature 391*, 481–484.

Green, D. M. and Swets, J. A. 1966. *Signal detection theory and psychophysics*. Wiley New York.

Hamker, F. H. 2005. The emergence of attention by population-based inference and its role in distributed processing and cognitive control of vision. *Journal of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance 100,* 1-2, 64–106.

Hamker, F. H. 2006. Modeling feature-based attention as an active top-down inference process. *BioSystems 86*, 91–99.

Harel, J., Koch, C., and Perona, P. 2007. Graph-based visual saliency. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, Cambridge, MA, 545–552.

Heidemann, G., Rae, R., Bekel, H., Bax, I., and Ritter, H. 2004. Integrating context-free and context-dependent attentional mechanisms for gestural object reference. *Machine Vision and Applications 16,* 1, 64–73.

Heinke, D. and Humphreys, G. W. 2003. Attention, spatial representation and visual neglect: Simulating emergent attention and spatial memory in the selective attention for identification model (SAIM). *Psychological Review 110,* 1, 29–87.

Heinke, D. and Humphreys, G. W. 2004. Computational models of visual selective attention. A review. In *Connectionist models in psychology*, G. Houghton, Ed. Psychology Press, 273 – 312.

Henderson, J. M., Brockmole, J. R., Castelhano, M. S., and Mack, M. 2007. Visual saliency does not account for eye movements during visual search in real-world scenes. In *Eye movements: A window on mind and brain*, R. van Gompel, M. Fischer, W. Murray, and R. Hill, Eds. Elsevier, Oxford, 537–562.

Horowitz, T. S. and Wolfe, J. M. 2003. Memory for rejected distractors in visual search? *Visual Cognition 10,* 3, 257–298.

Humphreys, G. W. and Müller, H. J. 1993. Search via recursive rejection (SERR): A connectionist model of visual search. *Cognitive Psychology 25*, 43–110.

Itti, L. 2002. Real-time high-performance attention focusing in outdoors color video streams. In *Proc. SPIE Human Vision and Electronic Imaging IV (HVEI)*. San Jose, CA.

Itti, L. 2004. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing 13,* 10 (Oct).

Itti, L. 2005. Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition 12,* 6 (Aug), 1093–1123.

Itti, L. and Baldi, P. 2009. Bayesian surprise attracts human attention. *Vision Research 49,* 10, 1295–1306.

Itti, L., Dhavale, N., and Pighin, F. 2003. Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Proc. of the SPIE 48th Annual International Symposium on Optical Science and Technology*. Vol. 5200.

Itti, L. and Koch, C. 2001a. Computational modeling of visual attention. *Nature Reviews Neuroscience 2,* 3 (Mar), 194–203.

Itti, L. and Koch, C. 2001b. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging 10,* 1 (Jan), 161–169.

Itti, L., Koch, C., and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence 20,* 11, 1254–1259.

Johansson, R., Westling, G., Backstrom, A., and Flanagan, J. 2001. Eye-hand coordination in object manipulation. *The Journal of Neuroscience 21,* 17, 6917–6932.

Johnson, A. and Proctor, R. 2003. *Attention: theory and practice*. Sage Publications.

Jonides, J. 1981. Voluntary versus automatic control over the mind's eye movements. In *Attention and Performance IX*, A. D. Long, Ed. Lawrence Erlbaum Associates, Hillsadale, NJ, 187–203.

Kadir, T. and Brady, M. 2001. Saliency, scale and image description. *Int'l J. of Computer Vision 45,* 2, 83–105.

Kahneman, D. and Treisman, A. 1992. The reviewing of object files: Object-specific integration of information. *Cognitive Psychology 24*, 175–219.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. 1996. *Essentials of Neural Science and Behavior*. McGraw-Hill/Appleton & Lange.

Kastner, S. and Ungerleider, L. G. 2001. The neural basis of biased competition in human visual cortex. *Neuropsychologia 39*, 1263–1276.

Koch, C. and Ullman, S. 1985. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology 4,* 4, 219–227.

Kootstra, G., Nederveen, A., and de Boer, B. 2008. Paying attention to symmetry. In *Proc. of the British Machine Vision Conference (BMVC)*. Leeds, UK.

Kunar, M., Flusberg, S., and Wolfe, J. 2008. The role of memory and restricted context in repeated visual search. *Perception and Psychophysics 70*, 314–328.

Land, M. F. 2006. Eye movements and the control of actions in everyday life. *Prog Retinal & Eye Res 25*, 296–324.

Lee, E. A. 2008. Cyber physical systems: Design challenges. Tech. Rep. UCB/EECS-2008-8, EECS Department, University of California, Berkeley. Jan.

Lee, K., Buxton, H., and Feng, J. 2003. Selective attention for cue-guided search using a spiking neural network. In *Proc. of the Int'l Workshop on Attention and Performance in Computer Vision (WAPCV)*. Graz, Austria, 55–62.

Levin, D. 1996. Classifying faces by race: the structure of face categories. *Journal of Experimental Psychology: Learning, Memory, & Recognition 22*, 1364–1382.

Li, Z. 2005. The primary visual cortex creates a bottom-up saliency map. In *Neurobiology of Attention*, L. Itti, G. Rees, and J. Tsotsos, Eds. Elsevier Academic Press.

Liu, T., Slotnick, S. D., Serences, J. T., and Yantis, S. 2003. Cortical mechanisms of feature-based attentional control. *Cerebral Cortex 13,* 12.

Livingstone, M. S. and Hubel, D. H. 1987. Psychophysical evidence for separate channels for the perception of form, color, movement, and depth. *Journal of Neuroscience 7,* 11, 3416–3468.

Logan, G. D. 1996. The CODE theory of visual attention: an integration of space-based and object-based attention. *Psychological Review 103*, 603–649.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int'l J. of Computer Vision (IJCV) 60,* 2, 91–110.

Maki, A., Nordlund, P., and Eklundh, J.-O. 2000. Attentional scene segmentation: Integrating depth and motion. *Computer Vision and Image Understanding (CVIU) 78,* 3, 351–373.

Marr, D. 1982. *VISION – A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company, New York (NY).

Maunsell, J. H. R. 1995. The brain's visual world: representation of visual targets in cerebral cortex. *Science 270*, 764–769.

May, S., Klodt, M., and Rome, E. 2007. GPU-accelerated Affordance Cueing based on Visual Attention. In *Proc. of Int'l Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 3385–3390.

Mazer, J. A. and Gallant, J. L. 2003. Goal-related activity in V4 during free viewing visual search. Evidence for a ventral stream visual salience map. *Neuron 40,* 6, 1241–50.

McMains, S. A. and Somers, D. C. 2004. Multiple spotlights of attentional selection in human visual cortex. *Neuron 42*, 677–686.

Mertsching, B., Bollmann, M., Hoischen, R., and Schmalz, S. 1999. The neural active vision system NAVIS. In *Handbook of Computer Vision and Applications*, B. Jähne, H. Haussecke, and P. Geissler, Eds. Vol. 3. Academic Press, 543–568.

Miau, F., Papageorgiou, C., and Itti, L. 2001. Neuromorphic algorithms for computer vision and attention. In *Proc. SPIE 46 Annual Int'l Symposium on Optical Science and Technology.* Vol. 4479. 12–23.

Milanese, R. 1993. Detecting salient regions in an image: From biological evidence to computer implementation. Ph.D. thesis, University of Geneva, Switzerland.

Milanese, R., Wechsler, H., Gil, S., Bost, J., and Pun, T. 1994. Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '94).* Conference: Seattle, 781–785.

Mitri, S., Frintrop, S., Pervölz, K., Surmann, H., and Nüchter, A. 2005. Robust object detection at regions of interest with an application in ball recognition. In *IEEE Proc. of the Int'l Conf. on Robotics and Automation (ICRA '05).* Conference: Barcelona, Spain, 126–131.

Mozer, M. C. 1987. Early parallel processing in reading: a connectionist approach. In *Attention and performance XII: The psychology of reading*, M. Coltheart, Ed. Hove, UK: Lawrence Erlbaum Associated Ltd., 83–104.

Muhl, C., Nagai, Y., and Sagerer, G. 2007. On constructing a communicative space in HRI. In *Proc. of the 30th German Conference on Artificial Intelligence (KI 2007)*, J. Hertzberg, M. Beetz, and R. Englert, Eds. Springer, Osnabrück, Germany.

Nagai, Y. 2009. From bottom-up visual attention to robot action learning. In *IEEE 8th Int'l Conf. on Development and Learning.*

Nakayama, K. and Mackeben, M. 1989. Sustained and transient components of focal visual attention. *Vision Research 29*, 1631–1647.

Nakayama, K. and Silverman, G. H. 1986. Serial and parallel processing of visual feature conjunctions. *Nature 320*, 264–265.

Navalpakkam, V. and Itti, L. 2006a. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR).*

Navalpakkam, V. and Itti, L. 2006b. Top-down attention selection is fine-grained. *Journal of Vision 6,* 11 (Oct), 1180–1193.

Navalpakkam, V., Rebesco, J., and Itti, L. 2004. Modeling the influence of knowledge of the target and distractors on visual search. *Journal of Vision 4,* 8, 690.

Navalpakkam, V., Rebesco, J., and Itti, L. 2005. Modeling the influence of task on attention. *Vision Research 45,* 2, 205–231.

Neisser, U. 1967. *Cognitive Psychology.* Appleton-Century-Crofts, New York.

Nickerson, S. B., Jasiobedzki, P., Wilkes, D., Jenkin, M., Milios, E., Tsotsos, J. K., Jepson, A., and Bains, O. N. 1998. The ARK project: Autonomous mobile robots for known industrial environments. *Robotics and Autonomous Systems 25,* 1-2, 83–104.

Nothdurft, H.-C. 2005. Salience of feature contrast. In *Neurobiology of Attention*, L. Itti, G. Rees, and J. K. Tsotsos, Eds. Elsevier, 233–239.

Ogawa, T. and Komatsu, H. 2004. Target selection in area V4 during a multidimensional visual search task. *Journal of Neuroscience 24,* 28, 6371–6382.

Oliva, A. 2005. Gist of the scene. In *Neurobiology of Attention*, L. Itti, G. Rees, and J. Tsotsos, Eds. Elsevier Academic Press, Chapter 41, 251–257.

Oliva, A., Torralba, A., Castelhano, M. S., and Henderson, J. M. 2003. Top-down control of visual attention in object detection. In *Int'l Conf. on Image Processing (ICIP).* Barcelona, Spain, 253–256.

Olshausen, B., Anderson, C., and van Essen, D. 1993. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience 13,* 11 (November), 4700–4719.

Olshausen, B. A. and Field, D. J. 2005. How close are we to understanding V1? *Neural Computation 17,* 8, 1665 – 1699.

Olshausen, B. A. and Field, D. J. 2006. What is the other 85% of V1 doing? In *23 Problems in Systems Neuroscience*, L. V. Hemmen and T. Sejnowslti, Eds. Oxford University Press.

Ouerhani, N. 2003. Visual attention: From bio-inspired modeling to real-time implementation. Ph.D. thesis, Institut de Microtechnique Université de Neuchâtel, Switzerland.

Ouerhani, N., Bur, A., and Hügli, H. 2005. Visual attention-based robot self-localization. In *Proc. of European Conference on Mobile Robotics (ECMR 2005)*. Ancona, Italy, 8–13.

Ouerhani, N. and Hügli, H. 2000. Computing visual attention from scene depth. In *Proc. of Int'l Conf. on Pattern Recognition (ICPR 2000)*. Vol. 1. IEEE Computer Society Press, 375–378.

Ouerhani, N., Jost, T., Bur, A., and Hügli, H. 2006. Cue normalization schemes in saliency-based visual attention models. In *Proc. Int'l Cognitive Vision Workshop*. Graz, Austria.

Ouerhani, N., von Wartburg, R., Hügli, H., and Müri, R. 2004. Empirical validation of the saliency-based model of visual attention. *Electronic Letters on Computer Vision and Image Analysis 3,* 1, 13–24.

Palmer, J., Ames, C., and Lindsey, D. 1993. Measuring the effect of attention on simple visual search. *J. of experimental psychology. Human perception and performance 19,* 1, 108–130.

Palmer, S. E. 1999. *Vision Science, Photons to Phenomenology.* The MIT Press, Cambridge, MA.

Parkhurst, D., Law, K., and Niebur, E. 2002. Modeling the role of salience in the allocation of overt visual attention. *Vision Research 42,* 1, 107–123.

Pashler, H. 1997. *The Psychology of Attention.* MIT Press, Cambridge, MA.

Pessoa, L. and Exel, S. 1999. Attentional strategies for object recognition. In *Proc. of the International Work-Conference on Artificial and Natural Neural Networks (IWANN '99)*, J. Mira and J. Sachez-Andres, Eds. Lecture Notes in Computer Science (LNCS), vol. 1606. Springer, Alicante, Spain, 850–859.

Peters, R., Iyer, A., Itti, L., and Koch, C. 2005. Components of bottom-up gaze allocation in natural images. *Vision Research 45*, 2397–2416.

Peters, R. J. and Itti, L. 2008. Applying computational tools to predict gaze direction in interactive visual environments. *ACM Trans. on Applied Perception 5,* 2, Article 8.

Phaf, R. H., van der Heijden, A. H. C., and Hudson, P. T. W. 1990. SLAM: A connectionist model for attention in visual selection tasks. *Cognitive Psychology 22*, 273–341.

Posner, M. and Cohen, Y. 1984. Components of visual orienting. In *Attention and Performance X*, H. Bouma and D. Bouwhuis, Eds. London: Erlbaum, 531–556.

Posner, M. I. 1980. Orienting of attention. *Quarterly Journal of Experimental Psychology 32*, 3–25.

Posner, M. I. and Petersen, S. E. 1990. The attentional system of the human brain. *Annual Review of Neuroscience 13*, 25–42.

Postma, E. 1994. Scan: A neural model of covert attention. Ph.D. thesis, Rijksuniversiteit Limburg, Wageningen.

Pylyshyn, Z. and Storm, R. 1988. Tracking multiple independent targets: evidence for a parallel tracking mechanism. *Spatial Vision 3*, 179–197.

Pylyshyn, Z. W. 2003. *Seeing and Visualizing: It's Not What You Think.* MIT Press.

Rae, R. 2000. Gestikbasierte Mensch-Maschine-Kommunikation auf der Grundlage visueller Aufmerksamkeit und Adaptivität. Ph.D. thesis, Universität Bielefeld, Germany.

Ramström, O. and Christensen, H. I. 2002. Visual attention using game theory. In *Proc. Workshop on Biologically Motivated Computer Vision (BMCV)*. Vol. 2525. Springer Verlag, Lecture Notes in Computer Science (LNCS).

Ramström, O. and Christensen, H. I. 2004. Object based visual attention: Searching for objects defined by size. In *Proc. of Int'l Workshop on Attention and Performance in Computational*

Vision (WAPCV), L. Paletta, J. K. Tsotsos, E. Rome, and G. W. Humphreys, Eds. Conference: Prague, Czech Republic, 9–16.

Rao, R., Zelinsky, G., Hayhoe, M., and Ballard, D. 2002. Eye movements in iconic visual search. *Vision Research 42*, 1447–1463.

Rasolzadeh, B., Björkman, M., Huebner, K., and Kragic, D. 2009. An active vision system for detecting, fixating and manipulating objects in real world. *International Journal of Robotics Research*. (in press).

Rauschenberger, R. 2003. Attentional capture by auto- and allo-cues. *Psychonomic Bulletin & Review 10,* 4 (Dec.), 814–842.

Rensink, R. A. 2000. The dynamic representation of scenes. *Visual Cognition 7*, 17–42.

Rensink, R. A., O'Regan, J. K., and Clark, J. J. 1997. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science 8*, 368–373.

Riesenhuber, M. and Poggio, T. 1999. Hierarchical models of object recognition in cortex. *Nature Neuroscience 2,* 11, 1019–1025.

Rosenholtz, R. 2001. Search asymmetries? What search asymmetries? *Perception & Psychophysics 63,* 3, 476–489.

Rotenstein, A., Andreopoulos, A., Fazl, E., Jacob, D., Robinson, M., Shubina, K., Zhu, Y., and Tsotsos, J. 2007. Towards the dream of intelligent, visually-guided wheelchairs. In *Proc. 2nd Int'l Conf. on Technology and Aging*. Toronto, Canada.

Rothenstein, A. and Tsotsos, J. 2006a. Attention links sensing to recognition. *Image & Vision Computing Journal, Special Issue on Cognitive Vision Systems 26,* 1, 114–126.

Rothenstein, A. and Tsotsos, J. 2006b. Selective tuning: Feature binding through selective attention. In *Proc. of International Conference on Artificial Neural Networks*. Athens, Greece.

Rybak, I., Gusakova, V., Golovan, A., Podladchikova, L., and Shevtsova, N. 1998. A model of attention-guided visual perception and recognition. *Vision Research 38*, 2387–2400.

Sabra, A. I. 1989. *The Optics of Ibn Al-Haytham*. The Warburg Institute, University of London.

Salah, A., Alpaydin, E., and Akrun, L. 2002. A selective attention based method for visual pattern recognition with application to handwritten digit recognition and face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 24,* 3, 420–425.

Sandini, G. and Metta, G. 2002. Retina-like sensors: motivations, technology and applications. In *Sensors and Sensing in Biology and Engineering*. Springer Verlag, New York, NY.

Schauerte, B., Richarz, J., Plötz, T., Thurau, C., and Fink, G. A. 2009. Multi-modal and multi-camera attention in smart environments. In *Proc. of Multimodal Interfaces and Workshop on Machine Learning for Multimodal Interaction*.

Scheier, C. and Egner, S. 1997. Visual attention in a mobile robot. In *Proc. of the IEEE Int'l Symposium on Industrial Electronics*. 48–53.

Scholl, B. J. 2001. Objects and attention: the state of the art. *Cognition 80*, 1–46.

Shulman, G., Remington, R., and McLean, J. 1979. Moving attention through visual space. *J. of Experimental Psychology. Human Perception and Performance 5,* 3, 522–526.

Siagian, C. and Itti, L. 2009. Biologically inspired mobile robot vision localization. *IEEE Transaction on Robotics 25,* 4 (July), 861–873.

Simons, D. J. and Levin, D. T. 1997. Change blindness. *Trends in Cognitive Sciences 1*, 261–267.

Styles, E. A. 1997. *The Psychology of Attention*. Psychology Press Ltd, East Sussex, UK.

Sumner, P. and Mollon, J. 2000. Catarrhine photopigments are optimized for detecting targets against a foliage background. *J. of Experimental Biology 203*, 1963–1986.

Sun, Y. and Fisher, R. 2003. Object-based visual attention for computer vision. *Artificial Intelligence 146,* 1, 77–123.

Tatler, B. W. 2007. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *J. of Vision 14,* 7, 1–17.

Tatler, B. W., Baddeley, R. J., and Gilchrist, I. D. 2005. Visual correlates of fixation selection: effects of scale and time. *Vision Research 45*, 643–659.

Tatler, B. W., Baddeley, R. J., and Vincent, B. T. 2006. The long and the short of it: Spatial statistics at fixation vary with saccade amplitude and task. *Vision Research 46*, 1857–1862.

Theeuwes, J. 2004. Top-down search strategies cannot override attentional capture. *Psychonomic Bulletin & Review 11*, 65–70.

Torralba, A. 2003a. Contextual priming for object detection. *International Journal of Computer Vision 53,* 2, 169–191.

Torralba, A. 2003b. Modeling global scene factors in attention. *Journal of Optical Society of America A. Special Issue on Bayesian and Statistical Approaches to Vision 20,* 7, 1407–1418.

Treisman, A. M. 1993. The perception of features and objects. In *Attention: Selection, awareness, and control*, A. Baddeley and L. Weiskrantz, Eds. Clarendon Press, Oxford, 5–35.

Treisman, A. M. and Gelade, G. 1980. A feature integration theory of attention. *Cognitive Psychology 12*, 97–136.

Treisman, A. M. and Gormican, S. 1988. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review 95,* 1, 15–48.

Tsotsos, J., Rodriguez-Sanchez, A., Rothenstein, A., and Simine, E. 2008. Different binding strategies for the different stages of visual recognition. *Brain Research 1225*, 119–132.

Tsotsos, J. K. 1987. A 'complexity level' analysis of vision. In *Proc. of International Conference on Computer Vision: Human and Machine Vision Workshop*. London, England.

Tsotsos, J. K. 1990. Analyzing vision at the complexity level. *Behavioral and Brain Sciences 13,* 3, 423–445.

Tsotsos, J. K. 1993. An inhibitory beam for attentional selection. In *Spatial Vision in Humans and Robots*, L. R. Harris and M. Jenkin, Eds. Cambridge University Press, 313–331.

Tsotsos, J. K., Culhane, S. M., Wai, W. Y. K., Lai, Y., Davis, N., and Nuflo, F. 1995. Modeling visual attention via selective tuning. *Artificial Intelligence 78,* 1-2, 507–545.

Tsotsos, J. K., Liu, Y., Martinez-Trujillo, J. C., Pomplun, M., Simine, E., and Zhou, K. 2005. Attenting to visual motion. *Journal of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance 100,* 1-2, 3–40.

Tsotsos, J. K., Verghese, G., Stevenson, S., Black, M., Metaxas, D., Culhane, S., Dickinson, S., Jenkin, M., Jepson, A., Milios, E., Nuflo, F., Ye, Y., and Mann, R. 1998. PLAYBOT: A visually-guided robot to assist physically disabled children in play. *Image and Vision Computing 16, Special Issue on Vision for the Disabled*, 275–292.

Tuytelaars, T. and Mikolajczyk, K. 2007. Local invariant feature detectors: A survey. *Foundations and Trens in Computer Graphics and Vision 3,* 3, 177–280.

van Oeffelen, M. P. and Vos, P. G. 1982. Configurational effects on the enumeration of dots: counting by groups. *Memory & Cognition 10*, 396–404.

Vecera, S. and Farah, M. 1994. Does visual attention select objects or locations? *Journal of experimental psychology. General 123,* 2, 146–160.

Verghese, P. 2001. Visual search and attention: a signal detection theory approach. *Neuron 31*, 523–535.

Vickery, T. J., King, L.-W., and Jiang, Y. 2005. Setting up the target template in visual search. *Journal of Vision 5,* 1, 81–92. doi:10.1167/5.1.8.

Vijayakumar, S., Conradt, J., Shibata, T., and Schaal, S. 2001. Overt visual attention for a humanoid robot. In *Proc. International Conference on Intelligence in Robotics and Autonomous Systems (IROS 2001)*. Hawaii, 2332–2337.

Vincent, B. T., Troscianko, T., and Gilchrist, I. D. 2007. Investigating a space-variant weighted salience account of visual selection. *Vision Research 47*, 1809–1820.

Viola, P. and Jones, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision (IJCV) 57,* 2 (May), 137–154.

von Helmholtz, H. 1896. *Handbuch der physiologischen Optik*. Von Leopold Voss Verlag, Hamburg, Germany. (an English Quote is included in Nakayama & Mackeben, 1989).

Walther, D. 2006. Interactions of visual attention and object recognition: computational modeling, algorithms, and psychophysics. Ph.D. thesis, California Institute of Technology, Pasadena, CA.

Walther, D., Edgington, D. R., and Koch, C. 2004. Detection and tracking of objects in underwater video. In *Proc. of Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Walther, D. and Koch, C. 2007. Attention in hierarchical models of object recognition. *Computational Neuroscience: Theoretical insights into brain funciton, Progress in Brain research 165*, 57–78.

Wells, A. and Matthews, G. 1994. *Attention and Emotion: A Clinical Perspective*. Psychology Press.

Wolfe, J. M. 1994. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review 1,* 2, 202–238.

Wolfe, J. M. 1998a. Visual search. In *Attention*, H. Pashler, Ed. Hove, U.K.: Psychology Press, 13–74.

Wolfe, J. M. 1998b. What can 1,000,000 trials tell us about visual search? *Psychological Science 9,* 1, 33–39.

Wolfe, J. M. 2001a. Asymmetries in visual search: An introduction. *Perception & Psychophysics 63,* 3, 381–389.

Wolfe, J. M. 2001b. Guided search 4.0: A guided search model that does not require memory for rejected distractors. *Journal of Vision, Abstracts of the 2001 VSS Meeting 1,* 3, 349a.

Wolfe, J. M. 2007. Guided search 4.0: Current progress with a model of visual search. In *Integrated models of cognitive systems*, W. D. Gray, Ed. Oxford University Press, New York, NY, Chapter 8.

Wolfe, J. M., Cave, K., and Franzel, S. 1989. Guided search: An alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human Perception and Performance 15*, 419–433.

Wolfe, J. M. and Gancarz, G. 1996. Guided search 3.0: Basic and clinical applications of vision science. Dordrecht, Netherlands: Kluwer Academic. 189–192.

Wolfe, J. M., Horowitz, T., Kenner, N., Hyle, M., and Vasan, N. 2004. How fast can you change your mind? The speed of top-down guidance in visual search. *Vision Research 44*, 1411–1426.

Wolfe, J. M. and Horowitz, T. S. 2004. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience 5*, 1–7.

Xu, T., Chenkov, N., Kühnlenz, K., and Buss, M. 2009. Autonomous switching of top-down and bottom-up attention selection for vision guided mobile robots. In *Proc. of Int'l Conf. on Intelligent Robots and Systems (IROS)*.

Xu, T., Pototschnig, T., Kühnlenz, K., and Buss, M. 2009. A high-speed multi-GPU implementation of bottom-up attention using CUDA. In *Proc. of the International Conference on Robotics and Automation, (ICRA)*.

Yantis, S. 2000. Goal-directed and stimulus-driven determinants of attentional control. In *Attention and Performance*, S. Monsell and J. Driver, Eds. Vol. 18. MIT Press, Cambridge, MA.

Yantis, S., ach, J. S., Serences, J., Carlson, R., Steinmetz, M., Pekar, J., and Courtney, S. 2002. Transient neural activity in human parietal cortex during spatial attention shifts. *Nature Neuroscience 5*, 995–1002.

Yantis, S. and Serences, J. T. 2003. Cortical mechanisms of space-based and object-based attentional control. *Current Opinion in Neurobiology 13*, 187–193.

Yarbus, A. L. 1967. *Eye Movements and Vision*. Plenum Press (New York).

Zeki, S. 1993. *A Vision of the Brain*. Blackwell Scientific., Cambridge, MA.

Zelinsky, G. J. and Sheinberg, D. L. 1997. Eye movements during parallel-serial visual search. *J. of Experimental Psychology: Human Perception and Performance 23,* 1, 244–262.

Received February 2007; revised January and July 2008; accepted: November 2008

Publication [8]

Simone Frintrop. General object tracking with a component-based target descriptor. In *Proceedings of the IEEE International Conference Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010.

General Object Tracking with a Component-based Target Descriptor

Simone Frintrop

Abstract— **In this paper, we present a component-based visual object tracker for mobile platforms. The core of the technique is a component-based descriptor that captures the structure and appearance of a target in a flexible way. This descriptor can be learned quickly from a single training image and is easily adaptable to different objects. The descriptor is integrated into the observation model of a visual tracker based on the well known Condensation algorithm. We show that the approach is applicable to a large variety of objects and in different environments with cluttered backgrounds and a moving camera. The method is robust to illumination and viewpoint changes and applicable to indoor as well as outdoor scenes.**

I. INTRODUCTION

Object tracking is an important task in machine vision as well as in mobile robotics. Applications include surveillance systems, mobile robots that guide or follow people, or human-robot interaction in which a robot interacts with a human and both have to concentrate on the same objects.

Many good approaches for object tracking have been proposed during the last years (see survey in [1]). However, the methods that are applicable for a certain task vary largely depending on requirements and setting. If the type of object is known in advance, model-based trackers may be applied. In these approaches, a model of the object is learned offline, usually from a large set of training images which show the object from different viewpoints and in different poses [2], [3]. These methods are especially well-suited for specialized tracking tasks such as person or face tracking. In some applications however, the object of interest is not known in advance, e.g., if a user shows an object to the system which shall be able to immediately capture the appearance of the object and track it. A long training phase is inacceptable in such cases, online learning methods are required.

In systems with a static camera, it is possible to apply methods like background subtraction [4]. For statistical investigations that do not require immediate response, like e.g. counting people, it is possible to process the data offline which extends the range of applicable algorithms considerably.

On the other hand, systems which shall be applied on a mobile platform usually have to operate in real-time and have to deal with more difficult settings. The background changes, illumination conditions vary, and platforms are often equipped with low-resolution cameras. Such conditions require robust and flexible tracking mechanisms. Mostly, feature-based tracking approaches are applied in such areas. They track an object based on simple features such as color cues or corners. An example is the Mean Shift algorithm [5] which classifies objects according to a color distribution or the CamShift algorithm which is based on the Mean Shift approach [6]. Other groups integrate color histograms into a particle tracker [7], [8] or combine a color model with a template tracker [9]. In previous work, we have used a cognitive observation model for visual tracking that was based on features inspired by human visual perception [10], [11]. Several ideas from this work have been integrated into the current approach. Over the last years, techniques which use interest points, like colored Harris corners [12] or SIFT features [13] for object tracking have been introduced. Note that these approaches usually rely on textured objects and a certain image resolution and quality to work well.

For feature-based tracking, it is especially important to detect discriminative features that distinguish the target well from the background. However, the discriminability of different parts of the object may differ strongly depending on the appearance of object and background. If a person wears a shirt in a color similar to the background, it has a low discriminability while the trousers on the other hand might have a high discriminability. To consider the different discriminability of parts, Beuter et al. train a top-down attention model to learn the face and the torso of a person separately [14]. Pérez et al [7], [8] determine different color histograms for different, rigidly linked parts of the target. Similarly, Adam et al. represent the target by a rigid layout of vertical and horizonal patches [15]. All of these approaches define a rigid layout of the parts in advance. In contrast to this, we suggest to automatically detect the different parts of a target in a flexible and object-dependent way.

In this paper, we present a component-based approach to visual tracking that is able to automatically detect the most discriminative parts of a target and to quickly learn its appearance from a single frame. Depending on the appearance of the object, the system determines a flexible number of components, each representing a discriminative part with respect to a certain feature channel. The resulting components form a target template that is used in the following frames to detect the most likely target position. A similarity measure determines the similarity between the target template and image regions in the following frames. Instead of computing the similarity for each pixel, we employ the component-based approach within a CONDENSATION-based person tracker [16]. For this purpose, the similarity measure is converted to a likelihood function that is used as observation model within the particle filter.

This approach leads to a robust and flexible tracker that is quickly applicable to track arbitrary objects in unknown

The author is with the Institute of Computer Science III, University of Bonn, 53117 Bonn, Germany frintrop@iai.uni-bonn.de

environments. Currently, the system works on camera data from a hand-held camera. Thus, it provides all conditions which are necessary to use it on a mobile robot: it is real-time capable and it is able to deal with background changes, viewpoint changes and varying illuminations.

We evaluated the approach in different settings and compared it to other color-based tracking methods. We tested the ability of the methods to deal with illumination changes, scale changes, occlusions, motion blur, background changes and more. It shows that on average the performance of the component-based tracking outperforms the other approaches considerably.

In the following, we first introduce the component-based descriptor (Sec. II). In Section III, we explain the visual tracking system and Section IV presents experimental results. We finally conclude in Section V.

II. A Multi-Component Target Descriptor

In this section, we introduce the multi-component descriptor that represents a target object. The descriptor consists of a collection of components that have a strong contrast within a certain feature dimension. These regions are automatically and object-dependently extracted from the target region. The components are color-based and the computation is motivated from the cognitive perception model VOCUS [17] that mimics human early visual processing.

Determining the multi-component descriptor consists of two steps. First, six intensity and color feature maps are computed (sec. II-A), second, components are automatically determined within the feature maps and combined to form the descriptor (sec. II-B). Finally, we describe how the target descriptor is matched to a region in a different frame to test if the target is present or not (sec. II-C).

A. Feature map computations

In this section, we describe how six intensity and color maps are computed as a basis for the component-based descriptor. An overview is displayed in Fig. 1.

First, the input image is converted to an image in the CIELAB color space (also $L^*a^*b^*$), smoothed with a Gaussian filter and subsampled twice to reduce the influence of noise. The resulting image is called I_{lab}. CIELAB has the dimension L for lightness and a and b for the color-opponent dimensions; it is perceptually uniform, i.e., a change of a certain amount in a color value is perceived as a change of about the same amount in human visual perception. Each of the 6 ends of the axes that confine the color space serve as a prototype color, resulting in two intensity prototypes for white and black and four color prototypes for red, green, blue, and yellow (cf. Fig. 1, top right).

Then, the computation of feature maps is started. We treat intensity and color computations separately since this results in a higher illumination invariance. The intensity computations can be performed directly from the L channel I_l. The color computations are performed on the color layer I_{ab} spanned by a and b. Now, we determine four color

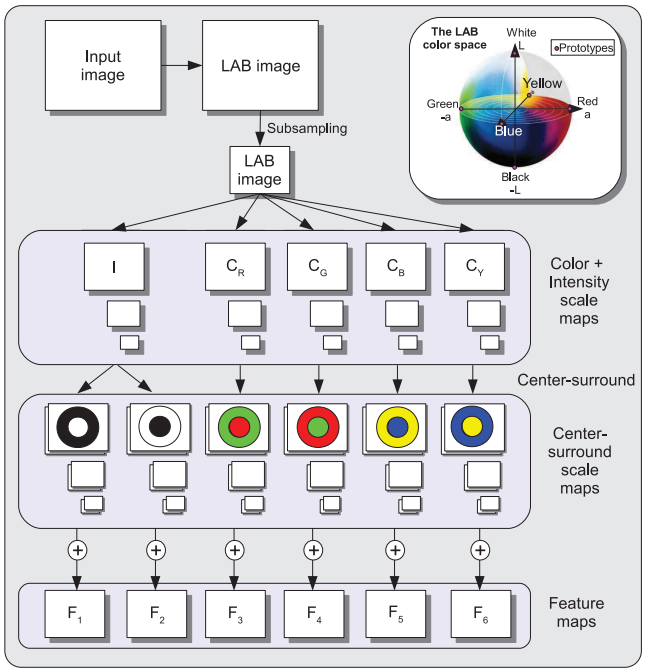


Fig. 1. The feature computations: from an input image, 6 feature maps are computed, showing bright-dark, dark-bright, red-green, green-red, blue-yellow, and yellow-blue contrasts.

specific maps C_i that represent the four colors red, green, blue and yellow.

For each of the color maps C_i, there is one prototype color P_i (cf. Fig. 1, top right) and each pixel $C_i(x, y)$ in a color map stores the Euclidean distance to the corresponding prototype color P_i:

$$C_i(x, y) = V_{max} - ||I_{ab}(x, y) - P_i|| \qquad i \in \{1, ..., 4\}, \quad (1)$$

where $V_{max} = 255$ is the maximal pixel value and the prototypes P_i are the ends of the a and b axes with coordinates $(0, 127), (127, 0), (255, 127), (127, 255)$ in an 8-bit I_{ab}.

Next, image pyramids with 3 levels are determined from I_l and C_i. This enables flexibility to scale changes. On each of these scale maps in the pyramids we perform *center-surround mechanisms*. These are filters that detect image contrasts between a center c and a surround region s, similar to ganglion cells in the human brain. Applied to our scale maps, the filters detect intensity and color contrasts. On the color maps, the filters react especially strong to red-green, green-red, blue-yellow, and yellow-blue contrasts. We use surround regions of two different sizes, resulting in six center-surround maps $S_{i,j}, j \in \{1, ..., 6\}$ for each color/intensity (details in [17]). Note that center surround applied to the intensity scale maps detects only bright-dark contrast. To additionally determine dark-bright contrasts, we compute the opposite difference $s - c$. To speed up processing, all center-surround filters are computed with integral images [18].

Finally, we sum up the 36 center-surround scale maps to obtain 6 feature maps $F_i = \sum_{j=1}^{6} S_{i,j}$. The feature maps for some example images are displayed in Fig. 2.

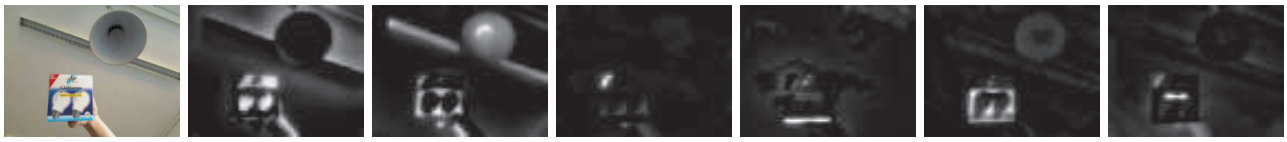


Fig. 2. An example image and the corresponding feature maps F_i.

B. Determining a target template and descriptor

Now, we determine a component-based template from the feature maps and derive a descriptor from the template. A component is a peak in one of the feature maps within the target region $\vec{R^*} = (x^*, y^*, w^*, h^*)$, where x^*, y^* denote the position and w^*, h^* the width and height of the region. The peaks are detected by first detecting local intensity maxima and then segmenting the region around the maxima with region growing. For easier computations, the regions are approximated by rectangular bounding boxes that we call $m_{i,j} = (x_{m_{i,j}}, y_{m_{i,j}}, w_{m_{i,j}}, h_{m_{i,j}})$, where i denotes the feature map and j the different maxima in a map. Hereby, the number of components per map is flexible and depends on the appearance of the object. Additionally, we add the whole target region as one of the $m_{i,j}$ to make the descriptor more robust.

The positions of the regions $m_{i,j}$ are stored relative to the center of $\vec{R^*}$ and represent a template $\vec{M_{R^*}} = \{m_{i,j} | i \in \{1, .., 6\}, j \in \{1, .., l_i\}\}$, where l_i is the number of components detected in feature map F_i (cf. Fig. 3, left). Now, we derive a descriptor vector from the $m_{i,j}$. For each $m_{i,j}$, we compute the ratio of the mean intensity value within $m_{i,j}$ and the mean value of the background:

$$\rho_{i,j} = \frac{mean(m_{i,j})}{mean(F_i \backslash m_{i,j})} \tag{2}$$

The mean is computed with integral images, to speed up processing and enable constant computation times for each region, independent of the size of the region. Thus, the target descriptor that we obtain is $\vec{d^*} = \{\rho_{i,j} | i \in \{1, .., 6\}, j \in \{1, .., l_i\}\}$.

C. Match descriptor to image region

In order to match the target descriptor $\vec{d^*}$ to an image region $\vec{R'}$ of arbitrary size and dimension, we first determine the factors f_w and f_h that represent the difference in size between the target region $\vec{R^*}$ and $\vec{R'}$: $f_w = R'_w / R^*_w, \quad f_h = R'_h / R^*_h$, where R'_w, R^*_w denote the width and R'_h, R^*_h the height of the regions. Now, an adapted template $\vec{M_{R'}}$ is computed by extending or compressing all $m_{i,j} \in \vec{M_{R^*}}$ with f_w and f_h: $w_{m'_{i,j}} = f_w * w_{m^*_{i,j}}, \quad h_{m'_{i,j}} = f_w * h_{m^*_{i,j}}$ (cf. Fig. 3, right). $\vec{M_{R'}}$ is now used to compute a descriptor $\vec{d'}$ equivalently as in eq. 2.

Finally, the descriptors $\vec{d^*}$ and $\vec{d'}$ are matched by computing the similarity of the vectors. As similarity measure, we use the Tanimoto coefficient:

$$T(\vec{d^*}, \vec{d'}) = \frac{\vec{d^*} \cdot \vec{d'}}{||\vec{d^*}||^2 + ||\vec{d'}||^2 - \vec{d^*} \cdot \vec{d'}}. \tag{3}$$

Fig. 3. Left: An illustration of the template $\vec{M_{R^*}}$ for the target region $\vec{R^*}$. The three colored rectangles denote the $m_{i,j}$. Note that each of them comes from a different feature map which is illustrated here by the different colors. Right: the template $\vec{M_{R'}}$ adapted to region $\vec{R'}$.

The Tanimoto coefficient produces values in the interval $[0, 1]$, the higher the value the higher the similarity. If the two vectors are identical, the coefficient is 1.

III. The Visual Tracking System

The tracking system uses the component-based descriptor from the previous section as observation model of a particle filter approach. It employs the standard Condensation algorithm [16] which maintains a set of weighted particles over time using a recursive procedure based on the following three steps: First, the system draws particles randomly from the particle set of the previous time step, where each particle is drawn with a probability proportional to the associated weight of the particle. Second, the particles are transformed (predicted) according to a motion model. Finally, all particles are assigned new weights according to an observation model and the object state is estimated.

Let us first introduce the notation. At each point in time $t \in \{1, .., T\}$, the particle filter recursively computes an estimate of the probability density of the object's location within the image using a set of J particles $\vec{\Phi}_t = \{\vec{\phi}^1_t, ...\vec{\phi}^J_t\}$ with

$$\vec{\phi}^j_t = (\vec{s}^j_t, \pi^j_t, \vec{d}^j_t), \quad j \in \{1, ..., J\}. \tag{4}$$

(here: $J = 500$). $\vec{s}^j_t = (x, y, v_x, v_y, w, h)$ is the state vector that specifies the particle's region with center (x, y), width w and height h – in the following, the region is also denoted as $\vec{R}^j_t = (x, y, w, h)$. The v_x and v_y components specify the current velocity of the particle in the x and y directions. Each particle additionally has a weight π^j_t determining the relevance of the particle with respect to the target, and the component-based descriptor $\vec{d}^j_t$ that describes the appearance of the particle region.

In the following, we first mention how the system is initialized (sec. III-A), second describe the motion model

(sec. III-B), and finally, specify the observation model as core of the system (sec. III-C).

A. Initialization

Before starting the tracking, the initial target region $\vec{R}^*$ has to be specified in the first frame. This can either be carried out manually or automatically using a separate detection module. We initialize manually here. Based on the initial target region $\vec{R}^*$, the component-based descriptor $\vec{d}^*$ is computed that describes the appearance of the object. The initial particle set

$$\vec{\Phi}_0 = \{(\vec{s}_0^j, \pi_0^j, \vec{d}_0^j) \,|\, j = 1, ..., J\} \tag{5}$$

is generated by randomly distributing the initial target location around the region's center (x^*, y^*). The velocity components v_x and v_y are initially set to 0 and the region dimensions of each particle are initialized with the dimensions of $\vec{R}^*$. The particle weights π_0^j are set to $1/J$.

B. Motion model

The object's motion is modeled by a simple first order autoregressive process in which the state of a particle depends only on the state of the particle in the previous frame:

$$\vec{s}_t^j = \mathbf{M} \cdot \vec{s}_{t-1}^j + \vec{Q}. \tag{6}$$

Here, $\mathbf{M}$ is a state transition matrix of a constant velocity model and $\vec{Q}$ is a random variable that denotes some white Gaussian noise. This enables a flexible adaption of position and size of the particle region as well as of its velocity. Thus the system is able to quickly react to velocity changes of the object.

C. Observation model

In visual tracking, the choice of the observation model is the most crucial step since it decides which particles will survive. It therefore has the strongest influence on the estimated position of the target. Here, we use the component-based descriptor to determine the feature description for the target and for each particle, enabling the comparison and weighting of particles.

First, we compute a descriptor $\vec{d}_t^j$ for each of the particles according to sec. II-B. That means, the target template $\vec{M}_{R^*}$ is adapted to the size of the current particle and the descriptor $\vec{d}_t^j$ is computed for the resulting template M_t^j. Then, the weight of a particle is computed based on the Tanimoto coefficient as

$$\pi_t^j = c \cdot e^{\lambda \cdot T(\vec{d}^*, \vec{d}_t^j)}. \tag{7}$$

This function prioritizes particles which are very similar to $\vec{d}^*$ by assigning an especially high weight. A value of $\lambda = 14$ has shown to be useful in our experiments. The parameter c is a normalization factor which is chosen so that $\sum_{j=1}^{J} \pi_t^j = 1$.

Finally, the current target state, including target position and size, can be estimated as a weighted average of the particles by

$$\vec{x}_t = \sum_{j=1}^{J} \pi_t^j \cdot \vec{s}_t^j. \tag{8}$$

IV. Experiments and Results

In this section, we compare three different approaches for visual object tracking. All methods use the same particle filter approach for tracking and a color-based observation model. The first approach is a standard method based on color histograms and was implemented according to [7][1]. The second approach that we call ROI tracking is a simplified version of the here presented method. It uses the same feature maps as in sec. II-A but no components. Instead, it considers the whole target region and computes a descriptor based on the ratio of the mean of the target region and the mean of the background as in eq. 2. Thus, it computes a 6-dimensional target descriptor.[2] The third approach is the here presented component-based tracking.

We test the three methods in seven different settings to illustrate different properties. In each setting, we tracked one object over a sequence of images (320×240 pixels, length of sequences: 125 – 388 frames). Examples of the settings together with the component-based descriptors are displayed in Fig. 4. The complete tracking results can be watched in a video on http://ivs.informatik.uni-bonn.de/research/tracking/.

For each estimated tracking trajectory, we computed the Euclidean distance to the real position of the target that was determined manually. Reference for the computation was the center of the object resp. the center of the estimated target position. These distances are displayed in Fig. 5. Since the distance of the estimation from the real position is not always meaningful (depending on the size of the object, the same distance might be still acceptably good or quite bad), we additionally determined whether the center of the estimation was on the target or not. This detection rate is displayed in Tab. I. The computation time varied between 69 and 90 ms per frame (av. 80 ms), depending on the complexity of the target template (on a 2.5 GHz dual core PC). This frame rate was sufficient for online tracking but a higher rate could be easily achieved by code optimization.

In the following, we describe the different settings.

A. Illumination Changes

In this example, we test the ability of the systems to deal with illumination changes. We tested a static scene in which only the illumination is varied by opening and closing the sun-blinds. It shows that the new component-based tracking is hardly effected by these changes, while both other approaches have problems. Note that the detection rate of histograms is better than the one of the ROI tracking

[1]The color histograms were implemented exactly as described in [7] (HSV color model, bin numbers $N_h = N_s = N_v = 10$), the particle filter was the same as for the other approaches (cf. sec. III) to concentrate the comparison on the observation model.

[2]We used almost the same method in [11], but we omitted the orientation features to make the approach comparable to the other methods which are purely color-based.

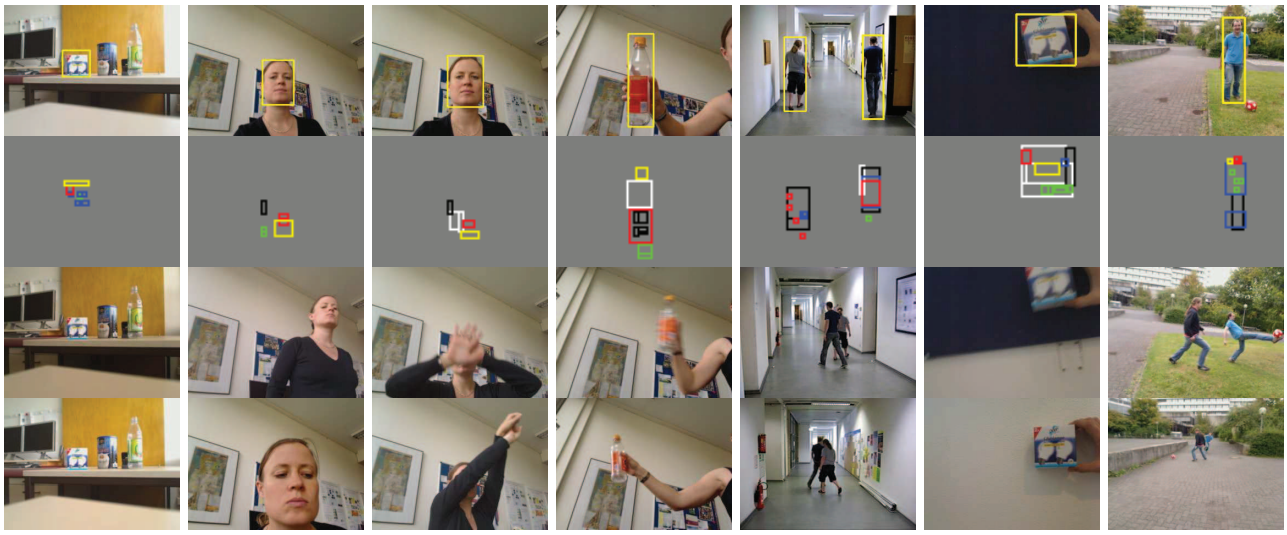


Fig. 4. The test sequences A - G. First row: the target region used for initialization (yellow rectangles). Second row: the component-based descriptors computed for the target region. Colors denote the feature map the component comes from (white: bright-dark map, black: dark-bright map, red: red-green map, ...). 3rd and 4th row: other example frames from the sequences. See also video on http://ivs.informatik.uni-bonn.de/research/tracking/

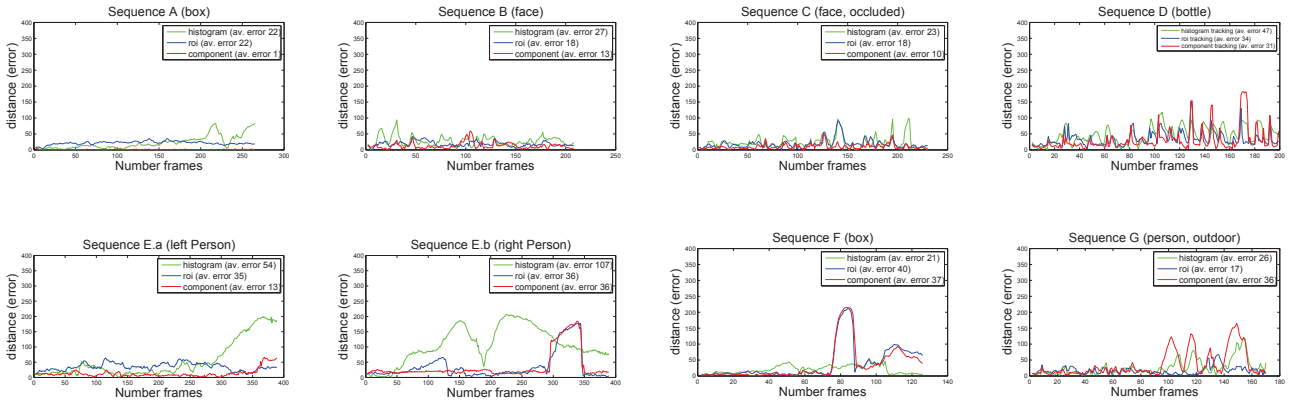


Fig. 5. Comparison of the trajectories of the three tracking methods with ground truth. The y axis shows the Euclidean distance of the center of the estimated region to the center of the real position of the target. average errors: histogram = 41, roi = 28, new component-based = 22.

while the average distance of the trajectories (cf. Fig. 5) is the same.

B. Object Motion and Scale Changes

Next, we test an object that is moving and changes strongly in scale. We use face tracking as example application. Again, the component-based method outperforms the other approaches clearly.

C. Temporal Object Occlusion

In this example, we test how the approaches deal with temporal occlusions of the object. The target is a face that is temporary occluded by hands and arms of the person. This is especially challenging since hands and face both have skin color. The fact that the results of all methods are better for this sequence than for seq. B shows that obviously the scale changes affect the methods stronger than a brief occlusion.

D. Quick Object Motion

Here, we test the ability of the methods to deal with extremely quick object motion. The object changes its direction abruptly and the motion is so quick that the object moves many pixels between consecutive frames: Rows 3 and 4 in column 4 of Fig. 4 show consecutive frames; the object position varies almost 1/3 of the image width. All methods show that the particle filter tracking needs some frames to follow the object if the motion is very fast. Thus, the target is briefly lost until the method adapts and redetects the target again. This results in relatively low detection rates of all methods (cf. Tab. I). From Fig. 5 it can be seen that the error grows quickly for each quick motion but is reduced briefly after when the target is redetected (see also video on http://ivs.informatik.uni-bonn.de/research/tracking/).

Seq.	Object	# Frames	detection rate [%]		
			Hist.	ROI	Comp. (our)
A.	Box	264	61	42	100
B.	Face	207	55	78	89
C.	Face	229	76	82	100
D.	Bottle	198	33	45	57
E.a	left Person	388	45	78	89
E.b	right Person	388	15	56	84
F.	Box	125	92	70	74
G.	Person	169	68	70	54
av.			56	65	81

TABLE I

COMPARISON OF THE THREE TRACKING METHODS BASED ON COLOR HISTOGRAMS (HIST.), SIMPLE REGION OF INTEREST TRACKING (ROI) AND THE HERE PRESENTED COMPONENT-BASED TRACKING (COMP.).

E. Moving Camera

While the previous examples have been tested with a static camera, the following three examples are recorded with a moving camera. This is considerably more challenging since it envolves illumination changes, motion blur, and background changes. The first example shows two people walking down a corridor, while the camera is following them. The persons cross their way twice. This is a typical setting for a service robot that shall follow a person and not confuse it with other people. We tested the tracking of each of the persons individually. In both cases, the component-based tracking clearly outperforms the other methods. Most difficulties has the histogram-based approach, especially when tracking the right person.

F. Moving Camera with Strongly Changing Background

The next example shows an extreme case of background change: the background changes from dark blue to white. Since the target object has similar colors (also mainly blue and white), the two tracking approaches that include the background (ROI and component-based) have some difficulties here. While including the background is usually helpful, it makes some problems in such an extreme case. We are currently working on ways to adapt the descriptor automatically to new environments.

G. Outdoor

Finally, we show an outdoor sequence that combines most of the previous challenges: the camera is moving, several objects (two people and a ball) are moving very quickly, the appearance of the target person changes strongly in scale, the shape of the person changes, e.g. when shooting the ball (cf. Fig. 4, 3rd row, right), and the illumination as well as the background change. Here, the purely color-based approaches, histogram and ROI, outperform the component-based tracking. The strong changes in shape are problematic in the latter case. We are currently working on ways to track the components of the target individually. This could help to cope with such difficulties. However, it can be seen from Fig. 5, that the approach is always able to redetect the target after some frames.

In average, the new component-based tracking has outperformed the other two methods considerably with an average error 22 and a detection rate of 81 %, compared to error 28 and detection rate 65 % for the ROI tracking and error 41 and detection rate 56 % for the histogram tracking.

V. CONCLUSION

We have presented a new approach for object tracking based on a component-based descriptor. The method grabs the appearance of an object in color and intensity together with a rough spatial layout which are quickly learned from a single training image. It can deal with different objects and settings, works in real-time, and is applicable on a moving platform. We have shown that it on average clearly outperforms other methods.

However, there is still room for improvements. We are currently working on ways to store the position of the components individually to achieve more flexibility to deformations and rotations of the objects. Additionally, we intend to adapt the target descriptor online if background and/or target appearance change strongly, as e.g. in [19].

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[2] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int'l J. of Computer Vision, Special Issue on Learning for Recognition and Recognition for Learning*, vol. 77, no. 1-3, pp. 259–289, 2008.

[3] K. Rohr, "Towards model-based recognition of human movements in image sequences," *CVGIP – Image Understanding*, vol. 59, no. 1, pp. 94–115, 1994.

[4] C. Wren, A. Azarbayejani, and A. Pentland, "Pfinder: Real-time tracking of the human body," *Trans. on PAMI*, vol. 19, no. 7, 1997.

[5] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," Proc. of CVPR, 2000.

[6] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," Intel Technology Journal, 1998.

[7] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," Proc. of ECCV, 2002.

[8] P. Pérez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proc. of the IEEE*, vol. 92, no. 3, 2004.

[9] V. Badrinarayanan, P. Pérez, F. L. Clerc, and L. Oisel, "Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues," in *Proc. of ICCV*, 2007.

[10] S. Frintrop and M. Kessel, "Most salient region tracking," in *Proc. of ICRA*, 2009.

[11] S. Frintrop, A. Königs, F. Hoeller, and D. Schulz, "Visual person tracking using a cognitive observation model," in *ICRA Workshop on People Detection and Tracking*, 2009.

[12] T. Mathes and J. H. Piater, "Robust non-rigid object tracking using point distribution manifolds," in *Proc. of DAGM*, 2006.

[13] F. Tang and H. Tao, "Object tracking with dynamic feature graph," in *Proc. of the IEEE Workshop on VS-PETS*, 2005.

[14] N. Beuter, O. Lohmann, J. Schmidt, and F. Kummert, "Directed attention - a cognitive vision system for a mobile robot," in *Proc. of ROMAN*, 2009.

[15] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. CVPR*, 2006.

[16] M. Isard and A. Blake, "Condensation - conditional desity propagation for visual tracking," *IJCV*, vol. 29, no. 1, pp. 5–28, 1998.

[17] S. Frintrop, "VOCUS: a visual attention system for object detection and goal-directed search," Ph.D. dissertation, 2005, in LNAI, Vol. 3899, Springer, 2006.

[18] S. Frintrop, M. Klodt, and E. Rome, "A real-time visual attention system using integral images," in *Proc. ICVS*, 2007.

[19] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis, "Sequential kernel density approximation and its application to real-time visual tracking," *IEEE Trans. on PAMI*, vol. 30, no. 7, pp. 1186–1197, 2008.

Publication [9]

Simone Frintrop, Achim Königs, Frank Hoeller, and Dirk Schulz. A component-based approach to visual person tracking from a mobile platform. *International Journal of Social Robotics, Springer*, 2(1), 2010.

Springer International Journal of Social Robotics manuscript No.
(will be inserted by the editor)

A Component-based Approach to Visual Person Tracking from a Mobile Platform

Simone Frintrop[1] · **Achim Königs**[2] · **Frank Hoeller**[2] · **Dirk Schulz**[2]

Received: date / Accepted: date

Abstract In this article, we present a component-based visual tracker for mobile platforms with an application to person tracking. The core of the technique is a component-based descriptor that captures the structure and appearance of a target in a flexible way. This descriptor can be learned quickly from a single training image and is easily adaptable to different objects. It is especially well suited to represent humans since they usually do not have a uniform appearance but, due to clothing, consist of different parts with different appearance. We show how this component-based descriptor can be integrated into a visual tracker based on the well known Condensation algorithm. Several person tracking experiments carried out with a mobile robot in different laboratory environments show that the system is able to follow people autonomously and to distinguish individuals. We furthermore illustrate the advantage of our approach compared to other tracking methods.

Keywords Visual Tracking · Component-based Tracking · Person Tracking

1 Introduction

An important skill for mobile service robots is the ability to detect and keep track of individual humans in their surrounding. Especially robots that are designed to provide services to individual persons need to be able to distinguish their client from the surrounding environment. Usually, such systems shall be able to learn the appearance of a target person quickly, possibly from a single snapshot. Additionally, to run on a mobile platform the approaches have to be real-time capable and robust to illumination changes, motion blur and quick viewpoint changes.

S. Frintrop
Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität,
53117 Bonn, Germany.
E-mail: frintrop@iai.uni-bonn.de

A. Königs, F. Hoeller, D. Schulz
Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE),
53343 Wachtberg, Germany.
E-mail: {koenigs,hoeller,schulz}@fgan.de

While many approaches have been proposed to track humans, most of them are not designed to distinguish individuals. This is especially true for laser-based systems that usually track the legs of people, or for model-based vision approaches that consider the shape of objects. Well suited to distinguish people are feature-based vision approaches. For these methods, it is especially important to detect discriminative features that distinguish the target well from the background. However different parts of complex objects, such as people, provide different discriminability from the background. If a person wears, for example, a shirt in a color similar to the background, it has a low discriminability while the trousers on the other hand might have a high discriminability. A good feature descriptor shall consider this variable discriminability and focus on the most discriminative parts. Since the structure of parts differs from target to target, it is preferable to automatically detect the different parts instead of using a rigid template.

In this paper, we present a component-based approach to visual tracking that is able to automatically detect the most discriminative parts of a target person and to quickly learn its appearance from a single frame. Depending on the appearance of the person (clothing, hair color, skin color etc.), the system determines a flexible number of components, each representing a discriminative part with respect to a certain feature channel. The resulting components form a target template that is used in the following frames to detect the most likely target position. A similarity measure determines the similarity between the target template and image regions in the following frames. Instead of computing the similarity for each pixel, we employ the component-based approach within a Condensation-based person tracker [20]. For this purpose, the similarity measure is converted to a likelihood function that is used as the observation model within the particle filter.

This approach leads to a robust and flexible tracker that is quickly applicable to track arbitrary people in unknown environments. It is able to work in real-time on a mobile platform. We evaluated the approach in different settings: first, we compared the approach to other color-based tracking approaches and show that the performance of the component-based tracking outperforms the other approaches considerably. Second, we tested the ability of the system to distinguish a target person from other people that cross their way in front of the robot. Finally, we showed that the robot is able to follow a person autonomously in different settings of our laboratory environment under varying lighting conditions and backgrounds.

The remainder of the article is organized as follows. After discussing related work in Section 2, we introduce the component-based descriptor in Section 3. In Section 4, we explain the visual tracking system. Section 5 briefly explains how the approach is integrated into a prototypical person following application and presents experimental results. We finally conclude in Section 6.

2 Related work

In mobile robotics, researchers have developed person tracking techniques for different sensors. A frequently used approach is to use laser range finders, as these sensors are available on many robots for collision avoidance purposes. Because laser sensors usually only provide distance information to objects in the environment, most laser-based approaches only keep track of the motion of people and do not try to distinguish between individuals [25,31]. However, several techniques have been developed that utilize the appearance of a person's legs in the data, to reduce the risk of track loss or

the confusion of tracks of different persons. For example, Arras et al. [2] use AdaBoost to train a detector for the legs of persons in laser range profiles and in more recent work [3] they suggest a two-leg constraint in combination with a specialized occlusion handling technique to increase the robustness against track loss. Taylor and Kleeman [34] use a switched dynamic model to even track the repetitive leg motion for this purpose.

Other authors improve the robustness of laser-based tracking by additionally taking camera information into account. Using this combination of sensors, the spatial tracking can still be performed on the laser data, while the camera immediately provides informative appearance information to distinguish between persons. For example, Bennewitz et al. [5] and Bellotto and Hu [4] use color histograms to discriminate between the persons being tracked. Schulz [30] uses a shape matching approach to distinguish between persons; a probabilistic exemplar approach is applied to track characteristic silhouettes of individuals over time. However, this requires a time-consuming learning process for the exemplar model of each new person.

In machine vision, people tracking is a well-studied problem. Two main approaches can be distinguished: *model-based* and *feature-based* methods. In model-based tracking, a model of the object is learned in advance, usually from a large set of training images which show the object from different viewpoints and in different poses [29]. Learning a model of a human is difficult because of the dimensionality of the human body and the variability in human motion. Current approaches include simplified human body models, e.g. stick, ellipsoidal, cylindric or skeleton models [8,37,24], or shape-from-silhouettes models [9]. While these approaches have reached good performance in laboratory settings with static cameras, they are usually not applicable in real-world environments on a mobile system. They usually do not operate in real-time and often rely on a static, uniform background. A model-based approach that works from moving cameras is shape matching. For example, Gavrila [17] suggests an exemplar-based technique that employs fast Chamfer matching to detect the shapes of pedestrians in images in real-time. The technique has been adopted for a particle filter tracking algorithm by Toyama and Blake [36]. However, it is not possible to adapt the rather large exemplar models on-line and, thus, the approach is not capable of distinguishing between persons during tracking. A modeling technique related to exemplars are implicit shape models [22] which, in comparison to pure exemplar approaches, improve the robustness against partial occlusions of objects. However, these models can also not be adapted online and are generally also not suitable to distinguish individual people. The final model-based technique, we want to mention, is tracking-by-detection, which has become increasingly popular over the last years. Typically, these approaches learn classifiers based on feature descriptors in order to detect and track humans in images [12, 1,38]. Due to carefully chosen object specific feature sets, very reliable detections are achieved that can directly be used as observations within a tracking algorithm. The combination of part detectors even allows for partial occlusions. However, the classifiers generally require an off-line learning phase on a rather large training set. Our descriptor, in contrast, does not allow to detect people, but is used to acquire a robust observation model for individual objects from a single image for tracking. On-line supervised learning techiques can be applied to train classifiers for a similar purpose [18, 32], but need a larger image sequence to acquire the models.

Feature-based tracking approaches on the other hand do not learn a model but track an object based on simple features such as color cues or corners. One approach for feature-based tracking is the Mean Shift algorithm [10,11] which characterizes objects by their color distribution. The algorithm tracks objects by carrying out a gradient

descent in the image that minimizes the dissimilarity between the local color statistics in the image and the object's color histogram. An extension of this method is the CamShift algorithm [7]. Other groups integrate color histograms into a particle tracker [27,28]. In previous work, we have used a cognitive observation model for visual tracking that was based on features inspired by human visual perception [14,16]. Several ideas from this work have been integrated into the current approach. Over the last years, techniques which use interest points, like colored Harris corners [23] or SIFT features [33] for object tracking have been introduced. Note that these approaches usually rely on textured objects and a certain image resolution and quality to work well. While these feature-based techniques are not especially designed for person tracking, they are commonly applied in this area.

Some people have also suggested to store different representations for different parts of the objects. For example, Pérez et al. determine different color histograms for different, rigidly linked parts of the target [27,28]. Beuter et al. train a top-down attention model to learn the face and the torso of a person separately [6]. We are however not aware of any work that determines the number and kinds of components of a target automatically to obtain a flexible descriptor as we will present in this work.

3 A Multi-Component Target Descriptor

In this section, we introduce the multi-component descriptor that represents a target. The computation consists of two steps. First, six intensity and color feature maps are computed (sec. 3.1), second, components are determined within the feature maps and combined to form the descriptor (sec. 3.2). Finally, we describe how the descriptor is matched to a region in a new frame to test if the target is present (sec. 3.3).

3.1 Feature map computations

In this section, we describe the computation of six intensity and color maps as a basis for the component-based descriptor. The computation of these feature maps is based on concepts from the human visual system in which color opponent cells determine the contrast of a center region and its surround [26]. The computation is the same as in the visual attention system VOCUS [13,15] and similar to Itti's attention system NVT [21].[1] An overview of the processing is displayed in Fig. 1.

First, the input image is converted to an image in the CIELAB color space (also $L^*a^*b^*$), smoothed with a Gaussian filter and subsampled twice to reduce the influence of image noise. We call the resulting image I_{Lab}. The CIELAB space has the dimension L for lightness and a and b for the color-opponent dimensions; it is perceptually uniform, which means that a change of a certain amount in a color value is perceived as a change of about the same amount in human visual perception. Furthermore, the space suits our purpose especially well since the four main colors red, green, blue and yellow are at the end of the axes a and b. This will show to be useful for our computations. Each of the 6 ends of the axes that confine the color space serves as one prototype color, resulting in two intensity prototypes for white and black and four color prototypes for red, green, blue, and yellow (cf. Fig. 1, left, top right corner).

[1] Differences to NVT include the use of a different color space and of integral images to speed up processing; more differences outlined in [13].

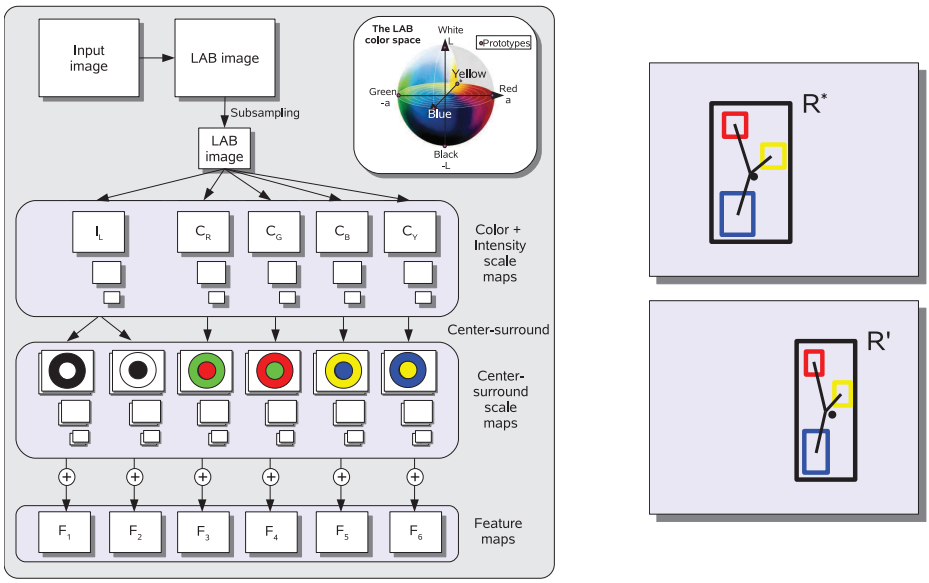


Fig. 1 Left: The feature computations: from an input image, 6 feature maps are computed, showing bright-dark, dark-bright, red-green, green-red, blue-yellow, and yellow-blue contrasts. Right, top: An illustration of the template $\mathbf{M_{R*}}$ for the target region $\mathbf{R^*}$. The three colored rectangles denote the $m_{i,j}$; the different colors illustrate the feature maps they result from. Right, bottom: the template $\mathbf{M_{R'}}$ adapted to region $\mathbf{R'}$.

Then, the computation of feature maps is started. We treat intensity and color computations separately since this results in a higher illumination invariance. The intensity computations can be performed directly from the L channel I_L. The color computations are performed on the color layer I_{ab} spanned by a and b. Now, we determine four color specific maps C_i that represent the four colors red, green, blue and yellow.

For each of the color maps C_i, there is one prototype color P_i (cf. Fig. 1, left, top right corner) and each pixel $C_i(x, y)$ in a color map stores the Euclidean distance to the corresponding prototype color P_i:

$$C_i(x, y) = V_{max} - ||I_{ab}(x, y) - P_i|| \qquad i \in \{1, ..., 4\}, \tag{1}$$

where $V_{max} = 255$ is the maximal pixel value and the prototypes P_i are the ends of the a and b axes with coordinates $(0, 127), (127, 0), (255, 127), (127, 255)$ in an 8-bit I_{ab}.

Next, image pyramids with 3 levels are determined from I_L and C_i. This enables flexibility to scale changes. On each of these scale maps in the pyramids we perform *center-surround mechanisms*. These are filters that detect image contrasts between a center c and a surround region s. Applied to our scale maps, the filters detect intensity and color contrasts. On the color maps, the filters react especially strong to red-green, green-red, blue-yellow, and yellow-blue contrasts. We use surround regions of two different sizes (radius 3 and 7 pixels, center 1 pixel), resulting in six center-surround maps $S_{i,j}, j \in \{1, ..., 6\}$ for each color/intensity (details in [13]). Note that center surround applied to the intensity scale maps detects only bright-dark contrast. To additionally determine dark-bright contrasts, we compute the opposite difference $s - c$. To speed up processing, all center-surround filters are computed with integral images [15].

Fig. 2 The initial frames used for the experiments in sec. 5.1 and corresponding feature maps.

Finally, we sum up the 36 center-surround scale maps to obtain 6 feature maps F_i: $F_i = \sum_{j=1}^{6} S_{i,j}$. The feature maps for some example images are displayed in Fig. 2.

3.2 Determining a target descriptor

The target descriptor consists of components that have a strong contrast within a certain feature dimension. It is derived from the feature maps. A component is a peak in one of the feature maps within the target region $\mathbf{R^*} = (x^*, y^*, w^*, h^*)$. The peaks are detected by first detecting local intensity maxima and then segmenting the region around the maxima with region growing. For easier computations, the regions are approximated by rectangular bounding boxes that we call $m_{i,j}$, where i denotes the feature map and j the different maxima in a map. Hereby, the number of components per map is flexible and depends on the appearance of the object. Additionally, we add the whole target region as one of the $m_{i,j}$ to make the descriptor more robust.

The positions of the regions $m_{i,j}$ are stored relative to the center of $\mathbf{R^*}$ and represent a template $\mathbf{M_{R^*}} = \{m_{i,j} | i \in \{1,..,6\}, j \in \{1,..,l_i\}\}$ (cf. Fig. 1, right top, and Fig. 5). Now, we compute a descriptor vector from the $m_{i,j}$. For each $m_{i,j}$, we compute the ratio of the mean intensity value within $m_{i,j}$ and the mean value of the background:

$$\rho_{i,j} = \frac{mean(m_{i,j})}{mean(F_i \backslash m_{i,j})} \tag{2}$$

The mean is computed with integral images, to speed up processing and enable constant computation times for each region, independent of the size of the region. Thus, the target descriptor that we obtain is $\mathbf{d^*} = \{\rho_{i,j} | i \in \{1,..,6\}, j \in \{1,..,l_i\}\}$.

3.3 Matching the descriptor to an image region

In order to match the target descriptor $\mathbf{d^*}$ to an image region $\mathbf{R'}$ of arbitrary size and dimensions, we first determine the factors f_w and f_h that represent the difference

in size between the target region $\mathbf{R^*}$ and $\mathbf{R'}$: $f_w = R'_w/R^*_w, \quad f_h = R'_h/R^*_h$, where R'_w, R^*_w denote the width and R'_h, R^*_h the height of the regions. Now, an adapted template $\mathbf{M_{R'}}$ is computed by extending or compressing all $m_{i,j} \in \mathbf{M_{R^*}}$ with f_w and f_h: $m'_w = f_w * m^*_w, \quad m'_h = f_h * m^*_h, \quad \forall m' \in \mathbf{M_{R'}}, m^* \in \mathbf{M_{R^*}}$ (cf. Fig. 1, right bottom). $\mathbf{M_{R'}}$ is now used to compute a descriptor $\mathbf{d'}$ equivalently as in eq. 2.

Finally, the descriptors $\mathbf{d^*}$ and $\mathbf{d'}$ are matched by computing the similarity of the vectors. As similarity measure, we use the Tanimoto coefficient:

$$T(\mathbf{d}^*, \mathbf{d}') = \frac{\mathbf{d}^* \cdot \mathbf{d}'}{||\mathbf{d}^*||^2 + ||\mathbf{d}'||^2 - \mathbf{d}^* \cdot \mathbf{d}'}. \tag{3}$$

The Tanimoto coefficient produces values in the interval $[0, 1]$, the higher the value the higher the similarity. If the two vectors are identical, the coefficient is 1.

4 The Visual Tracking System

The tracking system we present uses the component-based descriptor from Sec. 3 for the observation model of a particle filter. It employs the standard Condensation algorithm [20] which maintains a set of weighted particles over time using a recursive procedure based on three steps: First, the system draws particles randomly from the particle set of the previous time step, where each particle is drawn with a probability proportional to the associated weight of the particle. Second, the particles are transformed (predicted) according to a motion model. Finally, all particles are assigned new weights according to an observation model and the object state is estimated.

Let us first introduce the notation. At each point in time $t \in \{1, .., T\}$, the particle filter recursively computes an estimate of the probability density of the person's location within the image using a set of J (here $J = 500$) particles $\mathbf{\Phi}_t = \{\phi^1_t, ...\phi^J_t\}$ with $\phi^j_t = (\mathbf{s}^j_t, \pi^j_t, \mathbf{d}^j_t), \quad j \in \{1, ..., J\}$. Here, $\mathbf{s}^j_t = (x, y, v_x, v_y, w, h)$ is the state vector that specifies the particle's region with center (x, y), width w and height h – in the following, the region is also denoted as $\mathbf{R}^j_t = (x, y, w, h)$. v_x and v_y specify the current velocity of the particle in x and y directions. Each particle additionally has a weight π^j_t determining the relevance of the particle with respect to the target, and the component-based descriptor $\mathbf{d}^j_t$ that describes the appearance of the particle region.

In the following, we first mention how the system is initialized (sec. 4.1), second describe the motion model (sec. 4.2), and finally, specify the observation model as core of the system (sec. 4.3).

4.1 Initialization

To start the tracking process, the initial target region $\mathbf{R}^*$ has to be specified in the first frame. This can be carried out manually or automatically with a separate detection module. Here, we initialize manually. Based on the initial target region $\mathbf{R}^*$, the component-based descriptor $\mathbf{d}^*$ is computed that describes the appearance of the person. The initial particle set $\mathbf{\Phi}_0 = \{(\mathbf{s}^j_0, \pi^j_0, \mathbf{d}^j_0) \,|\, j = 1, ..., J\}$ is generated by randomly distributing the initial target location around the region's center (x^*, y^*). The velocity components v_x and v_y are initially set to 0 and the region dimensions of each particle are initialized with the dimensions of $\mathbf{R}^*$. The particle weights π^j_0 are set to $1/J$.

4.2 Motion model

The object's motion is modeled by a simple first order autoregressive process in which the state $\mathbf{s}_t^j$ of a particle depends only on the state of the particle in the previous frame:

$$\mathbf{s}_t^j = \mathbf{M} \cdot \mathbf{s}_{t-1}^j + \mathbf{Q}. \tag{4}$$

Here, $\mathbf{M}$ is a state transition matrix of a constant velocity model and $\mathbf{Q}$ is a random variable that denotes some white Gaussian noise. This enables a flexible adaption of position and size of the particle region as well as of its velocity.[2] Thus the system is able to quickly react to velocity changes of the object.

4.3 Observation model

In visual tracking, the choice of the observation model is the most crucial step since it decides which particles will survive. It therefore has the strongest influence on the estimated position of the target. Here, we use the component-based descriptor to determine the feature description for the target and for each particle, enabling the comparison and weighting of particles.

First, we compute a descriptor $\mathbf{d_t^j}$ for each of the particles according to sec. 3.2. That means, the target template $\mathbf{M_{R^*}}$ is adapted to the size of the current particle and the descriptor $\mathbf{d_t^j}$ is computed for the resulting template $\mathbf{M_t^j}$. Then, the weight of a particle is computed based on the Tanimoto coefficient as

$$\pi_t^j = c \cdot e^{\lambda \cdot T(\mathbf{d}^*, \mathbf{d}_t^j)}. \tag{5}$$

This function prioritizes particles which are very similar to $\mathbf{d}^*$ by assigning an especially high weight. A value of $\lambda = 14$ has shown to be useful in our experiments. The parameter c is a normalization factor which is chosen so that $\sum_{j=1}^{J} \pi_t^j = 1$.

Finally, the current target state, including target position and size, can be estimated as a weighted average of the particles by

$$\mathbf{x}_t = \sum_{j=1}^{J} \pi_t^j \cdot \mathbf{s}_t^j. \tag{6}$$

5 Experiments and Results

The experiments were carried out using a RWI B21 robot equipped with a simple USB web camera mounted on a pantilt unit (see Fig. 3, left). The camera captures 15 frames/sec, with a resolution of 320×240. The software runs on a 2GHz dual core PC onboard the robot. For the experiments, the tracking application was implemented within the software framework RoSe developed at FKIE [35]. This framework consists of roughly 30 modules which exchange information over a UDP-based communication infrastructure. It is specifically designed to allow for the easy assembly of multi-robot applications, which extensively use wireless ad-hoc communication. However, here we only required two modules on a single robot:

[2] The size of the region is not adapted by M but only by Q.

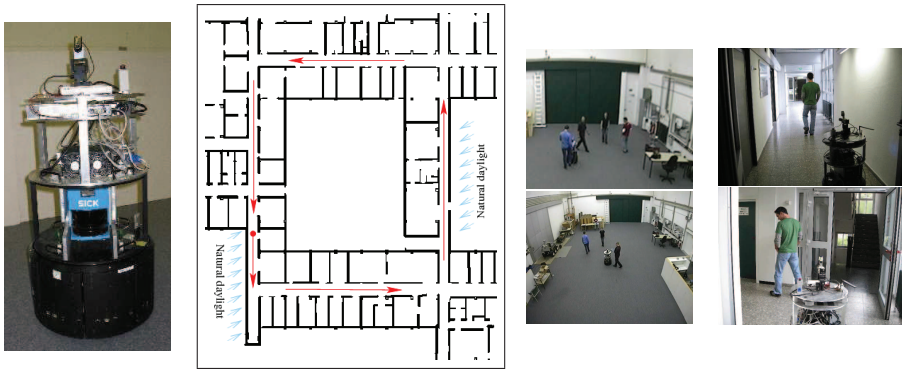


Fig. 3 Left: the RWI B21 robot *Blücher*. The images were taken using the small pantilt mounted webcam on top of the robot. Middle: An outline of the FKIE hallway environment. The red arrows indicate the corridors. Right: Experiments in our robot experimentation hall and in the corridors.

1) A visual tracking module, which captures the images and employs the tracking algorithm for tracking a single person within the image. Based on the pixel location of the person computed by the vision-based tracker, the module computes a heading direction relative to the robot, steers the pantilt unit in order to center the person within the image and commands the robot to follow the person. This is achieved by continuously instructing the reactive collision avoidance component of the robot to drive to a goal location a few meters ahead, in the direction of the moving person.

2) The collision avoidance component of the robot. It is specifically designed for the task of following moving persons based on motion tracking information. It does so by applying an expansive spaces tree algorithm, which carries out a search for admissible paths in time and space, based on information about static obstacles provided by a laser range scanner, as well as motion information, i.e. position and velocity vectors of moving obstacles and the person being followed, provided by the external tracking component [19].

We performed three series of experiments with this system within the robot experimentation hall and the hallways of the FKIE building (cf. Fig. 3). The first series evaluates the robustness of the component-based tracker compared to simpler feature-based techniques. In the second series, the robot autonomously controls the camera to track a target person while other persons are moving around in the field of view of the robot and try to distract it. In the third series, the robot uses the people tracker to autonomously follow a person.

All series were performed during normal working hours with people walking around. The lighting conditions varied strongly during the experiments: some areas show natural daylight, others artificial light. In some parts, the light was switched off resulting in poorly illuminated areas. These conditions resulted in several images with very poor quality. Furthermore, after quick camera movements the camera was out of focus for some frames and capturing images was sometimes delayed resulting in large changes between consecutive frames. To evaluate the tracking, we counted the number of detec-

Fig. 4 Some tracking results. Green points: particles that matched to target; cyan points: particles that didn't match. Rectangles show estimated target state. Yellow rectangle: more than 30% of particles match, otherwise the rectangle is blue.

tions manually. A detection occurs if the center of the target state was on the person[3]. In Fig. 4 we display some of the tracking results.

5.1 Experiment 1: Comparison with Other Feature-based Techniques

Most similar to the here presented approach are color-based trackers. Here, we compare our approach to three other color-based tracking methods. The first is the Camshift tracker [7] based on the MeanShift algorithm [11]. It is a statistical method of finding the peak of a probability distribution, usually obtained with a color histogram. Additionally to the implementation based on the HSV color space that is available from the OpenCV library[4], we used it with two other color spaces: RG chromaticity space and LAB space.

The second and the third method are both based on particle filters. The second approach is a standard method based on color histograms and was implemented according to [27]. The third approach that we call ROI (region of interest) tracking is a simplified version of the here presented method. It uses the same feature maps as in sec. 3.1 but no components. Instead, it considers the whole target region and computes a descriptor based on the ratio of the mean of the target region and the mean of the background as in eq. 2. Thus, it computes a 6-dimensional target descriptor.[5]

To be able to compare the approaches on the same data, several image sequences were acquired by tele-operating the robot and processed offline. We tested 5 different runs, each covering one circle through the hallways (approx. 160 m per run). Each run was performed with a different person as target, with different clothing (cf. Fig. 5). The runs consisted of 1000–1600 frames each. The results are displayed in Tab. 1. In all cases, the component-based tracker performed best, with an average detection rate of 90%. The simpler ROI tracking achieved 77% on average. The approaches based on color histograms (Camshift and histogram with particles) approaches perform considerably worse (33, 45, 40%, and 37%). This is mainly due to problems with illumination changes. For all approaches it turned out that the clothing of the person made a strong

[3] This approximation is actually too optimistic since the region might include a part of the background and still have its center on the target. It is reasonable here anyway since the center is the point the robot uses as target direction.

[4] OpenCV library: http://opencvlibrary.sourceforge.net/ For Camshift, it is usually necessary to adapt the parameters newly for each object. This is difficult for targets like persons which vary strongly in appearance due to different clothing. Since our tracker is applicable to different objects without adapting parameters, we used the Camshift algorithm with the standard parameter set of the OpenCV implementation for all test sequences to make the approaches comparable.

[5] We used almost the same method in [16], but omitted here the orientation features to make the approach comparable to the other methods which are purely color-based.

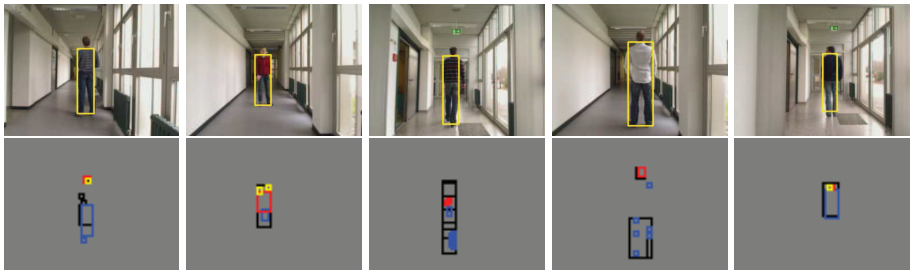


Fig. 5 Experiment 1: Top: initial frames and target regions $\mathbf{R}^*$ (yellow rectangles) used to learn the appearance of the 5 persons. Bottom: the templates $\mathbf{M}_{\mathbf{R}^*}$ that are determined for each of the targets. Each rectangle represents an $m_{i,j}$, its color represents the feature map it was extracted from.

	# Frames	correct detections [%]					
		Cam (HSV)	Cam (RG)	Cam (LAB)	Hist.	ROI	component
1	1477	51	88	39	42	85	95
2	1158	53	62	54	73	98	94
3	1596	5	28	50	17	60	85
4	1392	13	1	10	15	61	90
5	1519	46	47	46	38	80	84
	Average	33	45	40	37	77	90

Table 1 Experiment 1: Comparison of Camshift tracking with three different color spaces (HSV, RG, LAB), color histogram tracking with particles, ROI tracking, and our new component-based tracking. The rows show the results for the 5 persons in Fig. 5.

difference in performance: the larger the contrast and difference to the background, the easier the tracking.

5.2 Experiment 2: Tracking with Autonomous Camera Control

In the 2nd series of experiments, the robot was not moving itself, but autonomously controlled its camera to keep the target person in the center of the frame. We performed 4 runs with 4 different target persons. During all runs other persons were walking in the same area, occasionally occluding the target (cf. Fig. 3, 3rd col., and Fig. 4, a,b).

This experiment demonstrates the robustness of the tracking mechanism and especially the ability to discriminate individual persons. The results are shown in Tab. 2. Images in which the target was not visible were not considered for the detection rate but are mentioned in col. 4. It shows that the tracking works generally very well, the average detection rate is 91%. Most difficulties occurred in example 4, since here two people were sometimes confused.

5.3 Experiment 3: Autonomous Person Tracking

In the 3rd series of experiments, the robot followed a person autonomously. Three runs were performed in the robot experimentation hall and another four in the hallways of FKIE (cf. Fig. 3 and Fig. 4, c-e). The robot estimated the position of the person in each

	# Frames	detections [%]	# frames without target
1	278	91	0
2	509	92	9
3	437	99	0
4	491	82	0
Average	429	91	2.25

Table 2 Experiment 2: Results of component-based tracking on a stationary robot with autonomous camera control and several people walking around.

	# Frames	detections [%]	# frames without target
1	431	93	1
2	472	96	0
3	560	96	75
4	1533	88	13
5	1199	94	0
6	1612	95	8
7	1116	99	0
Average	989	94	14

Table 3 Experiment 3: Component-based tracking in online experiments used to autonomously drive a robot.

frame and drove autonomously into the direction of the estimated target state.[6] The camera was again controlled to center the target in the frame. The results are displayed in Tab. 3. In all of the runs, the detection rate was above 80%. The robot managed to keep the target person in its field of view very well. If the person was lost by the tracker, an audible signal told the person that it should wait for the robot to catch up again. One example in which the person was lost since it was too far away from the robot is displayed in Fig. 4 e. On the four runs through the hallways the sharp corners were the biggest challenge for the system. The 5th run was aborted on such a corner, because the robot lost the person and then was not sure enough if it detected the right person again. The average detection rate was 94%, showing that a robot equipped with the component-based tracker is able to follow a person autonomously.

6 Conclusion

In this paper, we have presented a component-based approach for visual tracking. We have applied the method to person tracking on a mobile platform which is especially challenging due to real-time constraints, a moving camera, and strong illumination and viewpoint changes. The appearance of a person is learned from an initially provided target region and the resulting target descriptor is used to search for the target in subsequent frames. Advantages of the system are that it determines automatically the most descriminative parts of a target, that it considers not only the appearance of the target but also of the background, and that it is quickly adaptable to a new target without a time-consuming learning phase.

[6] Here, control of the distance to the person is left to the laser-based collision avoidance. The robot approaches the person until a certain minimal distance is achieved.

We showed that the system is able to distinguish individuals and can follow a person autonomously through an environment. However, the task of person tracking in natural conditions is very challenging and there are still settings in which the system has difficulties. Persons with clothing similar to the background (especially camouflage), bright sunlight, and crowded environments are settings in which most systems fail. Adding additional features, e.g. motion cues, and asking for feedback from the target person in cases of ambiguity might help to tackle such problems. There are also cases in which the current approach has difficulties if the appearance of target and background change strongly, e.g. due to strong illumination changes. We are currently working on automatically detecting such changes and adapting the target descriptor accordingly.

References

1. M. Andriluka, S. Roth, and Schiele B. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.
2. K. O. Arras, O. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2D range data. In *Proc. Int'l Conf. on Robotics and Automation (ICRA'07)*, Rome, Italy, 2007.
3. K.O. Arras, S. Grzonka, M. Luber, and W. Burgard. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *Proc. of Int'l Conf. on Robotics and Automation*, 2008.
4. N. Bellotto and H. Hu. Multisensor data fusion for joint people tracking and identification with a service robot. In *Proc. of the IEEE Int'l Conf. on Robotics and Biomimetics*, Sanya, China, 2007.
5. M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *Int'l Journal of Robotics Research*, 24, 2005.
6. N. Beuter, O. Lohmann, J. Schmidt, and F. Kummert. Directed attention - a cognitive vision system for a mobile robot. In *18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009.
7. G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal, 1998.
8. C. Breglera, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *Int'l Journal of Computer Vision (IJCV)*, 2004.
9. K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part II: Applications to human modeling and markerless motion. *Int'l Journal of Computer Vision (IJCV)*, 2005.
10. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, 2002.
11. D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. Proc. Conf. Computer Vision and Pattern Recognition, 2000.
12. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.
13. S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. PhD thesis, University of Bonn, Germany, July 2005. Published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer Verlag.
14. S. Frintrop and M. Kessel. Most salient region tracking. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA'09)*, Kobe, Japan, 2009.
15. S. Frintrop, M. Klodt, and E. Rome. A real-time visual attention system using integral images. In *Proc. of Int'l Conf. on Computer Vision Systems*, 2007.
16. S. Frintrop, A. Königs, F. Hoeller, and D. Schulz. Visual person tracking using a cognitive observation model. In *ICRA Workshop on People Detection and Tracking*, 2009.
17. D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. *Int. Conference on Computer Vision (ICCV)*, 1999.
18. Helmut Grabner and Horst Bischof. On-line boosting and vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
19. F. Hoeller, D. Schulz, M. Moors, and F. E. Schneider. Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps. In *Proc. of the Int'l Conf. on Robots and Systems (IROS)*, pages 1260–1265. IEEE, 2007.

20. M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int'l J. of Computer Vision (IJCV)*, 29(1):5–28, 1998.
21. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
22. B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int'l J. of Computer Vision, Special Issue on Learning for Recognition and Recognition for Learning*, 77(1-3):259–289, 2008.
23. T. Mathes and J. H. Piater. Robust non-rigid object tracking using point distribution manifolds. In *Proc. of the 28th Annual Symposium of the German Association for Pattern Recognition (DAGM)*, 2006.
24. I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *Int'l Journal of Computer Vision*, 53(3), 2003.
25. M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Int'l Conf. on Robotics and Automation (ICRA)*, 2002.
26. Stephen E. Palmer. *Vision Science, Photons to Phenomenology.* The MIT Press, Cambridge, MA, 1999.
27. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. Proc. of European Conf. on Computer Vision, 2002.
28. P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proc. of the IEEE*, 92(3), 2004.
29. K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP – Image Understanding*, 59(1):94–115, 1994.
30. D. Schulz. A probabilistic exemplar approach to combine laser and vision for person tracking. In Proc. of the Int'l Conf. on Robotics Science and Systems, 2006.
31. D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. Int'l Journal of Robotics Research, 22(2), 2003.
32. Xuan Song, Jinshi Cui, Hongbin Zha, and Huijing Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *European Conference on Computer Vision (ECCV)*, 2008.
33. F. Tang and H. Tao. Object tracking with dynamic feature graph. In *Proc. of the IEEE Workshop on VS-PETS*, 2005.
34. G. Taylor and L. Kleeman. A multiple hypothesis walking person tracker with switched dynamic model. In *Conf. on Robotics and Automation (ACRA)*, 2004.
35. A. Tiderko, T. Bachran, F. Hoeller, D. Schulz, and F. E. Schneider. RoSe – a framework for multicast communication via unreliable networks in multi-robot systems. *Robotics and Autonomous Systems*, 56(12):1017–1026, 2008.
36. K. Toyama and A. Blake. Probabilistic tracking with exemplars in a metric space. *Int'l Journal of Computer Vision (IJCV)*, 48(1):9–19, 2002.
37. R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding (CVIU), special issue Modeling People*, 2006.
38. Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision (IJCV)*, 75(2), 2007.

Simone Frintrop got her diploma in computer science in 2001 from the University of Bonn, Germany, was a Ph.D. student at the Fraunhofer institute AiS in St. Augustin, Germany, and got her Ph.D. in 2005. 2005-2006 she was a postdoctoral researcher at the Royal Institute for Technology (KTH) in Stockholm, Sweden. Currently, she works as Senior Scientist in the Intelligent Vision Systems Group at the University of Bonn, where she investigates cognitive methods for intelligent vision systems.

Achim Königs received his diploma in computer science in 2008 from the University of Bonn, Germany. Currently, he works as Ph.D. student at the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE), where he investigates intelligent vision systems in conjunction with unmanned systems.

Frank Hoeller finished his diploma thesis in computer science in 2006 at the University of Bonn, Germany. Currently, he works as Ph.D. student at the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE). His main studies concern autonomous navigation and planning for unmanned systems.

Dirk Schulz received his Doctorate degree in Computer Science from the University of Bonn in 2002. In 2003 he worked as a postdoctoral researcher at the University of Washington in Seattle, USA. From 2004 to 2007 he was a postdoctoral researcher at the Computer Science department of the University of Bonn. Since 2008 he is head of the Unmanned Systems group at the Fraunhofer Institute for Communication, Information Processing and Ergonomics (FKIE) in Wachtberg, Germany. His primary research interests include networked multi-robot systems as well as state estimation and sensor fusion techniques with applications in robotics and intelligent environments.

Publication [10]

Simone Frintrop and Armin B. Cremers. Visual landmark generation and redetection with a single feature per frame. In *Proceedings of the IEEE International Conference Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010.

Visual Landmark Generation and Redetection with a Single Feature Per Frame

S. Frintrop and A. B. Cremers

***Abstract*— In this paper we show that visual landmark generation and redetection is possible with a single feature per frame. The approach is based on the assumption that highly discriminative regions are easily redetectable in subsequent frames as well as in frames visited from different viewpoints. We investigate which feature detectors fit for this purpose and under which conditions the discriminability applies. The approach is tested in a topological localization scenario in which the best feature is tracked over several frames to build landmarks. We show that we can represent a large environment with a few salient landmarks and that a large percentage of these landmarks is robustly redetectable from different viewpoints.**

I. INTRODUCTION

Self localization and navigation belong to the key competences of mobile robots and have been a topic of intensive research during the last decades. Vision-based approaches are of special interest in many applications, since cameras are light-weight, low-cost, passive sensors, that additionally offer rich information about the environment [1], [2], [3]. Visual localization and navigation is often based on landmarks, that means on objects or regions in the environment that serve as reference points for the robot. Ideally, they shall be easily redetectable from different viewpoints, under changing illumination conditions, and in the presence of disturbances such as walking people.

The first step of visual landmark detection is usually the feature detection. However, not all detected features are useful landmark candidates. Especially corner features are often detected at intersections of objects and thus not stable [4]. Furthermore, most feature detectors obtain a feature repeatability of 50 – 80%, depending on the scene and the transformation between frames [5]. That means, a large amount of the detected features is not redetected in a following frame. Only a few of the features are stable enough to survive the tracking over several frames. To find stable landmarks, a common approach is to extract a large amount of features (usually several hundred per frame), track them over several frames and keep only the most stable ones [4].

However, detecting, matching and storing of hundreds of features per frame and the comparison to a large image database is costly. Robots usually have to operate in real-time and additionally have to share resources between different modules and tasks. While there have been successful approaches to deal even with large amount of features [6], [7], it is certainly preferable if it is possible to solve the task with less features. Desirable would be to know in advance which features will turn out to be stable and thus will be good candidates for landmarks.

The authors are with the Institute of Computer Science III, University of Bonn, 53117 Bonn, Germany {frintrop}/{abc}@iai.uni-bonn.de

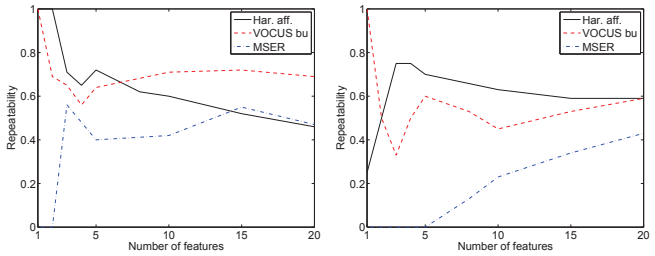


Fig. 1. Repeatability depending on number of features per frame. Features are selected by their quality as defined in sec. II-A.4. Examples determined on data sets 1 and 5 of Fig. 3 for 4 different viewpoints (after 50, 100, 150, and 200 frames) for Harris-affine regions, MSERs, and bottom-up salient regions (VOCUS bu). Two typical cases occur: either the best feature is very poor or extremely stable.

When investigating sparse sets of features (1–20 features per frame, features selected by their quality as defined in sec. II-A.4), we found two typical cases for the distribution of the repeatability values: Before converging to stable repeatability values, the repeatability of the best feature was either very poor or extremely high, often reaching 100% repeatability (cf. Fig. 1). This behaviour depended on the scene and was observed for all the investigated detectors. The proportion of poor versus high performance cases however differed among the detectors.

Outgoing from this observation, we pose the following questions: is it possible to exploit the fact that the best feature often is extremely stable? Is it possible to generate landmarks and redetect them reliably with only one feature per frame? For topological localization, it is in principle enough to have one landmark every few meters. The robot does not have to know its exact position and it is not necessary to see a landmark in each frame, as long as the scene is recognized reliably from time to time. A certain redundancy is necessary anyway since some landmarks may be occluded or removed upon revisiting that place, but as long as a few stable landmarks per environment remain, this is sufficient.

In this paper, we show that topological localization is possible with a single feature per frame. First, we investigate which feature detectors are suitable to be restricted to a sparse set and which quality measure suits to determine the best feature. We investigate Harris-affine regions [5], maximally stable extremal regions (MSERs) [8] and a saliency detector [9]; we finally chose the last one for

further investigations and show that it is possible to build stable landmarks from the most salient feature. In a scene classification experiment, we show that a reliable redetection of landmarks from different viewpoints is possible and that a test sequence can be reliable allocated to the correct scene.

Feature selection has been investigated before in several ways. In applications in which training data is available, machine learning methods often determine the best features for a class of objects from a pool of training images [10], [11]. The reduced set however contains usually still several dozens of features per frame or object and reduces only the features in the database, not the ones obtained during testing. Other approaches compare descriptors applied to the detected regions and keep only the most discriminative ones [12]. The main difference in our approach is that we start much earlier with the preselection, namely already during feature selection. In applications in which no training data is available, e.g. visual SLAM (simultaneous localization and mapping), some people use thresholds to reduce the number of features. E.g., [2] only add features to their map if the number of features visible in the robot view is below a threshold and [3] keep only landmarks that perform well over a sequence of frames. Preliminary investigations on the repeatability of a single stable feature have been made in [13]. Here, we extend this study by using detectors that are known to perform well in other applications (Harris-affine, MSERs), by introducing a more adequate repeatability measure, and by performing more detailed experiments. Completely new in this paper is the integration of the single-feature approach into a topological localization scenario.

II. Feature Detection

In this section, we discuss and evaluate the feature detection. First, we describe the investigated feature detectors and the quality measure to determine the best feature (sec. II-A). Second, we present the performance measure repeatability and extend the definition to image sequences (sec. II-B). Finally, we investigate in several experiments which feature detector provides the most stable feature in tracking and redetection situations (sec. IV-A).

A. Feature Detectors

1) Harris-Affine Regions: Harris-affine regions are computed by detecting interest points with the Harris detector in scale-space and determining an elliptical region for each point based on the second moment matrix of the intensity gradient [5].[1] For each pixel $\vec{x} = (x, y)$, the Harris detector determines its *cornerness* $c(\vec{x})$ (also strength or Harris response) as $c(\vec{x}) = \det(M) - \alpha trace^2(M)$, where M is the second moment matrix describing the local neighborhood of $\vec{x}$. This detector is applied to multiple scales and the characteristic scale is chosen to obtain scale-invariance. Finally, the affine region is determined according to [14]. If the cornerness exceeds a certain threshold, the pixel is defined as a corner.

2) MSERs: Maximally Stable Extremal Regions (MSERs) were introduced in [8] and have shown high repeatability results under various image transformations [5].[2] The MSER algorithm first detects several nested sets of extremal regions $Q_1, ..., Q_k$. Each Q_i is a region such that for all pixels $p \in Q_i, q \in \partial Q_i : I(p) > I(q)$ (MSER+) or $I(p) < I(q)$ (MSER-), where ∂Q_i is the boundary of Q_i, consisting of pixels that are adjacent but do not belong to Q_i, and $I(p)$ is the intensity value of p. A region Q_i is maximally stable iff the stability $q(i) = |Q_{i+\Delta} - Q_{i-\Delta}|/|Q_i|$ has a local minimum at i. Usually, a fixed Δ is used. This however results often in a set of regions with the same stability value (e.g. $q(i) = 0$) making it impossible to determine a single best MSER. Increasing Δ results in fewer regions with higher repeatability but usually lower $q(i)$, while a too large Δ might result in no MSERs in certain images. In our approach, we increase the Δ automatically until the MSER with the lowest $q(i)$ is non-ambiguous.

3) Biologically-inspired salient regions: Biologically-inspired attention systems compute the saliency of regions based on concepts of the human visual system [15]. They have shown to outperform other methods such as intensity contrasts, local oriented edge density, or entropy in terms of predicting human eye movements [16]. Here, we use the attention system VOCUS [9] that is real-time capable (20 ms for a 320×240 pixel image, on a 2.5 GHz PC [17]) and has a top-down part to search for targets.

VOCUS creates a saliency map by computing image contrasts and uniqueness of a feature. The feature computations for the features intensity, orientation, and color are performed on 3 different scales with image pyramids. Two intensity feature maps, for on-off and off-on contrasts, are computed by *center-surround mechanisms*. Similarly, 4 orientation maps ($0°, 45°, 90°, 135°$) and 4 color maps (green, blue, red, yellow) are computed (cf. [9]).

The core of the saliency detector is the *uniqueness weight* that is applied before feature channels are fused: a feature which occurs seldomly in a scene is assigned a higher saliency than a frequently occurring feature. This is a mechanism which enables humans to instantly detect outliers. The uniqueness $\mathcal{W}$ of map X is computed as $\mathcal{W}(X) = X/\sqrt{m}$, where m is the number of local maxima that exceed a threshold. Note that this weighting, together with the parallel investigation of different feature channels, distinguishes this detector from standard detectors such as Harris corners or MSERs because it considers the global instead of the local discriminability of a region.

The weighted feature maps are summed up to 3 conspicuity maps I (intensity), O (orientation) and C (color) and combined to the *saliency map*: $S_{bu} = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C)$. The salient regions, the *VOCUS-ROIs*, are the local maxima in S above a threshold, extended to a region with a region growing approach [18].

4) Sorting features: To determine the best features, we need a measure for the quality. This depends on the detector,

[1] We used the detector from http://www.robots.ox.ac.uk/∼vgg/research/affine

[2] We used the MSER code from the VLFeat library: http://www.vlfeat.org

for Harris-affine regions we chose the cornerness, for MSERs the stability, for VOCUS-ROIs the saliency. This results for each detector in an ordered list of features $F_i = (F_1, ..., F_n)$, where F_1 is the best feature and F_j has a higher quality than F_{j+1}.

B. Performance Measure: Repeatability

The performance measure to compare the stability of features is the repeatability that is defined as follows:

$$R(I_j, I_k) = \frac{\#\textit{ features in } I_j \textit{ with correspondence in } I_k}{\#\textit{ features in } I_j}\,.$$

for parts of the scene visible in both frames I_j and I_k. To be a valid correspondence, about 50% of the regions have to overlap. This allows a relatively large overlap error but a powerful descriptor is still able to match such regions successfully (cf. [5]). A symmetric measure can be obtained as follows:[3]

$$R_{sym}(I_j, I_k) = (R(I_j, I_k) + R(I_k, I_j))/2\,.$$

To extend the repeatability definition to image sequences or sets, we distinguish two different versions: we define the *tracking repeatability* as the average repeatability between consecutive frames:

$$R_T(I_{1:t}) = \frac{\sum_{i=2}^{t} R_{sym}(I_{i-1}, I_i)}{(t-1)}\,.$$

for an image sequence $I_{1:t} = I_1, ..., I_t$. It is called tracking repeatability because it is mainly of interest when features are tracked over frames. The *viewpoint repeatability* on the other hand is defined as the average repeatability between a frame I_i and the remaining images of the sequence or set:

$$R_V(I_i, I_{1:t}) = \frac{\sum_{j=1, Ij\neq Ii}^{t} R_{sym}(I_i, I_j)}{(t-1)}\,.$$

It is called viewpoint repeatability because, in contrast to tracking, the viewpoint between considered frames might change strongly, usually the more the longer the sequence.

III. Landmark Generation

While a feature is a 2D region in an image, a landmark is a region in the 3D world that can be observed from different viewpoints. To create landmarks, the detected feature is tracked over several frames. The resulting list of features represents a landmark. The *length* of a landmark is the number of elements in the list, which is equivalent to the number of frames the feature was detected in.

To compute the landmarks, we match new features to features from previous frames whereas we allow gaps of up to 2 frames. We finally consider only landmark with a length $\geq k$ (here: $k = 5$). This enables to determine which landmarks are stable over time.

To match two features, we use the SIFT descriptor [12] that has outperformed most other descriptors in terms of matching performance [19]. Usually, SIFT descriptors are computed at intensity extrema in scale space [12] or at Harris-Laplacians [5]. Here, we calculate one SIFT descriptor for each VOCUS-ROI. The center of the ROI provides the position and the size of the ROI determines the size of the descriptor grid. The grid should be larger than the ROI to allow catching information about the surrounding but should also not include too much background and stay within the image borders[4]. The procedure to generate landmarks is illustrated in Fig. 2.

IV. Experiments

In our experiments, we investigate three questions. First: Which is the best feature detector for our purpose? This experiment investigates the repeatability in tracking situations as well as under strong viewpoint changes. Second: Is it possible to create stable landmarks from a single feature per frame? And third: Can localization be performed based on such a sparse landmark representation?

A. Which is the best feature detector for our purpose?

To test which feature detector suits best for our purpose, we investigated the repeatability of features on 7 image sequences of 200-400 frames of size 320×240 (cf. Fig. 3). In all sequences, strong viewpoint changes occur. Data sets 1–4 show natural scenes in an office environment and contain objects which were especially designed to be salient for humans: a green exit sign, a magnet clamp, a red circle containing a warning remark, and, in data set 4, a fire extinguisher and a red piece of paper at the wall. The last 3 data sets show natural, cluttered office environment scenes. We investigated the tracking repeatability as well as the viewpoint repeatability on these data sets. The results are displayed in Table I. As to be expected, the tracking repeatability is almost always higher than the viewpoint repeatability. Worth to note is also that the viewpoint repeatability naturally goes down the more the viewpoint changes.

It turns out that the Harris regions as well as the salient VOCUS-ROIs perform well in most cases, whereas the MSERs show a considerably lower performance. The VOCUS-ROIs outperform the Harris regions on average since the attention system is able to capture the uniqueness of features in more cases. The low performance of the MSERs can be explained as follows: usually, MSERs are stable, if all possible MSERs in a scene are considered (as in [5]). But, since all MSERs have an equal stability value, it is hard to determine a stable subset or even a best feature. So, if reduction of the number of features is desired, the other detectors seem to be the better choice.

We decided to use the salient VOCUS-ROIs for our application, first, because they yielded the highest repeatability

[3]The symmetric measure in [5] divides instead by the smaller of the number of regions in both frames. This might however result in problems if the number of features in the 1st frame is a subset of the features in the 2nd frame. The measure would report a repeatability of 100%, even if the number of features in the 2nd frame is considerably larger. This is especially a problem for small numbers of features.

[4]We chose a grid size of 1.5 times the maximum of width and height of the ROI.

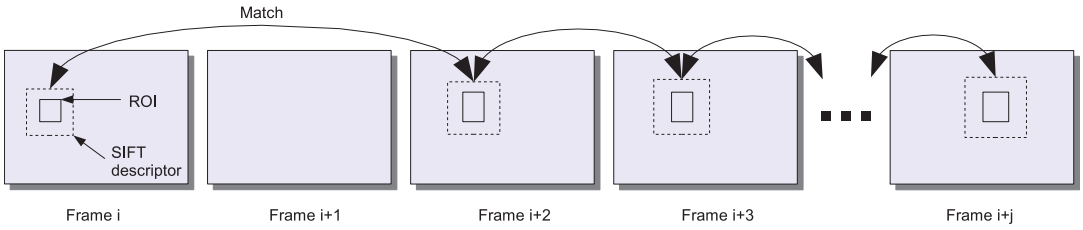


Fig. 2. The process to generate landmarks: For each feature (ROI, solid rectangle), a SIFT descriptor is computed (area in dashed rectangle). The descriptors of the ROIs of consecutive frames are compared. If they match, a landmark is created. Gaps of up to 2 frames are allowed and only landmarks of length $\geq k = 5$ are considered.

Fig. 3. Data sets. 1st row: 1st frame, 2nd row: last frame of sequence

data set	# frames	Tracking repeatability [%]			Viewpoint repeatability [%]		
		Harris	MSER	VOCUS-ROI	Harris	MSER	VOCUS-ROI
1.	259	96	25	97	96	18	97
2.	210	76	80	100	78	89	100
3.	315	94	20	90	77	5	82
4.	254	95	33	83	92	21	29
5.	254	85	5	100	19	11	100
6.	209	76	13	82	61	1	14
7.	341	86	10	84	19	9	72
av.		87	27	91	63	22	71

TABLE I

THE TRACKING REPEATABILITY $R_T(I_{1:t})$ AND THE VIEWPOINT REPEATABILITY $R_V(I_1, I_{2:t})$ OF THE BEST FEATURE F_1 (SELECTED ACCORDING TO SEC. II-A.4) ON THE DATA SETS OF FIG. 3.

and second, because it is possible to adapt the attention system to search for expected regions in a top-down manner. We plan to exploit this in future work.

B. Is it possible to create stable landmarks from a single feature per frame?

In this section, we investigate whether the VOCUS-ROIs can be used to create stable landmarks. A stable landmark should be visible over several frames and should be redetectable under viewpoint and illumination changes. We tested our approach in 5 scenes of a typical office building: 3 corridors on different levels of the same building (scene 2,3,4) and two open areas (scene 1 and 5) (cf. Fig. 4). The corridors, especially scene 3 and 4, are very similar, resulting in matching ambiguities. The experiments were performed during normal working hours, i.e. people walked around, doors were opened or closed etc. In each of the scenes, we recorded two image sequences (denoted a and b in the following) with a mobile camera mounted on a moving vehicle. Each track had a length of about $100\,m$, images had a resolution of 320×240.

First, we test whether a single feature per frame is sufficient at all to build landmarks. Remember that a feature has to be seen and matched over at least 5 frames to become a valid landmark. Thus, if repeatability is too low, the system will not create any landmarks. The results are shown in Table II. We obtained between 9 and 62 landmarks per scene, depending on the length of the sequence. Each landmark consists of 7 – 16 ROIs, on average 10 ROIs. That means, a feature that was used to create a landmark was on average visible over 10 frames. This shows that it is possible to create landmarks even from a single feature per frame. The approach can also be applied for tasks like visual SLAM (simultaneous localization and mapping) in which no previous training is possible.

Next, we investigate whether these landmarks can be redetected under viewpoint and illumination changes. Especially for a sparse landmark representation this is not obvious and has to be investigated further.

To test the redetection of landmarks, we divided the image

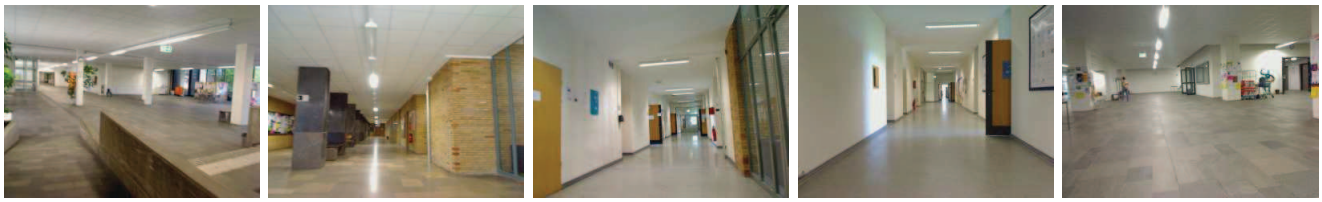


Fig. 4. Example frames of the 5 scenes we investigated for scene recognition

TABLE II
LANDMARK GENERATION

Scene	# Frames	# landmarks	av. # ROIs per LM
1.a	539	13	8
1.b	598	26	10
2.a	1194	31	7
2.b	1144	56	9
3.a	828	32	12
3.b	749	17	10
4.a	1720	62	16
4.b	1064	48	11
5.a	580	26	12
5.b	568	9	7

TABLE III
LANDMARK REDETECTION. LEFT COLUMN: S_i/S_j MEANS THAT LANDMARKS WERE OBTAINED FROM REFERENCE SEQUENCE S_i AND REDETECTED IN TEST SEQUENCE S_j.

Scene	redetected LMs [%]
1.a/1.b	84
2.a/2.b	83
3.a/3.b	75
4.a/4.b	61
5.a/5.b	73
1.b/1.a	69
2.b/2.a	79
3.b/3.a	94
4.b/4.a	38
5.b/5.a	78
average	73

sequences into train and test sequences. In a first run, the sequences denoted by a (1.a, 2.a, ..., 5.a) are used as training data $S_i, i \in \{1, .., 5\}$, the ones denoted by b as test sequences $S_j, j \in \{1, .., 5\}$. In a second run, we applied the sequences vice versa. The redetection ratio was determined by matching the detected VOCUS-ROIs of each frame of test sequence S_j to all landmarks obtained from the training sequences $S_i, i \neq j$. (Remember that only one of these sequences is from the same environment as S_j, the other sequences are from different environments.) Some matching examples are displayed in Fig. 5; the percentage of redetected landmarks is shown in Tab. III. It shows that generally the majority of the landmarks, on average 73%, is redetected in a test sequence. Thus, stable landmarks can be created from a single feature per frame and reliably redetected.

C. Can we perform topological localization with such a sparse landmark representation?

In this section, we show that the sparse landmark representation that we obtained in the previous section can be used to reliably localize a system in an office scenario. We use the same sequences as in the previous section and show that we can reliably assign the correct location to a sequence of images. To show this, we cross-validated the matching performance of all sequences to each other, i.e. we considered one sequence S_i as training data and another sequence S_j as test data. For each sequence combination (S_i, S_j) we compute the confidence that the test sequence S_j was obtained in the same environment as the reference sequence S_i:

$$C(S_i, S_j) = \frac{M(S_i, S_j)}{\sum_{k=1, k\neq j}^{N} M(S_k, S_j)}, \quad \forall i \neq j$$

where N is the number of sequences, here $N = 10$, and $M(S_i, S_j)$ denotes the number of ROI matches between S_i and S_j.

The confidence values are shown in Tab. IV. It can be seen that the confidence values for test sequences from the same environment as the reference sequences are considerably higher (bold numbers). In most cases, they are between 95 and 100%. Only the matching confidences for scenes 3 and 4, two very similar corridors, are a little lower. The similarity of the two scenes results in several false detections. Still, the confidence for the correct sequence is always more than three times as high as the confidence for each other sequence. The final decision of the robot for a test sequence S_j is:

$$Estimated\ scene = argmax_i\ C(S_i, S_j) \qquad (1)$$

Based on this decision rule, the system determines the correct scene for all of the test sequences. Thus, we have shown that it is possible to reliably localize a system based on a single feature per frame. This is also applicable if no training phase is possible as in visual SLAM.

V. DISCUSSION AND CONCLUSION

In this paper, we have shown that visual localization and scene recognition is possible with a very sparse landmark representation. Focusing on the most salient feature in a frame enables to select the most discriminative regions in an environment as landmark candidates. While this approach does not detect landmarks in each part of the environment (if there is nothing salient, no landmarks are found), it works well as long as the environment contains some discriminative parts. Especially human-made environments have plenty of such salient objects: fire extinguishers, exit signs, doors or posters can serve as valuable landmarks. The advantage of such landmarks is that they are easily redetected from

TABLE IV

THE CONFIDENCE VALUES FOR THE LANDMARK MATCHING. ROWS DENOTE THE REFERENCE SEQUENCES S_i, COLUMNS THE TEST SEQUENCES S_j. BOLD NUMBERS HIGHLIGHT THE HIGHEST VALUE IN A COLUMN.

	1.a	1.b	2.a	2.b	3.a	3.b	4.a	4.b	5.a	5.b
1.a		**100**	0	0	0	0	0	0	0	0
1.b	**100**		0	0	0	0	0	0	0	0
2.a	0	0		**99**	2	1	0	1	0	0
2.b	0	0	**95**		1	1	0	0	1	0
3.a	0	0	3	1		**82**	11	19	1	1
3.b	0	0	1	0	**75**		7	11	0	0
4.a	0	0	0	1	8	5		**68**	0	0
4.b	0	0	1	0	14	1	**82**		0	0
5.a	0	0	0	0	1	0	0	0		**98**
5.b	0	0	0	0	1	0	0	0	**98**	

Fig. 5. Some examples of matching ROIs. Top: test sequence, bottom: reference sequence. Four successful matches and one false match (right) are shown.

different viewpoints. We show in several experiments that the one-feature-per-frame approach is well suited for landmark generation and redetection.

While the approach works well in the presented setting, it could be even improved with active camera control and top-down feature search. This would enable the robot to actively search for salient landmarks. In future work, we plan to integrate the one-feature approach to a SLAM setting with active camera control as the one in [3]. We also plan to investigate how the approach copes with long-term changes in the environment.

REFERENCES

[1] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int'l J. of Robotics Research*, vol. 21, no. 8, pp. 735–758, August 2002.

[2] A. Davison and D. Murray, "Simultaneous localisation and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 7, 2002.

[3] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *IEEE Trans. on Robotics, Special Issue on Visual SLAM*, vol. 24, no. 5, Oct 2008.

[4] J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[5] K. Mikolajczyk and C. Schmid, "A comparison of affine region detectors," *International Journal of Computer Vision (IJCV)*, vol. 65, no. 1-2, pp. 43–72, 2005.

[6] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[7] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based localization by indexing scale-invariant features," *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 36, no. 2, 2006.

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. of the British Machine Vision Conference*, 2002.

[9] S. Frintrop, "VOCUS: a visual attention system for object detection and goal-directed search," Ph.D. dissertation, Rheinische Friedrich-Wilhelms-Universität Bonn, Germany, July 2005, published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer.

[10] A. Pronobis and B. Caputo, "Confidence-based cue integration for visual place recognition," in *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS'07)*, 2007.

[11] F. Li and J. Kosecka, "Probabilistic location recognition using reduced feature set," in *ICRA*, 2006, pp. 3405–3410.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] S. Frintrop, "The high repeatability of salient regions," in *Proc. of ECCV workshop "Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environments"*, 2008.

[14] T. Lindeberg and J. Garding, "Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure," *Image and Vision Computing*, vol. 15, no. 6, pp. 415–434, 1997.

[15] A. M. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, pp. 97–136, 1980.

[16] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 19. Cambridge, MA: MIT Press, 2006, pp. 547–554.

[17] S. Frintrop, M. Klodt, and E. Rome, "A real-time visual attention system using integral images," in *Proc. of the 5th Int'l Conf. on Computer Vision Systems (ICVS)*, Bielefeld, Germany, March 2007.

[18] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 16, no. 6, pp. 641 – 647, 1994.

[19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions of Pattern Analysis and Machine intelligence*, vol. 27, no. 10, 2005.

Publication [11]

Simone Frintrop and Patric Jensfelt. Attentional landmarks and active gaze control for visual SLAM. *IEEE Transactions on Robotics, Special Issue on Visual SLAM*, 24(5), 2008.

Attentional Landmarks and Active Gaze Control for Visual SLAM

Simone Frintrop and Patric Jensfelt

***Abstract*—This paper is centered around landmark detection, tracking and matching for visual SLAM (Simultaneous Localization And Mapping) using a monocular vision system with active gaze control. We present a system specialized in creating and maintaining a sparse set of landmarks based on a biologically motivated feature selection strategy. A visual attention system detects salient features which are highly discriminative, ideal candidates for visual landmarks which are easy to redetect. Features are tracked over several frames to determine stable landmarks and to estimate their 3D position in the environment. Matching of current landmarks to database entries enables loop closing. Active gaze control allows us to overcome some of the limitations of using a monocular vision system with a relatively small field of view. It supports (i) the tracking of landmarks which enable a better pose estimation, (ii) the exploration of regions without landmarks to obtain a better distribution of landmarks in the environment, and (iii) the active redetection of landmarks to enable loop closing in situations in which a fixed camera fails to close the loop. Several real-world experiments show that accurate pose estimation is obtained with the presented system and that active camera control outperforms the passive approach.**

***Index Terms*—Mobile robotics, visual SLAM, landmark selection, visual attention, saliency, active camera control**

I. Introduction

WHAT do I see? This is one of the most important questions for a robot that navigates and localizes itself based on camera data. What is "seen" or "perceived" at a certain moment in time is firstly determined by the images acquired by the camera and secondly by the information extracted from the images. The first aspect is usually determined by the hardware, but if a steerable camera is available, it is possible to actively direct the camera to obtain useful data. "Useful" refers here to data which supports improving the current task, e.g. localization and map building. The second aspect is especially important in tasks based on visual data since the large amount of image data together with real-time constraints make it impossible to process everything. Selecting the most important data is one of the most challenging tasks in this field.

SLAM is the task of simultaneously estimating a *model* or *map* of the environment and the robot's position in this map. The map is not necessarily a 3D reconstruction of the world, it is a representation that allows the robot to localize itself. Based on range sensors such as laser scanners, SLAM has reached a rather mature level [1], [2], [3], [4], [5]. Visual SLAM instead attempts to solve the problem with cameras as external sensors [6], [7], [8], [9], [10], [11]. This is desirable because cameras are low-cost, low-power and lightweight sensors which may be used in many applications where laser scanners are too expensive or too heavy. In addition, the rich visual information allows the use of more complex feature models for position estimation and recognition. On the other hand, visual SLAM is considerably harder, for example for the reasons given above.

A key competence in visual SLAM is to choose useful landmarks which are easy to track, stable over several frames, and easily re-detectable when returning to a previously visited location. This *loop closing* is important in SLAM since it decreases accumulated errors by distributing information from areas with lower uncertainty to those with higher. Furthermore, the number of landmarks should be kept under control since the complexity of SLAM typically is a function of the number of landmarks in the map. Landmarks should also be well distributed over the environment. Here, we suggest the application of a biologically motivated attention system [12] to find salient regions in images. Attention systems are designed to favor regions with a high uniqueness such as a red fire extinguisher on a white wall. Such regions are especially useful for visual SLAM because they are discriminative by definition and easy to track and redetect. We show that salient regions have a considerably higher repeatability than Harris-Laplacians and SIFT keypoints.

Another important part of our system is the gaze control module. The strategy to steer the camera consists of three behaviours: a *tracking* behaviour identifies the most promising landmarks and prevents them from leaving the field of view. A *redetection* behaviour actively searches for expected landmarks to support loop-closing. Finally, an *exploration* behaviour investigates regions with no landmarks, leading to a more uniform distribution of landmarks. The advantage of the active gaze control is to obtain more informative landmarks (e.g. with a better baseline), a faster loop closing, and a better distribution of landmarks in the environment.

The contributions of this paper are first, a landmark selection scheme which allows a reliable pose estimation with a sparse set of especially discriminative landmarks, second, a precision-based loop-closing procedure based on SIFT descriptors, and finally, an active gaze control strategy to obtain a better baseline for landmark estimations, a faster loop closing, and a more uniform distribution of landmarks in the environment. Experimental results are presented to show the performance of the system. This paper builds on our previous work [8], [13], [14] and combines all this knowledge into one system.

S. Frintrop is with the Institute of Computer Science III, Rheinische Friedrich-Wilhems-Universität, 53117 Bonn, Germany e-mail: frintrop@iai.uni-bonn.de

P. Jensfelt is with the Centre for Autonomous Systems (CAS), Royal Institute of Technology, 10044 Stockholm, Sweden patric@csc.kth.se

In the following, we first give an overview over related work (sec. II), then we introduce the SLAM architecture (sec. III). Sec. IV, V, and VI describe the landmark selection and matching processes and VII introduces the active camera control. Sec. VIII shows the performance of the SLAM system in several real-world scenarios and illustrates the advantages of active camera control. Finally, we finish with a conclusion.

II. Related Work

As mentioned in the introduction, there has been large interest in solving the visual SLAM problem during the last years [6], [7], [8], [9], [10], [11]. One of the most important issues in this field are landmark selection and matching. These mechanisms directly affect the ability of the system to reliably track and redetect regions in a scene and to build a consistent representation of the environment. Especially in loop closing situations, matching of regions has to be largely invariant to viewpoint and illumination changes.

The simplest kind of landmarks are artificial landmarks like red squares or white circles on floor or walls [15], [16]. They have the advantage that their appearance is known in advance and the re-detection is easy. While a simple solution if the main research focus is not on the visual processing, this approach has several obvious drawbacks. First, the environment has to be prepared before the system is started. Apart from the effort this requires, this is often not desired, especially since visual landmarks are also visible for humans. Second, landmarks with uniform appearance are difficult to tell apart which makes loop closing hard. Another approach is to detect frequently occurring objects like ceiling lights [17]. While this approach does not require a preparation of the environment, it is still dependent on the occurrence of this object.

Because of these drawbacks, current systems determine landmarks which are based on ubiquitous features like lines, corners, or blobs. Frequently used is the *Harris corner detector* [18] which detects corner-like regions with a significant signal change in two orthogonal directions. An extension to make the detector scale-invariant, the *Harris-Laplacian detector* [19], was used by Jensfelt et al. for visual SLAM [8]. Davison and Murray [6] find regions with a version of the Harris detector to large image patches (9×9 to 15×15) as suggested by Shi and Tomasi [20]. Newman and Ho [21] used *maximally stable extremal regions (MSERs)* [22] and in newer work [9] *Harris affine regions* [23]. In previous work, we used a combination of attention regions with Harris-Laplacian corners [13].

Here, we show that attention regions alone can be used as landmarks which simplifies and speeds up the system. Many attention systems have been developed during the last two decades [24], [25], [12]. They are all based on principles of visual attention in the human visual system and adopt many of their ideas from psychophysical and neuro-biological theories [26], [27], [28]. Here, we use the attention system VOCUS [12], which is capable to operate in real-time [29].

Attention methods are well suited for selecting landmark candidates since they favor especially discriminative regions in a scene, nevertheless, their application to landmark selection has rarely been studied. Nickerson et al. detect landmarks in hand-coded maps [30], Ouerhani et al. built a topological map based on attentional landmarks [31], and Siagian and Itti use attentional landmarks in combination with the gist of a scene for outdoor Monte-Carlo Localization [32]. The only approach we are aware of which uses an approach similar to a visual attention system for landmark detection for SLAM, is presented in [33]. They use a saliency measure based on entropy to define important regions in the environment primarily for the loop closing detection in SLAM. However, the map itself is built using a laser scanner.

Landmarks can only be detected and re-detected if they are in the field of view of the robot's sensor. By actively controlling the viewing direction of the sensors much can be gained. The idea of actively controlling the sensors is not new. Control of sensors in general is a mature discipline that dates back several decades. In vision, the concept was first introduced by Bajcsy [34], and made popular by Active Vision [35] and Active Perception [36]. In terms of sensing for active localization, Maximum Information Systems are an early demonstration of sensing and localization [37]. Active motion to increase recognition performance and active exploration was introduced in [38]. More recent work has demonstrated the use of similar methods for exploration and mapping [39]. Active exploration by moving the robot to cover space was presented in [40] and in [41] the uncertainty of the robot pose and feature locations were also taken into account. In [42] an approach for active sensing with ultrasound sensors and laser-range finders in a localization context is presented. When cameras are used as sensors, the matching problem becomes more difficult but includes also a higher information content. In the field of object recognition, [43] show how to improve the recognition results by moving the camera actively to regions which maximize discriminability.

In the field of visual SLAM, most approaches use cameras mounted statically on a robot. Probably the most advanced work in the field of active camera control for visual SLAM is presented by Davison and colleagues. In [6], they present a robotic system which chooses landmarks for tracking which best improve the position knowledge of the system. In more recent work [44], [11], they apply their visual SLAM approach to a hand-held camera. Active movements are done by the user, according to instructions from a user-interface [44], or they use the active approach to choose the best landmarks from the current scene without controlling the camera [11].

III. System Overview

This paper describes a system for visual SLAM using an attention-based landmark selection scheme and an active gaze control strategy. This section gives an overview of the components in the system. The visual SLAM architecture is displayed in Fig. 1. Main components are a *robot* which provides camera images and odometry information, a *feature detector* which finds regions of interest (ROIs) in the images, a *feature tracker* which tracks ROIs over several frames and builds landmarks, a *triangulator* which identifies useful landmarks, a *database* in which triangulated landmarks are stored, a *SLAM module* which builds a map of the environment, a *loop closer* which

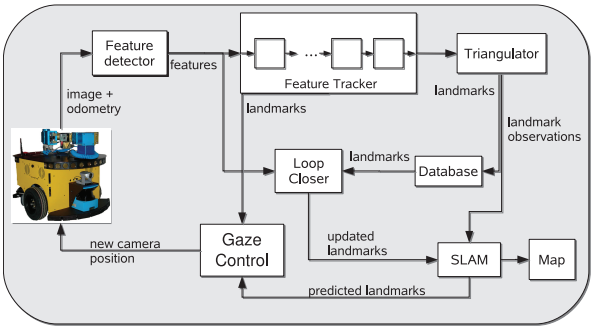


Fig. 1. The active visual SLAM system estimates a map of the environment from image data and odometry.

matches current ROIs to the database and a *gaze control module* which determines where to direct the camera to. The robot used in the experiments is an ActivMedia PowerBot equipped with a Canon VC-C4 pan/tilt/zoom camera mounted in the front of the robot at a height of about 0.35m above the floor. The ability to zoom is not used in this work.

When a new frame from the camera is available, it is provided to the *feature detector*, which finds ROIs based on a visual attention system. Next, the features are provided to the *feature tracker* which stores the last n frames, performs matching of ROIs in these frames and creates landmarks. The purpose of this buffer is to identify features which are stable over several frames and have enough parallax information for 3D initialization. These computations are performed by the *triangulator*. Selected landmarks are stored in a *database* and provided to the EKF-based *SLAM module* which computes an estimate of the position of landmarks and integrates the position estimate into the map. Details about the robot and the SLAM architecture can be found in [8]. Notice that the inverse depth representation for landmarks [45] would have allowed for an undelayed initialization of the landmarks. However the main purpose of the buffer in this paper is for selecting what landmarks are suitable for inclusion in the map and it would thus still be used had another SLAM technique been applied.

The task of the *loop closer* is to detect if a scene has been seen before. Therefore, the features from the current frame are compared with the landmarks in the database. The *gaze control module* actively controls the camera. It decides whether to track currently seen landmarks, to actively look for predicted landmarks, or to explore unseen areas. It computes a new camera position which is provided to the robot.

IV. Features and Landmarks

As mentioned before, landmark selection and matching belong to the most important issues in visual SLAM. A *landmark* is a region in the world. It has a 3D location and an appearance. A *feature* on the other hand is a region in an image. It has only a 2D location in the image and an appearance. The distance to the feature is initially not known since we use a monocular vision system. To build landmarks, features are detected in each frame, tracked over several frames and finally, the 3D position of the landmark is estimated by triangulation.

Feature selection is performed with a *detector* and the matching with a *descriptor*. While these two mechanisms are often not distinguished in the literature (people talk e.g. about "SIFT-features"), it is important to distinguish between them. A stable detector is necessary to redetect the same regions in different views of a scene. In applications like visual SLAM with time and memory constraints, it is also favorable to restrict the amount of detected regions. A powerful descriptor on the other hand has to capture the image properties at the detected region of interest and enable a stable matching of two regions with a high detection and low false detection rate. It has to be able to cope with viewpoint variations as well as with illumination changes. In this section, first the feature detection is introduced which finds ROIs in images (IV-A), then the descriptors which describe ROIs (IV-B), and finally the strategy to match two ROIs based on the descriptors (IV-C).

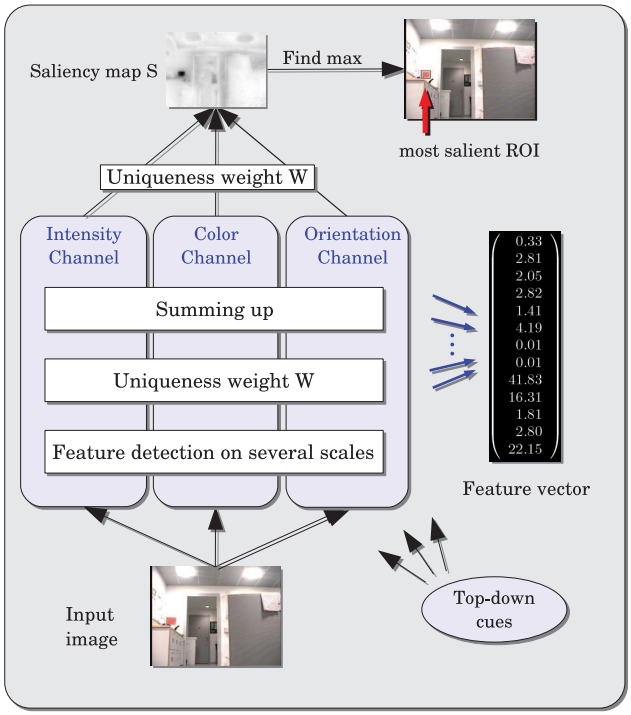
A. Attentional Feature Detection

An ideal candidate for selecting a few, discriminative regions in an image is a visual attention system. Computational attention systems select features motivated from mechanisms of the human visual system: several feature channels are considered independently and strong contrasts and the uniqueness of features determine their overall saliency. The resulting regions of interest have the advantage that they are highly discriminative, since repeated structure is assigned low saliency automatically. Another advantage is that there are usually only few regions detected per image (on average between 5 to 20), reducing the amount of features to be stored and matched considerably.

The attention system we use is VOCUS (Visual Object detection with a CompUtational attention System) [12]. VOCUS consists of a bottom-up part which computes saliency purely based on the content of the current image and a top-down part which considers pre-knowledge and target information to perform visual search. Here, we consider only the bottom-up part of VOCUS, however, top-down search can be used additionally if a target is specified.[1] For the approach presented here, any real-time capable attention system which computes a feature vector for each region of interest could be used.

An overview of VOCUS is shown in Fig. 2. The bottom-up part detects salient image regions by computing image contrasts and the uniqueness of a feature. The computations for the features intensity, orientation, and color are performed on 3 different scales with image pyramids. The feature intensity is computed by *center-surround mechanisms*; in contrast to most other attention systems [24], [31], on-off and off-on contrasts are computed separately. After summing up the scales, this yields 2 intensity maps. Similarly, 4 orientation maps $(0\,^\circ, 45\,^\circ, 90\,^\circ, 135\,^\circ)$ are computed by Gabor filters and 4 color maps (green, blue, red, yellow) which highlight salient

[1] In [46] we found that in tracking situations, bottom-up matching outperforms top-down search, for loop-closing, top-down search is preferable. But since using the top-down mechanism requires a target, rather precise expectations about expected landmarks are necessary. If the system searches for many expected landmarks in each frame this slows down the system considerably since the top-down search has to be applied for each expectation.



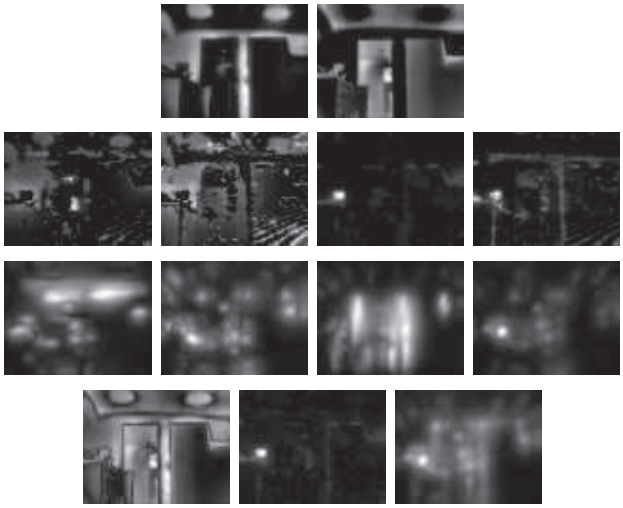


Fig. 2. Left: the visual attention system VOCUS detects regions of interest (ROIs) in images based on the features intensity, orientation, and color. For each ROI, it computes a feature vector which describes the contribution of the features to the ROI. Right: The feature and conspicuity maps for the image on the left. Top-left to bottom-right: intensity on-off, intensity off-on, color maps green, blue, red, yellow, orientation maps $0\,^\circ$, $45\,^\circ$, $90\,^\circ$, $135\,^\circ$ and conspicuity maps I, C, O. Since the red region sticks out as a unique peak in the feature map *red*, this map is weighted strongly by the uniqueness weight function and the corresponding region becomes the brightest in the saliency map (left, top).

regions of a certain color. Before the features are fused, they are weighted according to their *uniqueness*: a feature which occurs seldomly in a scene is assigned a higher saliency than a frequently occurring feature. This is a mechanism which enables humans to instantly detect outliers like a black sheep in a white herd [26], [27]. The uniqueness $\mathcal{W}$ of map X is defined as

$$\mathcal{W}(X) = X/\sqrt{m}, \tag{1}$$

where m is the number of local maxima that exceed a threshold and '$/$' is here the point-wise division of an image with a scalar. The maps are summed up to 3 conspicuity maps I (intensity), O (orientation) and C (color) and combined to form the *saliency map*:

$$S = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C) \tag{2}$$

From the saliency map, the brightest regions are extracted as *regions of interest (ROIs)*. This is done by first determining the maxima (brightest points) in the map and then finding for each maximum a surrounding region with *seeded region growing*. This method finds recursively all neighbors with sufficient saliency. For simpler storing of ROIs, we approximate the region here by a rectangle.

The output of VOCUS for one image is a list of ROIs, each defined by 2D location, size and a feature vector (see next section). The feature and conspicuity maps for one example image are displayed in Fig. 2, right.

Discussion on Feature Detection: The most common feature detectors for visual SLAM are corner-like features as SIFT keypoints [47] or Harris-Laplacian points [19]. These approaches are usually based on the idea that many features are extracted and a few of them show to be useful for tracking and matching.[2] Matching these features between frames to find stable ones, matching to existing landmarks, storing landmarks in the database, and matching current features to the database requires considerable time. With intelligent database management based on search trees, it is possible to store and access a large amount of features in real-time [8], [48], [49]. Nevertheless, solving the task equally well with less features is favorable and enables to use computational power and storage for other processes. To enable the system to use only few features, it is necessary to have a detector which computes discriminative features and is able to prioritize them.

We claim that an attention system is especially well suited to detect discriminative features and that the repeatability of salient regions is higher than the repeatability of non-salient regions and of features detected by standard detectors. The *repeatability* is defined as the percentage of regions which are redetected in a subsequent frame (cf. [23]). While an exhaustive analysis is beyond the scope of this paper, a

[2]We obtained in average 400 – 500 Harris-Laplace features per frame. Computing these features together with a SIFT descriptor required 250 ms per frame.

few experiments shall illustrate this.[3] The precondition for the following experiments is that one or a few object(s) or region(s) in the scene are salient (a *salient* region differs from the rest of the scene in at least one feature type).

In the experiment in Fig. 3, we compare an image sequence showing many white and one green object. For humans, the green object visually pops out of the scene, so it does for VOCUS. We compared the performance of VOCUS with two other detectors: Harris-Laplace corners and SIFT keypoints, i.e. extrema in DoG scale space, since these are the most commonly used detectors in visual SLAM scenarios.[4] To make the approaches comparable, we reduced the number of points by sorting them according to their response value and using only the points with the strongest response. We compared whether this response can be used to obtain a similar result as with salient regions.

We determined the repeatability of regions over 10 frames for different amounts of detected features.[5] The result of the comparison is shown in Fig. 3. The highest repeatability is naturally obtained for the most salient region: it is detected in each frame. The strongest Harris-Laplace feature and the strongest SIFT keypoint on the other hand are in a subsequent frame only detected at the same position in 20% of the images. We compared the repeatability up to 11 features per frame since this is the average number of features detected by the attention system in our experiments. It shows that the repeatability of attentional ROIs is consistently higher than the one of the other detectors. It remains to mention that the repeatability of Harris-Laplace features and SIFT points goes up when computing more features, repeatability rates of about 60% have been reported for Harris-Laplacians in [23]. Note that our point here is that with attentional ROIs it is possible to select very few discriminative features with high repeatability, which is not possible with the other, locally operating detectors.

To show that the results in these simple experiments also extend to longer image sequences and to more natural settings, some videos showing qualitative results can be found on http://www.informatik.uni-bonn.de/∼frintrop/research/saliency.html. While these experiments illustrate the advantages of salient regions for visual SLAM, more detailed experiments will be necessary to investigate the differences of the different detectors in different settings.

Another aspect to mention is the accuracy of the detectors. The Harris-Laplace detector is known to be very precise and to obtain sub-pixel accuracy. Attention regions on the other hand are not as precise, their position varies sometimes a few pixels from frame to frame. This is partially due to the segmentation process which determines the region. In previous work, we therefore combined Harris-Laplace corners and attention regions [13]. Tracking of landmarks with this approach was accurate and the matching process based on two descriptors resulted in a very low false detection rate. A problem however was that the detection rate also was very low: both detectors had to detect a feature in the same area and both descriptors had to agree on the high reliability of a match.

Using only attention regions with reasonable accuracy is possible with an improved outlier rejection mechanism during the triangulation process (cf. sec. V); this made the system considerably simpler and about 8 times faster.

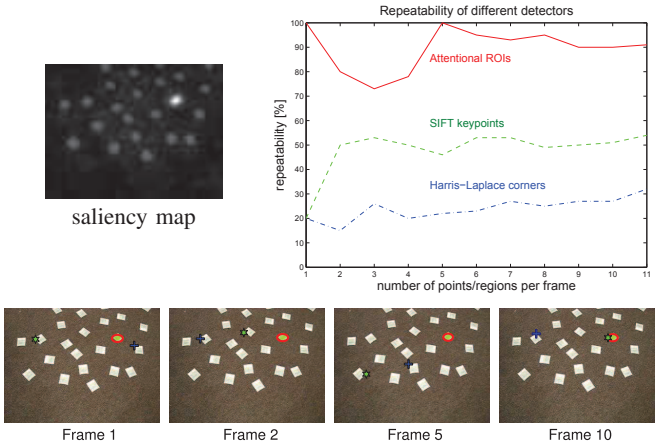


Fig. 3. Comparison of the repeatability of attentional ROIs (red ellipses), Harris-Laplace corners (blue crosses), and SIFT keypoints (green stars) on 10 frames of a sequence with a visually salient object (bottom: some example frames with detected features. top left: saliency map of 1st frame). The most salient attention region is detected in all frames (100% repeatability), the strongest point of the other detectors reaches only 20% (see also videos on http://www.informatik.uni-bonn.de/∼frintrop/research/saliency.html).

B. The Descriptors

To compare if two image regions belong to the same part in the world, each region has to have a description vector. The most simple vector is a vector consisting of the pixel values of the region and possibly some surrounding. The similarity of two vectors can then be computed by cross-correlation. However, this results in high-dimensional vectors and matching does not perform well under image transformations.

An evaluation of more powerful descriptors is provided in [50]. The best performance was obtained for the SIFT descriptor (scale invariant feature transform [47]) and the GLOH descriptor (gradient location-orientation histogram) – an extension of the SIFT descriptor. The SIFT descriptor is also probably the most used descriptor in visual tasks for mobile robots [51], [7], [8], [10].

In this work, we use two kinds of descriptors: first, we determine an attentional descriptor for tracking ROIs between consecutive frames. The attentional descriptor can be obtained almost without cost from the feature maps of VOCUS. Since it is only an 13-element vector, matching is faster than with the

[3]We did not compare the detectors on standard datasets as in [23] because these have been designed for tasks like object recognition and do not contain especially salient regions. Therefore, the advantages of salient regions cannot be shown there.

[4]We used the publically available PYRA real-time vision library for both detectors (http://www.csc.kth.se/∼celle/).

[5]For this comparison, VOCUS was adapted to compute all local maxima from the saliency map to make it comparable to the Harris detector. In normal usage it determines only regions which have a saliency of at least 50% of the most salient region.

SIFT descriptor. It is less powerful, but in tracking situations sufficient. Second, we use the SIFT descriptor to match ROIs in loop closing situations.

The *attentional descriptor* is determined from the values of the 10 feature and 3 conspicuity maps of VOCUS. For each ROI, a feature vector $\vec{v}$ with 13 entries is determined, which describes how much each feature contributes to the ROI (cf. Fig. 2). The value v_i for map X_i is the ratio of the mean saliency in the target region $m_{(ROI)}$ and in the background $m_{(image-ROI)}$:

$$v_i = m_{(ROI)}/m_{(image-ROI)}. \tag{3}$$

This computation does not only consider which features are the strongest in the target region but also which features separate the region best from the rest of the image (details in [12]).

The *SIFT descriptor* is a $4 \times 4 \times 8 = 128$ dimensional descriptor vector which results from placing a 4×4 grid on a point and calculating a pixel gradient magnitude at $45°$ intervals for each of the grid cells. Usually, SIFT descriptors are computed at intensity extrema in scale space [47] or at Harris-Laplacians [19]. Here, we calculate one SIFT descriptor for each ROI. The center of the ROI provides the position and the size of the ROI determines the size of the descriptor grid. The grid should be larger than the ROI to allow catching information about the surrounding but should also not include too much background and stay within the image borders.[6]

C. Feature Matching

Feature matching is performed in two of the visual SLAM modules: in the feature tracker and in the loop closer.

In the tracker, we apply simple matching based on attentional descriptors. Two vectors $\vec{v}$ and $\vec{w}$ are matched by calculating the similarity $d(\vec{v}, \vec{w})$ according to a distance similar to the Euclidean distance [13]. This simple matching is sufficient for the comparably easy matching task in tracking situations.

In the loop closer, SIFT matching is applied to achieve a higher matching stability. Usual approaches to perform matching based on SIFT descriptors are *threshold-based matching*, *nearest neighbor-based matching* and *nearest neighbor distance ratio matching* [50]. For each ROI in the image, we use threshold-based matching to find a fitting ROI in the database. Then, we apply nearest neighbor matching in the other direction to verify this match.[7]

The distance d_S of two SIFT descriptors is calculated as the sum of squared differences (SSD) of the descriptor vectors. Thresholding on the distance between two descriptors is a bit tricky. Small changes on the threshold might have unexpected effects on the detection quality since the dependence of distance and matching precision is not linear (cf. Fig. 4).

Therefore, we suggest a slightly modified thresholding approach. By learning the dependence of distance and matching precision from training data, it is possible to set directly a threshold for the precision from which the corresponding distance threshold is determined.

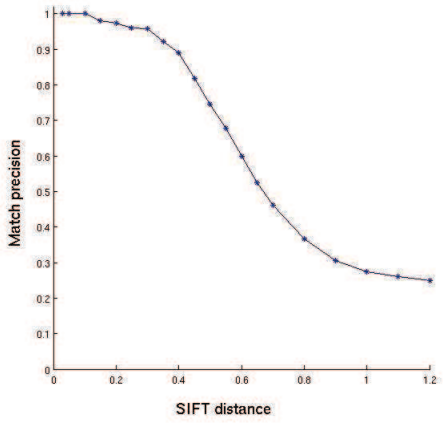


Fig. 4. The dependence of the distance of two SIFT descriptors and their matching precision (cf. eq. 4) determined from training data.

This is done as follows: for a large amount of image data, we gathered statistics regarding the distribution of correct and false matches. 698 correct matches and 2253 false matches were classified manually to obtain ground truth. We used data from two different environments, one was the office environment shown in Fig. 11, the other a different environment not used in the experiments. The training data for the office environment was obtained one year earlier than the test data for the current experiments.[8] Since the d_S are real values, we discretized the domain of d_S into $t = 20$ values. For the t distinct distance threshold values θ_j, we compute the *precision* as

$$p(\theta_j) = \frac{c(\theta_j)}{c(\theta_j) + f(\theta_j)}, \qquad \forall\, j \in \{1..t\} \tag{4}$$

where $c(\theta_j)$ and $f(\theta_j)$ denote the number of correct and false matches. The resulting distribution is displayed in Fig. 4.

To finally determine if two ROIs match, the distance of the SIFT descriptors is computed and the corresponding matching precision is determined according to the distribution in Fig. 4. If the precision is above a threshold, the ROIs match.[9]

Discussion on Feature Matching: The precision-based matching has several advantages over the usual thresholding. First, it is possible to choose an intuitive threshold like "98% matching precision".[10] Second, linear changes on the threshold result in linear changes on the matching precision. Finally,

[6]We chose a grid size of 1.5 times the maximum of width and height of the ROI.

[7]Mikolajczyk and Schmid show that the nearest neighbor and nearest neighbor distance ratio matching are more powerful than threshold-based matching but also point out that they are difficult to apply when searching in large databases [50].

[8]Correct matches are naturally much more difficult to obtain than false matches since there is a extremely large amount of possible false matches. To enable a reasonable amount of correct matches, we considered only distances below 1.2. As can be seen in Fig. 4, this does not affect the final matching mechanism as long as a precision of at least 0.3 is desired.

[9]For our system, we chose a threshold of 0.98. We chose a high threshold because an EKF SLAM system is sensitive to outliers.

[10]Note however that the precision value refers to the training data, so in test data the obtained precision might be lower than the specified threshold. However, the threshold gives a reasonable approximation of the precision on test data.

for every match a precision value is obtained. This value can be directly used by other components of the system to treat a match according to the precision that it is correct. For example, a SLAM subsystem which can deal with more uncertain associations could use these values.

The SIFT descriptor is currently one of the most powerful descriptors, however, people have worked on improving the performance, e.g. by combining it with other descriptors. While intuitively a good idea, we suggest to be careful with this approach. In previous work, we matched ROIs based on the attentional and the SIFT descriptor [14]. While obtaining good matching results, we found out that using only the SIFT descriptor results in a higher detection rate for the same amount of false detections. While surprising at first, this might be explained as follows: a region may be described by two descriptors, the perfect descriptor d_1 and the weaker descriptor d_2. d_1 detects all correct matches and rejects all possible false matches. Combining d_1 with d_2 cannot improve the process, it can only reduce the detection rate by rejecting correct matches.

V. The Feature Tracker

In the feature tracker, *landmarks* are built from ROIs by tracking the ROIs over several frames. The *length* of a landmark is the number of elements in the list, which is equivalent to the number of frames the ROI was detected in.

To compute the landmarks, we store the last n frames in a buffer (here: $n = 30$). This buffer enables to determine which landmarks are stable over time and therefore good candidates for the map. The output from the buffer is thus delayed by n frames but in return quality assessment can be utilized before using the data. New ROIs are matched with their attentional feature vector to previously detected landmarks and to ROIs from the previous frame to build new landmarks (details in [14]). At the end of the buffer, we consider the length of the resulting landmarks and filter out too short ones (here ≤ 3). Finally, the triangulator attempts to find an estimate for the location of the landmark. In this process, also outliers, i.e. bearings that fall far away from the estimated landmark location, are detected and removed from the landmark. These could be the result of mismatches or a poorly localized landmark.

VI. Loop Closing

In the loop closing module, it is detected if the robot has returned to an area where it has been before. This is essential to update the estimations of landmark and robot positions in the map. *Loop closing* is done by matching the ROIs from the current frame to landmarks from the database. It is possible to use position prediction of landmarks to determine which landmarks could be visible and thus prune the search space, but since this prediction is usually not precise when uncertainty grows after driving for a while, we perform "global loop closing" instead without using the SLAM pose estimate, as in [33]. That means, we match to all landmarks from the database. For the environments in our test it is possible to search the whole database in each iteration. However, for larger environments it would be necessary to use e.g. a tree-structure to organize the database, perform global loop closing less frequently or distribute the search over several iterations.

Fig. 6. Falsely matched ROIs (rectangles): in both cases, lamps are matched to a different lamp. Top: current frame. Bottom: frame from the database.

ROIs are matched to the landmarks from the database with the precision matching based on SIFT descriptors described in sec. IV-C. When a match is detected, the coordinates of the matched ROI in the current frame are provided to the SLAM system, to update the coordinates of the corresponding landmark. Additionally, the ROI is appended to the landmark in the database. Some examples of correct matches in loop closing situations are displayed in Fig. 5. False matches occur seldomly with this approach. If they do, the ROIs usually correspond to almost identical objects. Two examples are shown in Fig. 6.

VII. Active Gaze Control

The active gaze control module controls the camera according to three behaviours:

- Redetection of landmarks to close loops
- Tracking of landmarks
- Exploration of unknown areas

The strategy to decide which behaviour to choose is as follows: Redetection has the highest priority, but it is only chosen if there is an expected landmark in the possible field of view (def. see below). If there is no expected landmark for redetection, the *tracking* behaviour is activated. Tracking should only be performed if more landmarks are desired in this area. As soon as a certain amount of landmarks is obtained in the field of view, the *exploration* behaviour is activated. In this behaviour, the camera is moved to an area without landmarks. Most times, the system alternates between tracking and exploration, the redetection behaviour is only activated every once in a while (see sec. VII-A and Fig. 8). An overview over the decision process is displayed in Fig. 7. In the following, we describe the respective behaviours in more detail.

Fig. 5. Some examples of correctly matched ROIs, displayed as rectangles. Top: current frame. Bottom: frame from the database.

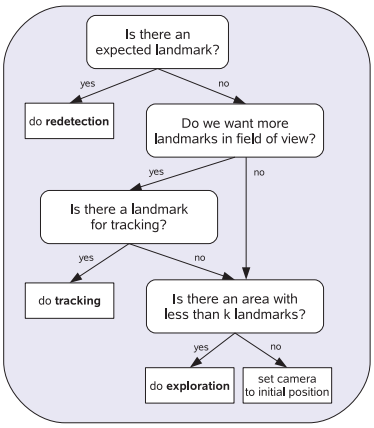


Fig. 7. The three camera behaviours *Redetection*, *Tracking*, *Exploration*.

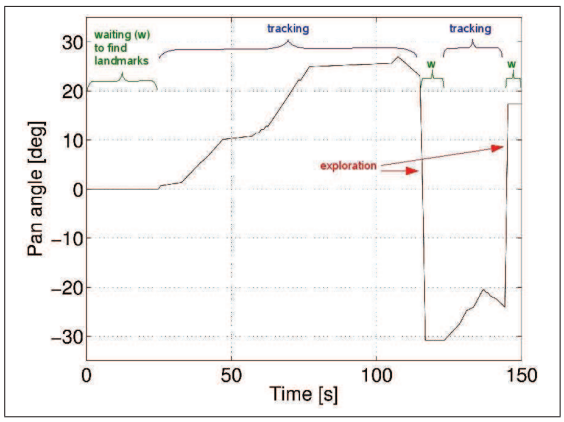


Fig. 8. The camera pan angle as a function of time. The camera behaviour alternates here between tracking and exploration.

A. *Redetection of Landmarks*

In redetection mode, the camera is directed to expected landmarks. *Expected landmarks*

(a) are in the potential field of view of the camera,[11]
(b) have low-enough uncertainty in the expected positions relative to the camera,[12]
(c) have not been seen recently, [13]
(d) had no matching attempt recently.

If there are several expected landmarks, the most promising one is chosen. Currently, we use a simple approach: the longest landmark is chosen because a landmark which has been observed frequently is more likely to be redetected than a seldomly observed one. In future work, we consider integrating information theory to choose the landmark that will result in the largest information gain, as e.g. in [44].

When a landmark has been chosen, the camera is moved to focus it and pointed there for several (here 8) frames, until it is matched. Note, that redetection and matching are two independent mechanisms: active redetection only controls the camera, matching is permanently done in the loop closer, also if there is no expected landmark.

If no match is found after 8 frames, the system blocks this landmark and chooses the next expected landmark or continues with tracking or exploration.

B. *Tracking of Landmarks*

Tracking a landmark means to follow it with the camera so that it stays longer within the field of view. This enables better triangulation results. This behaviour is activated if the preconditions for redetection do not apply.

First, one of the ROIs in the current frame has to be chosen for tracking. There are several aspects which make a landmark useful for tracking. First, the length of a landmark is an important factor for its usefulness since longer landmarks are more likely to be triangulated soon. Second, an important factor is the horizontal angle of the landmark: points in the direction of motion result in a very small baseline over several

[11]The potential field of view of the camera is set to $\pm 90°$ horizontally and $7m$ distance. This prevents considering landmarks which are too far away, since these are probably not visible although they are in the right direction.

[12]The uncertainty is considered as too high if it exceeds the image size, i.e. if the uncertainty of the landmark in pan-direction, projected to the image plane, is larger than the width of the image. Note, that these are actually the most useful landmarks to redetect, but on the other hand the matching is likely to fail. Passive matching attempts for these landmarks are permanently done in the loop closer, only the active redetection is prevented.

[13]The redetection behaviour focuses on landmarks which have not been visible for a while (here: 30 frames) to prevent switching the camera position constantly. The longer a landmark had not been visible, the more useful is usually its redetection.

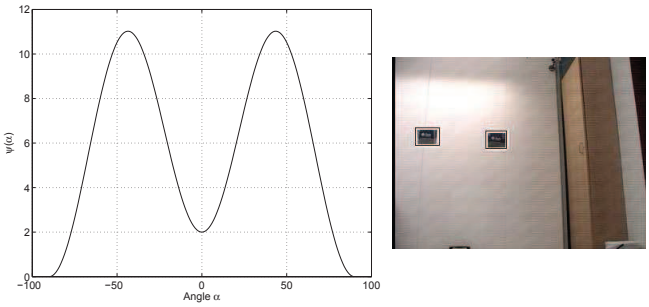


Fig. 9. Left: function $\psi(\alpha)$ with $k_1 = 5$ and $k_2 = 1$. Right: One test image with two (almost) identical ROIs, differing only by their position in the image. The center ROI has the angle $\alpha_1 = 0.04$ resulting in $\psi(\alpha_1) = 2.06$. The left ROI has a larger angle $\alpha_2 = 0.3$ resulting in $\psi(\alpha_2) = 5.09$ ($> \psi(\alpha_1)$). The tracking behaviour selects the left ROI for tracking and prevents it from moving out of the image.

frames and hence often in poor triangulations. Points at the side usually give much better triangulation results, but on the other hand they are more likely to move outside the image borders soon so that tracking is lost.

We define a usefulness function capturing the length l of the landmark and the angle α of the landmark in the potential field of view as

$$U(L) = \psi(\alpha)\sqrt{l} \tag{5}$$

where

$$\psi(\alpha) = k_1\,(1.0 + \cos(4\alpha - 180)) + k2\,(1.0 + \cos(2\alpha)). \tag{6}$$

The function is displayed in Fig. 9, left, and an example is shown in Fig. 9, right. Like in redetection mode, integrating the information gain could improve this estimation. After determining the most useful landmark for tracking, the camera is directed into the direction of the landmark.[14] The tracking stops when the landmark is not visible any more or when it was successfully triangulated.

C. Exploration of Unknown Areas

As soon as there are enough (here more than 5) landmarks in the field of view, the exploration behaviour is started, i.e., the camera is directed to an area within the possible field of view without landmarks. We favor regions with no landmarks over regions with few landmarks since few landmarks are a hint that we already looked there and did not find more landmarks.

We look for a region which corresponds to the size of the field of view. If the camera is currently pointing to the right, we start by investigating the field directly on the left of the camera and vice versa. We continue the search in that direction, in steps corresponding to the field of view. If there is no landmark, the camera is moved there. Otherwise we switch to the opposite side and investigate the regions there. If no area without landmarks is found, the camera is set to the initial position.

[14]The camera is moved slowly (here 0.1 radians per step), since this changes the appearance of the ROI less than large camera movements. This results in a higher matching rate and prevents to loose other currently visible landmarks.

To enable building of landmarks over several frames, we let the camera focus one region for a while (here 10 frames). As soon as a landmark for tracking is found, the system will automatically switch behaviour and start tracking it (cf. Fig. 8).

VIII. Experiments and Results

We tested the system in two different environments: an office environment and an atrium area at the Royal Institute of Technology (KTH) in Stockholm. In both environments, several test runs were performed, some at day, some at night to test differing lighting conditions. Test runs were performed during normal work days, therefore they include normal occlusions like people moving around. The matching examples in Fig. 5 show that loop closing is possible anyway.

For each run, the same parameter set was used. During each test run, between 1200 and 1800 images with 320×240 pixels were processed. In the office environment, the robot drove the same loop several times. This has the advantage that there are many occasions in which loop closing can take place. Therefore, this is a good setting to investigate the matching capability of the system. On the other hand, the advantage of the active camera control is not obvious here since loop closing is already easy in passive mode. To test the advantages of the active camera mode, the atrium sequence fits especially well. Here, the robot drove an "eight", making loop closing difficult in passive mode because the robot approaches the same area from three different directions. Active camera motion makes it possible to close the loop even in such difficult settings.

The current system allows real-time performance. Currently, it runs on average at ~ 90 ms/frame on a Pentium IV 2 GHz machine. Since the code is not yet optimized, a higher frame rate should be easily achievable by standard optimizations. Although VOCUS is relatively fast with ~ 50 ms/frame since it is based on integral images [29], this part requires about half of the processing time. If a faster system is required, a GPU implementation of VOCUS is possible, as realized in [52].

The experiments section has two parts. First, we investigate the quality of the attentional landmarks. Second, we compare active and passive camera control.

A. Visual SLAM with Attentional Landmarks

In this section, we investigate the quality of landmark detection, of data association in loop closing situations, and the effect on the resulting maps and robot trajectories. We show that we obtain a high performance with a low number of landmarks. Loop closing is obtained easily even if only few landmarks are visible and if they are seen from very different viewpoints.

In the first experiment, the same trajectory was driven three times in the office environment. Fig. 10 shows the robot trajectory which was determined from pure odometry (left) and from the SLAM process (right). Although the environment is small compared to other scenarios of the literature, it is well visible that the odometry estimation becomes wrong quickly. The estimated end position differs considerably from the real end position. The SLAM estimate on the other hand (right), is much more accurate. During this run, the robot acquired 17

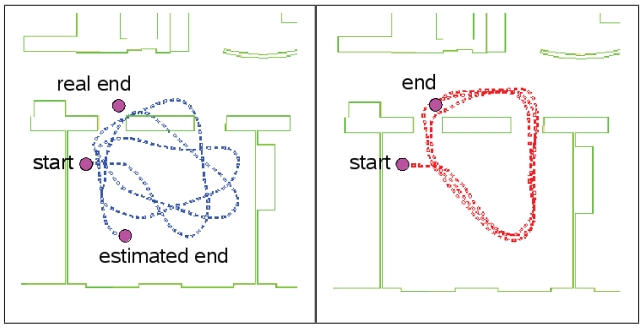


Fig. 10. A test run in the office environment. The robot trajectory was estimated once from only odometry (left) and once from the SLAM system (right).

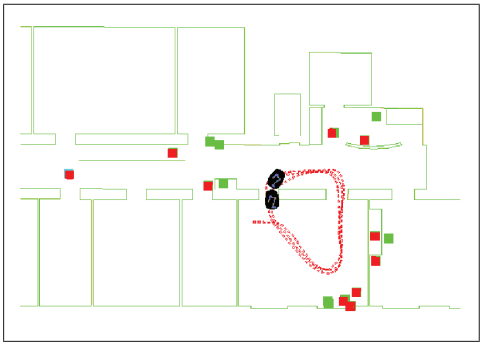


Fig. 11. Estimated robot trajectory with final robot position (the "first" robot is the real robot, whereas the robot behind visualizes the robot position at the end of the buffer. The latter is used for trajectory and landmark estimation). Green dots are landmarks, red dots are landmarks which were redetected in loop-closing situations.

environment	camera control	# landmarks	# correct matches	# false matches
office	passive	17	21	0
office	active	36	31	2
office	passive	18	23	1
office	passive	21	21	0
office	active	34	16	1
atrium	active	57	14	1
atrium	active	61	15	3
atrium	active	50	8	2
atrium	passive	19	1	1

TABLE I

MATCHING QUALITY FOR DIFFERENT TEST RUNS IN TWO ENVIRONMENTS. 2ND COLUMN: PASSIVE/ACTIVE CAMERA CONTROL. 3RD COLUMN: THE NUMBER OF MAPPED LANDMARKS. 4TH/5TH COLUMN: THE NUMBER OF TIMES A CURRENT LANDMARK WAS MATCHED TO AN ENTRY IN THE DATABASE. MATCHES ARE ONLY COUNTED, IF THE CORRESPONDING LANDMARK HAD NOT BEEN SEEN FOR AT LEAST 30 FRAMES. NOTE THAT A LANDMARK CAN ALSO BE MATCHED SEVERAL TIMES.

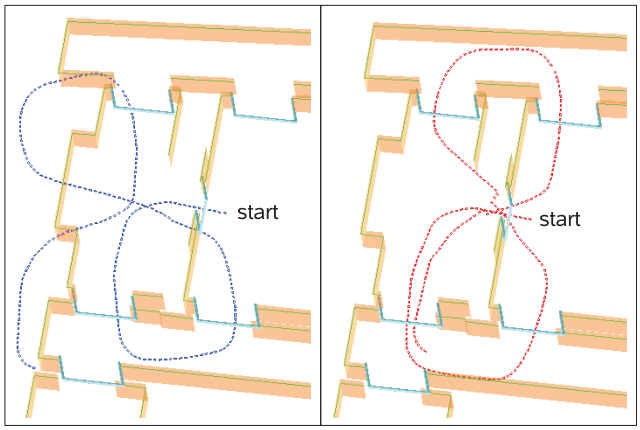


Fig. 12. A test run in the atrium area. The robot trajectory was estimated once from only odometry (left) and once from the SLAM system (right).

landmarks, found 21 matches to the database (one landmark can be detected several times) and all of the matches were correct (cf. Tab. I, row 1). The estimated landmark positions and the matches are displayed in Fig. 11. Notice that more than half of the landmarks are redetected when revisiting an area. More results from the office environment are shown in row 2–5 of Tab. I. The three occurring false matches belong always to the same object in the world: the lamp in Fig. 6 left.

More experiments were performed in the atrium environment. A comparison between the estimated robot trajectory from odometry data and from the SLAM system is visualized in Fig. 12. In this example, the system operated in active camera mode (cf. sec. VIII-B). Also here, the big difference in accuracy of the robot trajectory is visible. The corresponding number of landmark detections and matches is shown in Tab. I, row 6. Results from additional runs are shown in rows 7-9. Note that the percentage of matches with respect to the number of all landmarks is smaller in the atrium area than in the office environment since a loop can be only closed at a few places. Also in this environment, all the false matches belong to identical lamps (cf. Fig. 6 right).

In the presented examples, the few false matches did not lead to problems, the trajectory was estimated correctly anyway. Only the falsely matched landmarks are assigned a wrong position. But note that more false matches might cause problems for the SLAM process. The detection quality could be improved by collecting evidence for a match from several landmarks.

B. *Passive versus Active Camera Control*

In this section, we compare the passive and the active camera mode of the visual SLAM system. We show that with active camera control, more landmarks are mapped with a better distribution in the environment, more database matches are obtained, and that loop closing occurs earlier and even in situations where no loop closing is possible in passive mode.

From Tab. I, it can be seen that the test runs with active camera control result in more mapped landmarks than the runs with passive camera. Although this is not necessarily an advantage — we claim actually that the sparseness of the map is an advantage — it is favorable if the larger number results from a better distribution of landmarks. That this is the case here can be seen e.g. in the example in Fig. 13: landmarks show up in active mode (right), where there are no landmarks in passive mode (left).

Loop closing occurs usually earlier in active mode. For example in Fig. 11, the robot is already able to close the loop when it enters the doorway (position of front robot in figure)

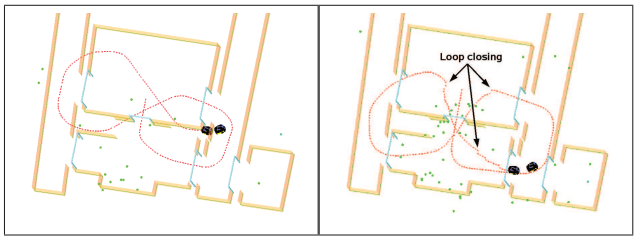


Fig. 13. Atrium environment: the estimated robot trajectory in passive (left, cf. Tab. I row 9) and active (right, cf. Tab. I row 8) camera mode (the 1st robot is the real robot, the 2nd a virtual robot at the end of the buffer). Landmarks are displayed as green dots. In passive mode, the robot is not able to close the loop. In active mode, loop closing is clearly visible and results in an accurate pose estimation (see also videos on http://www.informatik.uni-bonn.de/∼frintrop/research/aslam.html).

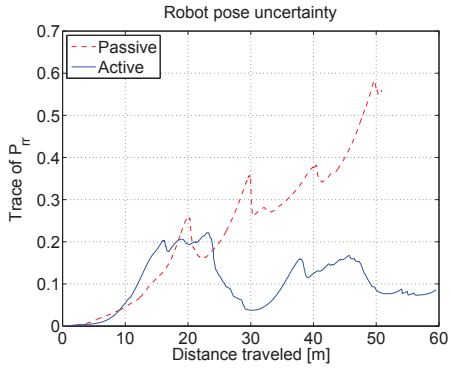


Fig. 14. The robot pose uncertainty computed as the trace of P_{rr} (covariance of robot pose) for passive and active camera mode.

by directing the camera to the landmark area on its left. In passive mode, loop closing only occurs when the robot itself moved to face this area. An earlier loop closing leads to an earlier correction of measurements and provides time to earlier go back to other behaviours like exploration.

In active mode, the robot closed a loop several times in the atrium. This is visible from the small jumps in the estimated trajectory in Fig. 13 right. The final pose estimate is much more accurate here than in passive mode. Fig. 14 displays a comparison of the robot pose uncertainty in passive and active mode, computed as the trace of P_{rr} (covariance of robot pose). The two loop closing situations in active mode around meter 30 and 50 reduce the pose uncertainty considerably, resulting at the end of the sequence in a value which is much lower than the uncertainty in passive mode.

IX. Discussion and Conclusion

In this paper, we have presented a complete visual SLAM system, which includes feature detection, tracking, loop closing and active camera control. Landmarks are selected based on biological mechanisms which favor salient regions, an approach which enables focusing on a sparse landmark representation. We have shown that the repeatability of salient regions is considerably higher than the one of regions from standard detectors. Additionally, we presented a precision-based matching strategy, which enables to intuitively choose a matching threshold to obtain a preferred matching precision.

The active gaze control module presented here enabled to obtain a better distribution of landmarks in the map and to re-detect considerably more landmarks in loop closing situations than in passive camera mode. In some cases, loop closing is actually only possible by actively controlling the camera.

While we obtain a good pose estimation and a high matching rate, further improvements are always possible and planned for future work. For example, we plan to collect evidence for a match from several landmarks together with their spatial organization as already done in other systems. Also determining the salience of a landmark not only in the image but in the whole environment would help to focus on even more discriminative landmarks. Using the precision value of a match could be very helpful to improve the system performance too. Adapting the system to deal with really large environments could be achieved by removing landmarks which are not redetected to keep the number of landmarks low, by database management based on search trees, indexing [53], [49], and by using hierarchical maps as in [11]. Also testing the system in outdoor environments is an interesting challenge for future work.

Acknowledgment

The present research has been sponsored by SSF through the Centre for Autonomous Systems, VR (621-20 06-4520), the EU project “CoSy” (FP6-004150-IP), and the university of Bonn through Prof. A. B. Cremers. This support is gratefully acknowledged. We also want to thank Mårten Björkman for providing the PYRA real-time vision library.

References

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 229–241, 2001.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, 2002.

[3] J. Folkesson and H. Christensen, “Graphical SLAM - a self-correcting map,” in *Proc. of the IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2004.

[4] F. Dellaert, “Square root SLAM: Simultaneous location and mapping via square root information smoothing,” in *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[5] U. Frese and L. Schröder, “Closing a million-landmarks loop,” in *Proc. of the IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[6] A. Davison and D. Murray, “Simultaneous localisation and map-building using active vision,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 7, pp. 865–880, 2002.

[7] L. Goncavles, E. di Bernardo, D. Benson, M. Svedman, J. Ostrovski, N. Karlsson, and P. Pirjanian, “A visual front-end for simultaneous localization and mapping,” in *Proc. of the Int’l Conf. on Robotics and Automation, (ICRA)*, 2005.

[8] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, “A framework for vision based bearing only 3D SLAM,” in *Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2006.

[9] K. Ho and P. Newman, “Detecting loop closure with scene sequences,” *Int’l J. of Computer Vision and Int’l J. of Robotics Research. Joint issue on computer vision and robotics*, 2007.

[10] M. Cummins and P. Newman, “Probabilistic appearance based navigation and loop closing,” in *Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2007.

[11] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardos, “Mapping large loops with a single hand-held camera,” in *Proc. of Robotics: Science and Systems (RSS)*, 2007.

[12] S. Frintrop, "VOCUS: A visual attention system for object detection and goal-directed search," Ph.D. dissertation, Universität Bonn, Germany, 2005, ser. LNAI. Springer, 2006, vol. 3899.

[13] S. Frintrop, P. Jensfelt, and H. Christensen, "Simultaneous robot localization and mapping based on a visual attention system," in *Attention in Cognitive Systems*, ser. LNAI. Springer, 2007, vol. 4840.

[14] S. Frintrop and P. Jensfelt, "Active gaze control for attentional visual SLAM," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2008.

[15] P. Zhang, E. E. Milios, and J. Gu, "Underwater robot localization using artificial visual landmarks," in *Proc. of IEEE Int'l Conf. on Robotics and Biomimetics*, 2004.

[16] U. Frese, "Treemap: An O($log\, n$) algorithm for indoor simultaneous localization and mapping," *Autonomus Robots*, vol. 21, no. 2, pp. 103–122, 2006.

[17] F. Launay, A. Ohya, and S. Yuta, "A corridors lights based navigation system including path definition using a topologically corrected map for indoor mobile robots," in *Int'l Conf. on Robotics and Automation (ICRA)*, 2002.

[18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[19] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2001.

[20] J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1994.

[21] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2005.

[22] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. of the British Machine Vision Conference (BMVC)*, 2002.

[23] K. Mikolajczyk and C. Schmid, "A comparison of affine region detectors," *Int'l J. of Computer Vision (IJCV)*, vol. 65, no. 1-2, pp. 43–72, 2006.

[24] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 20, no. 11, pp. 1254–1259, 1998.

[25] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 507–545, 1995.

[26] A. M. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, pp. 97–136, 1980.

[27] J. M. Wolfe, "Guided search 2.0: A revised model of visual search," *Psychonomic Bulletin and Review*, vol. 1, no. 2, pp. 202–238, 1994.

[28] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews*, vol. 3, no. 3, pp. 201–215, 2002.

[29] S. Frintrop, M. Klodt, and E. Rome, "A real-time visual attention system using integral images," in *Proc. of the Int'l Conf. on Computer Vision Systems (ICVS)*, 2007.

[30] S. B. Nickerson, P. Jasiobedzki, D. Wilkes, M. Jenkin, E. Milios, J. K. Tsotsos, A. Jepson, and O. N. Bains, "The ARK project: Autonomous mobile robots for known industrial environments," *Robotics and Autonomous Systems*, vol. 25, no. 1-2, pp. 83–104, 1998.

[31] N. Ouerhani, A. Bur, and H. Hügli, "Visual attention-based robot self-localization," in *Proc. of Europ. Conf. on Mobile Robotics (ECMR)*, 2005.

[32] C. Siagian and L. Itti, "Biologically-inspired robotics vision monte-carlo localization in the outdoor environment," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[33] P. Newman and K. Ho, "SLAM- loop closing with visually salient features," in *Proc. of the Int'l Conf. on Robotics and Automation (ICRA)*, 2005.

[34] R. Bajcsy, "Active perception vs. passive perception," in *Proc. of Workshop on Computer Vision: Representation and Control*. IEEE Press, 1985.

[35] Y. Aloimonos, I. Weiss, and A. Bandopadhay, "Active vision," *Int'l J. of Computer Vision (IJCV)*, vol. 1, no. 4, pp. 333–356, 1988.

[36] R. Bajcsy, "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 996–1005, 1988.

[37] B. Grocholsky, H. F. Durrant-Whyte, and P. Gibbens, "An information-theoretic approach to decentralized control of multiple autonomous flight vehicles," in *Sensor Fusion and Decentralized Control in Robotic Systems III*, 2000.

[38] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 15, no. 5, pp. 417–433, 1993.

[39] R. Sim and J. J. Little, "Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters," in *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[40] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *In Proc. of the IEEE Int'l Symp. on Computational Intelligence in Robotics and Automation*, 1997.

[41] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, "An experiment in integrated exploration," in *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[42] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.

[43] T. Arbel and F. P. Ferrie, "Entropy-based gaze planning," in *Proc. of IEEE Workshop on Perception for Mobile Agents*, 1999.

[44] T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D. W. Murray, "Active control for single camera SLAM," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2006.

[45] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proc. of Robotics: Science and Systems (RSS)*, 2006.

[46] S. Frintrop and A. B. Cremers, "Top-down attention supports visual loop closing," in *Proc. of European Conference on Mobile Robotics (ECMR)*, 2007.

[47] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[48] S. Obdrzalek and J. Matas, "Sub-linear indexing for large scale object recognition," in *Proc. of the British Machine Vision Conference (BMVC)*, 2005.

[49] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[50] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. of Pattern Analysis and Machine intelligence (PAMI)*, vol. 27, no. 10, pp. 1615–1630, 2005.

[51] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int'l J. of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[52] S. May, M. Klodt, E. Rome, and R. Breithaupt, "GPU-accelerated affordance cueing based on visual attention," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[53] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. of the Int'l Conf. on Computer Vision*, 2003.

Simone Frintrop Simone Frintop got her Ph.D. from the University of Bonn, 2005. She was a postdoctoral researcher at the Computer Vision and Active Perception lab (CVAP) at the School of Computer Science and Communications (CSC) at the Royal Institue of Technology (KTH), Stockholm, Sweden until 2006. She now works in the Intelligent Vision Systems Group at the Institute of Computer Science III, University of Bonn, Germany, where she is currently a Senior Scientific Assistant.

Patric Jensfelt Patric Jensfelt received his M.Sc. in Engineering Physics in 1996 and Ph.D. in Automatic Control in 2001, from the Royal Institute of Technology, Stockholm, Sweden. Between 2002 and 2004 he worked as a project leader in two industrial projects. He is currently an assistant professor with the Centre for Autonomous System (CAS) and the principal investigator of the European project CogX at CAS. His research interests include mapping and localiation and systems integration.

Publication [12]

Simone Frintrop and Armin B. Cremers. Top-down attention supports visual loop closing. In *Proceedings of European Conference on Mobile Robotics (ECMR)*, Freiburg, Germany, 2007.

Top-down Attention Supports Visual Loop Closing

Simone Frintrop Armin B. Cremers
Institute of Computer Science III,
Rheinische Friedrich-Wilhelms-Universität Bonn, Germany

***Abstract*— In this paper, we present a method to improve the loop closing behaviour for visual SLAM. Landmarks consist of a combination of attention regions and Harris-Laplace corners. The attention regions are detected by a visual attention system which combines image-based, bottom-up and target-related, top-down information. The ability to perform target-directed search is used to search for expected landmarks.**

We analyze the amount of correct and false matches for bottom-up and top-down matching depending on different matching thresholds. It shows that whereas bottom-up matching is useful for situations in which the scene changes only slightly like during tracking, top-down matching has advantages in loop closing situations by detecting a much higher amount of correctly matched landmarks.

***Index Terms*— Visual SLAM, loop closing, saliency, visual attention**

I. Introduction

An essential task of mobile robots which explore unknown environments is *SLAM (Simultaneous localization and mapping)*, the task of building a map and staying localized within it at the same time [3, 4, 18]. Special interest during the last years has been on *visual SLAM*, which uses cameras as main sensors [1, 10, 12, 17]. In contrast to laser-scanners, cameras are low-cost, low-power, and lightweight sensors which may be used in many applications where laser scanners are too expensive or too heavy. Additionally, the rich visual information of camera images holds potential for better data association and more accurate 3D representations of the environment. Challenges in this field are the high amount of data which requires intelligent landmark selection strategies and the sensitivity of image data to illumination and viewpoint changes which requires robust tracking and matching methods. Additionally, when performing *bearing-only SLAM* with a single camera, depth estimation is difficult because it has to be estimated by triangulation from several frames.

One of the most challenging problems in SLAM is the *data association*, the task of associating current observations with map elements. In visual SLAM, this means to match currently detected visual landmarks to landmarks from a database. For consecutive frames, this problem is relatively easy, especially if additional odometry information is used, since usually images change only slightly between frames and since the odometry provides the system with rather accurate position estimates. The problem becomes much more difficult when the robot revisits a location after some time. This *loop closing* has to deal with illumination variations and viewpoint changes, and since the odometry estimation is much less accurate, large areas have to be considered for matching.

The choice of the feature detector is important to obtain useful landmarks which are on the one hand robust and easy to redetect and which have, on the other hand, high positional stability to obtain precise depth estimations when triangulating. Often, the landmarks are selected by a human expert or the kind of landmark is determined in advance, e.g., ceiling lights [17], artificial landmarks [2], Harris corners [12], SIFT features [13], or maximally stable extremal regions (MSERs) [15]. As pointed out by [19], there is a need for methods which enable a robot to choose landmarks autonomously. A good method should pick the landmarks which are most suitable for the current situation. An especially useful method to find landmarks autonomously depending on the current surrounding are visual attention systems [20, 11, 5]. They select regions that "pop out" in a scene due to strong contrasts and uniqueness. The advantage of these methods is that they determine globally which regions in the image are discriminative instead of locally detecting predefined properties. In previous work, we have shown that a combination of attention regions with Harris-Laplace corners is especially useful to obtain both, positional stability and good discrimination for loop closing [7, 6].

In this paper, we focus on an improvement of the loop closing module of our visual SLAM system. All approaches we are aware of match landmarks in a bottom-up manner, i.e., the same feature detection methods are applied to two frames and the detected features are compared afterwards [16, 15, 12, 8]. In contrast to this, we change the feature computations depending on the kind of landmarks we currently expect: we use the ability of the attention system to search in a top-down, target-directed manner for expected landmarks by explicitly supporting expected features. Information about which landmarks are expected is provided by the SLAM module, based on the estimate robot pose and the map.

We compare in real-world experiments the new top-down matching with the conventional bottom-up matching. It turns out that whereas the bottom-up matching shows advantages in easy matching situations like tracking, the top-down matching outperforms the bottom-up matching clearly in difficult matching situations with changing viewpoints. Therefore, the new method is more useful for loop-closing situations.

In the following, we first give an overview over the whole visual SLAM system (sec. II). Then, we describe the feature detection (sec. III), the feature matching (sec. IV), the feature tracking (sec. V), and the loop closing (sec. VI). Finally, we present several experiments on real-world data in sec. VII before we conclude (sec. VIII).

II. System Overview

The visual SLAM architecture is displayed in Fig. 1. The main components are a *robot* which provides camera images

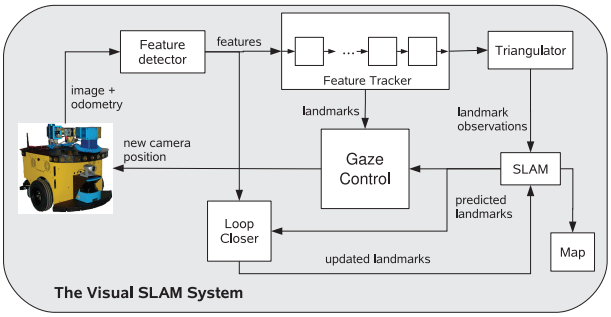


Fig. 1. The visual SLAM system

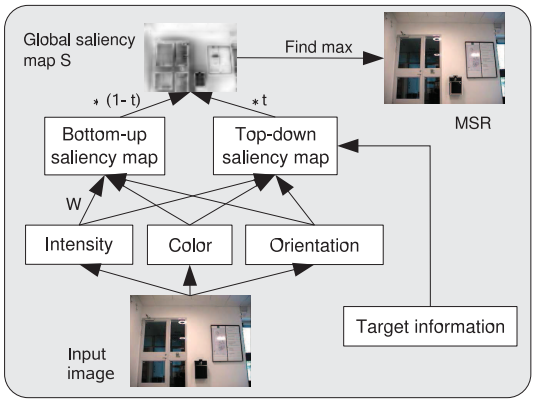


Fig. 2. ROI detection: The visual attention system VOCUS.

and odometry information, a *feature detector* which finds regions of interest (ROIs) in the images, a *feature tracker* which tracks ROIs over several frames and builds landmarks, a *triangulator* which identifies useful landmarks, a *SLAM module* which builds a map of the environment, a *loop closer* which matches current ROIs to the database and, as main part of the current paper, a *gaze control module* which determines where to direct the camera to.

When a new frame from the camera is available, it is provided to the *feature detector*, which finds ROIs based on a visual attention system and Harris-Laplace corners inside the ROIs. Next, the features are provided to the *feature tracker* which stores the last n frames, performs matching of ROIs and Harris corners in these frames and creates landmarks. The purpose of this buffer is to identify features which are stable over several frames and have enough parallax information for 3D initialization. These computations are performed by the *triangulator*. Selected landmarks are stored in a database and provided to the SLAM module which computes an estimate of the position of landmarks and integrates the position estimate into the map. Details about the robot and the SLAM architecture can be found in [12].

The task of the *loop closer* is to detect if a scene has been seen before. Therefore, the features from the current frame are compared with the features from the landmarks in the database. To narrow down the search space, the SLAM module provides the loop closer with expected landmark positions. Only landmarks that should be currently visible are considered for matching.

Finally, the *gaze control module* actively controls the camera. It decides whether to track currently seen landmarks, to actively look for predicted landmarks, or to explore unseen areas. It computes a new camera position which is provided to the robot. Details on this module can be found in [8].

III. The Feature Detector

The feature selection is based on two different kinds of features: attentional ROIs and Harris-Laplace corners. In [7] we have shown that this combination is useful, since it combines the advantages of both approaches: the attentional ROIs focus the processing on salient image regions which are thereby well redetectable. The corners on the other hand provide well localized points as required for precise depth estimation for structure from motion with a small baseline. Additionally, the combination improves the matching of landmarks (cf. sec. IV).

A. ROI Detection

The ROIs are detected with the attention system VOCUS (Visual Object detection with a CompUtational attention System) [5] (Fig. 2). It consists of a bottom-up part similar to [11], and a top-down part enabling goal-directed search; global saliency is determined from both cues.

1) Bottom-up computations: The bottom-up part detects salient image regions by computing image contrasts and uniqueness of a feature. The feature computations for the features intensity, orientation, and color are performed on 3 different scales with image pyramids. The feature intensity is computed by *center-surround mechanisms*; on-off and off-on contrasts are computed separately. After summing up the scales, this yields 2 intensity maps. Similarly, 4 orientation maps $(0\,^\circ, 45\,^\circ, 90\,^\circ, 135\,^\circ)$ are computed by Gabor filters and 4 color maps (green, blue, red, yellow) which highlight salient regions of a certain color. Each feature map i is weighted with a uniqueness weight $\mathcal{W}(i) = i/\sqrt{m}$, where m is the number of local maxima that exceed a threshold. This promotes pop-out features. The maps are summed up to 3 conspicuity maps I (intensity), O (orientation) and C (color) and combined to form the *bottom-up saliency map* $S_{bu} = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C)$. Details on the feature computations in [5].

To achieve real-time performance, the feature computations in VOCUS are efficiently performed on *integral images* [21]. After once creating an integral image in linear time with respect to the number of pixels, a rectangular feature value of arbitrary size is computed with only 4 references. This results in a fast computation (50ms for a 400×300 pixel image, 2.8GHz) that enables real-time performance (details in [9]).

If no top-down information is available, S_{bu} corresponds to the global saliency map S. In S, the *most salient regions (MSRs)* are determined: first the local maxima (seeds) in S are found and second all neighboring pixels over a saliency threshold (here: 25% of the seed) are detected recursively with *region growing*. A ROI is defined as the smallest rectangle including the MSR. It is an approximation, to allow easier storing of features.

For each *MSR*, a bottom-up feature vector $\vec{v_{bu}}$ with $(2 + 4 + 4 + 3 = 13)$ entries (one for each feature and conspicuity map) is determined. The feature value v_i for map i is the ratio

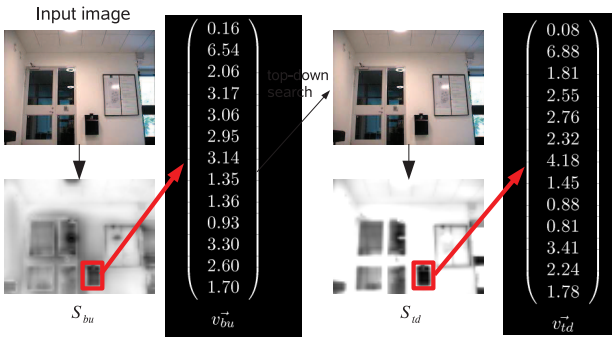


Fig. 3. Procedure to create a top-down vector $v_{td}^{\rightarrow}$: First, the bottom-up saliency Map S_{bu} is created from the input image. Then, for each MSR in S_{bu} the corresponding bottom-up vector $v_{bu}^{\rightarrow}$ is created. This vector is used to apply top-down search to the input image, yielding in a top-down saliency map S_{td}. The feature vector describing the corresponding MSR in S_{td} is the vector $v_{td}^{\rightarrow}$. The values in the vectors stand for the feature maps intensity on-off, intensity off-on, orientations $0\,^{\circ}, 45\,^{\circ}, 90\,^{\circ}, 135\,^{\circ}$, colors green, blue, red, yellow and for the conspicuity maps I, O, C.

of the mean saliency in the target region $m_{(MSR)}$ and in the background $m_{(image-MSR)}$: $v_i = m_{(MSR)}/m_{(image-MSR)}$. This computation does not only consider which features are the strongest in the target region, it also regards which features separate the region best from the rest of the image. Fig. 3 shows the feature vector $v_{bu}^{\rightarrow}$ which corresponds to the wastebin. It tells us, e.g., that the region is dark on a bright background, since the highest value is the 2nd value of the vector, which represents the off-on intensity.

2) Top-down computations: In top-down mode, VOCUS aims to detect a target, i.e., input to the system is the image and some target information, provided as feature vector $\vec{v}$. In *search mode*, VOCUS multiplies the feature and conspicuity maps with the corresponding weights of $\vec{v}$. The resulting maps are summed up, yielding the *top-down saliency map* S_{td}. Finally, S_{bu} and S_{td} are combined by: $S = (1-t)*S_{bu}+t*S_{td}$, where t determines the contributions of bottom-up and top-down (details in [5]). Here, we use $t = 0$ for bottom-up and $t = 1$ for top-down computations.

Fig. 3 shows a bottom-up and a top-down saliency map: the bottom-up saliency map highlights all regions which might be of interest, regardless of a certain target. The top-down map highlights especially the target region (the black wastebin) and suppresses regions which do not look similar.

If the similarity of two ROIs shall be compared (see sec. IV), we cannot compare a top-down ROI with a bottom-up ROI because the feature values result from different computations. Instead, we additionally compute a top-down vector $v_{td}^{\rightarrow}$ for each bottom-up ROI. This is done by using the bottom-up vector $v_{bu}^{\rightarrow}$ as target information and search for this region within the same image. This results in a top-down saliency map in which the top-down MSR within the target region, defined by the bottom-up ROI, is determined. Fig. 3 shows the procedure to create such a top-down vector $v_{td}^{\rightarrow}$.

B. Harris-Laplace detector:

To detect features with high position stability inside the ROIs, we used the Harris-Laplace feature detector [14] – an extension of the Harris corner detector to Laplacian pyramids which enables scale invariance. For convenience, we talk briefly about Harris corners in the following. The method finds a few (av. 1.6) points per ROI. To allow matching of points, a SIFT descriptor is computed for each detected corner [13].

IV. Feature Matching

Feature matching is performed between consecutive frames (in the feature tracker) and with features from the database (in the loop closer). The general matching procedure is the same in both modules. It is based on two criteria: proximity and similarity. First, the features in the new frame have to be close enough to the predicted position. Second, the similarity of the features is determined. This is done differently for attentional ROIs and for Harris corners: the matching of Harris corners is based on the SIFT descriptor by determining the Euclidean distance between the descriptors. When the distance is below a threshold, the points match.

For the attentional ROIs, we consider the size of the ROIs and the similarity of the feature values. We set the allowed deviation in width and height of the ROI to 10 pixels to allow some variations. This is required, because the ROIs might differ slightly in shape depending on image noise and illumination variations.

The similarity of two feature vectors $\vec{v}$ and $\vec{w}$ is determined by eq. 1; the smaller the distance $d(\vec{v}, \vec{w})$, the higher the similarity of the ROIs. If $d(\vec{v}, \vec{w})$ is below a certain threshold δ, the ROIs match (see sec. VII for the choice of δ). The computation is similar to the Euclidean distance of the vectors, but it treats the feature map values (v_1,..,v_{10}) differently than the conspicuity map values (v_{11},..., v_{13}). The reason is as follows: the conspicuity values provide information about how important the respective feature maps are. For example, a low value for the color conspicuity map v_{13} means the values of the color feature maps (v_7,...,v_{10}) are not discriminative and should be assigned less weight than the other values. Therefore, we use the conspicuity values to weight the feature values. We found out that this matching procedure outperforms the simple Euclidean distance of the feature vectors.

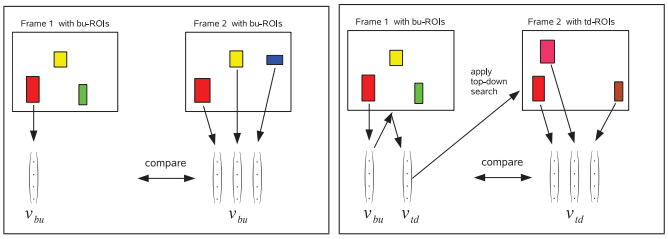
We distinguish two matching approaches: *bottom-up* and *top-down matching*. They differ in the kind of vectors which are used to determine the similarity. We describe both in the following.

A. Bottom-up matching

For bottom-up matching, each ROI from a frame f_1 is compared to each ROI from a frame f_2. If the matching distance $d(\vec{v}, \vec{w})$ for the vectors $\vec{v}$ and $\vec{w}$ of two ROIs is below the matching threshold δ, the ROIs are considered as a match. If several ROIs from f_2 match to the same ROI from f_1, the best match with the smallest distance is chosen. The bottom-up matching procedure is illustrated in Fig. 4, left.

The bottom-up matching works especially well, if the two frames differ only slightly. This is the case for tracking. For loop-closing, the bottom-up matching works well if the viewpoint of the landmark differs only slightly to the viewpoint it had when seeing the landmark for the first time. For more different viewpoints, the top-down matching is preferable.

$$d(\vec{v},\vec{w}) = \sqrt{\frac{v_{11}w_{11}\sum_{i=1,2}(v_i-w_i)^2 + v_{12}w_{12}\sum_{i=3,..,6}(v_i-w_i)^2 + v_{13}w_{13}\sum_{i=7,..,10}(v_i-w_i)^2}{v_{11}w_{11}+v_{12}w_{12}+v_{13}w_{13}}} \quad (1)$$



(a) bottom-up matching (b) top-down matching

Fig. 4. Left: bottom-up matching. To find a match for a ROI from frame 1, it is compared to each ROI from frame 2. Right: top-down matching. To find a match for a ROI from frame 1, its top-down feature vector v_{td} is used as target information to search for this ROI in frame 2. The resulting ROIs all look similar to the ROI from frame 1.

B. Top-down matching

For top-down matching, we determine for each ROI a top-down feature vector $\vec{v_{td}}$, as described in sec. III-A.2. These vectors are later used for comparison.

To find a match for ROI r_1 from frame f_1 in frame f_2, the vector $\vec{v_{td}}$ which describes r_1 is used to apply top-down search to f_2. From the resulting top-down saliency map, the most salient ROIs are extracted and their top-down feature vectors are compared to $\vec{v_{td}}$. As for the bottom-up matching, the ROIs are considered as a match if the matching distance d is below the matching threshold δ, and if several ROIs from f_2 match to r_1, the best match with the smallest d is chosen. The top-down matching procedure is illustrated in Fig. 4, right. The colors and the shape of the ROIs illustrate their similarity. Top-down matching compares a ROI only to similar regions, whereas the bottom-up matching compares it with all salient regions.

Top-down matching pays off especially if the appearance of two frames differs strongly. Since this is usually the case in loop closing situations, we apply the top-down matching for loop-closing. To search for an expected ROI does not mean that all computations of VOCUS have to be repeated for each expected ROI. The most time consuming computations, the computations of the feature maps, do not have to be done again. They are the same for the bottom-up computations and for each expected ROI. Therefore, these computations are still possible in real-time.

V. The Feature Tracker

In the feature tracker, the frames are stored in a buffer with length n (here: $n = 30$) and features are tracked over several frames. This buffer provides a way to determine which landmarks are stable over time and thus good candidates to use in the map. The output from the buffer is thus delayed by n frames but in return quality assessment can be utilized before using the data. The matching is performed not only between consecutive frames, but allows for gaps of several (here: 2) frames where a ROI is not found. We call frames which are at most 3 frames behind the current frame *close frames*.

A *landmark* is a list of tracked features. Features can be ROIs (ROI-landmark) or Harris corners (Harris-landmark). The *length* of a landmark is the number of elements in the list, which is equivalent to the number of frames the feature was detected in. The procedure to create landmarks is the following: when a new frame comes into the buffer, each of its ROIs is matched to all existing landmarks of close frames. We apply bottom-up matching here. If the matching is successful, the new ROI is appended to the end of the best matching landmark. Additionally, the ROIs that did not match any existing landmarks are matched to the unmatched ROIs of the previous frame. If two ROIs match, a new landmark is created consisting of these two ROIs. The same procedure is used to create the Harris-landmarks.

At the end of the buffer, the landmarks are transferred to the triangulator, which first checks whether the landmarks are long enough (≥ 5). Then, the Harris corners inside of ROIs are determined, and it is checked whether the corresponding Harris-landmarks are long enough and stable enough. Finally, the Harris-landmarks which survive the process are reported to the SLAM module.

VI. The Loop Closer

The loop closer obtains landmark predictions from the SLAM module and checks if these landmarks are visible in the current frame. In bottom-up matching mode, it compares each ROI from the expected landmarks to each ROI of the current frame. In top-down mode, it takes each ROI from each expected landmark, uses it as target information, and searches for it with top-down attention within the current frame. Then, the resulting top-down ROIs are compared to the ROIs from the expected landmarks with top-down matching. If there are several matches in the current frame, the best match is taken.

If there is a ROI-match, all of the Harris corners within the matching ROIs are compared based on their SIFT descriptor. If there is also a match, the corresponding landmark is reported to the SLAM module, to update the map. The combination of ROI and Harris matching enables a reliable matching with almost no false positives.

VII. Experiments and Results

In this section, we illustrate the differences between bottom-up and top-down matching and the advantages of the top-down matching for loop closing. We investigated the system behaviour twice for the same data obtained from the trajectory displayed in Fig. 5. The robot drove through a room, left the room, drove through the corridor, and entered the room again through a different door. After entering the room, it faced the same region as in the beginning. At this point, it should be

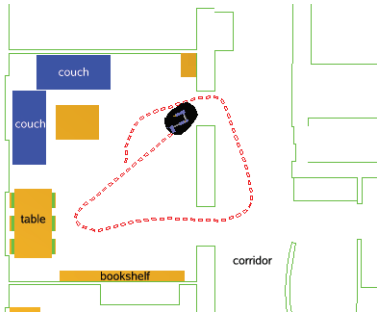


Fig. 5. The robot environment and the driven trajectory.

able to detect that it closed a loop. Although the loop is very small compared to some other SLAM-scenarios, it is sufficient here to show that top-down matching outperforms bottom-up matching in loop-closing situations. The effect of larger loops would be a higher uncertainty of robot and landmark positions, resulting in larger search areas in the images, in the worst case the whole image. In these cases, the advantage of top-down matching is expected to be even more important.

The visual SLAM system runs online in real-time, but for our experiments we needed offline data to enable experiments on the same data for both matching methods. Therefore, we stored the image sequence, consisting of 283 images, as well as the odometry information. We ran the system twice on this sequence, once the loop closing was implemented with the bottom-up matching and once with the top-down matching. Note, that in offline mode the gaze control module cannot be used. But since gaze control and top-down matching are two largely independent mechanisms (gaze control controls the camera actively whereas top-down matching focuses the processing actively to regions of interest within the current image), this does not affect the current experiments.

Each ROI of each expected landmark was considered for matching. Fig. 6 shows the matching results for different thresholds δ. It shows, that the increase of false matches (red, dashed line) for increasing thresholds is about the same for bottom-up and top-down matching, whereas the increase of correct matches (blue, solid line) is steeper for the top-down matching. That means, more correct matches are obtained in top-down mode.

To illustrate the correspondence between false and correct matches in more detail, Fig. 7 displays the correct matches depending on the number of false matches. This figure is similar to a ROC (receiver operating characteristic) curve, but note that here the axes denote numbers of matches instead of ratios. This is sufficient here, because in contrast to recognition tasks, where the ratio of correct matches is important, we are not interested in detecting all possible matches; some matches are sufficient to close the loop. However, a higher detection rate is still preferable, because it speeds up the loop closing process and makes it more stable.

We expected the top-down matching to outperform the bottom-up matching. Interestingly, this was not always the case. For low thresholds which accept only very few or no false detections at all, the bottom-up matching showed to be better and provided more correct matches. The turning point is between 8 and 15 false matches, where both bottom-up and top-down matching perform equally. For higher thresholds which accept more false matches, the top-down matching outperformed the bottom-up matching, resulting in a considerably higher number of correct matches: for 50 false matches, the bottom-up matching detected 183 correct matches whereas the top-down matching achieved 261 correct matches, which is an increase of 42%.

Note that this number of false ROI matches is not the number of false landmark matches which is reported to SLAM. First, several of the matched ROIs belong to the same landmark, since each ROI from an expected landmark is matched to current ROIs. For example, the 50 false matches of the top-down matching belonged to only 5 different ROI-landmarks with 10 matches on average and the 261 correct matches belonged to 9 ROI-landmarks with 29 matches on average. Since there are usually considerably more matches from correctly matching landmarks then from not-matching landmarks, the number of matching ROIs per landmark is an additional hint whether a landmark is redetected. We plan to consider this for future work. Second, since we additionally use the sift matching of the Harris corners, we are able to get rid of almost all of the remaining false ROI-landmark matches. In this example, only one Harris-landmark was classified wrongly with the top-down matching. Interestingly, this false match does not result from a false ROI match but from a wrong association of a Harris corner in the top-right corner of a ROI to one in the bottom-right corner.

To investigate the difference between the cases in which the bottom-up matching performed better and the ones in which top-down matching performed better, we had a closer look at the matches. It turned out that "easy" matches are better redetected with bottom-up matching. Easy matches are those in which ROIs are seen under almost the same conditions (i.e. from almost the same viewpoint and under almost the same lightning conditions) as when they were detected the first time. One example of such an easy match is displayed in Fig. 8, left. More difficult matches are better redetected with top-down matching. In some of these examples, the ROI is seen from a quite different viewpoint as the example in Fig. 8, right. These examples are of course more interesting, since usually the robot does not face a landmark from exactly the same position as before, so a viewpoint tolerance is necessary.

Since the matches in tracking situations are usually easy matches, we suggest to use the bottom-up matching in the feature tracker and top-down matching in the loop closer.

VIII. Conclusions

In this paper, we have presented a method to improve the loop closing behaviour for visual SLAM. The visual attention system, which detects regions of interest in a frame, is tuned in a top-down manner to search for expected landmarks. Whereas in easy matching situations the bottom-up matching is preferable, the top-down matching outperforms the bottom-up approach clearly in difficult matching situations: especially when the viewpoint changes, the top-down matching enables a

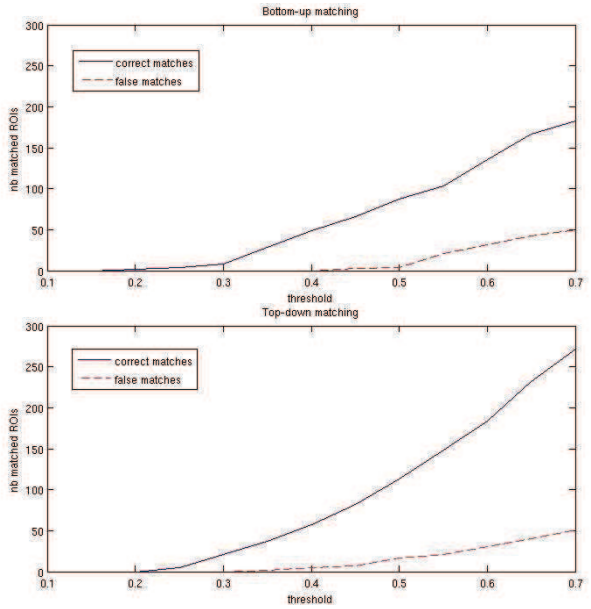


Fig. 6. Correct and false ROI matches for bottom-up (top) and top-down matching (bottom) depending on the matching threshold δ.

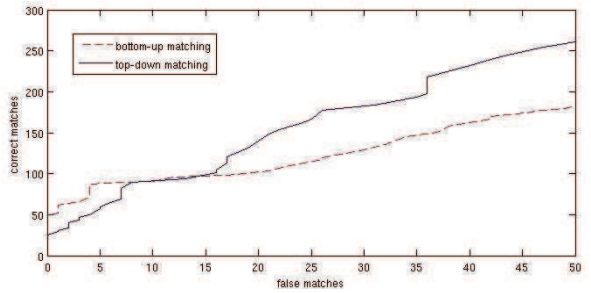


Fig. 7. Correct matches for bottom-up and top-down matching depending on the error rate: For a low number of false detections, bottom-up matching results in more correct matches. If more false matches are acceptable, top-down matching provides more correct matches.

more stable redetection with a considerably higher amount of correct matches. Remaining false detections are removed with an additional SIFT matching of Harris corners. This makes the method useful for loop closing situations. In future work, we plan to make the matching even more robust by considering the matching stability of features over time and the constellation of landmarks to each other within frames. Another topic of research will be to investigate the limits of the method, i.e., to check how strongly the viewpoint may differ to still enable redetection.

Fig. 8. The rectangles denote correctly matched ROIs. Top: current frame. Bottom: frame with expected ROI from database. Left: "easy" matching situation. Right: "difficult" matching situation from different viewpoint.

REFERENCES

[1] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. of the ICCV*, oct 2003.

[2] Andrew J. Davison and David W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Trans. PAMI*, 2002.

[3] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Automat.*, 17(3):229–241, 2001.

[4] U. Frese, P. Larsson, and T. Duckett. A multigrid algorithm for simultaneous localization and mapping. *IEEE Trans. Robot.*, 21(2):1–12, 2005.

[5] Simone Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. PhD thesis, 2005. Published 2006 in LNAI, Vol. 3899, Springer.

[6] Simone Frintrop, Patric Jensfelt, and Henrik Christensen. Attentional Landmark Selection for Visual SLAM. In *Proc. Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[7] Simone Frintrop, Patric Jensfelt, and Henrik Christensen. Pay attention when selecting features. In *Proc. Int'l Conf. on Pattern Recognition (ICPR 2006)*, 2006.

[8] Simone Frintrop, Patric Jensfelt, and Henrik Christensen. Attentional robot localization and mapping. In *ICVS Workshop on Computational Attention & Applications (WCAA)*, 2007.

[9] Simone Frintrop, Maria Klodt, and Erich Rome. A real-time visual attention system using integral images. In *Proc. of Int'l Conf. on Computer Vision Systems (ICVS)*, 2007.

[10] L. Goncavles, E. di Bernardo, D. Benson, M. Svedman, J. Ostrovski, N. Karlsson, and P. Pirjanian. A visual front-end for simultaneous localization and mapping. In *Proc. of ICRA*, pages 44–49, apr 2005.

[11] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11), 1998.

[12] Patric Jensfelt, Danica Kragic, John Folkesson, and Mårten Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. of ICRA'06*, Orlando, FL, May 2006.

[13] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of ICCV*, pages 1150–57, 1999.

[14] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. of ICCV*, pages 525–531, 2001.

[15] Paul Newman and Kin Ho. SLAM-loop closing with visually salient features. In *Proc. Int'l Conf. on Robotics and Automation, (ICRA 2005)*, 2005.

[16] Nabil Ouerhani, Alexandre Bur, and Heinz Hügli. Visual attention-based robot self-localization. In *Proc. of European Conf. on Mobile Robotics (ECMR 2005)*, 2005.

[17] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *Int'l J. of Robotics Research*, 19(11), 2000.

[18] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Y. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Int. J. Robot. Res.*, 23(7-8):693–716, 2004.

[19] Sebastian Thrun. Finding landmarks for mobile robot navigation. In *Proc. of ICRA*, 1998.

[20] John K. Tsotsos, Sean M. Culhane, Winky Yan Kei Wai, Yuzhong Lai, Neal Davis, and Fernando Nuflo. Modeling visual attention via selective tuning. *AI*, 78(1-2), 1995.

[21] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int'l Journal of Computer Vision (IJCV)*, 57(2):137–154, May 2004.

Simone Frintrop

Short Curriculum Vitae

Education and Employment

June 2007 – 2014	**Senior scientific assistant (Akademische Rätin)**, and head of Cognitive Vision Group (since 2013) at the Institute for Computer Science III (Prof. Dr. A.B. Cremers), Rheinische Friedrich-Wilhelms-Universität Bonn.
Sept. 2006 – June 2007	**Scientific assistant (Wissenschaftliche Mitarbeiterin)**, at the Institute for Computer Science III (Prof. Dr. A.B. Cremers), Intelligent Vision Systems Group (IVS), Rheinische Friedrich-Wilhelms-Universität Bonn.
Aug. 2005 – July 2006	**Postdoctoral researcher**, at the "Computer Vision and Active Perception" lab (CVAP) (Prof. Dr. H.I. Christensen), at the Royal Institute of Technology (KTH) in Stockholm, Sweden.
July 2005	**PhD defense (Disputation)**, grade 1.0 (very good, "magna cum laude").
Feb. 2002 – June 2005	**PhD student**, at the team ARC (Autonomous Robot Architectures, Prof. Dr. J. Hertzberg) at the Fraunhofer Institute AIS (Autonomous Intelligent Systems) in St. Augustin, Germany.
Oct. 2001	**Diploma degree (∼Master)**, in computer science with grade 1.0 (very good).
Aug. 2000	**Working student**, for the JOANNEUM RESEARCH Institute (Graz, Austria).
1994 – 2001	**Studies**, of computer science at the Rheinische Friedrich-Wilhelms-Universität Bonn.
1994	**Abitur**, (university entrance diploma) in Mettmann, Germany.
1982 – 1994	**Schools**, in Wetter, Zweibrücken, and Mettmann, Germany.

Declaration of Authorship

Hereby, I declare that this cumulative habilitation thesis titled „Cognitive Approaches for Mobile Vision Systems” is my own work. I confirm that I have acknowledged all main sources of help and have clearly attributed where I have consulted the published work of others. Where ever work was performed in cooperation with others, this has been clearly stated.

Simone Frintrop