2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Interactive Segmentation for Manipulation in Unstructured Environments

Jacqueline Kenney Thomas Buckley Oliver Brock

***Abstract*— To perform successful manipulation, robots depend on information about objects in their environment. In unstructured environments, such information cannot be given to the robot *a priori*. It is thus critical for the robot to be able to continuously acquire task-specific information about objects. Towards this goal, we present a robust perceptual skill for identifying, tracking, and segmenting objects in a cluttered environment. We increase the robot's perceptual capabilities by closely coupling them with the robot's manipulation skills. The robot's interaction with objects in the environment creates a perceptual signal, i.e. motion, that renders segmentation and tracking robust and reliable. In addition, the resulting perceptual signal reveals the type of segmentation most relevant to manipulation, namely a segmentation of rigidly connected physical bodies. We demonstrate our approach with experiments on a real world mobile manipulation platform with multiple objects in a cluttered scene.**

I. Introduction

Competent manipulation in unstructured environments is a prerequisite for many important applications, ranging from household robotics to cooperative manufacturing. Current manipulation systems perform most reliably when accurate object models are available and when environmental conditions can be tightly controlled. These restrictive assumptions do not hold in our everyday environments, where the robot must perform manipulation tasks without object models in continuously changing surroundings. We call these unstructured environments because humans have not intentionally imposed structure on the space to help the robot. In these environments, manipulation skills can only rely on assumptions that hold irrespective of the specific circumstances.

In this paper, we take a first step towards manipulation in unstructured environments by presenting a perceptual skill to identify, segment, and track rigid objects in cluttered scenes, such as the one shown in Figure 1.

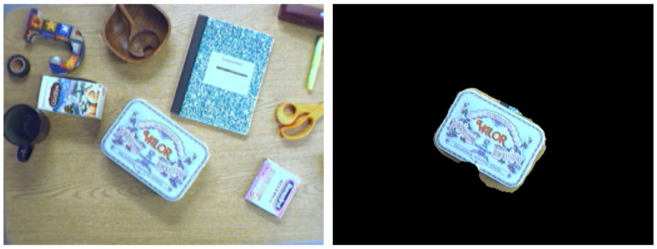


Fig. 1. A cluttered table top (left) and the segmentation of an object obtained from interaction.

Our goal to perform image segmentation in support of manipulation in unstructured environments imposes important requirements. First, an adequate segmentation algorithm must not depend on prior knowledge about the scene or the objects it contains. Any prior knowledge or assumption may become invalid in unstructured environments. Second, a successful algorithm has to continuously identify, segment, and track new objects as they appear in the scene. This capability will enable the robot to move around in the environment to interact with objects. Third, image segmentation must be computationally efficient. This is necessary to obtain feedback about objects at rates adequate for manipulation. Finally, the context of manipulation provides us with a clear definition of segmentation. To support manipulation, a segmentation algorithm should identify rigidly connected objects in the scene, as these are the entities the robot manipulates and interacts with.

The computer vision community has developed many successful segmentation techniques; we will discuss them in more detail in the next section. Most of these techniques are not well-suited for manipulation. Some depend on prior knowledge about the scene, such as object models or the number of moving objects. Others process entire video sequences or require problem-specific training; as a result, they do not support the incremental segmentation required in our application. Yet others are too computationally expensive to provide timely feedback for manipulation.

Image differencing methods [9], [21], on the other hand, are computationally efficient and enable online, incremental segmentation. These methods segment a scene by measuring the pixel-wise intensity change in two consecutive frames of a video sequence. This implies, however, the limiting assumption that objects must be in motion to be segmented.

In this paper we develop an image segmentation algorithm suitable for manipulation in unstructured environments. We overcome the limiting assumption of image differencing by leveraging manipulation capabilities to move objects. Following the work of Fitzpatrick et al. [8], we view the camera as part of an embodied agent that can interact with its environment. This interaction causes objects to move, making them easily segmentable by image differencing. We refer to the general idea of the mutual support of perceptual and manipulation processes as interactive perception [11]. We develop a segmentation method that is efficient, robust,

The authors are members of the Robotics and Biology Laboratory, Department of Computer Science, University of Massachusetts Amherst.

The authors gratefully acknowledge support by the National Science Foundation (NSF) under grants CNS-0454074, IIS-0545934, CNS-0552319,CNS-0647132, CNS-0812986, and by QNX Software Systems Ltd. in the form of a software grant.

978-1-4244-2789-5/09/$25.00 ©2009 IEEE

and sufficiently general to support manipulation in unstructured environments. We validate our method in real-world experiments on a mobile manipulation platform.

II. Related Work

Segmentation is a well-studied problem and there are a wide variety of approaches presented in the literature [9], [21]. Broadly, these approaches can be grouped into those that segment objects in a single image and those that segment objects in sequences of images. Segmentation methods for single images rely on thresholding, edge detection, clustering, or region growing to group pixels based on brightness, color, or texture [9]. These methods are based on the assumption that the boundaries of objects correspond to discontinuities in these properties—and that these discontinuities do not occur anywhere else. This assumption does not capture the type of object we are interested in. In fact, a single rigid object can consist of many different regions, according to this definition. Due to this fundamental mismatch, we do not discuss segmentation on single images any further.

Motion-based image segmentation methods detect changes in a sequence of images and use the resulting information to perform segmentation. Approaches in this category can be further divided into statistical methods, methods based on optical flow, wavelet transforms, factorization methods, and image differences [21]. Statistical methods treat segmentation as a classification problem in which each pixel is classified as belonging to a particular cluster or object [7], [16], [17], [18]. Optical flow methods identify distinct image regions based on their perceived motion in the image plane [22]. Wavelet transforms perform analysis on different frequencies of an image, increasing the efficiency of detecting motion for segmentation [12], [19]. Factorization techniques rely on methods from linear algebra to extract information about motion and structure of objects from the motion of individual features tracked throughout a sequences of images [6], [10].

None of the approaches to segmentation discussed so far are suited for segmentation in service of manipulation in unstructured environments. The reasons vary with each individual method and category. Factorization approaches, for example, are computationally complex and can only be applied to entire image sequences [6], [10]. Statistical methods rely on prior knowledge about the scene [17], [18] or the objects contained in it [16]. Other approaches restrict the type of motion to translation only [12], [19] or do not work for multiple objects [7].

There is another image segmentation method that does not exhibit the shortcomings discussed so far: image differencing. These methods segment an image based on the pixel-wise intensity change between two consecutive frames in an image sequence [1], [3], [4], [5], [8], [13], [14], [15], [20]. This leads to computationally efficient and robust segmentation of moving objects; stationary objects cannot be segmented. Fitzpatrick et al. [1], [8], [13] overcome this shortcoming of image differencing by using a robot arm to interact with the observed scene. This interaction causes object motion in an otherwise static scene, thereby enabling segmentation based on image differencing.

The work presented in this paper extends the work by Fitzpatrick and his collaborators in several ways. Our method is able to identify, segment, and track multiple objects; we believe this to be a prerequisite for manipulation in cluttered environments. Furthermore, our method accumulates segmentation information over time, producing increasingly accurate segmentations throughout the robot's interaction with the environment.

III. Interactive Segmentation

The goal of this work is to support manipulation in unstructured environments with the perceptual capability to continuously detect, segment, and track objects. The detection of a new object tells the robot that either a moving object (possibly its own end-effector) has entered the field of view, that one of the objects in the scene has started moving, or that one of the moving objects has made contact with another object. The segmentation of an object enables the robot to identify its spatial extent; this is necessary for determining appropriate manipulation strategies. Finally, the tracking of objects provides important feedback about the ongoing manipulation. We believe that these perceptual capabilities are prerequisites for robust manipulation in unstructured environments.

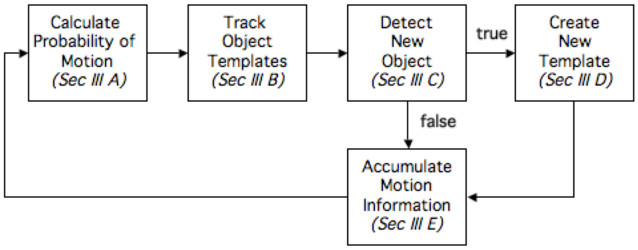
Our method exploits the simple insight that manipulation itself can facilitate the acquisition of perceptual information in service of manipulation. By interacting with objects, the robot can cause them to move, thereby creating a perceptual signal that segments rigid objects. Note that in this paper we are not concerned with articulated objects, but we believe that algorithms developed in our laboratory [11] will allow a simple extension of the work presented here. This is the subject of future work.

Our method raises the question of how to initially detect objects in order to know where and how to interact with them. The experiments in this paper assume knowledge of object location, but we believe that this assumption can be easily eliminated. One possible solution is to randomly explore the space with the end effector, using force feedback to detect contact with an object. Another solution is to find a rough initial location using blob detection, and then to use visual servoing and force feedback to move toward the object and make contact. This will be addressed in future work.

To obtain a segmentation of a scene into rigid objects, the robot incrementally processes a sequence of images. The five stages are shown in Figure 2. The subsequent sections describe of these stages in detail. Information about implementation choices and parameters are in Section IV.

A. Motion Detection

In order to use a motion signal for segmentation, we must first determine where motion occurs in an image. We do this by assessing how likely the color of a pixel is, given our previous observations. An unlikely observation provides us with high confidence of motion.



Calculate Probability of Motion (Sec III A)
Track Object Templates (Sec III B)
Detect New Object (Sec III C)
true
Create New Template (Sec III D)
false
Accumulate Motion Information (Sec III E)

Fig. 2. The process for identifying and tracking multiple objects

We use the first n frames of the image sequence to build a set of Gaussian models, $N(\mu,\sigma)$, for each pixel of the image. The variance of these Gaussians effectively represent a noise model of our sensor. For each pixel value p, the probability of motion is zero if p falls within $T_{in}\sigma$ of the distribution mean, one if it is more than $T_{out}\sigma$ away from the mean and a linearly interpolated value between zero and one if it falls in between. The result is an image where each pixel holds a gray scale value between 0 and 255 representing the probability that it represents motion in the image.

After processing each frame, the mean μ of the Gaussian is set to the current value of the pixel. However, the variance σ remains unchanged.

B. Template Tracking

Now that we have a motion signal, we want to understand which parts belong to previously encountered objects. These objects are represented as templates and by matching templates from prior frames to the new motion evidence, we can track the motion of those objects.

To do this we created a feature-based alignment process. When a template is created, n_f features are selected to represent the corresponding object. These features are chosen at random from the group of pixels in the template with high probability of motion. This type of pixel is locally distinctive, otherwise no motion could have been detected. This makes them well-suited for template alignment.

The system searches a set of local transformations for the highest-quality alignment between the object template and the current image of motion evidence. The transformations include translations in the x and y direction between $-x$ and x pixels and a rotation of between $-\theta$ and θ degrees.

The alignment quality function relies on motion and color information. For motion information, the number of coinciding features with motion evidence is considered, relative to the number of overall features. The quality function also evaluates the similarity of the color values for each feature pair. The overall quality equally weighs these two factors.

C. New Object Detection

Now we want to identify any motion that cannot be described by the existing object templates. This extra motion is a candidate for a new object. The corresponding motion evidence for each existing template is removed from the current motion image, leaving only the unexplained motion information.

The amount of this motion leftover is analyzed to determine if it represents a new object. If it does not, there will be very few pixels with a non-zero probability of motion. This number should be less than a given threshold, T_{motion}. On the other hand, if there is a new object in the scene, there will be greater than T_{motion} pixels with a non-zero probability of motion. We know this is true because all points on an object will begin to move at once. See Figure 3 for an illustration.

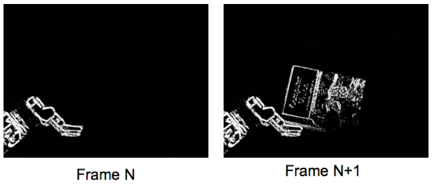


Fig. 3. Motion information in two consecutive frames. Frame N+1 contains a new object and an entire region of new pixels with a non-zero probability of motion. We perceive that significant jump in the amount of motion evidence to detect new objects.

It should be noted that our algorithm treats the motion of the manipulator arm differently by assuming that it is the first object to move in the scene. Because it enters the scene gradually, the above method does not work. Instead, we wait until the total amount of motion evidence exceeds a threshold T_{manip} to declare it as a new object.

D. Template Creation

Once a new object has been detected, the corresponding motion evidence is used to create a template. The template is found by removing all motion observed at the previous frame from the current motion image. This is illustrated in Figure 4.

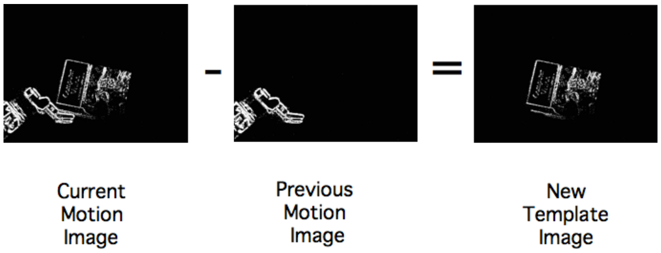


Fig. 4. An example of how to create a new template.

E. Accumulation

At this point in the algorithm, we perform an accumulation step to incorporate the current motion information into the existing templates. This increases the quality of each template over time by continuing to learn about object boundaries and detecting new areas based on changing directions of motion.

To do this for each individual template, we take the current motion image and remove all of the other existing templates. This leaves motion information corresponding to only the current template. We update the pixel values in the area of the template with a new value according to the following equation:

$$p_{acc} = (1-\alpha)p_{n-1} + (\alpha)p_n,$$

where p_{n-1} is the probability of motion for that pixel in the previous frame, p_n is the probability of motion for that pixel in the current frame and p_{acc} is the new accumulated pixel value. This update incorporates a percentage of the new motion information into the template value at each frame.

F. *Segmentation*

Segmentation is the final process of obtaining an image region from the motion evidence. This image region then corresponds to a particular object. This is performed using a maximum-flow/minimum-cut graph algorithm on the image of motion evidence, as proposed by [8]. We make a graph representation of each template by creating a node for every pixel. An edge is added to the graph for all pairs of neighboring pixels with weight w. Pixels with a probability of motion greater than T_{seg} are also connected to a special foreground node with a weight w_f. All other pixels are connected to the background node with weight w_b.

The maximum flow algorithm then finds a minimum-cost cut of the graph into two connected components. These components correspond to the the foreground and background of the template image. The implementation used is due to Boykov and Kolmogorov and is described in detail in [2].

IV. Experiments

We validate the segmentation algorithm presented in the previous section with real world experiments. We pursue three major objectives. First, we would like to demonstrate that segmentation based on motion information can successfully segment types of scenes for which color segmentation would fail (see Sections IV-A and IV-B). This will validate our use of object motion as the appropriate source of perceptual information for segmentation in the context of manipulation. It will also emphasize the importance of combining perception with manipulation, as manipulation permits us to generate this perceptual signal in static scenes.

Our second objective is to demonstrate that our segmentation method is well-suited to support manipulation in unstructured environments. The experiment in Section IV-C will demonstrate that our method successfully and continuously detects, segments, and tracks a number of objects whose motion is caused by the robot itself. While in our experiments we do not pursue a specific manipulation objective, we demonstrate our ability to monitor the progress of manipulation.

Lastly, we demonstrate that the quality of the segmentation and the robustness of the pose estimation of objects can be improved by accumulating motion information throughout the robot's interaction with its environment (see Section IV-D). This illustrates the benefit of performing segmentation incrementally and increases our confidence that the perceptual skills presented in this paper are capable of addressing the sensing uncertainty inherent to unstructured environments.

All experiments are performed using our robotic platform, the UMass Mobile Manipulator (UMan). UMan's holonomic mobile base is equipped with a seven degree-of-freedom manipulator arm, and a three-fingered hand (see Figure 5). The objects manipulated by UMan are laying on a table in front of it. A static overhead Web camera with a resolution of 640x480 and a frame rate of 12 Hz provides visual data to the robot.

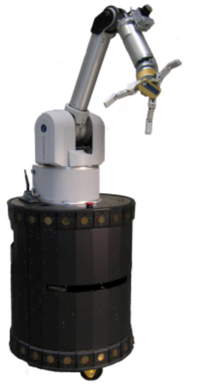


Fig. 5. UMan

Our system requires several different parameter values. We set these values empirically, and it should be noted that they remain constant for all of the experiments described in this section. The parameters that are particular to our setup are the motion thresholds for object detection, $T_{manip} = 7000$ pixels and $T_{motion} = 400$ pixels. In the future, these thresholds could be set automatically, or using proprioception. The following other parameters are used: $n = 10$,$T_{in} = 5$, $T_{out}15$, $n_f = 50$, $\alpha = 0.2$.

A. *Similar Object and Background*

In an unstructured environment, a robot is likely to encounter objects that have a color or texture that is similar to that of their background. This situation is particularly difficult for segmentation techniques that rely on color information because strong visual evidence for the boundary of the object does not exist.

We demonstrate that our method can successfully segment objects in this case through the example shown in Figure 6. The appearance of the wooden block is very similar to the wooden table, but by interacting with the object, the robot is able to create motion information that is sufficient for performing a correct segmentation. This shows that motion information is the appropriate signal to use in this situation and reminds us that we are able to use this signal because of the robot's direct interaction with the environment.

B. *Multi-colored Object*

A robot is also likely to find multi-colored and textured objects in an unstructured environment. In order to enable manipulation of those objects, the many regions that compose these objects must be segmented as one.

Figure 6 shows an experiment demonstrating that our method can segment entire rigid objects composed of many regions. UMan successfully segments an entire striped folder, even though the colors of the regions are distinctly different. This is because our technique labels regions as objects if they begin moving simultaneously. Despite their different colors, all regions of the folder begin moving at the same time and are identified as belonging to the same rigid object. The motion signal is able to capture information that is relevant for segmentation to support manipulation.

C. *Multiple Objects*

A robot exploring an unstructured environment will also need to acquire information about several different objects in

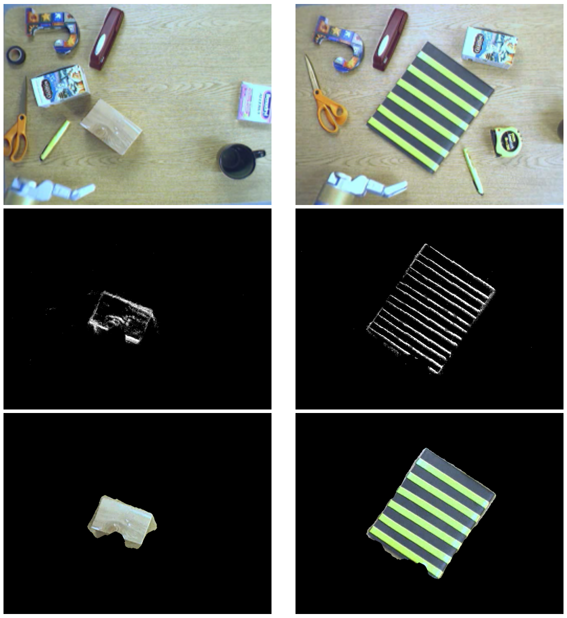


Fig. 6. Left: The segmentation of an object very similar to the background. Right: The segmentation of a multi-color object. The top images show the original scenes , the center images show the motion templates and the bottom images show the final segmentations.

the environment. In this scenario it is useful to continuously use interactive segmentation to segment multiple objects in the scene. It is also important to be able to observe and detect motion caused by objects that are not moved directly by the end effector. This capability increases the information the robot can learn from interactions with the environment.

Figure 7 shows an experiment where UMan interacts with an object that moves and strikes another object. This example shows that our approach is able to segment both objects separately. It is able to identify multiple objects because we track each existing template throughout the interaction and can reason about whether other motion information is caused by a new object.

D. Accumulation vs. No Accumulation

We also test the effectiveness of our accumulation method by comparing final segmentations of objects with accumulated motion templates versus non-accumulated templates. This experiment shows that accumulation is important for improving the quality of segmentations.

Figures 8 and 9 show the difference between both the motion templates and the final segmentations. The motion information for the non-accumulated templates is sparse and does not capture the entire object. The hand template is missing information for all of the bottom portion of the wrist and the book template is missing information on both the right and left lower corners. Alternatively, the accumulated templates include more information about both objects. The bottom of the wrist is completely present, and the edges of the book are thicker and straighter, plus the missing corner information has been filled in.

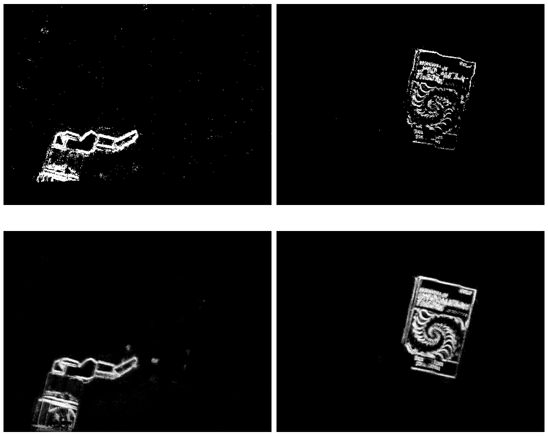


Fig. 8. A comparison of no accumulation vs. accumulation for motion templates. The top row shows the templates without accumulation and the bottom row shows the templates with accumulation.

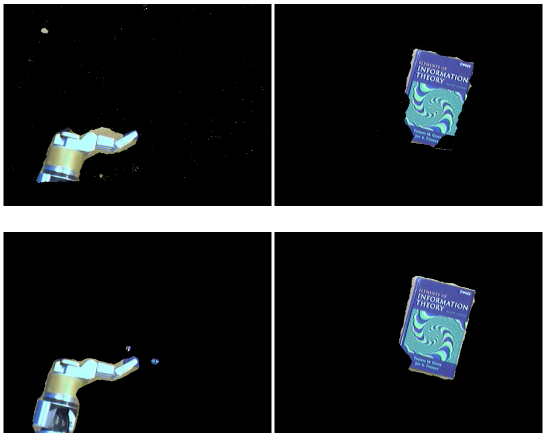


Fig. 9. A comparison of no accumulation vs. accumulation for segmentations. The top row shows the segmentations without accumulation and the bottom row shows the segmentations with accumulation.

A comparison of the segmented images reveals similar results. The segmentation of the hand without accumulation is missing the entire bottom portion of the wrist, and the book is missing both bottom corners. The segmentations with accumulation include these missing areas and have straighter, better defined edges around the hand and book.

This experiment shows that with accumulation our approach is able to improve the quality of object segmentation throughout an interaction and overcome the uncertainty introduced by an unstructured environments.

V. Conclusions

To perform manipulation in unstructured environments, a robot must acquire task-relevant information about objects and the environment. Towards this long-term objective, we presented a perceptual skill for the detection, segmentation, and tracking of multiple moving objects in the robot's field of view (including the robot's end-effector). Our method leverages the robot's ability to interact with its environment to reveal perceptual information that would otherwise remain hidden to a passive observer. This careful composition of perception and manipulation renders the segmentation problem

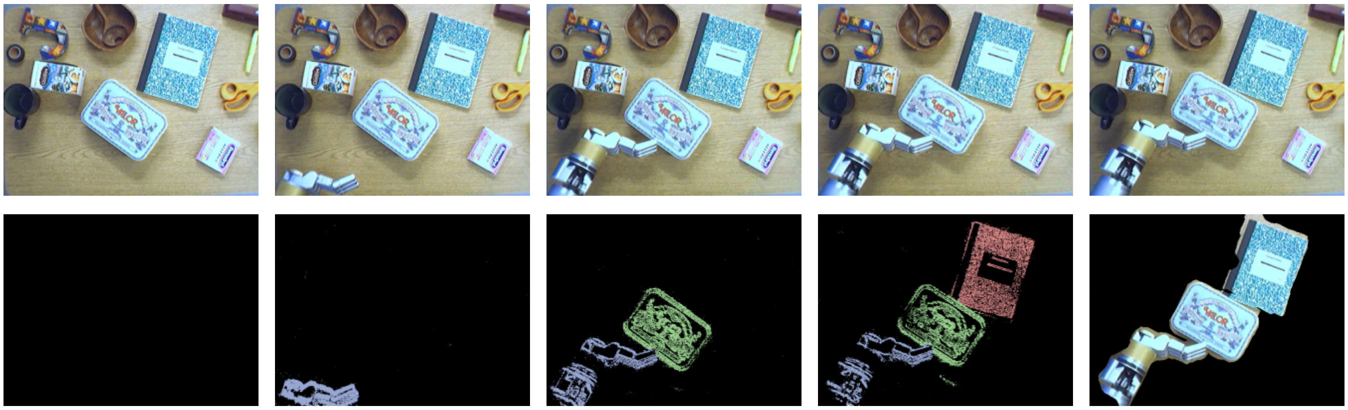


Fig. 7. A sample run of an experiment with multiple objects. The top row shows the manipulator as it moves through the scene. The bottom row shows the motion templates generated by our system (shown here combined into one image) and the last frame shows the segmentation results.

simple, robust, and general.

This method enables the robot to obtain a scene segmentation that is relevant to the task of manipulation: because the proposed segmentation method relies on motion caused by interaction, it segments the scene into rigid bodies—exactly those physical entities the robot is manipulating. Due to this interactive nature, we refer to this perceptual skill as interactive segmentation, a specific skill in the broader category of interactive perception [11]. We presented real-world experiments on a mobile manipulator to validate our approach in cluttered scenes.

We see this skill as a basic method for acquiring information about a new environment. In future work we hope to use it as a starting point for identifying objects without having to interact with them every time. To do this, interactive segmentation can be used to build a collection of training examples of object segmentations. These can be used to build an object recognition system to identify objects without interaction. When a robot enters a new environment, however, there is no way of recognizing all of the possible objects it may see, so interactive segmentation provides a robust skill to fall back on when objects cannot be identified another way. Because the robot can always use interaction if necessary, we believe that this combination can be a robust method for obtaining information about new environments.

ACKNOWLEDGMENTS

The authors thank Dov Katz for insightful discussions.

REFERENCES

[1] A. Arsenio, P. Fitzpatrick, C. Kemp, and G. Metta. The whole world in your hand: active and interactive segmentation. In *Proceedings of the Third International Workshop on Epigenetic Robotics*, 2003.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *In Proceedings of Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2001.

[3] A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(4):575–584, 2005.

[4] F.-H. Change and Y.-L. Chen. Real time multiple objects tracking and identification based on discrete wavelet transform. *Pattern Recognition*, 39(6):1126–1139, 2006.

[5] A. Colombari, A. Fusiello, and V. Murino. Segmentation and tracking of multiple video objects. *Pattern Recognition*, 40(4):1307–1317, 2007.

[6] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[7] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[8] P. Fitzpatrick. First contact: an active vision approach to segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, 2003.

[9] D. A. Forsyth and J. Ponce. *Computer Vision – A Modern Approach*. Prentice Hall, 2002.

[10] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[11] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, USA, 2008.

[12] M. Kong, J.-P. Leduc, B. Ghosh, and V. Wickerhauser. Spatio-temporal continuous wavelet transforms for motion-basedsegmentation in real image sequences. In *Proceedings of the International Conference on Image Processing*, 1998.

[13] G. Metta and P. Fitzpatrick. Early integration of vision and manipulation. *Adaptive Behavior*, 11(2):109–128, 2003.

[14] L. P. K. Michael G. Ross. Learning static object segmentation from motion segmentation. In *Twentieth National Conference on Artificial Intelligence*, 2005.

[15] S. J. Pundlik and S. T. Birchfield. Motion Segmentation at Any Speed. In *British Machine Vision Conference*, 2006.

[16] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Particle Filtering for Geometric Active Contours with Application to Tracking Moving and Deforming Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[17] H. Shen, L. Zhang, B. Huang, and P. Li. A MAP Approach for Joint Motion Estimation, Segmentation, and Super Resolution. In *IEEE Transactions on Image Processing*, 2007.

[18] R. Stolkin, A. Greig, M. Hodgetts, and J. Gilby. An em/e-mrf algorithm for adaptive model based tracking in extremely poor visibility. *Image and Vision Computing*, 26(4):480–495, 2008.

[19] L. Wiskott. Segmentation from motion: Combining gabor- and mallat-wavelets to overcome the aperture and correspondence problems. *Pattern Recognition*, 32(32):1751–1766, 1999.

[20] R. L. S. Y. X. Yang. Efficient spatio-temporal segmentation for extracting moving objects in video sequences. *Consumer Electronics, IEEE Transactions on*, 53(3):1161–1167, 2007.

[21] L. Zappella. Motion sgmentation from feature trajectories. Master's thesis, University of Girona, Spain, 2008.

[22] J. Zhang, F. Shi, J. Wang, and Y. Liu. 3d motion segmentation from straight-line optical flow. In *Multimedia Content Analysis and Mining*, pages 85–94. Springer Berlin / Heidelberg, 2007.