An Object-driven Online Segmentation System for Mobile Robots

Xin Wang
Delft University of Technology
The Netherlands
xin.wang@tudelft.nl

Pieter Jonker
Delft University of Technology
The Netherlands
p.p.jonker@tudelft.nl

Abstract

In this paper we propose an object-driven online segmentation system for mobile robots. Among existing tracking and segmentation methods, the methods themselves are highly emphasized while properties of objects are usually not exploited enough. We propose an adaptive selection mechanism based on the properties of the objects to automatically choose an optimal tracking algorithm. For textured objects, a tracking algorithm which combines Lucas-Kanade tracker and model based detector using Random Forests classifier is adopted; for uniform objects, a color based tracker with a smooth constraint enforcement is used to ensure a robust performance. Moreover, a two-step object segmentation using Gaussian Mixture Models and graph cuts is applied to obtain detailed shape information. The experimental results on a variety of objects show the effectiveness of the adaptive tracking selection mechanism. The system also yields very promising performance in very challenging conditions with occlusions, illumination changes as well as cluttered background.

1 Introduction and Related Work

One of the important tasks of mobile robots is to explore interesting objects, learn, and manipulate them in an unstructured environment. In order to meet all these object-driven requirements, the proposed system should have the ability to adapt its tracker to various objects with different properties and precisely locate the objects despite viewpoint changes, illumination variations, occlusions, cluttered background and then extract detailed shape information for further tasks such as object recognition, object grasping, etc.

Even the best existing systems still exhibit limitations once dealing with such constraints of the real world settings [1]. Numerous uniform object tracking algorithms were proposed [2, 3, 4], among which color information is a strong cue. However, when encountering textured objects, their performance decreased considerably. The state of the art trackers [5, 6, 7] use distinctive features to cope with illumination changes, occlusions as well as cluttered background do not specifically target uniform objects. [8] utilized multiple cues to overcome disadvantages of using a single feature. However, the advantages of each feature were averaged. [9] combined texture and color information while the complexity of the algorithm made computation load too high for realtime robotics applications. [10] used an online appearance learning and adaptive algorithm to attain a robust tracking result. However, the prior knowledge about the properties of the objects is ignored. Instead of finding a universal tracking algorithm that works for every single object, we employ an adaptive tracking selection mechanism which is driven by the properties of the objects. Besides, we are interested in a service robot system that can be used in Robocup@home applications and autonomously obtain knowledge about the objects. Therefore, an online method with automatic segmentation of interesting objects is indispensable. For this task, existing background subtraction methods such as [11] will fail because of the constant change of the background. Motion-based online segmentation [12] is also not an option since the objects in the environment are static without any motion information.

In our paper, we present a complete system for robust online object segmentation which can overcome all above mentioned difficulties based on [13]. Fig.1 is the schematic overview of the system. The input is a bounding box manually selected around an interesting object. In order to examine the properties of the object, the system first extracts the object and discards the contour. Then HOG features are generated to determine if the object is textured or uniform. If the textureness is below a threshold, the system will switch to uniform object tracking, otherwise textured object tracking is employed. For uniform object tracking, a smooth constraint is added to Hue-Saturation segmentation to enforce similarity among neighborhood regions. For textured object tracking, motion based detection and model based detection are combined to track the object. Therefore, we obtain the location of the object of interest. In the final step, for every object model we refine object segmentation using Gaussian Mixture Models (GMMs) and graph cuts. As a result, detailed shape and contour information is extracted.

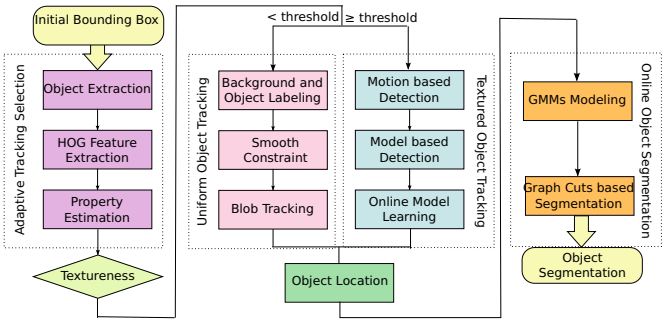

Figure 1. System scheme

2 Approach and Implementation

2.1 Adaptive Tracking Selection Mechanism

Methods that combine multiple cues do not distinguish between different situations and also bear the disadvantage of heavy computation. Therefore we use one of the most distinctive attributes, the textureness, and treat different objects with different tracking methods.

Given a bounding box, we need to extract the object from the initial bounding box using interactive segmentation GrabCut [14] in order to accurately investigate the property of the object. Both textured objects and uniform objects have contour, hence the contour is not an essential factor for measuring textureness and is discarded.

Histogram of Oriented Gradients (HOG) features [15] are used here to represent textureness of the object. HOG features are generated within the object using cell size 8x8, after which the property of the object is deduced by the amount of HOG features. Thus, we switch to either textured object tracking or uniform object tracking according to a specified threshold.

2.2 Online Object Segmentation

For the task of observing objects from different viewpoints, we need to update the object models online so that it can adapt to the constant change in object appearance. We employ the state of the art algorithm [6], which uses 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. Therefore the object tracking problem turns into a foreground and background classification problem.

For uniform object tracking, color is a strong cue. The other benefit of using color information lies in its low computation cost. However, the color based methods are sensitive to lighting conditions and ignorant of smoothness among neighborhood regions. Therefore smooth constraint was added to enforce similarity.

The back projection image based on hue histogram is compared with the hue-saturation joint histogram, which is shown in Fig.2. As seen in this figure, the hue-saturation histogram achieves better result than only using hue histogram. Then we can label the pixel to be either object, background, or undefined and assign the label confidence to each pixel according to histgram probability distribution. The labeled image only preserves region property, thus we use a smooth constraint to enforce similarity. For each pixel p, the similarity with its neighborhood pixel q is calculated. If they are very similar and the label confidence of q is very high, the neighborhood pixel q will affect the pixel p and p will have the same label as q. The label confidence of q will also change accordingly.

After iterations until no change, the object is determined. In Fig.2, the back projection image after smooth constraint has less background noise and neighborhood similarity enforced. Afterwards, we use blob tracking to detect the bounding box of each new frame.

For robotics applications, object segmentation will provide a cue for further tasks such as the object recognition, object manipulation. Most of the existing segmentation algorithms need interaction from users [14]. To fully automate the process we use the object model from tracking. To decrease the computation time, segmentation is performed only for key frames. For other frames, the confident segmentation from previous frame is used. If the displacement of current image patch and previous image patch and the scale difference are larger than a specified threshold, the frame is considered to be a key frame.

Hue Hue-Saturation Smooth Constraint

Figure 2. Back projection images comparison

We first adopt GMMs for hard constraints to construct the object and background models. With regards to a pixel x_p, the GMMs are defined as

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

where Gaussian density $N(x|\mu_k, \Sigma_k)$ is 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, B, and K is the number of components. K needs to be tuned according to scene, and more textured scenes require higher values of K. For textured objects $K = 5$ and for uniform object $K = 1$. Since we have the initial model, we can assign each pixel to each component in object GMMs and background GMMs. Then we use energy minimization as a soft constraint to optimize the segmentation. The energy minimization equation is

$$\begin{aligned} E(L) &= \lambda R(L) + B(L) \\ &= \lambda \sum_{p \in P} R_p(l_p) + \sum_{(p,q) \in N} B_{(p,q)} \cdot \delta(l_p, l_q) \end{aligned} \quad (2)$$

where $R_p(l_p)$ describes the region property based on GMMs models; $B_{(p,q)}$ describes the coherence of similarity within a region. 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 $c = \arg\min_L E(L)$ to have the foreground object and background. Here we confine background to be a region surrounding the image patch instead of using the region of the whole image to speed up the computation. We also use the output of the segmentation result as a refined input of the online model for more precise tracking.

3 Experiments and Results

For experiments we randomly picked up 75 different objects and put them in diverse scenes with various conditions. In total we tested on 6920 frames and used two criteria: center distance error (E) and score (S) defined in Eqn.3 and Eqn.4. The ground truth was manually labeled frame by frame. The implementation is written in C++ incorporated into MATLAB

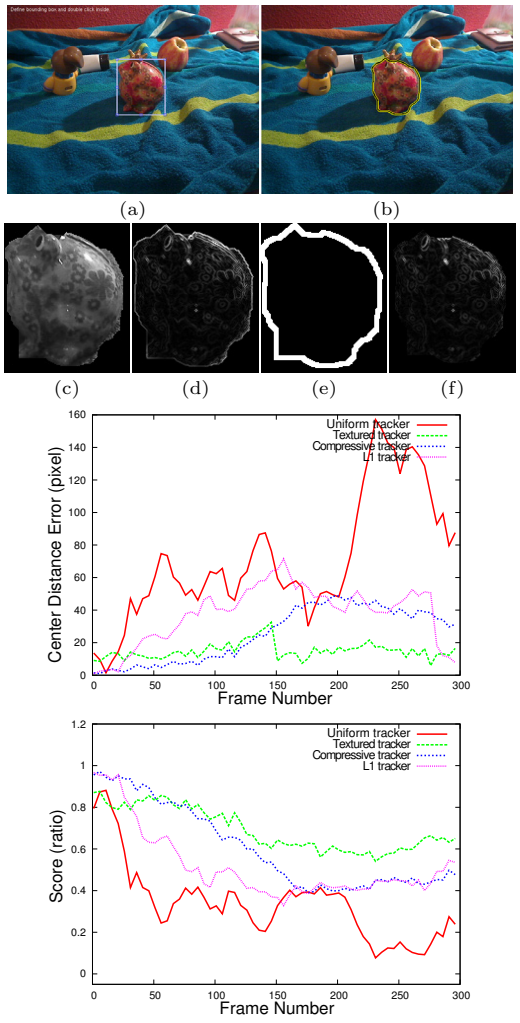

Figure 3. Performance on a textured object

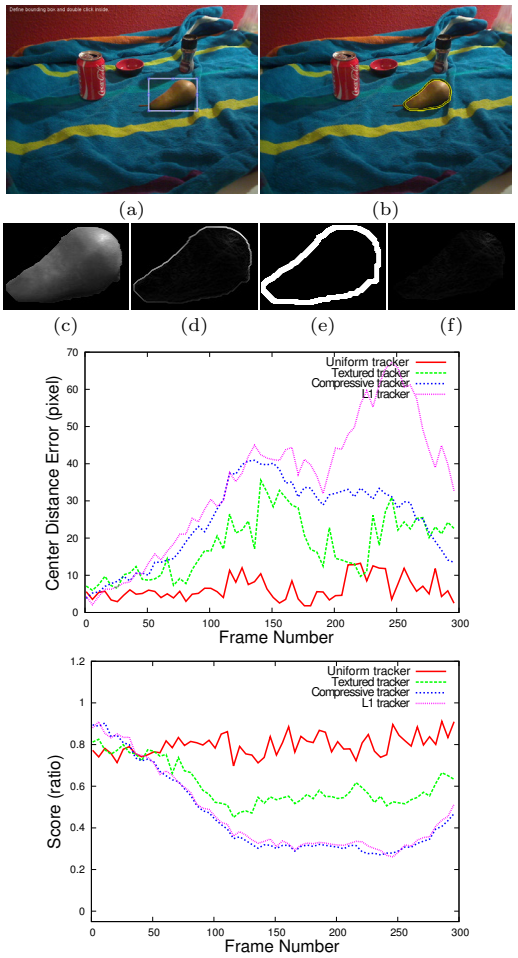

Figure 4. Performance on a uniform object

interface.

$$error = \sqrt{(x\_truth - x\_tracked)^2 + (y\_truth - y\_tracked)^2} \tag{3}$$

$$score = \frac{area(bb\_truth) \bigcap area(bb\_tracked)}{area(bb\_truth) \bigcup area(bb\_tracked)} \tag{4}$$

We used 25 objects in 1500 frames for testing the adaptive selection mechanism and determining the optimal threshold. Two examples of system performing on a textured object and a uniform object are shown in Fig.3 and Fig.4, respectively. (a) is the original image with object of interest selected by a bounding box, (b) and (c) are the segmented object, (d) encodes texture information, (e) is the extracted contour and (f) is the texture of the object without contour inference. The two (f) images show that the uniform object has a low textureness (textureness: 0.1256) compared to the textured object (textureness: 0.4269). The two graphs compare the performance of 4 trackers: uniform tracker, textured tracker [6] (TLD), an improved realtime L1 tracker [7] (L1), and compressive tracker [16] (CT). For the textured object, the uniform tracker performs worst since it mainly relies on color information, thus can not cope with very textured objects with complex color distribution. The textured tracker achieves very promising results with respect to textured objects. CT fails when scale changes. For the uniform object, uniform tracker outperforms the other trackers with a center distance error always below 15 pixels and very high stable score. L1 has a growing distance error as more frames get processed. Textured tracker experiences performance decrease for uniform objects. Consequently, we can deduce that combining uniform tracker and textured tracker dependent on the properties of the objects is an effective method to achieve robust performance.

For optimal textureness threshold determination, we used a selection error equation as

$$L(t) = \frac{\sum_{i=1}^{N} e(x_i \leq t) + \sum_{i=1}^{N} e(x_i > t)}{N} \tag{5}$$

where

$$e(x_i \leq t) = \begin{cases} 1 & \text{if } x_i \leq t \wedge S_u(i) < S_t(i) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$e(x_i > t) = \begin{cases} 1 & \text{if } x_i > t \wedge S_u(i) > S_t(i) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

For each object with its estimated textureness x_i, $S_u(i)$ and $S_t(i)$ are the average scores obtained respectively by the uniform and textured tracker. N is the total number of tested objects. For a given textureness threshold t, an object with textureness below it activates to uniform object tracking, while above it textured object tracking is activated. By comparing the ground truth of the training dataset with tracking algorithms performance, the selection error can be calculated. The textureness threshold with lowest selection error value is the optimal threshold. As we can

Figure 6. Online segmentation results

see from Fig.5, the optimal textureness threshold is chosen to be 0.2.

Table 1. Segmentation performance evaluation

(S) indicates a single object, and (M) indicates multiple objects

Scene vs Object	Textureless (S)	Textured (S)	Textureless (M)	Textured (M)
Uniform	98.2%	94.8%	96.2%	95.0%
Textured	98.2%	94.6%	94.6%	88.4%
Total	98.2%	94.7%	95.4%	91.7%

Table 1 presents segmentation performance of uniform objects and textured objects in various scenes. The columns represent two types of objects, uniform and texutured, and the rows the types of scenes: textureless background with a single object, textured background with a single object, textureless background with multiple objects, textured background with multiple objects. The overall performance is very promising with very high precision rates above 90%. We also derive the conclusion that in most cases, it is easier to segment the objects from textureless scenes than from textured scenes and it is easier to segment the object within single object background than multiple objects background. In case of large viewpoint changes and occlusions, the algorithms can still achieve robust segmentation performance, which is demonstrated in Fig.6. The computation rate is 10.80 fps for uniform objects and 5.32 fps for textured objects. Some further optimizations will ensure it can be used in realtime robotics applications.

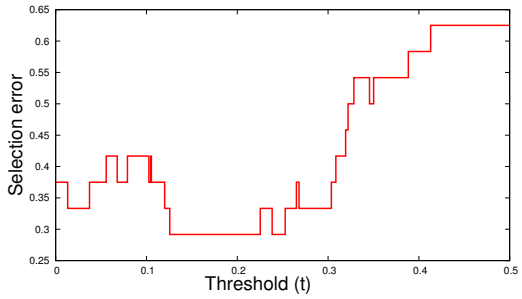

Figure 5. Selection error plot

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, Dec. 2006.

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

[3] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust mean-shift tracking with corrected background-weighted histogram," *Computer Vision, IET*, vol. 6, no. 1, pp. 62 –69, january 2012.

[4] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, May 2012.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–511 – I–518 vol.1.

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

[7] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, june 2012, pp. 1830 –1837.

[8] R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1631 –1643, oct. 2005.

[9] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust object tracking using joint color-texture histogram," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 07, pp. 1245–1263, 2009.

[10] D. A. Klein, D. Schulz, S. Frintrop, and A. B. Cremers, "Adaptive real-time video-tracking for arbitrary objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 772–777.

[11] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, aug. 2004, pp. 28 – 31.

[12] J. Mooser, S. You, and U. Neumann, "Real-time object tracking for augmented reality combining graph cuts and optical flow," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 1–8.

[13] X. Wang, M. Rudinac, and P. P. Jonker, "Robust online segmentation of unknown objects for mobile robots," in *7th International Conference on Computer Vision Theory and Applications, VISAPP*, 2012.

[14] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH '04*. New York, NY, USA: ACM, 2004, pp. 309–314.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, june 2005, pp. 886 –893 vol. 1.

[16] M.-H. Y. Kaihua Zhang, Lei Zhang, "Real-time compressive tracking," in *Proceedings of the 12th European Conference on Computer Vision*, 2012, pp. 864–877.