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.

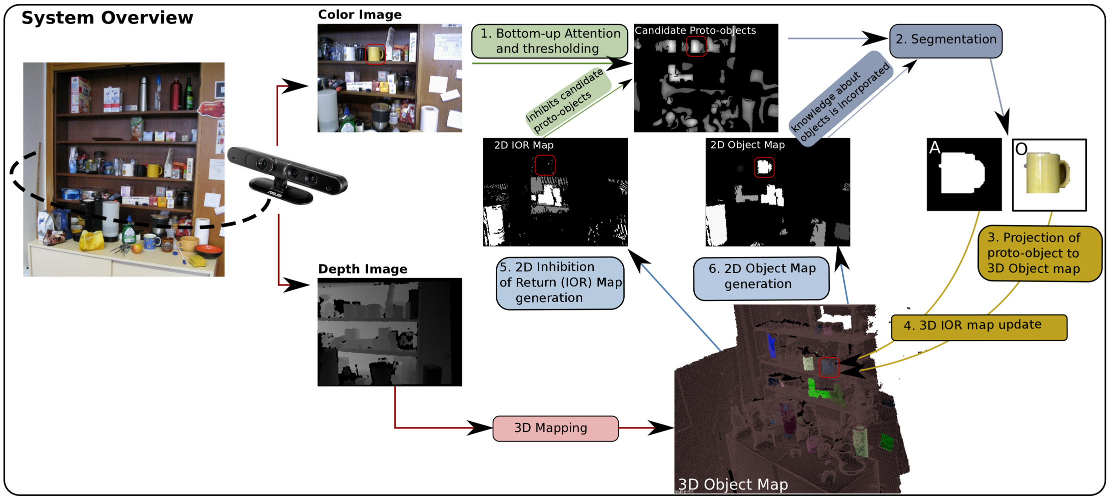

System Overview
Color Image
1. Bottom-up Attention and thresholding
Candidate Proto-objects
2. Segmentation
Inhibits candidate proto-objects
Knowledge about objects is incorporated
2D IOR Map
2D Object Map
A
O
3. Projection of proto-object to 3D Object map
5. 2D Inhibition of Return (IOR) Map generation
6. 2D Object Map generation
4. 3D IOR map update
Depth Image
3D Mapping
3D Object Map

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.

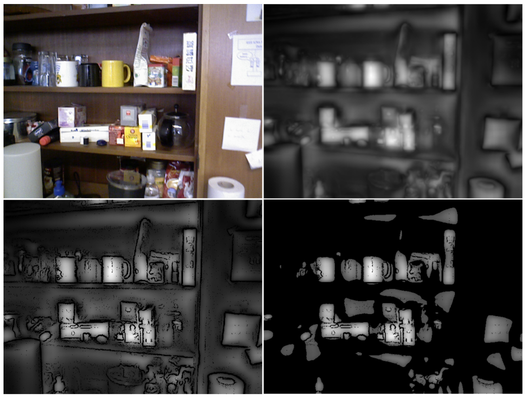


Figure 2: Top left to bottom right: original RGB image; its corresponding saliency map SM; saliency map after adaptive thresholding SM'; the SM'' map after the final thresholding.

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/

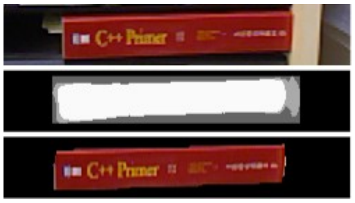


Figure 3: Top: a book as example object. Middle: initialization of GrabCut, the grayscale values correspond to the four possible likelihoods FG (white), PR_FG (light gray), PR_BG (dark gray), and BG (black). Bottom: the segmentation result.

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

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

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 $\text{map}: p \in \mathbb{Z}^2, T \in \mathbb{R}^4, D \in \mathbb{Z}^{m \times n} \rightarrow c \in \mathbb{Z}^3$, 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:

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

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.

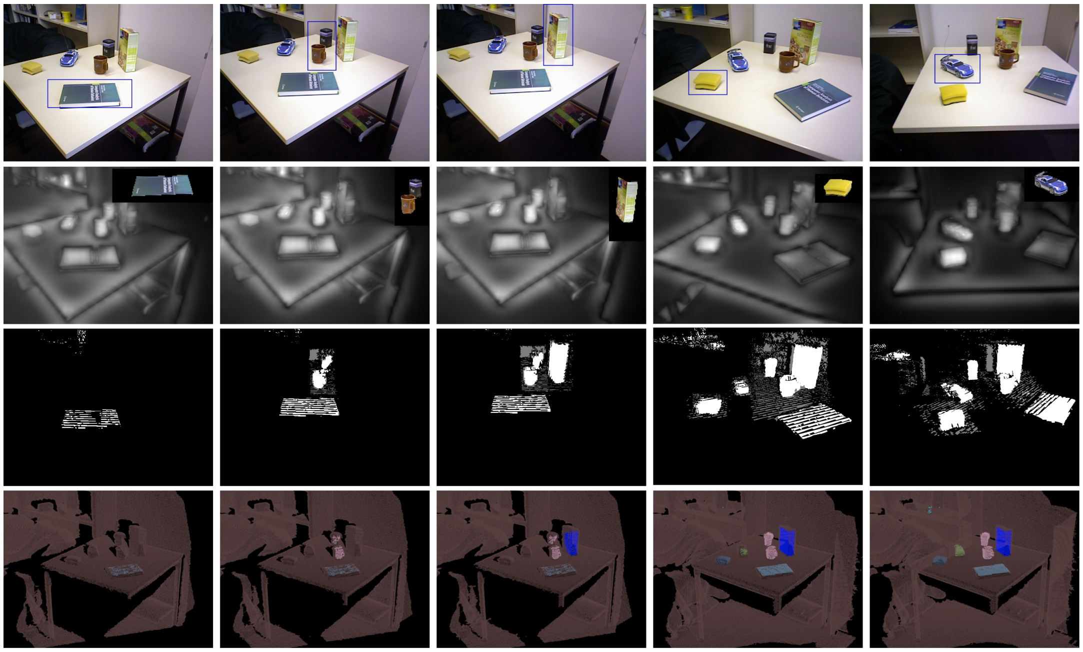


Figure 4: Table Top sequence at different points in time (columns). From top to bottom: (i) image of the scene with currently attended object (blue rectangle); (ii) the saliency map and the segmented part from the currently attended object; (iii) inhibition of return maps; white: object-based IOR, gray: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

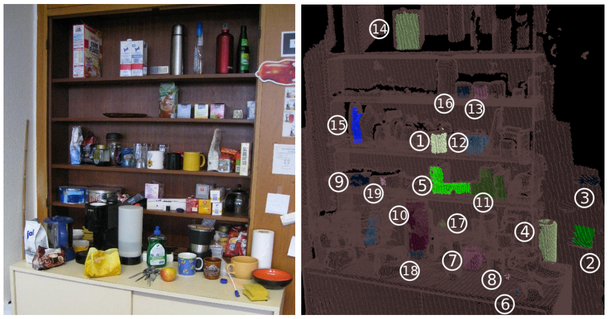
To evaluate our system we recorded two video sequences in an office environment with an RGB-D camera that provides depth as well as color information. The first sequence shows a setting of objects on a table top (cf. Fig. 4). The complexity of this setting corresponds to the complexity of scenes in current 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 (~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.