Simultaneous Robot Localization and Mapping Based on a Visual Attention System

Simone Frintrop* and Patric Jensfelt** and Henrik Christensen***

* Comp. Science III, University of Bonn, Germany, frintrop@iai.uni-bonn.de
** CSC, KTH, Stockholm, Sweden, patric@csc.kth.se
*** GeorgiaTec, Atlanta, USA hic@cc.gatech.edu

Abstract. Visual attention regions are useful for many applications in the field of computer vision and robotics. Here, we introduce an application to simultaneous robot localization and mapping. A biologically motivated attention system finds regions of interest which serve as visual landmarks for the robot. The regions are tracked and matched over consecutive frames to build stable landmarks and to estimate the 3D position of the landmarks in the environment. Matching of current landmarks to database entries enables loop closing and global localization. Additionally, the system is equipped with an active camera control, which supports the system with a tracking, a re-detection, and an exploration behaviour. We present experiments which show the applicability of the system in a real-world scenario. A comparison between the system operating in active and in passive mode shows the advantage of active camera control: we achieve a better distribution of landmarks as well as a faster and more reliable loop closing.

1 Introduction

In the field of robotics, *visual SLAM* (Simultaneous Localization And Mapping) has recently been a topic of much research [7, 3, 15, 18, 16, 4]. The task is to build a map of the environment and to simultaneously stay localized within the map. In contrast to common laser-based approaches, visual SLAM aims at solving the problem only based on camera data. The *map* consists of landmarks and their relative position to each other and to the robot. It is not intended as reference for a human but as internal representation of the environment for the robot.

A key competence in visual SLAM is to choose useful visual 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 one of the most important problems in SLAM since it decreases accumulated errors. Furthermore, there should be a limited number of landmarks since the complexity of SLAM typically is a function of the number of landmarks in the map. On the other hand, landmarks should be distributed over the environment.

Often, the landmarks are selected by a human expert or the kind of landmark is determined in advance, e.g., ceiling lights [28] or Harris-Laplace corners [18]. As pointed out by [27], there is a need for methods which enable a robot to choose landmarks autonomously. A good method should pick the landmarks

which are best suitable for the current situation. An adequate method to find landmarks autonomously depending on the current surrounding are visual attention systems [31, 17, 10]. They select regions that "pop out" in a scene due to strong contrasts and uniqueness, as the famous black sheep in a white herd. The advantage of these methods is that they determine globally which regions in the image discriminate instead of locally detecting predefined properties.

In this paper, we present a visual SLAM system based on an attentional landmark detector. Regions of interest (ROIs) are detected by the attention system VOCUS [10], and are tracked and matched over consecutive frames to build stable landmarks. The 3D position of the landmarks in the environment is estimated by structure from motion and the landmarks are integrated into the map. When the robot returns to an already visited location, this loop closing is detected by matching current landmarks to database entries. This enables the updating of the current robot position as well as the other landmark entries in the map. Additionally, active camera control improves the quality and distribution of detected landmarks with three behaviours: a *redetection* behaviour actively searches for expected landmarks to support loop-closing. A *tracking* behaviour identifies the most promising landmarks and prevents them from moving out of the field of view. Finally, an *exploration* behaviour investigates regions with no landmarks, leading to a more uniform landmark distribution.

The applicability of the system in a real-world scenario is shown in real-world experiments in an office environment. The advantages of the active vs. the passive camera control are shown by comparing the system performance for both operating modes.

2 Related Work

In robotics, SLAM (Simultaneous localization and mapping) has been a topic of significant interest over the last decade [8, 9, 29]. The usual approach in these systems is to use range sensors like laser scanners. As mentioned in the introduction, there has recently been large interest within the robotics community to solve the SLAM problem with cameras as sensors, since cameras are low-cost, low-energy and light-weight sensors which might be used in many applications where laser scanners are too expensive or too heavy [6, 15, 18, 16, 4]. There is also interest in the computer vision community for visual SLAM systems, since the techniques may equally be applied to hand-held cameras [7, 3].

Concerning visual attention systems, there have been many attention systems developed during the last two decades [17, 31, 26, 2, 10]. They are all based on principles of visual attention in the human visual system and adopt many of their ideas from psychological theories, like the feature integration theory [30] and the Guided Search model [34]. Most systems focus on bottom-up computations, that means they consider only image-based information to compute the saliency regions. Recently, there have been some systems which are able to include top-down cues, that means they are able to search for a object of interest in a scene [21, 11, 10]. Here, we use the attention system VOCUS [11, 10], which is able to

perform visual search and has the additional advantage that it is capable to operate in real-time [14].

Although attention methods are well suited for selecting landmark candidates, the application of attention systems to landmark selection has rarely been studied. Nickerson et al. detect landmarks in hand-coded maps [23], Ouerhani et al. built a topological map based on attentional landmarks [24], and Siagian and Itti use attentional landmarks in combination with the gist of a scene for outdoor Monte-Carlo Localization [25]. 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 [22]. 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.

The idea of active sensing is not new [1], and has been extensively investigated for robot exploration. However, in the field of visual SLAM, most approaches use static cameras. Probably the most advanced work in the field of active camera control for visual SLAM is presented by the group around Davison. In [5, 6] they present a robotic system, which chooses landmarks for tracking which best improve the position knowledge of the system. In more recent work [32, 3], they apply their visual SLAM approach to a hand-held camera. Active movements are done by the user, according to instructions from user-interface [32], or they use the active approach to choose the best landmarks from the current scene without controlling the camera [3].

3 System Overview

The visual SLAM architecture (Fig. 1) consists of 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 *SLAM module* which builds a map of the environment, a *loop closer* which matches current ROIs to the database, and 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*. This module finds ROIs based on the visual attention system VOCUS 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-Laplace 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 to SLAM module in [18]).

The task of the *loop closer* is to detect if a scene has been seen before. The features from the current frame are compared with the features from the

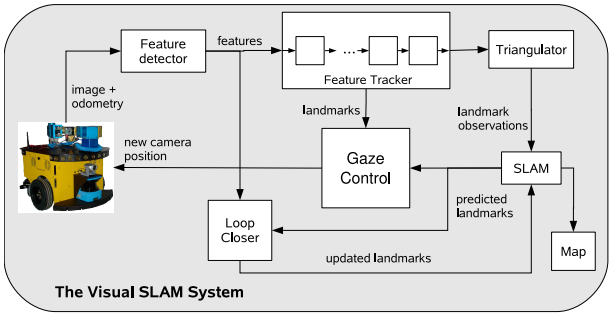


Fig. 1. The visual SLAM system builds a map based on image data and odometry

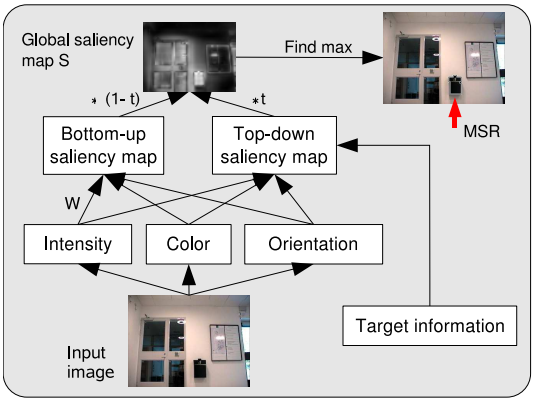
landmarks in the database. To narrow down the search space, the SLAM module provides the loop closer with expected landmark positions. Only landmarks that should be currently visible are considered for matching. Finally, the *gaze control module* controls the camera actively. It decides whether to actively look for predicted landmarks, to track currently seen landmarks, or to explore unseen areas. It computes a new camera position which is provided to the robot.

4 Feature Selection

The feature selection is based on two different kinds of features: attentional ROIs and Harris-Laplace corners. The attentional ROIs focus the processing on salient image regions which are thereby well redetectable. Harris-Laplace corners on the other hand provide well-localized points which enables a precise depth estimation when performing structure from motion. Additionally, the combination of two kinds of features makes the matching of regions for loop closing more stable. At the moment, we are investigating ways to use only the attention regions, which would simplify and speed up the system.

ROI Detection: Regions of interest (ROIs) are detected with the attention system VOCUS (Visual Object detection with a CompUtational attention System) [10] (Fig. 2). It consists of a bottom-up part similar to [17], and a top-down part enabling goal-directed search; global saliency is determined from both cues.

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



Feature	vector $\boldsymbol{v}$
intensity on/off	0.16
intensity off/on	6.54
orientation 0°	2.06
orientation 45°	3.17
orientation 90°	3.06
orientation 135°	2.95
color green	3.14
color blue	1.35
color red	1.36
color yellow	0.93
conspicuity I	3.30
conspicuity O	2.60
conspicuity C	1.70

Fig. 2. Left: the visual attention system VOCUS. The red arrow points to the most salient region (MSR). Right: feature vector for the MSR.

to 3 conspicuity maps I (intensity), O (orientation) and C (color) and combined to form the *bottom-up saliency map* S_{bu} (cf. Fig. 3, top left):

$$S_{bu} = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C)$$

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 in S (seeds) are found and second all neighboring pixels over a saliency threshold (here: 25% of the seed) are detected recursively with *region growing*. A *region of interest (ROI)* is defined as *height* $*$ *width* of the MSR. For each *MSR*, a feature vector $\boldsymbol{v}$ 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 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. 2 right shows a feature vector which corresponds to the MSR of the image on the left. It tells us, e.g., that the region is dark on a bright background (off-on intensity).

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 $\boldsymbol{v}$. In *search mode*, VOCUS multiplies the feature and conspicuity maps with the weights of $\boldsymbol{v}$. The resulting maps are summed up, yielding the *top-down saliency map* S_{td} (cf. Fig. 3, bottom left). 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 [10]). Here we use t=0, since the experiments are restricted to bottom-up

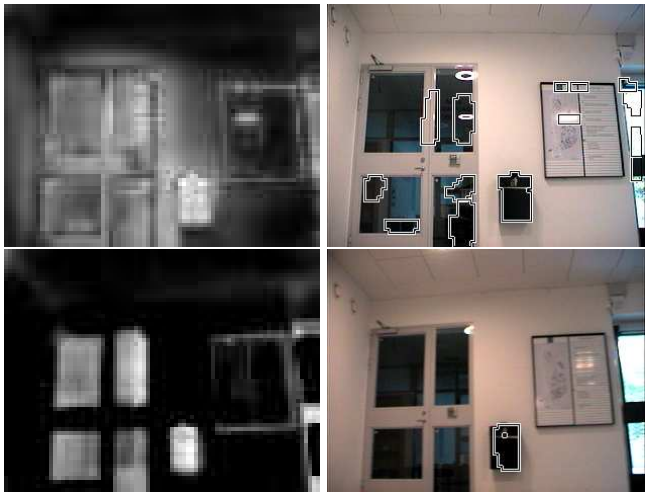


Fig. 3. Saliency maps (top, left: S_{bu}, bottom, left: S_{td}) and MSRs in a loop closing example: Top: scene at beginning of sequence (all MSRs shown). Bottom: revisited scene, 592 frames later, searching for the black waste bin with top-down attention (only most salient MSR shown).

mode. In [12] we show how top-down information can be included into the loop closer of the visual SLAM system: if a previously found landmark is expected to be visible in the current frame, the feature vector of the landmark is used as target information for top-down search. A matching procedure, similar to the one described in the next section, is used to determine whether the expected landmark was actually redetected.

The feature computations are efficiently performed on *integral images* [33]. 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 400×300 pixel image, 2.8GHz) that enables real-time performance (details in [14]).

Harris-Laplace corners: To detect features with high position stability inside the ROIs, we used the Harris-Laplace feature detector [20] – an extension of the Harris corner detector to Laplacian pyramids which enables scale invariance. This resulted in a few (average 1.6) points per ROI (cf. Fig. 4 bottom right). To allow matching of points, a SIFT descriptor is computed for each detected corner [19].

5 Matching and Tracking of Features

Feature matching is performed in the feature tracker (for creating landmarks) and in the loop closer (to detect if this landmark has been seen before). The

matching is based on two criteria: proximity and similarity. First, the features in the new frame have to be close enough to the predicted position. Secondly, the similarity of the features is determined. This is done differently for attentional ROIs and for Harris-Laplace corners: the matching of Harris-Laplace 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 first, the ROIs might differ slightly in shape depending on image noise and illumination variations and, second, seeing a region in the environment from different viewpoints changes the size of the region in the image.

The similarity of two feature vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ is determined by

$$d(\boldsymbol{v},\boldsymbol{w}) = \sqrt{\frac{v_{11}w_{11}\sum_{i=1,2}(v_i-w_i)^2 + v_{12}w_{12}\sum_{i=3,..,6}(v_i-w_i)^2 + v_{13}w_{13}\sum_{i=7,..,10}(v_i-w_i)^2}{v_{11}w_{11}+v_{12}w_{12}+v_{13}w_{13}}}.$$

The smaller the distance $d(\boldsymbol{v},\boldsymbol{w})$, the higher the similarity of the ROIs. If $d(\boldsymbol{v},\boldsymbol{w})$ is below a certain threshold δ, the ROIs match (see sec. 7 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.

If the distance d is below a certain threshold δ, the ROIs match. We use different values for tracking ($\delta = 3.0$) and loop closing ($\delta = 1.7$). When tracking, the estimated position from odometry is usually accurate, and we can afford a more relaxed threshold than for loop closing where the position estimation is less accurate. In [13], we investigated the choice of the threshold in detail.

In the feature tracker, the features are tracked over several frames. We store the last n frames in a buffer (here: $n = 30$). 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*.

Creating Landmarks: A *landmark* is a list of tracked features. Features can be ROIs (ROI-landmark) or Harris-Laplace corners (Harris-landmark). The *length* of a landmark is the number of elements in the list, which is equivalent to the

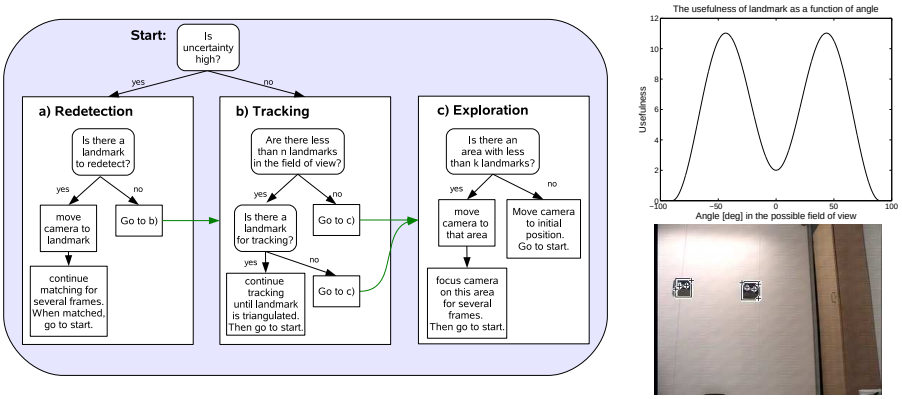


Fig. 4. Left: the three camera behaviours. Right top: usefulness function w. Bottom: example image with two ROI-landmarks and several Harris-landmarks. The landmarks of the left ROI are more useful, since they are not in the center of the field of view.

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. 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. At the end of the buffer, we consider the length of the resulting landmarks and filter out too short ones (here ≤ 5).

6 Active Gaze Control

The active gaze control is divided into three behaviours: a) redetection of landmarks to close loops, b) tracking of landmarks, and c) exploration of unknown areas. The strategy to decide which behaviour to choose is as follows (Fig. 3): Redetection has the highest priority, but it is only chosen if the position uncertainty is over a certain value. If the uncertainty is low or if there is no expected landmark for redetection, the *tracking* behaviour is activated. Tracking is only performed if there are not yet enough landmarks in this area. As soon as a certain amount of landmarks is obtained in the field of view, the *exploration* behaviour takes over. It moves the camera to an area with no detected landmarks. In the following we describe the behaviours in more detail.

Redetection: The redetection of landmarks is performed if the current robot pose uncertainty is high and there are old landmarks that are or could be made visible through active camera control. This information is provided by the SLAM module. If there is an expected landmark and the robot pose uncertainty is high, the camera is moved to focus on the expected landmark. If we have more than

one expected landmark, we have to choose the potentially most useful landmark for redetection. Here, we consider only the length of the current ROI-landmark: the longer this landmark, the better. The new camera position is maintained until a match is performed or until a waiting threshold is exceeded.

Tracking: 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. 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 ROI- and Harris-landmarks are important factors for the usefulness of a landmark, 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 frames and result often in poor triangulation results. 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.

Therefore, we determine the usefulness of a landmark by first considering the length of the ROI-landmark, second the angle of the landmark in the potential field of view, and third the length of the Harris-landmark. The length of the ROI-landmarks is considered by sorting out landmarks below a certain size (here: 5). The usefulness of the angle of a ROI is determined by the following function:

$$w = (k_1 \, (1.0 + \cos(4 \, (\alpha - 180))) + k_2 \, (1.0 + \cos(2\alpha))) \tag{1}$$

where α is the angle and $k_1 = 5$ and $k_2 = 1$. The function is displayed in Fig. 4 (top right). The usefulness is highest for points at $\alpha = 45°$ and $\alpha = -45°$ and lowest at $\alpha = 0°$ and $\alpha = \pm 90°$. Since points which are at the border of the field of view are likely to move out of view very soon, they are considered even worse than points in the center.

The usefulness U of a Harris-landmark is then determined by: $U = w\sqrt{l}$, where l is the length of the landmark. In Fig. 4, we demonstrate the effect of U. The bottom-right image shows two identical regions on the wall, both are detected by VOCUS and have several Harris-Laplace corners which were detected inside the ROI. The main difference between the landmarks is that one of them is almost in the center of the image and the other one at the border (the camera points straight ahead). The values w and U are higher for the landmark on the left. This leads to choosing the left landmark for tracking since it is likely that it provides a better baseline for triangulation.

After determining the most useful landmark for tracking, the camera is moved into the direction of the landmark. It is moved slowly (here 0.1 radians per step), since this turned out to be more successful than moving it quickly to center the landmark. This corresponds to the pursuit eye movements of humans when following a target. On the other hand, the quick camera motion for the redetection and exploration behaviour corresponds to saccades (quick eye movements) in human viewing behaviour, which are performed when searching for a target or exploring a scene. The tracking ends when the landmark is not visible any more

(because it left the field of view or because the matching failed) or when the landmark was successfully triangulated.

Exploration: In exploration mode, the camera is moved to an area in the possible field of view where the map contains no landmarks. To avoid too many camera movements and to enable building of landmarks over several frames, the camera focuses one region for a while (here 10 frames). As soon as a landmark for tracking is found, the system switches automatically the behaviour.

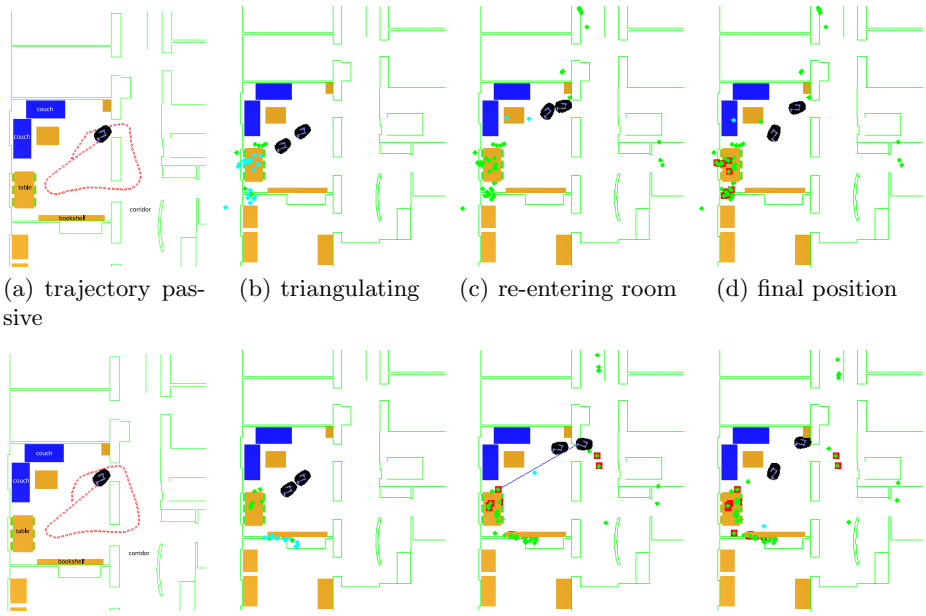
7 Experiments and Results

To illustrate the performance of the robot equipped with the visual SLAM system, the robot drove a loop within our office environment. It drove through a room, entered the corridor, entered the same room through a different door and closed a loop. During this trajectory, it built a map of the environment. Remember that in visual SLAM systems, the map consists of visual landmarks, that means the green and blue dots in Fig. 5 form the maps. In order to show the advantages of active gaze control, we let the robot drive the same trajectory once in passive and once in active mode. The trajectories are displayed in Fig. 5 ((a): passive. (e): active). Although the loop is very small compared to some other SLAM-scenarios, it is sufficient here to show that active camera control outperforms the passive approach. However, applying the system to scenarios with larger loops is an interesting topic for future work.

The second column shows a snapshot after driving about 2m. In passive mode, many landmarks are extracted in the table area, but nothing is found in the area of the bookshelf (b). In contrast to this, the active mode enables a better distribution of landmarks: as soon as 5 landmarks are triangulated in the table area, the exploration behaviour is activated, the camera is moved to the left. There, the system finds a ROI to track and keeps fixating the region until several landmarks are triangulated in the bookshelf area (f).

The next column shows the situation when the robot enters the room again through the second door. The robot faces the couch area and has in passive mode no chance to recognize any existing landmarks although they are close (c). On the other hand, in active mode there are expected landmarks within the possible field of view as indicated by the blue line in (g). The camera is directed to the table area and matches current ROIs to expected landmarks. Successful matches are displayed as big red squares (g). Two examples of these matches are displayed in Fig. 6. So, in active mode loop closing is already performed at a stage where this is not possible in passive mode, and a decrease of uncertainty is achieved earlier.

Finally, the end of the sequence is displayed in the last column. Here, the robot has achieved a position, in which matching is also possible in passive mode (d). On the other hand, in active mode the robot has already started to additionally match the landmarks in the bookshelf area (h). Altogether, by active gaze control we achieve a better distribution of landmarks and a faster and more reliable loop closing.



(a) trajectory passive (b) triangulating (c) re-entering room (d) final position

(e) trajectory active (f) active exploration (g) active redetection (h) final position

Fig. 5. Comparison of visual SLAM with passive (top) vs active (bottom) camera control. Two robots in one image correspond to the robot at the beginning and at the end of the buffer, i.e., the robot further ahead on the path is the real robot, the one behind is the virtual robot position 30 frames later. Currently visible landmarks are displayed as cyan dots, currently not visible landmarks in green. Landmarks matched to database entries are larger and displayed in red (d,g,h). When the robot tries to redetect a landmark, the estimated direction of the landmark is displayed as a blue line (g).

8 Conclusion

In this paper, we have presented a visual SLAM system based on an attentional landmark detector. The attentional regions are especially useful landmarks for tracking and redetection. Three behaviours for active camera control help to handle some of the problems of visual SLAM: landmarks with a better baseline are preferred and a better distribution of landmarks is achieved.

Needless to say, there is a lot which could be done to improve the performance of the system. The redetection rate of landmarks could be improved by considering not only one expected landmark for matching, but all in the current field of view. Monitoring redetection over several frames is another possibility to exclude false matches. Extending the system to larger environment is also not trivial, since the complexity of the SLAM system grows with the number of landmarks. Removing landmarks which are not redetectable from the map would

Fig. 6. Two matches of ROIs (rectangles) and Harris-Laplace points (crosses) between a current frame (top) and a scene from the database (bottom).

help to keep the number of landmarks low. Working with hierarchical maps as in [3], in which many local maps are built which do not exceed a certain size, is another possibility to cope with large environments.

9 ACKNOWLEDGMENTS

Large parts of the present research have been sponsored by the European Commission through the project NEUROBOTICS (EC Contract FP6-IST-001917) and the Swedish Foundation for Strategic Research through its Center for Autonomous Systems. The support is gratefully acknowledged. Additionally, we thank Prof. A.B. Cremers, for funding the research of S. Frintrop to complete the presented experiments.

References

1. Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *International Journal of Computer Vision (IJCV)*, 1(4):333–356, 1988.
2. G. Backer, B. Mertsching, and M. Bollmann. Data- and model-driven gaze control for an active-vision system. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(12):1415–1429, 2001.
3. 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.
4. M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, April 2007.

5. A. Davison and D. Murray. Mobile robot localisation using active vision. In *Proc. of ECCV*, May 1998.
6. A. Davison and D. Murray. Simultaneous localisation and map-building using active vision. *IEEE Trans. PAMI*, 2002.
7. A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. of the ICCV*, oct 2003.
8. 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.
9. U. Frese, P. Larsson, and T. Duckett. A multigrid algorithm for simultaneous localization and mapping. *IEEE Trans. Robot.*, 21(2):1–12, 2005.
10. S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Germany, July 2005. Published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer Verlag Berlin/Heidelberg.
11. S. Frintrop, G. Backer, and E. Rome. Goal-directed search with a top-down modulated computational attention system. In *Proc. of the Annual meeting of the German Association for Pattern Recognition (Jahrestagung der Deutschen Arbeitsgemeinschaft für Mustererkennung) DAGM 2005*, Lecture Notes in Computer Science (LNCS), pages 117–124, Conference: Wien, Austria, Sept. 2005. Springer.
12. S. Frintrop and A. B. Cremers. Top-down attention supports visual loop closing. In *accepted for Proc. of European Conference on Mobile Robotics (ECMR 2005)*, 2007.
13. S. Frintrop, P. Jensfelt, and H. Christensen. Attentional Landmark Selection for Visual SLAM. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS '06)*, Beijing, China, October 2006.
14. S. Frintrop, M. Klodt, and E. Rome. A real-time visual attention system using integral images. In *Proc. of the 5th International Conference on Computer Vision Systems (ICVS)*, Bielefeld, Germany, March 2007.
15. 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.
16. K. Ho and P. Newman. Detecting loop closure with scene sequences. *International Journal of Computer Vision and International Journal of Robotics Research. Joint issue on computer vision and robotics*, 2007.
17. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
18. P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. of ICRA'06*, Orlando, FL, May 2006.
19. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of ICCV*, pages 1150–57, 1999.
20. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. of ICCV*, pages 525–531, 2001.
21. V. Navalpakkam, J. Rebesco, and L. Itti. Modeling the influence of task on attention. *Vision Research*, 45(2):205–231, 2005.
22. P. Newman and K. Ho. SLAM- loop closing with visually salient features. In *Proc. of the International Conference on Robotics and Automation, (ICRA 2005)*, Barcelona, Spain, April 2005.

23. 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.
24. N. Ouerhani, A. Bur, and H. Hügli. Visual attention-based robot self-localization. In *Proc. of European Conference on Mobile Robotics (ECMR 2005)*, pages 8–13, Ancona, Italy, Sept. 2005.
25. C. Siagian and L. Itti. Biologically-inspired robotics vision monte-carlo localization in the outdoor environment. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS 2007) (to appear)*, 2007.
26. Y. Sun and R. Fisher. Object-based visual attention for computer vision. *Artificial Intelligence*, 146(1):77–123, 2003.
27. S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. of ICRA*, 1998.
28. 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.
29. S. Thrun, Y. Liu, D. Koller, A. 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.
30. A. M. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.
31. 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*, 78(1-2):507–545, 1995.
32. T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D. W. Murray. Active control for single camera slam. In *Proc. of ICRA 2006*, 2006.
33. P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, May 2004.
34. J. M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238, 1994.