# Saliency-based Object Discovery on RGB-D Data with a Late-Fusion Approach

Germán M. García[1], Ekaterina Potapova[2], Thomas Werner[1], Michael Zillich[2], Markus Vincze[2] and Simone Frintrop[1]

*Abstract*— **We present a novel method based on saliency and segmentation to generate generic object candidates from RGB-D data. Our method uses saliency as a cue to roughly estimate the location and extent of the objects present in the scene. Salient regions are used to glue together the segments obtained from over-segmenting the scene by either color or depth segmentation algorithms, or by a combination of both. We suggest a late-fusion approach that first extracts segments from color and depth independently before fusing them to exploit that the data is complementary. Furthermore, we investigate several mechanisms for ranking the object candidates. We evaluate on one publicly available dataset and on one challenging sequence with a high degree of clutter. The results show that we are able to retrieve most objects in real-world indoor scenes and clearly outperform other state-of-the art methods.**

## I. Introduction

Object discovery is the task of finding objects in scenes without having a priori knowledge of what the objects look like or what class they belong to. It has recently attracted a lot of attention in the robotics and vision communities [2], [19], [11], [15] and is essential for many robotic tasks such as object manipulation and scene exploration.

The task of discovering objects is a chicken-and-egg problem: how to look for an object before knowing how it looks like and which features it has? The vision and robotics communities have developed different approaches to address this problem. Vision approaches usually operate on color images and generate a pool of object candidates, also known as object proposals, based on various types of image features which are combined by a machine learning method [2], [19]. The idea is to generate promising candidate regions as pre-processing for recognition, whose number is significantly smaller than the number of sliding windows used by default. Since usually around 1000 candidates are generated, these approaches are less useful for systems which have to operate in real-time and which potentially aim to interact with the objects. In the robotics community, it is therefore preferred to generate a small set of object candidates. Many groups use the 3D information of the scene by either operating directly on the depth data from an RGB-D device [23] or by first reconstructing the scene and then doing the discovery of objects in the 3D reconstruction [11], [15]. Other approaches use information about changes over time to segregate objects from background [11] or interact with possible object candidates to determine what is an object [28]. While these are helpful approaches to resolve ambiguities, it is certainly desirable to be able to find objects also without or before interaction, and if possible already from a single view without the need to regard a scene over a longer time.

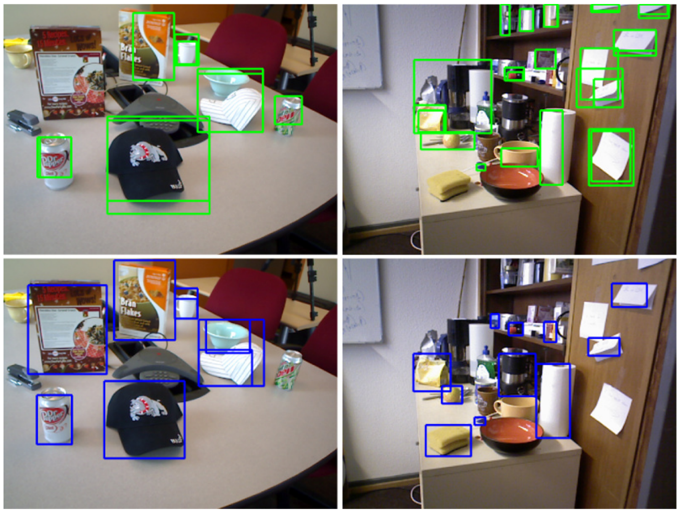

Fig. 1. Object candidates detected in color (top) and in depth (bottom) data. It shows that color and depth are complementary: some objects are only visible in color, some only in depth data. Fusion of both yields the best results.

In this paper, we present a method for object discovery that combines color and depth data to generate object hypotheses. We follow a late-fusion approach that first determines color and depth candidates independently, and fuses them afterwards by an SVM-learned sorting method. This exploits the fact that the data is complementary and some objects are most easily extracted in color and others in depth data. For example, in Fig. 1, the cereal box is fully detected in the depth but not in the color data due to the inhomogeneous texture (left), while the flat papers at the wall are only perceivable in the color image (right).

Our discovery method is based on the simple principle

The research leading to these results has received funding from the German research foundation (DFG), under project No. FR 2598/5-1, from the European Community, Seventh Framework Programme (FP7/2007-2013), under grant agreement No. 600623, STRANDS No. 610532, SQUIRREL, and the Austrian Science Foundation (FWF) under grant agreement No. TRP 139-N23, InSitu. We acknowledge the support.

[1] are with the Institute of Computer Science III, University of Bonn, Germany {martin,wernert,frintrop}@iai.uni-bonn.de

[2] are with the Automation and Control Institute, TU Vienna, Austria {potapova,zillich,vincze}@acin.tuwien.ac.at

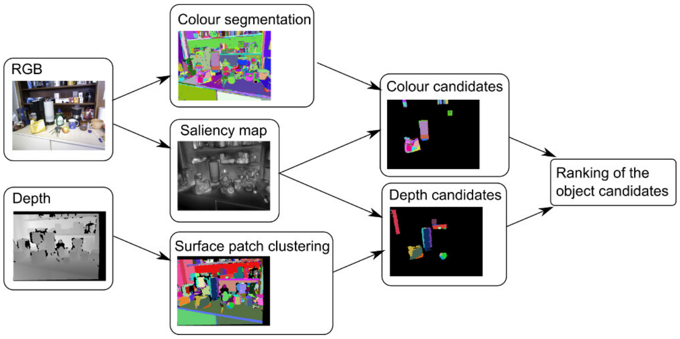

Fig. 2. Overview of the proposed method for object discovery.

that objects usually differ from their surroundings[1]. For this, we use an attention mechanism that finds salient regions in the scene and uses them as an indicator of the location and extent of objects. Their precise boundaries are delimited by a segmentation algorithm and segments are selected depending on their overlap with the salient blobs extracted from the saliency map. The segmentation algorithms we investigated here are the color-based segmentation of Felzenszwalb and Huttenlocher [5], the depth based surface patch clustering of Richtsfeld et al. [26], and the RGB-D supervoxels [22]. We suggest to combine the first two methods in a late-fusion approach and show in our experiments that this outperforms the early fusion approach in which candidates are directly generated in the RGB-D data, using color and depth early. Interestingly, the late fusion approach has a rough correspondence to human vision, where color and depth are processed largely independently in two different pathways [20].

The final step of our method is to rank the object candidates according to their likelihood to represent an object. We propose three ranking mechanisms, of which SVM-based ranking gave the best results. An overview of our approach is shown in Fig. 2. We show on two datasets, the well-known Washington dataset [18] and the challenging Coffee Machine sequence [9], that our approach is able to retrieve most objects in real-world indoor scenes and outperforms other state-of-the-art methods for object discovery.

## II. Related Work

Object discovery is a topic which started only recently to develop strongly. The methods can be roughly classified into approaches that operate on color images, those that operate on depth, and those that use additional information to detect objects.

Among the approaches that operate on color frames, the objectness measure of Alexe et al. [2] is one of the best known approaches. The authors sample object proposals guided by an objectness measure, learned from features such as saliency, edge density, and superpixels straddling. Rahtu et al. [24] introduced a cascade learning using similar features. Feng et al. [6] proposed to detect objects by measuring how well the candidate can be composed by the rest of the image. Manén et al. [19] proposed a randomised Prim algorithm to group superpixels into object proposals according to properties such as color homogeneity. In [8], we have presented a predecessor of the current work, which focused on images and did not use depth data.

In the robotics community, many groups have operated on depth data. Johnson-Roberson et al. do object segmentation on full point clouds [14]. The segmentation is seeded at salient points in the image that are mapped to the full point cloud. In [27], 3D object models are built by matching scans from partial views from which they subtract points that correspond to planar surfaces: floor, walls, etc. Karpathy et al. [15] rely on a reconstruction of the scene where several 3D features of objects are used to segment them: compactness, symmetry, smoothness, convexity.

Little work has been done on object discovery approaches that operate on both color and depth data. Mishra et al. [21] proposed a framework for segmentation, where an object is segmented using the concept of "simple objects and border ownership", which is defined using depth, color and/or motion information about the scene. In previous work [10], we have detected objects in RGB-D data by observing a scene over time and incrementally updating 3D object models. The difference to the here presented approach is that object candidates were purely generated based on color and not on depth data; depth was only used to create the 3D map. In the work of Potapova et al. [23], the authors developed a method to segment objects from RGB-D images. A 3D symmetry-based saliency operator is used to select attention points. The object boundaries are then found by adding surface patches that preserve the compactness and the color model of the object hypothesis.

The final class of approaches summarized are methods that use additional information about the scene to find objects. For example, Herbst et al. [11] discover objects by analysing the changes that have occurred in a scene at different points in time: elements that do not match to the 3D reconstruction of the scene are most likely objects. Collet et al. [4] incorporate domain knowledge to support the discovery process. Other researches [16],[28] used interaction with the scene do detect objects as parts of the space that are moving together when pushed or carried around.

Our goal here is to optimize the discovery on single RGB-D frames to lay a solid foundation for algorithms that use additional information such as temporal data or interaction.

## III. Saliency-based Object Discovery

Our approach for generating object candidates follows a simple principle: objects differ usually from their surroundings. This difference can be in color, intensity, texture, depth, or other features. A well-known method to detect regions that visually differ from their surroundings is saliency computation. It originates from the human visual attention system that is able to detect regions that differ from their surroundings quickly and effortlessly. Therefore, we use saliency as a cue to locate and roughly estimate the extent

[1] This principle is only violated in camouflage cases, which are also difficult to detect for humans.

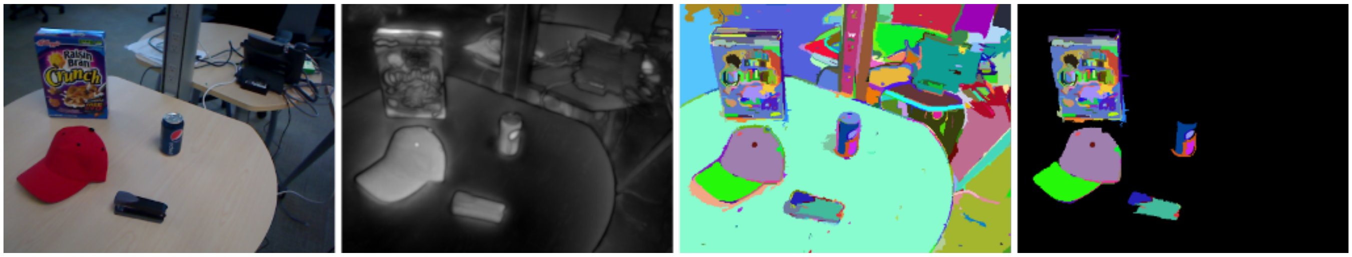

Fig. 3. From left to right: original image, saliency map, image segmentation, and some of the candidates generated.

of the objects. In parallel, a segmentation of the scene is computed, and saliency is used to glue the segments together into object hypotheses/candidates.

The idea to combine saliency and segmentation has a cognitive motivation in the psychological work of Rensink [25]: in human perception, so-called *proto-objects* are detected by segmentation processes that bundle parts of the visual field; such processes are believed to exist on all levels of the visual system [29]. Second, these proto-objects are combined by focused attention to form coherent objects. While these attention mechanisms consist of bottom-up and top-down parts, top-down information is not always available. Thus, we focus on bottom-up attention here, which corresponds to saliency computation.

An overview of our approach is shown in Fig. 2. It operates on RGB-D data, but if only color data is available, the method works also well. The combination gives however the best performance, since the channels are complementary. From the color image, we compute a saliency map. In parallel, we compute segmentations in the color as well as in the depth channels. By using saliency to select segments, we obtain a set of object candidates. Since in many applications it is preferable or even required to restrict the processing to a small number of candidates, e.g. to meet real-time requirements, it is important to rank the candidates according to their quality to be able to select the $n$ best ones. We investigated different ranking strategies in Sec. III-D.

### A. Detection of Salient Blobs

We extract salient blobs from the color image which we will need later on to select segments that form the object candidates. The salient blobs are computed in two steps. First, we compute a saliency map. Second, we extract the most salient blobs from this map. The saliency map is computed with the VOCUS2 saliency system[2], which is a re-implementation of the VOCUS system [7]. It computes pixel-precise saliency maps with state-of-the-art performance on current benchmarks for salient object detection, but it performs also well on cluttered real-world scenes as common in robotics applications. The main structure is similar as in traditional saliency systems such as [13], i.e., the system computes different features in parallel on different scales before fusing these feature maps to a single saliency map $sal(x, y)$. Contrast in each feature channel is computed by Difference-of-Gaussian filters, but instead of subtracting layers of the pyramid as in [13], we build specific center and surround pyramids which are used for subtraction. This enables more flexibility in determining the center-surround ratios. Additionally, we use a more sophisticated scale-space consisting of 5 octaves and 2 scales in a Gaussian pyramid, and we compute feature contrasts in intensity and color channels (orientation is less useful for salient object detection because it highlights the boundaries of objects, and this makes it more difficult to select salient blobs from the saliency map). The saliency maps we obtain are quick to compute (about 60 ms per $640 \times 480$ frame on a standard desktop computer on unoptimized code), don't have a center or border bias, and are detailed enough to let us obtain good estimates of the location and extent of objects: see for example the saliency map in Fig. 3.

For the second step, the extraction of salient blobs from the saliency map, we determine the set of local maxima $\{l_1, ..., l_n\}$. A local maximum is here a (collection of) pixel(s) which is larger than all neighboring pixels. Next, for each local maximum in the saliency map $l = (x_l, y_l)$, where $(x_l, y_l)$ are the pixel coordinates of the point, we do seeded region growing [1] to obtain a salient region $s_l$. The region growing recursively investigates all neighbors of $l_i$ and adds them to the salient blob $s_l$, as long as the saliency of the pixel is above some percentage of the saliency of the seeding point $sal(x_l, y_l)$. Thus, for every candidate point $p = (x_p, y_p)$, we compute whether $sal(x_l, y_l) \geq sal(x_p, y_p) \geq sal(x_l, y_l) \times t$, with $0 < t < 1$. This procedure is repeated for different values of $t$ (we use 0.6 and 0.7), and the complete set of salient regions $\{s_1, ..., s_m\}$ is stored for the next step.

### B. Segmentation

In parallel to the salient region extraction, the original image and depth data are segmented. Here, we present four alternatives for segmentation: color segmentation, surface clustering on the depth data, RGB-D supervoxel extraction, and a combination of the first two methods. Since our main approach differs depending on the segmentation we use, we call the four variants of our method accordingly $M1 - M4$. To illustrate the candidate generation, we show in Fig. 4 the results for our four methods on a sample frame from sequence 3 of the Washington dataset. In the following, we describe these variants in more detail.

*1) Method 1 (M1): Felzenszwalb Segmentation:* We chose the Felzenszwalb and Huttenlocher algorithm [5] for seg-

[2] Description and code: http://www.iai.uni-bonn.de/∼frintrop/vocus2.html

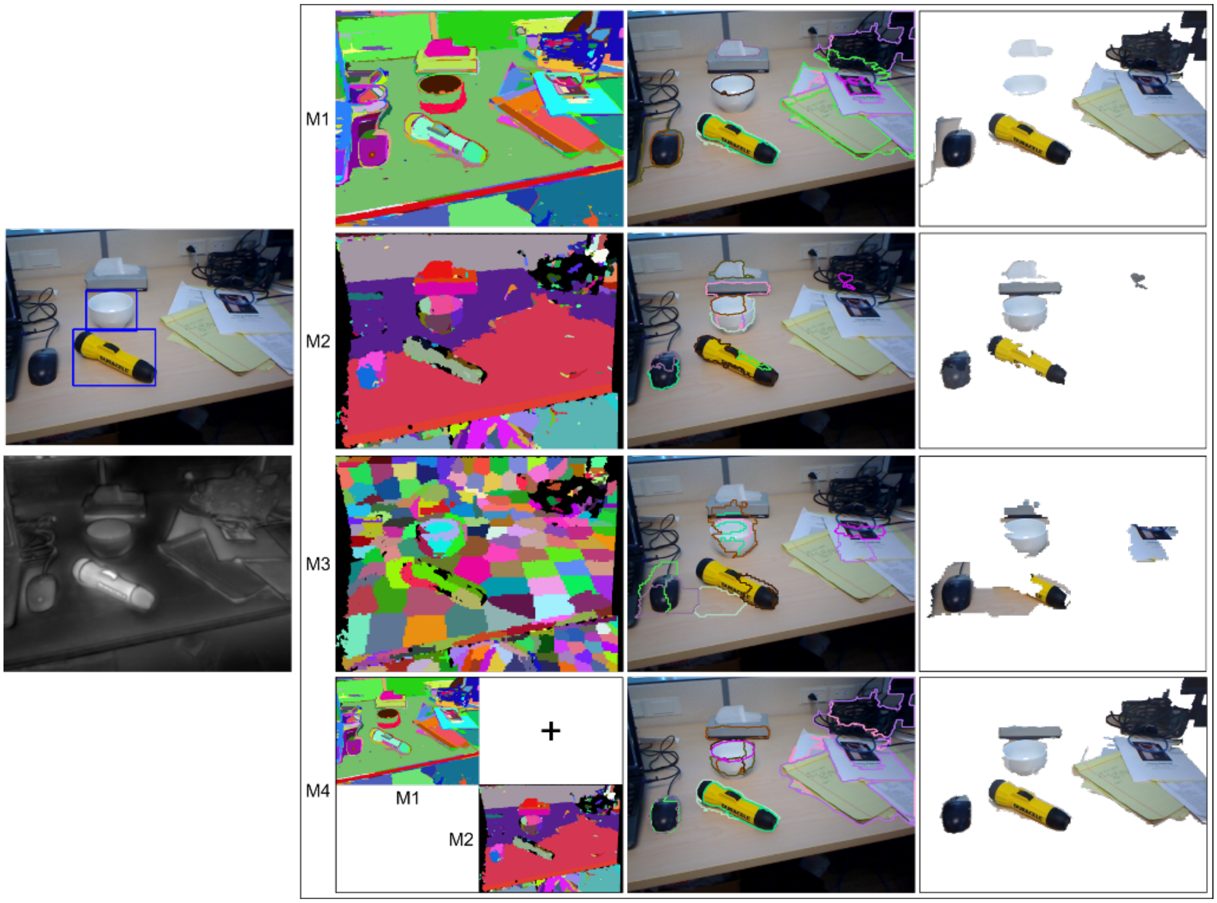

Fig. 4. Left side: the original image with ground truth. Below, the saliency map. The four rows on the right correspond to the top 10 candidates according to the SVM ranking for each of the methods M1, M2, M3, M4. The first column shows the segmentations, the second displays the contours of the object candidates, and the third, the actual candidates.

menting color images into perceptually coherent segments. The authors proposed a method that constructs a graph based on the pixel neighborhoods, and iteratively merges groups of pixels into regions, keeping a trade-off between the internal variability of the regions and the difference between neighboring components. Therefore, it relies mainly on one parameter, $k$, that determines the scale of observation. We set it in all our experiments to 200, to slightly over-segment the images. The candidates that we obtain with this method are shown exemplarily in the first row of Fig. 4.

*2) Method 2 (M2): Surface Clustering:* Here, we use a method that purely relies on depth information to produce image segments. Similar to Richtsfeld et al. [26] we cluster neighboring points into uniform planar patches without discontinuities based on their normals. Normal clustering starts at the point with lowest curvature and greedily assigns neighboring points as long as they fit to the initial plane model. The algorithm iteratively creates planar surface patches until all points belong to some plane or are identified as noise. An example of the segments and the candidates obtained with this method is shown in Fig. 4.

*3) Method 3 (M3): Supervoxel Segmentation:* As the third segmentation method, we used Voxel Cloud Connectivity Segmentation [22] that generates volumetric over-segmentations of 3D point cloud data[3]. The algorithm implements a local region growing variant of k-means clustering which incrementally expands supervoxels from a set of seed points distributed evenly in space. Expansion from the seed points is based on a measure consisting of distance, color, and normal similarity.

*4) Method 4 (M4): Fusion of Color and Depth candidates:* As an alternative to method M3, we propose to merge together the candidates obtained by methods M1 and M2. As we will show in the evaluation, color and depth candidates are complementary, so that we obtain better results with this late-fusion method than with the early fusion in M3.

### *C. Candidate Generation: Saliency + Segmentation*

Now, we have salient blobs and segments and we have to determine which segments form an object candidate. The selection of segments works in the same way for each of the segmentation methods: for each salient region $s$, we pick the segments which overlap at least $o\%$ with $s$. We set this overlap to $o = 30\%$ w.r.t. the segment.

[3] We used the implementation from the PCL library `http://pointclouds.org/documentation/tutorials/supervoxel_clustering.php`

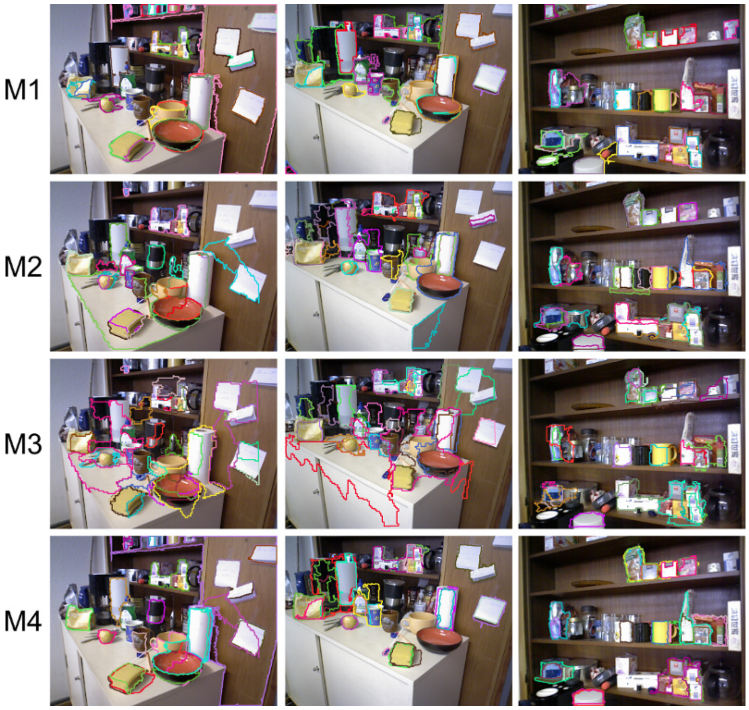

Fig. 5. The top 30 object candidates from M1,M2,M3 and M4 for several frames of the Coffee Machine sequence.

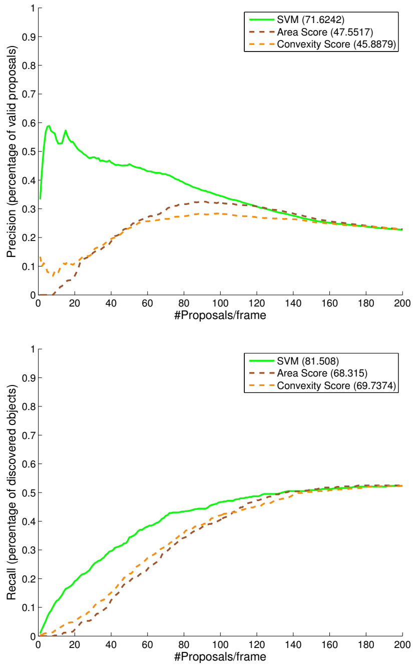

Fig. 6. Precision and recall curves for M4 using the three ranking methods on the Coffee Machine Sequence. Numbers in parentheses denote the AUC values.

To summarize the steps: first, saliency is computed on the input image; second, salient regions are extracted from the saliency map that roughly determine the extent and location of the objects; finally, the salient regions are used to glue together the segments obtained from one of the four different segmentation algorithms.

### *D. Ranking of the candidates*

A critical issue is how to rank the object candidates to be able to select the best ones. As mentioned before, this is important especially in robotics applications to meet real-time constraints and to select the most promising candidates for interaction. A set of 1000 candidates is just impractical to operate on.

We investigated three different approaches for ranking the object candidates. First, we used the average saliency of a candidate for ranking. To avoid a size bias towards small objects which have naturally a higher average saliency, we incorporate the size of the candidate, $p$, into the ranking score: $score(p) = \text{avg\_saliency(p)} * \sqrt{area(p)}$. We call this ranking method (*R1*).

Our second ranking approach (*R2*) sorts candidates according to their 3D convexity: convex candidates get higher score than non-convex ones. The convexity is computed according to [23]: given a set of object points $\{p_i\}$, $V$ is the corresponding object's convex hull, and $v_j$ is a set of visible faces from the current viewpoint. Convexity measure $\kappa$ is calculated as the mean of the shortest distances from the object points to the visible surfaces of the object's 3D convex hull:

$$\kappa = \frac{1}{n} \sum_{p_i} d_{min}(p_i, V), \tag{1}$$

where $n$ is the number of object points and $d_{min}(p_i, V)$ is the shortest distance from the point to any visible face

$$d_{min}(p_i, V) = \min_j d(p_i, v_j). \tag{2}$$

The lower the convexity measure, the more convex the object proposal is.

Our third approach (*R3*) ranks object candidates using several features extracted from the candidate mask: (1-7) Hu's image moments [12], which are invariant to rotation and scale; (8) 3D convexity measure (described above); (9) object proposal area normalized to the image area; (10) average saliency of the proposal; (11) perimeter of the object proposal mask normalized to the image area; (12) normalized average depth of the proposal. Given this set of features we trained a support vector machine (SVM) [3] to classify between object/non-object. Training was done on the ground truth annotated scenes of the Washington dataset. For every feature vector, the SVM then outputs the probability of the object candidate being object/non-object. This probability is used as a ranking score to sort candidates. To train the SVM, the Washington dataset was divided into two parts. One part was then used for training and the other for testing and vice versa.

## IV. Evaluation

We evaluate all the methods on one publicly available dataset, the Washington Dataset [18], and one of our own, the

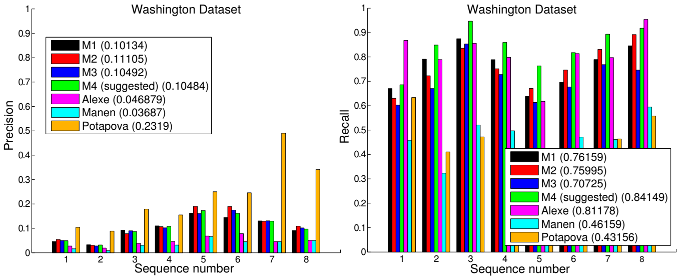

Fig. 7. Average recall (left) and precision (right) values on the Washington dataset. Numbers in parentheses denote the average recall/prec. over all sequences.

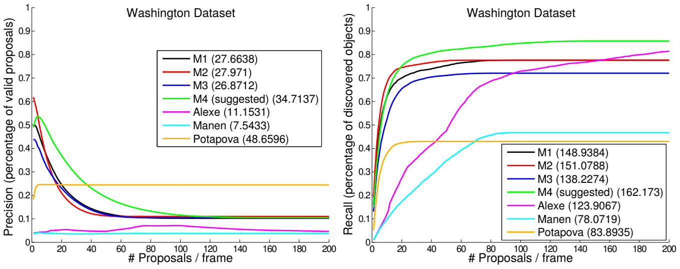

Fig. 8. Precision and recall values over number of proposals/candidates on the Washington dataset. Numbers in parentheses denote the AUC values.

Coffee Machine sequence, which first appeared in [9]. The latter is a challenging scene for object discovery with high clutter, and a total of 80 distinct objects appearing throughout the sequence and up to 43 objects per frame. It lasts for 436 frames, and ground truth was annotated on every 30th frame. Some frames with the object candidates generated with our four methods are depicted in Fig. 5. The Washington dataset [18] contains 8 sequences recorded with a Kinect camera on household environments, and is intended to test object recognition algorithms. Thus, it contains labeled ground truth where different object classes/instances appear, and serves to evaluate our generic object candidates. Note, however, that not all the objects that appear are labeled. The ground truth is provided as bounding boxes, so, in order to measure the overlap we fit a bounding rectangle on each object mask we generate. Although the dataset was not designed for object discovery it is to our knowledge the most suitable RGB-D dataset with annotated ground that is freely available.[4]

On both datasets, we measure precision as the number of correct object candidates over the total number generated, and recall as the number of correct candidates over the total present in the ground truth. We consider candidates as correct if they satisfy the Pascal criterion, i.e.,intersection over union is greater than 0.5.

### *A. Ranking of the Candidates*

In the first part of the evaluation, we compare the three ranking methods explained in Section III-D. Namely, the saliency/area score (*R1*), the convexity score (*R2*), and the SVM score (*R3*). To evaluate *R3*, we split the Washington sequences into two sets of four sequences each, and used one for training and the other for evaluation. We used the model learned in the first set to evaluate in our own Coffee sequence.

We show the results obtained in the Coffee sequence for our method M4 in Fig. 6. The results obtained in the other sequences were analogous. The effect we expect by a better ranking is that precision and recall values are increased for a smaller number of candidates. This happens mostly for *R3*. The convexity score showed some small improvement in the Coffee sequence and slightly worse performance than *R1* throughout the Washington dataset. It can be seen in the precision plot how good candidates are chosen first with the SVM ranking method; this is an effect that also occurs in the Washington dataset. As a result, the recall curve raises faster as well.

In the remaining evaluation, we used the ranking of our object candidates that turned out to be the best, R3.

### *B. Washington Dataset*

Here, we show our results on the Washington dataset [18]. We compare our four proposed methods, M1 (color), M2 (depth), M3 (RGB-D), and M4 (color+depth), to the method of Potapova et al. [23], to the objectness measure of Alexe et al. [2], and the method of Manén et al. [19].

We show in Fig. 7 average precision and recall obtained in each of the sequences by all methods. The results show that, with an average recall of 84%, our method M4 clearly outperforms all the other methods. Even when only using color (M1) or only depth (M2), the approaches outperform the other methods clearly (76% vs 75%), except the objectness measure of Alexe et al. (81%). This is due to the high number of candidates that objectness generates (1000). In terms of precision, the method of Potapova et al. [23] turns out to be the best: it produces very few object hypotheses but these are often correct. For the other methods, all values are quite low and similar to each other. The generally low precision values come partly from the fact that few objects are present in the scenes, and not all objects are labeled as ground truth in this dataset (e.g. in Fig. 4, the mouse is detected but not labeled as ground truth, resulting in a false detection).

When looking at the results more closely, the plots show that, in terms of recall, the color segmentation approach (M1) is better than the one using depth clustering in four sequences, and the other way round in the other four. The combination of M1 and M2 object candidates in M4 shows that to some extent, color and depth are complementary, and boosts the recall in every sequence, outperforming the RGB-D segmentation approach of M3. M4 has the highest recall in six out of eight sequences.

A complementary view for this results is shown in Fig. 8, where precision and recall values are plotted over the number of object candidates. It averages over all eight sequences in the Washington dataset. This is useful for deciding how many object candidates to generate. There, one can see that by

[4] There is also the recent dataset of [17], but the sequences do not add difficulty to the task of object discovery.

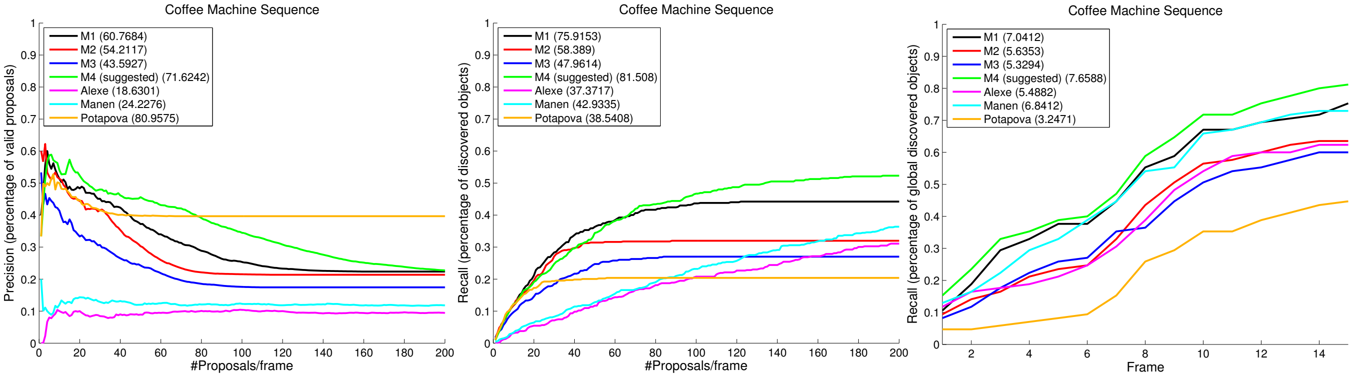

Fig. 9. Coffee Machine sequence results. Left and middle: precision and recall over number of proposals/candidates. Right: global number of discovered objects over time. Numbers in parentheses denote the AUC values.

taking about 20 candidates from method M4, approximately 75% of the objects are detected.

### C. Coffee Machine Sequence

The high recall obtained by most methods in the Washington dataset shows that the benchmark is relatively easy. Thus, we evaluate all the methods in a sequence [9] that contains many more objects (on average 36 per frame, some frames have up to 48 objects) and plenty of clutter.

As before, we show in Fig. 9 (left+middle) the precision and recall values over the number of candidates. The difficulty of the sequence is reflected in lower recall values than for the Washington dataset for all the methods. Despite this difficulty, all methods achieve a considerably higher precision (e.g., for M4 an AUC of 71 vs. AUC 34 in Fig. 8). This is due to the fact that in this dataset, all objects were labeled for the ground truth (cf. argumentation in Sec. IV-B).

In Fig. 9 (right) we show an additional plot that shows the number of discovered objects over time: for this, the identity of the objects in the scene is kept consistent in the ground truth. That means, we consider the amount of objects in the scene rather than in individual frames. This is interesting since for many applications it is not necessary to detect every object in every frame, but it is sufficient to detect an object in one of several frames. The values are achieved by computing up to 200 candidates/frame for each of the methods. The plot shows that after 15 frames, the M4 method retrieved about 80% of all the objects in the scene.

The top 30 candidates from M1 to M4 are shown in Fig. 5 for several frames. There, one can see examples of objects that would not be retrieved by the depth clustering method, for example the notes on the wall. Also, objects that are far away from the camera are a challenge to the surface clustering algorithm. Especially in those cases, visual information can be of help. On the other hand, some of the depth-based object candidates have more precise boundaries: see for example the sponge on the lower right part of the image in the middle column. The M1 method includes a shadow as part of the object, whereas this boundary is clearly defined for the M2 method. As for M3, the boundaries of the objects are not as precise as in the color and depth methods.

## V. Conclusion

We have presented a method for object discovery that is based on saliency and segmentation and that works on single RGB-D frames. It relies on a pre-segmentation of the scene by either using color, depth information, or both. Given an input frame, it produces a set of object candidates that can be used for either recognition, or interaction of a robot. We show that a method that treats color and depth segments separately improves results w.r.t. both methods independently, which suggests that both modalities are complementary. Also, the results were better than when using a single segmentation approach that integrates color and depth. In order to have the best candidates first, we proposed and evaluated three different approaches for ranking them. We showed that, especially in sequences with high clutter, our algorithm clearly outperforms state-of-the-art approaches for object discovery.

## References

[1] R. Adams and L. Bischof. Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 16(6):641 – 647, 1994.

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

[3] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):1–27, May 2011.

[4] Alvaro Collet, Bo Xiong, Corina Gurau, Martial Hebert, and Siddhartha S. Srinivasa. Exploiting Domain Knowledge for Object Discovery. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision (IJCV)*, 59(2), 2004.

[6] Jie Feng, Yichen Wei, Litian Tao, Chao Zhang, and Jian Sun. Salient Bbject Detection by Composition. In *IEEE International Conference on Computer Vision ICCV*, pages 1028–1035, 2011.

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

[8] Simone Frintrop, Germán Martín García, and Armin B Cremers. A cognitive approach for object discovery. In *ICPR*, 2014.

[9] 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.

[10] Germán Martín García, Simone Frintrop, and Armin B. Cremers. Attention-based Detection of Unknown Objects in a Situated Vision Framework. *German Journal of Artificial Intelligenz, Springer*, 2013.

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

[12] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[13] 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.

[14] M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic. Attention-based Active 3D Point Cloud Segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[15] A. Karpathy, S. Miller, and L. Fei-Fei. Object discovery in 3D scenes via shape analysis. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[16] Dov Katz, Moslem Kazemi, J. Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[17] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3d scene labeling. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[18] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

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

[20] A David Milner and Melvyn A Goodale. Two visual systems reviewed. *Neuropsychologia*, 46(3):774–785, 2008.

[21] Ajay K. Mishra and Yiannis Aloimonos. Visual Segmentation of Simple Objects for Robots. In *Robotics: Science and Systems (RSS)*, 2011.

[22] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Wörgötter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[23] Ekaterina Potapova, Karthik M Varadarajan, Andreas Richtsfeld, Michael Zillich, and Markus Vincze. Attention-driven object detection and segmentation of cluttered table scenes using 2.5 d symmetry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[24] Esa Rahtu, Juho Kannala, and Matthew B. Blaschko. Learning a Category Independent Object Detection Cascade. In *IEEE International Conference on Computer Vision, ICCV*, pages 1052–1059, 2011.

[25] R.A. Rensink. The Dynamic Representation of Scenes. *Visual Cognition*, 7:17–42, 2000.

[26] Andreas Richtsfeld, Thomas Morwald, Johann Prankl, Michael Zillich, and Markus Vincze. Segmentation of unknown objects in indoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[27] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard. Unsupervised Learning of 3D Object Models from Partial Views. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[28] D. Schiebener, A. Ude, and T. Asfour. Physical Interaction for Segmentation of Unknown Textured and Non-textured Rigid Objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[29] B.J. Scholl. Objects and Attention: the State of the Art. *Cognition*, 80:1–46, 2001.