ROBUST ONLINE SEGMENTATION OF UNKNOWN OBJECTS FOR MOBILE ROBOTS

Xin Wang, Maja Rudinac and Pieter P. Jonker
Delft Biorobotics Lab, Delft University of Technology, Delft, The Netherlands

Keywords: Unknown Environment, Saliency Detection, Tracking, Online Object Segmentation, Mobile Robots, Convergent Vision System.

Abstract: In this paper we present a novel vision system for object-driven and online learning based segmentation of unknown objects in a scene. The main application of this system is for mobile robots exploring unknown environments, where unknown objects need to be inspected and segmented from multiple viewpoints. In an initial step, objects are detected using a bottom-up segmentation method based on salient information. The cluster with the most salient points is assumed to be the most dominant object in the scene and serves as an initial model for online segmentation. Then the dominant object is tracked by a Lucas-Kanade tracker and the object model is constantly updated and learned online based on Random Forests classifier. To refine the model a two-step object segmentation using Gaussian Mixture Models and graph cuts is applied. As a result, the detailed contour information of the dominant unknown object is obtained and can further be used for object grasping and recognition. We tested our system in very challenging conditions with multiple identical objects, severe occlusions, illumination changes and cluttered background and acquired very promising results. In comparison with other methods, our system works online and requires no input from users.

1 INTRODUCTION

One of the most challenging problems in robotics is the exploration of unknown environments. Robots need to be able to navigate in the environments, explore present objects and learn them online. The first problem that needs to be solved is how to efficiently localize the unknown objects in the environment and obtain their detailed information such as shape. This is necessary for tasks such as grasping, recognition or learning of unknown objects.

For the localization of unknown objects in a scene, no top-down knowledge can be used. Object detection methods based on point clouds calculated from stereo images (Björkman and Kragic, 2010) provide good results in the case of the textured objects. However, they fail in the case of objects with uniform color which are widely present in environments. As a solution to this challenging problem, we therefore consider bottom-up visual-attention methods. The saliency method presented in (Itti et al., 1998) was used, for instance, in (Rasolzadeh et al., 2010) to guide the attention of a robot. An attention method based on local symmetry in the image was proposed in (Kootstra et al., 2010) to fixate on objects in the scene. Finally, the method (Rudinac and Jonker, 2010) provides fast segmentation of objects based on their saliency. Since it assumes no prior information about the scene and only requires input from a single camera, we will further exploit it in the initial step.

Once, the initial position of object is calculated, the robots should be able to navigate around the objects to inspect them from multiple viewpoints. Therefore, very fast and robust object detection methods must be applied. There are various challenges the object detection method needs to cope with: scale changes, viewpoint changes, variable illuminations, occlusion and background clutter. Many object detection methods have been proposed and studied. The most popular ones are motion based tracking, background subtraction, feature based detection, color based detection and contour based detection. In the state of the art, the Adaptive Boosting Classification (Kalal et al., 2009) and sparse coding (Mairal et al., 2010) are extensively used in online active vision approaches. They use the initial model to generate training models to confront viewpoint changes as well as the occlusion. However, they need input from users and do not provide detailed information about the contour and the shape of the object.

In our application, we are interested in a mobile robot system that can autonomously explore un-

Wang X., Rudinac M. and P. Jonker P..
ROBUST ONLINE SEGMENTATION OF UNKNOWN OBJECTS FOR MOBILE ROBOTS.
DOI: 10.5220/0003866803650374
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2012), pages 365-374
ISBN: 978-989-8565-03-7

known environments. Therefore, an online detection method that allows automatic segmentation of unknown objects is indispensable. Most of the state of the art methods require user defined object model, which is unusable in our case. The robot has the task to navigate around the unknown objects to inspect them from different viewpoints. For this online segmentation task, existing background subtraction methods (Zivkovic, 2004) will fail because of a constant change of the background. Motion based online segmentation (Mooser et al., 2007) is not an option since the objects in the environment are static without any motion information. Thus a model based tracker which can update online is needed. However histogram based online segmentation such as Camshift (Bradski, 1998) can not handle textured objects. Therefore we require an object-driven segmentation method which is able to work in case of complex scenes and objects.

In this paper, we present a novel system for robust online segmentation of unknown objects which can overcome all above mentioned difficulties. The main contributions are as follows. Firstly, we implement a vision system that can autonomously perceive objects in unknown environments without any prior knowledge. Secondly, we propose a robust online segmentation method by utilizing different object detection methods in order to achieve a good performance in spite of viewpoint changes, illumination changes, background clutter as well as occlusion. Our method also provides refined information about the objects such as shapes and contours instead of only locations. Thirdly, in our setup the camera moves around the static objects, which is in contrast to most active vision applications where static cameras track or segment the moving objects. Furthermore, we tested the system on foveated vision setup and achieved very promising results.

This paper is organized as follows. In Sec. 2 we provide a general outline of the proposed system; in Sec. 3 we explain the algorithms in detail; in Sec. 4 we show system evaluation and results analysis.

2 GENERAL FRAMEWORK

In this section we present a general framework of the system, depicted in Figure 1. We propose several steps: detection of a dominant unknown object, initial model generation, tracking to update the object model and detailed object segmentation. We assume that no initial knowledge on a scene or objects is given.

In the initial step it is necessary to detect the approximate positions of unknown objects. For initial segmentation, we propose a bottom-up segmentation based on the salient information in the static scene. After the saliency map of the scene is calculated, saliency points in the map are detected and clustered into salient regions, where every region represents a potential unknown object. A cluster with the most salient points is assumed to be the most dominant object in the scene and its initial model is extracted to be used for later segmentation. Details can be found in Sec. 3.1 A camera is then maneuvered around the dominant object to explore it from different viewpoints. In each frame, the dominant object is tracked by motion based tracker, and the model of the object is rebuilt and constantly updated using Random Forests based classification. By combining the detection results of the motion tracking and the model tracking the location of the object in the new frame is derived. More detailed information is given in Sec. 3.2.1 In the final step, for every viewpoint and updated object model we do refined object segmentation. The Gaussian Mixture Models(GMMs) is used to create the object model and the background model. Finally the graph cuts is used to obtain the optimal segmentation as is described in Sec. 3.2.2. As a result, detailed contour information of the dominant object is extracted.

3 APPROACH AND IMPLEMENTATION

3.1 Salient Object Detection

In order to be able to learn novel objects in unstructured environments, an initial step is to correctly segment the objects without any prior knowledge about the objects or their background. In our previous research (Rudinac and Jonker, 2010), we proposed a method for fast object segmentation based on the salient information in the scene. In the original method (Hou and Zhang, 2007), saliency was detected using a spectral residual approach on three different color channels, red-green, yellow-blue, and the illumination channel. The saliency map was further calculated as the inverse Fourier transform of each spectral residual, and the results were combined to obtain a more robust saliency map. The bright spots in the saliency map represent points of interest. In order to detect those peaks, we applied the MSER blob detector (Matas et al., 2004) directly on the saliency map. Once the interesting points were detected, close points were clustered together using Parzen window estimation, leading to the segmentation of objects in

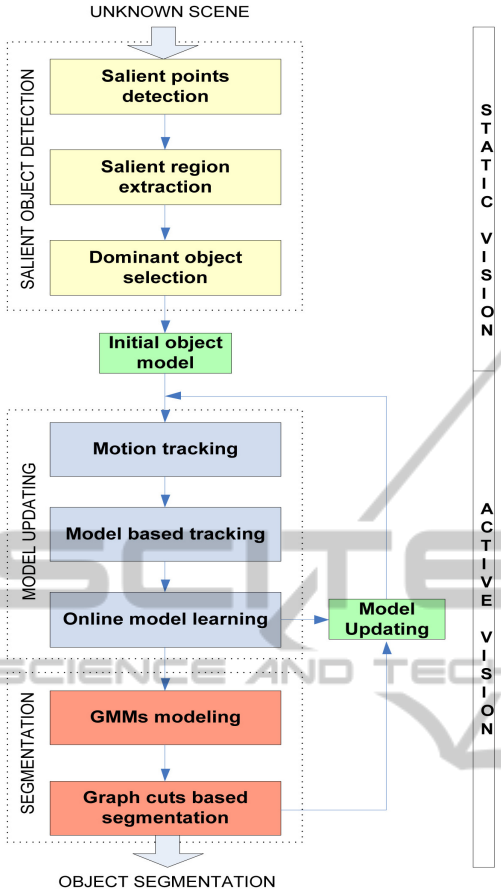
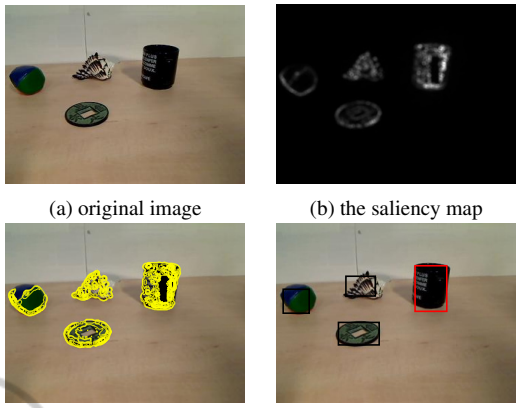


Figure 1: General outline of the system.

the scene.

The described method was designed for still images and here we propose an extension to process video. Given that the spectral residual process represents the difference between the original scene in Figure 2(a) and the scene average, acquiring information about the scene average from successive frames will improve the saliency map. The saliency map is displayed in Figure 2(b). Therefore, for each frame we detect MSER points on the saliency map in the standard way and merge the result with those from previous frames to obtain more stable salient points. In our setup we used 5 successive frames. The number of merging frames must be carefully chosen, since too many frames could lead to the segmentation larger than the object. To solve this problem, we use an active segmentation method in addition to the initial segmentation.

Once we obtained stable salient points from successive frames, for each detected point the contour describing the MSER region is calculated (Matas et al., 2004). The resulting contours can be seen as yellow points in Figure 2(c). These contours are then



(a) original image (b) the saliency map

(c) salient points (d) the dominant object

Figure 2: Initial localization of objects using saliency.

clustered leading to the segmentation of objects in the scene. For clustering, we use an adapted Parzen window estimation (Tax, 2001), which automatically fits a probability density function to the contour centers. For each point we calculate the probability $P(x)$ defined by Equation 1 where x_i and σ represent the Gaussian kernel center and the kernel size, while S is the number of contour centers and $m = 2$, since every contour center has a two-dimensional coordinate. Subsequently, outlier points that have low probability values and belong to isolated clusters are removed, as defined in Equation 2. Finally, the positions of the contour centers and their probability values are clustered using the Mean-shift method (Comaniciu et al., 2002). As a result, we find the regions of interest around each object in the scene, see Figure 2(d). The cluster with the most salient points represents the dominant region in the scene, the red bounded object in Figure 2(d) which will further be segmented.

$$P(x) = \frac{1}{S}\sum_{i=1}^{S}\frac{1}{\sqrt{2\pi^m\sigma^m}}\exp-\frac{\|x-x_i\|^2}{2\sigma^2} \quad (1)$$

$$\log(P(x)) < \log(\tfrac{1}{S}\textstyle\sum_{i=1}^{S}P(x_i)) - 3var(\log(\tfrac{1}{S}\textstyle\sum_{i=1}^{S}P(x_i)) \quad (2)$$

3.2 Online Object Segmentation

In the Sec. 3.1, we proposed a method which segments the unknown objects in the scenes and selects the most dominant one that will further be inspected by a robot from multiple viewpoints. Based on the location of the initial model, the robot should develop a self-learning system by observing objects from different perspectives and perceive its environment without any prior knowledge. One of the necessary steps towards such a system is an object-driven and on-line

learning segmentation method. In our application, objects are static while the robot navigates around objects to explore them from different viewpoints. Pure motion based and background modeling based online segmentation methods will fail in this situation. In this paper, we propose a robust online object segmentation method to cope with this situation. From the initial position located by saliency, we build up the object model using texture features and update the model frame by frame to efficiently track the object. Then we segment the interested object inferred from the model using GMMs and graph cuts. We will now explain the two steps for tracking and segmentation.

3.2.1 Build and Update the Object Model

With respect to the task of observing objects from different viewpoints, we need to online build up a training data set to model the object from initial object information and update the model so that it can adapt to the constant change in object appearance. Both methods (Kalal et al., 2010) and (Lepetit and Fua, 2006) for adaptive online tracking use Local Binary Pattern (LBP) variants to represent the texture of the object. The LBP features are randomly distributed on an image patch, thus the spatial information among the features is kept. Then the image patches are used to train a Random Forests classifier (Breiman, 2001). Therefore the object tracking problem turns into a foreground and background classification problem. The drawback of (Lepetit and Fua, 2006) lies in that it needs to offline generate an affine transformation training data set from the original image to build up the tracking model. (Kalal et al., 2010) goes a step further and just requires a user defined bounding box around the object and further updates the model online. However, they do not provide any detail on the object shape. In our system we propose a fully automatic system which also provides shape information.

Assuming that we have an object model M that contains a variety of model elements $(m_1, m_2, \ldots, m_N)$, each m_i uses a group of features $(f_{i1}, f_{i2}, \ldots, f_{iK})$ to encode the different appearance of the object. The combination of model elements can provide a more comprehensive and robust description of the object than a single model element. Using probability theory we deduce the probability of features based on a given object model element $P(f_{i1}, \ldots, f_{ik}|o_i), i = 0, 1, \ldots, N$.

Given a potential candidate C, we use

$$P(C) = \prod_{i}^{N} P(c_i|f_{i1}, \ldots, f_{iK}) \tag{3}$$

to denote the classification of C based on features.

According to the Bayes Theorem

$$P(c_i|f_{i1}, \ldots, f_{iK}) = \frac{P(f_{i1}, \ldots, f_{iK}|c_i)P(c_i)}{P(f_{i1}, \ldots, f_{iK})} \tag{4}$$

We assume the uniform prior $P(c_i)$ and the denominator to be the normalized constant to ensure that the sum of probabilities is one.

Then Equation 3 transforms into

$$P(C) \propto \prod_{i}^{N} P(f_{i1}, \ldots, f_{iK}|c_i) \tag{5}$$

Since we have the criterion to denote the object

$$P(O) = \prod_{i}^{N} P(f_{i1}, \ldots, f_{iK}|o_i) \tag{6}$$

We can assign C to the classification of object or background. Random Forests have the structure of fast and generalized classification, thus we use it to build and update the model. Here, the model elements are represented by trees and the features are nodes of the trees.

First we cover the input salient region with an image patch $x_0 \in X$, where $X = \{x_t, t = 0, 1, \ldots, T\}$ depicts the trajectory of the object, in which t is the frame number increased by time. We use LBP as local texture feature descriptor and randomly generate the features on the image patch to maintain the spatial information, therefore we have the first object model and features distribution $P(f_{i1}, \ldots, f_{ik}|o_i), i = 0, 1, \ldots, N$. We can then construct the Random Forests which has N trees. By using the Lucas-Kanade tracker, the new location of the image patch and scale of the object in the new frame are known. Every new frame is scanned using an image patch. Within every image patch we use the generated features to compare with the model. From the viewpoint of Random Forests, the search is carried out for each tree and if the search reaches the leaf the image patch is considered to be a potential object according to the given model element. Finally we use majority votes from all the trees to decide if it is a confident object. Among all confident objects in the frame, we select the most confident ones and cluster them by distance measurement using normalized cross-correlation. Then by combining the image patch location and scale obtained by Lucas-Kanade tracking and the image patch location obtained by detector we derive the image patch of object x_t in the new frame.

Updating the model is an online learning procedure to cope with viewpoint changes. If the image patches detected by the detector are close to the object, they are considered to be a positive data set and add to the branch of the trees, otherwise they will be treated as a negative data set and pruned from the

trees. In this way, a robust and "memorized" model is updated.

3.2.2 Refining of the Object Model by Segmentation

Although the position of the object is known, the information about its contour, edge or shape is still unknown. In our application, the object segmentation will be a cue for further tasks such as the object recognition, scene understanding, object grasping as well as convergent vision, and therefore a detailed contour of the object is necessary. For these reasons we need to further refine the object model and perform detailed segmentation. With respect to existing segmentation methods in which most of them need interaction from users (Rother et al., 2004) and (Vezhnevets, 2005). In order to automatize the process we use the object model from previous part and to decrease the computation time the segmentation is not carried out frame by frame. Object segmentation is performed only in key frames while for other frames, we use the confident segmentation from previous frame. The key frame is determined by comparison of the current image patch x_t with the previous image patch x_{t-1}. If the displacement and the scale difference are larger than a specified threshold, the frame t is considered to be a key frame.

We opt for a use of *RGB* color images and in the object modeling part, we combine both texture information of an intensity image and color information . We first apply the hard constraints to label the image and then use soft constraints to optimize the segmentation.

The task of the hard segmentation is to split the scene into an object and a background and we adopt the GMMs for a construction of the object and background models. The GMMs is a linear combination of Gaussians that gives complex densities and better characterization than histogram based methods, thus it provides good performance even when the object has complicated texture and color. For a known image patch x_t calculated by previous steps, we assume that within the image patch the properties of the object are preserved, while all pixels outside the patch have the attributes of background. Based on this, we derive the object GMMs and background GMMs in a following way.

With regards to a pixel $x_p, p = 1,2,\ldots,P$, the GMMs are defined as

$$P(x_p) = \sum_{k=1}^{K} \pi_k N(x_p|\mu_k,\Sigma_k) \tag{7}$$

where Gaussian density $N(x|\mu_k,\Sigma_k)$ is called one component with mean vector u_k and covariance matrix Σ_k. π_k is the weight. Here the mean vector u_k is composed of three values R, G and B while K is the number of components. K needs to be adapted to the scene, and more textured scenes require higher values of K. Typically $K = 5$.

Since we have the initial model, we can assign each pixel to each component in object GMMs and background GMMs. Therefore we have the label for all the pixels in the image.

After hard segmentation, we use energy minimization to optimize the segmentation. The energy minimization equation is

$$\begin{aligned} E(L) &= \lambda R(L) + B(L) \\ &= \lambda \sum_{p\in P} R_p(l_p) + \sum_{(p,q)\in N} B_{(p,q)} \cdot \delta(l_p,l_q) \end{aligned} \tag{8}$$

where $L = (l_1,\ldots,l_p,\ldots,l_P)$ is the label set for each pixel. $l_p = 1$ represents that p is assigned to object and $l_p = 0$ represents that p is assigned to the background. q is one of neighboring elements of p and $\delta(l_p,l_q)$ is defined as

$$\delta(l_p,l_q) = \begin{cases} 1 & \text{if } l_p \neq l_q \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where

$$R_p(l_p) = -\log P(x_p) \tag{10}$$

describes the region property based on GMMs models.

$$B_{(p,q)} = \exp(-\beta||I_p - I_q||^2) \tag{11}$$

describes the coherence of similarity within a region according to a distance between two pixel. Where I_p is the *RGB* value for a given pixel. λ is a parameter that relatively balance region property based on GMMs versus region property based on similarity.

Segmentation can be now estimated as a global minimization using graph cuts (Boykov and Jolly, 2001)

$$c = \arg\min_{L} E(L) \tag{12}$$

Then we have the foreground object and background. It is worth noting that the computation cost of segmentation using graph cuts will be a challenge for online applications. In our case, we confine background to be a region surrounding the image patch instead of using the region of the whole image. By doing this, we lower down the computation cost. We also use the output of the segmentation result as a refined input of the online model for more precise tracking.

4 TESTING AND RESULTS

4.1 Experimental Setup

In order to test the whole system, we used ground truth data obtained from 4400 image frames in 4 different test scenarios. They are following: a single object placed in the scene with uniform background, multiple objects placed in the scene with uniform background, a single object placed in the scene with textured background and multiple objects placed in the scene with texture background. We used different objects which varied in shape (simple vs complex) and in appearance (uniform color vs textured). It is also worth noticing that all of the experiments were carried out in different illumination conditions with natural light as well as artificial light. Moreover, we tested our system in difficult cases such as the objects with occlusion, as well as similar objects appearing in the same scene.

Here we also need to emphasize that in most state of the art online segmentation methods, the cameras are fixed to capture the motion of the objects in the scene. On the contrary, in our experiments the objects are static and the camera moves around the object, which is a more challenging case. There are two types of such active vision setups, one where the camera moves around the objects to "see" them from different viewpoints, and the other where the camera moves to keep the objects in the center of the view, so called foveated vision system. We performed experiment using both setups.

In both experimental setups, we used a Logitech Quickcam Pro 9000 with a resolution of $320 * 240$ and image capture rate 25 frames per second on x86 CPU at 2.8 GHz. We carried out two types of experiments, one with single camera moving around the objects and one with the camera fixed on top of motors in order to track the objects in the center of the view. For the latter, we adopted two Dynamixel RX-28 motors. In total the robot vision system has two degrees of freedom that can move the camera up and down, left and right. Figure 3 shows the setup of the whole system. As can be seen from the figure, in total there are two cameras and three motors since our future research is on convergent vision systems. For our experiment, we just used one camera controlled by two motors.

The input from saliency detection will influence how the object model is built up and updated and on the other, the input from the object model will affect the GMMs and further the graph-cut based segmentation performance. The three parts are strongly interrelated, and for that reason we present total segmentation results.

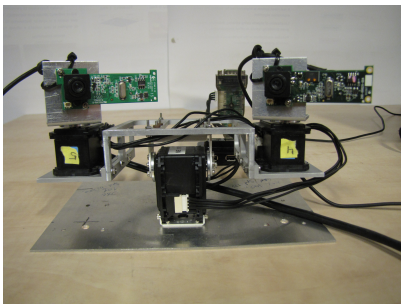


Figure 3: Convergent vision setup.

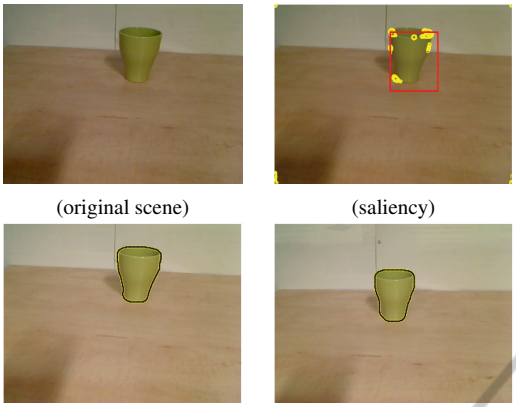
4.2 Saliency Detection and Online Segmentation Results

For testing we randomly picked up 50 objects with different color, texture and shape and put them in 88 different scenes with in total 4400 frames. Saliency detection selected 30 dominant objects from these different scenes. Since saliency provides segmentation of all objects in the scene, other objects could be inspected as well. Because the properties of the objects have influence on the segmentation results, we categorized the objects into 4 different types: objects with uniform color and simple shape, objects with uniform color and complex shape, objects with texture and simple shape, objects with texture and complex shape. Afterwards we used them in 4 different test scenarios as mentioned before.

In order to clearly demonstrate the performed tests, with regard to the types of objects and scenes, and to show the saliency detection and segmentation results, we show a number of figures with both single and multiple objects in the scenes. In each figure, we show the original image, the image after saliency detection, the image after object segmentation and one more example of the object segmentation from a different viewpoint. Figure 4 shows a single object with uniform color and simple shape in textureless scene, while Figure 5 depicts a single object with uniform color and complex shape in textured scene.

For the same reason, we also showed a number of figures of the multiple objects scenes. Figure 6 shows the textureless scene with multiple objects and the dominant object with texture and complex shape, while Figure 7 shows the textured scene with multiple objects and the dominant object with uniform color and complex shape.

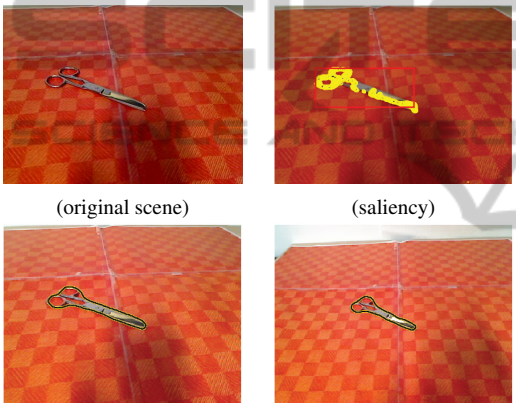
Table 1 presents the segmentation performance of a single object placed in an textureless or textured background. The rows represent the different types of objects and the columns the types of scenes. Table 2 shows the segmentation performance of multiple objects placed in textureless and texture environ-



(original scene) (saliency)

(segmentation1) (segmentation2)

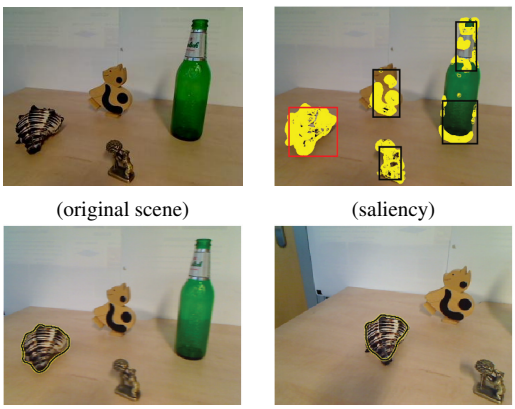
Figure 4: A single object with uniform color and simple shape in textureless scene.



(original scene) (saliency)

(segmentation1) (segmentation2)

Figure 5: A single object with uniform color and complex shape in textured scene.

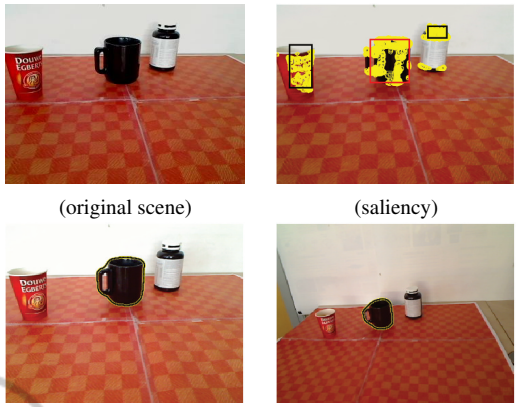


(original scene) (saliency)

(segmentation1) (segmentation2)

Figure 6: Textureless scene with multiple objects and the dominant object with texture and complex shape.

ment. Rows and columns are defined in a same manner as in Table.1. Both tables give the overall performance from all test frames. As can be seen from very high precision rates above 90%, the proposed algorithm gives a very robust segmentation of various types of objects in different scenes. We also come to the conclusion that in most cases, it is easier to segment the objects from textureless than from textured scenes and it is easier to segment the dominant object within single object background than multiple objects background. We can also notice that the multiple object cases show only a slight drop in precision rates. From the perspective of different types of objects, the uniform and simple shape objects make the task of saliency detection nontrivial. On the other hand, the objects with uniform color and complex shape increase the segmentation difficulty. Regarding very textured objects, saliency detection provides good results but in modeling an over-segmentation can occur, since the number of GMMs components might be low. The case of multiple objects with textured and complex shape is the most difficult one. However, our method gives a very good performance in all aforementioned situations, and even in case of large viewpoint changes.



(original scene) (saliency)

(segmentation1) (segmentation2)

Figure 7: Textured scene with multiple objects and the dominant object with uniform color and complex shape.

Besides testing the active vision of moving the camera around the objects, we also tested the foveated vision setup. We carried out experiments in 8 different scenes with various objects and in total 400 images. The test results show an overall precision rate of 95.5%, which proves effectiveness of the method on foveated active vision setup as well. One example is shown in Figure 8.

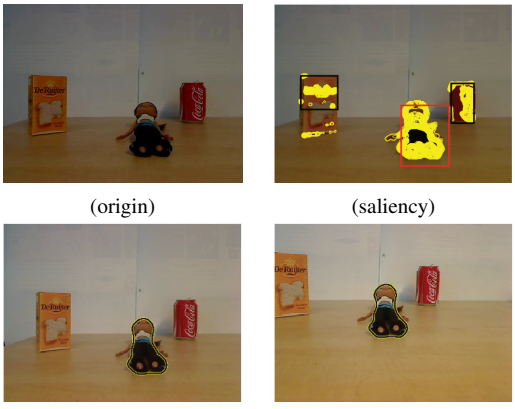
To test robustness of segmentation in more challenging conditions, we performed tests on similar objects appearing in the same scene, occluded objects as well as the motion of objects themselves. The testing result of perceiving objects from different viewpoints is shown in Figure 9. As we can see from this figure,

Table 1: Segmentation results of a single object placed in the textureless and textured scene.

objects vs scene	textureless %	textured %
uniform color and simple shape	98	96
uniform color and complex shape	98.4	93.6
texture and simple shape	98.4	96.4
texture and complex shape	98	92.8
total	98.2	94.7

Table 2: Segmentation results of multiple objects placed in the textureless and textured scene.

object vs scene	textureless %	textured %
uniform color and simple shape	94	97
uniform color and complex shape	98.3	93
texture and simple shape	95.6	90.4
texture and complex shape	90.8	86.4
total	94.68	91.7



(origin) (saliency)

(segmentation) (foveation)

Figure 8: Online segmentation results on foveated vision setup.

the algorithm has good segmentation performance despite the viewpoint changes. In Figure 10, regardless of occlusion, the algorithm can correctly extract the dominant object. Even with similar object occluded in front of the dominant object which is shown in Figure 11, the segmentation result is still good. And Figure 12 proves that the motion of the dominant object does not affect the performance.

During testing, we observed different situations that were difficult to cope with and those reduced the overall performance rate. We noticed that the segmentation results depend on the property of the object we choose. The transparent and reflective object normally give bad performance, as shown in Figure 13(a) and 13(b). The saliency detection will also affect the online segmentation results if the selected salient region only detects a part of the object, which could happen in the case of multiple object scenarios containing both uniform color and textured objects or if the objects are too close to each other. Another problem that rises, is in the case of very textured objects, the selected number of GMMs components might not be sufficient to efficiently segment the object. The failed case is shown in Figure 13(c). One way to solve this problem is to introduce the measure of the textureness of the object, since the more salient points detected usually means the more texture of the object. Then we can adaptively select the number of GMMs components according to this measure. We will investigate this solution in our future work. Finally, if the color or texture of the object is very similar to the background, it is difficult for the algorithm to extract it. Such example is shown in Figure 13(d). Also, sometimes the shadow might become a part of the object.

To conclude, the proposed method is very robust with very high performance rates above 90% in both textureless and textured scenes, and it can efficiently segment both simple and complex shapes as well as objects with uniform color or texture. Additionally, it can cope with viewpoint changes, similar objects appearing in the same scene, occlusion as well as the motion of object. Finally the system provides good results in both movable camera and foveated vision setups.

5 CONCLUSIONS

In this paper, we propose a novel vision system for robust online segmentation of unknown objects. Our system automatically detects unknown objects in the scene based on the saliency information, selects the most salient object, builds up and updates the object model online with movable camera, and finally refines object model using GMMs and graph cuts. The obtained outputs are the contours of the dominant object in different viewpoints. We tested our system in challenging conditions and the test results with the total segmentation precision above 90% show that our method performs well in spite of large viewpoint changes, illumination changes, occlusion as well as the case of similar object appearing in the same scene. The promising results inspire us to apply our system on mobile robots to autonomously explore, track and

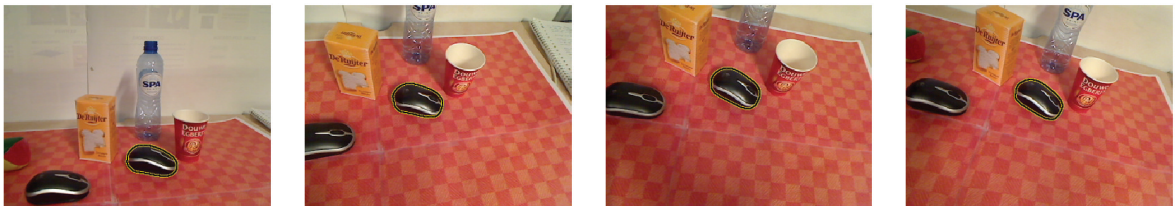


Figure 9: Online segmentation results with viewpoint changes.

Figure 10: Online segmentation results under occlusion.

Figure 11: Online segmentation results with similar objects appearing in the same scene and occlusion.

Figure 12: Online segmentation results with the motion of the dominant object.

(a) (b) (c) (d)

Figure 13: Failed cases.

segment unknown objects in unknown environments. The output of our system also provides a strong cue for further tasks such as object recognition, manipulation and learning. The test results on foveated vision setup also cast insight into convergent vision systems.

ACKNOWLEDGEMENTS

This research was sponsored by the Dutch government through the Point One project PNE09003 (Bobbie)

REFERENCES

Björkman, M. and Kragic, D. (2010). Active 3D scene segmentation and detection of unknown objects. In *ICRA*, pages 3114–3120.

Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105 –112 vol.1.

Bradski, G. R. (1998). Computer Vision Face Tracking For Use in a Perceptual User Interface. *Intel Technology Journal*, (Q2).

Breiman, L. (2001). Random Forests. *Machine Learning*, 45:5–32.

Comaniciu, D., Meer, P., and Member, S. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619.

Hou, X. and Zhang, L. (2007). Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR07). IEEE Computer Society*, pages 1–8.

Itti, L., Koch, C., and Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

Kalal, Z., Matas, J., and Mikolajczyk, K. (2009). Online learning of robust object detectors during unstable tracking. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1417 –1424.

Kalal, Z., Matas, J., and Mikolajczyk, K. (2010). P-N learning: Bootstrapping binary classifiers by structural constraints. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:49–56.

Kootstra, G., Bergström, N., and Kragic, D. (2010). Using symmetry to select fixation points for segmentation. In *Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23-26 August 2010*, pages 3894–3897.

Lepetit, V. and Fua, P. (2006). Keypoint Recognition Using Randomized Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1465–1479.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online Learning for Matrix Factorization and Sparse Coding. *J. Mach. Learn. Res.*, 11:19–60.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.*, 22(10):761–767.

Mooser, J., You, S., and Neumann, U. (2007). Real-Time Object Tracking for Augmented Reality Combining Graph Cuts and Optical Flow. In *Mixed and Augmented Reality*, pages 145 –152.

Rasolzadeh, B., Björkman, M., Huebner, K., and Kragic, D. (2010). An active vision system for detecting, fixating and manipulating objects in the real world. *The International Journal of Robotics Research*, 29(2-3):133–154.

Rother, C., Kolmogorov, V., and Blake, A. (2004). "GrabCut": interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 309–314, New York, NY, USA. ACM.

Rudinac, M. and Jonker, P. P. (2010). Saliency detection and object localization in indoor environments. In *Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23-26 August 2010*, pages 404–407.

Tax, D. (2001). *One-class classification*. phd, Delft University of Technology, Delft.

Vezhnevets, A. V. (2005). "GrowCut"-Interactive Multi-Label N-D Image Segmentation By Cellular.

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28 – 31 Vol.2.