Interactive Segmentation of Articulated Objects in 3D

Dov Katz Oliver Brock
Robotics and Biology Laboratory
Technische Universität Berlin

***Abstract*— To enable autonomous manipulation of previously unseen objects, robots must possess perceptual capabilities for modeling objects in their environment and for observing their motion during manipulation. Towards this objective, we present a robust perceptual skill for identifying, tracking, and segmenting objects in unstructured scenes. This skill is based on a principled and computationally efficient framework for integrating information from multiple visual clues to yield an accurate and robust segmentation, even under arbitrary object motion. Further, we present an interactive perception skill that provides an additional perceptual clue that greatly increases the generality and robustness of segmentation. The resulting perceptual skill is suitable for autonomous manipulation in unstructured environments. We validate this statement in real-world experiments on a mobile manipulation platform with multiple rigid articulated objects.**

I. INTRODUCTION

We present a perceptual skill in support of manipulation in unstructured environments. In such environments, successful manipulation requires that a robot be able to identify and segment the articulated objects to be manipulated; furthermore, the robot must be able to track the motion of the manipulated objects so as to obtain visual feedback throughout the manipulation. As this perceptual skill targets applications in unstructured environments, it must not rely on prior knowledge of the objects to be manipulated.

The assumption that no prior knowledge is available poses substantial challenges for visual perception. First, without any prior knowledge the information contained in a static image does not suffice for identification and segmentation of manipulable objects. Segmentations of images containing a single object often consist of multiple distinct regions, as objects can contain parts with different colors or textures. Furthermore, some object parts may be difficult to distinguish from the background. Second, without any prior knowledge the information contained in a single image does not suffice to identify the degrees of freedom of an object. Knowledge of the degrees of freedom is necessary for manipulation of articulated objects, such as scissors, doors, or drawers. Third, the lack of prior knowledge makes it difficult to track the manipulated object as it undergoes arbitrary 3D translation and rotation. During motion, the silhouette of an object can undergo discontinuous changes, making tracking of the contour difficult. To the best of our knowledge, no perceptual skill exists that overcomes these challenges.

In this paper, we describe a perceptual skill that identifies articulated objects, segments them from the background, and tracks the obtained segmentation throughout arbitrary motions of the object, including the motion of internal degrees of freedom of articulated objects. To achieve this, we rely on two insights. First, we leverage and combine multiple visual clues to perform image segmentation. Among these, the most important clue for segmentation is derived from deliberate interactions of the robot with the object to be segmented. Through this interactive perception, the robot reveals the visual signal required to appropriately identify pertinent object boundaries and to discover the kinematic relationships between object parts. Second, we chose to focus on segmenting individual objects, rather than segmenting an entire scene. This allows us to overcome many of the limitations of traditional segmentation algorithms.

Fig. 1. Our Mobile Manipulator interacts with a train toy, and identifies a set of rigid bodies: train engine, train car, and wooden figures. These rigid bodies are segmented from the background throughout the interaction.

The proposed perceptual skill identifies, segments, and tracks objects of arbitrary size, shape, color, and texture in uncontrolled lighting conditions, while being computationally efficient. It complements our prior work on perceiving the kinematic structure of an arbitrary object with a dense representation of the object. It is therefore key in enabling manipulation. The versatility and robustness of the skill depend on two assumptions. First, it assumes that objects possess reliably perceivable texture (this limitation is shared by all vision algorithms). Second, the skill currently relies on scripted physical interactions with objects. This limitation will be removed in future work, similar to our work on perceiving planar articulated objects [15].

Fig. 2. An example of everyday articulated objects. Our robot has interacted and segmented these objects in multiple configurations. The objects differ in size, shape, color, texture and their kinematic structure. Segmenting them in 3D provides the robot with necessary information for manipulation.

II. RELATED WORK

Segmentation algorithms fall into one of two categories [10], [27]. *Image segmentation* divides an entire image into spatially contiguous regions that share a particular property. *Object segmentation* extracts a single object from the background.

A. Image Segmentation

Image segmentation methods process an image to identify boundaries between regions that share a particular property. All of the methods in this category are based on the assumption that the boundaries of objects correspond to discontinuities in color, texture, brightness or depth—and that these discontinuities do not occur anywhere else.

Most methods rely on thresholding, edge detection, clustering, or region growing to group pixels based on brightness, color, or texture [10]. Other methods use parallax [2] to compute depth values for each pixel and subsequently apply segmentation algorithms to this depth image.

The fundamental assumption underlying these methods—namely, that discontinuities of an image property indicates object boundaries—does not capture the notion of object for the purpose of manipulation. In manipulation, an object is a connected set of bodies. The boundary of such an object and the boundaries determined by mentioned image properties rarely coincide, especially when attempting to segment an entire image containing many objects.

B. Object Segmentation

Object segmentation focuses on segmenting a single object in an image, rather than segmenting the entire image. By focusing on a single object and thus a local region of the image, the difficulty of finding good global segmentation parameters is avoided. The key to object segmentation is the selection of an initial hypothesis about the image region occupied by the object to be segmented. This region is generally specified by a fixation point, assumed to lie inside the object boundary. Starting from this fixation point, methods determine a local segmentation enclosing the fixation point.

The easiest way of specify a fixation point is human input (clicking on the image) [20]. However, this method is not suitable for autonomous manipulation. Kootstra et al. [11] suggest to identify fixation points based on object symmetry, where symmetry is determined using classical computer vision techniques. And Campbell et al. [3] assume that the camera can keep the object of interest in the middle of the frame, so that the center of the image serves as a fixation point. These methods for determining fixation points are computationally efficient and provide good results. However, they rely on assumptions that may not hold during manipulation tasks: objects may not be symmetric or may not be located in the center of the image. Furthermore, these method do not overcome the fundamental limitation of image segmentation: the properties used to segment an object do not necessarily coincide with "object-ness".

If, for the purpose of manipulation, we define objects as connected sets of bodies, successful segmentation must reveal this connectedness of bodies in the scene (here, we limit ourselves to *rigid* bodies). Consequently, we can obtain information about "object-ness" as well as adequate fixation points by analyzing a sequence of images in which objects move relative to each other. This can be accomplished using optical flow, statistical methods, wavelet transforms, and factorization methods. Optical flow methods identify distinct image regions based on their perceived motion in the image plane [28]. Statistical methods treat segmentation as a classification problem in which each pixel is classified as belonging to a particular cluster or object [6], [21], [22], [23]. Wavelet transforms analyze the different frequencies of an image to detect motion [17], [25]. Factorization techniques use information about the structure and motion of objects based on the motion of features tracked throughout a sequence of images [5], [12], [26].

All of these methods make assumptions about the object or the environment that limit their applicability to manipulation. The assumptions differ by method. Statistical methods, for example, rely on prior knowledge about the scene [22], [23] or the objects it contains [21]. Factorization approaches are computationally complex and depend on a relatively long sequence of images [5], [12]. Other approaches restrict the type of motion to translation only [17], [25] or do not work for multiple objects [6]. The most important limitation, however, shared by all segmentation methods that rely on relative motion is that they do not have control over whether an object is in motion.

Interactive segmentation methods overcome the limitations of the methods above by interacting with objects, for example by pushing them. This interaction generates a visual signal (the motion of connected bodies, i.e., an object) that directly corresponds to "object-ness." Segmentation is then computed based on the pixel-wise intensity changes between consecutive frames [1], [9], [19], [16]. This approach has the advantage of being very simple, robust, and computationally inexpensive. It is, however, limited to objects undergoing planar motion and ignores the rich information provided by color.

The perceptual skill presented in this paper combines the advantages of all of the aforementioned methods. It uses deliberate interactions to generate motion and uses this visual signal together with a variety of visual cues to determine candidates for fixation points. These points are tracked over time and segmented into clusters, corresponding to rigid bodies. It then computes an object segmentation around each cluster using an established method for object segmentation [20]. Our method continuously segments objects undergoing arbitrary three-dimensional motion, makes no prior assumptions about the object (other than it containing trackable features—a white, textureless object on the same background cannot be visually identified), works for objects of different sizes and shapes, under varying lighting conditions, and is computationally efficient. It is thus well-suited for manipulation in unstructured environments.

III. Interactive Segmentation of Articulated Objects

The goal of this work is to support manipulation in unstructured environments with a perceptual capability to continuously detect, segment, and track objects. To achieve this, our method exploits the insight that manipulation itself can facilitate the acquisition of perceptual information. By physically causing objects to move, the robot can generate a perceptual signal that enables object detection, segmentation, and tracking.

Our algorithm is composed of three components. The first component collects perceptual information that provides the input to our perceptual skill. This component initializes and tracks visual point features throughout the robot's interactions with the environment. The second component analyzes the trajectories of these features to formulate hypotheses about the presence of rigid bodies in the scene. The result is a clustering of features and their trajectories, where each cluster corresponds to a presumed rigid body. The third component of our algorithm uses the features associated with a single rigid body as fixation points for object segmentation. As we know the features' trajectories through time, we can use repeated segmentation to track objects during manipulation.

A. Collecting Perceptual Evidence for Segmentation

The first component of the algorithm collects perceptual information. The robot observes its interactions with the environment by tracking a large number of point features using the Lucas-Kanade feature tracker. During its interaction, the robot records the features' image coordinates (u,v) and their color values c for each time t in feature observations $f_i(t) = \{u,v,c\}$. Feature tracking is a simple and computationally efficient operation. It only requires that the scene contains sufficient texture to support visual tracking. It makes no assumption about the shape, size, or color of objects, about their motion, or the motion of the camera.

Feature tracking in unstructured scenes is highly unreliable. Features can jump between image regions, are lost, swapped, or drift along edges in the image. The remainder of the algorithm will automatically eliminate this noisy data, rendering the algorithm suitable for manipulation in unstructured environments.

To cause motion of objects in the scene, the robot must interact with the environment. In this paper, we will assume that this initial interaction is given or generated by random, force-controlled motion. In future work, we will eliminate this assumption, following prior work in manipulation of planar objects. In that work, the robot learns to generate such goal-directed interactions autonomously [15]. Please also note that our algorithm does not differentiate between objects that move by themselves and objects moved by the robot.

B. Obtaining Rigid Body Hypotheses

The proposed segmentation algorithm uses the obtained feature trajectories to determine hypotheses about groups of features that could lie on the same rigid body. To formulate these hypotheses, the algorithm leverages the simple insight that features associated with a single rigid body share a set of spatial, temporal, and appearance properties, some of which will remain constant or evolve consistently as the object or the camera moves.

Our algorithm stores the feature trajectories obtained in the previous step in a graph $G = (V,E)$. A vertex $v \in V$ corresponds to a feature f_i and contains the feature observations $f_i(t)$. The weight $w_{i,j} \in [0..1]$ of an edge $e_{i,j} \in E$ connecting vertices v_i and v_j will indicate the belief that the associated features belong to the same rigid body.

To compute this belief, we employ a set of predictors. Each predictor $P(f_i, f_j)$ estimates the belief that two features f_i and f_j belong to the same rigid body for a specific property, such as color or change in relative distance. The weight of the edge $e_{i,j}$ is then given by the product of the believes of all predictors: $\prod_k P_k(f_i, f_j)$. In our experiments we use $k = 6$ such predictors, each described below.

The **Relative Motion** predictor determines the probability of two features f_i and f_j being on the same rigid body by evaluating their relative distance over time. If the relative distance varies little over time, i.e., if $|\max_t \delta(f_i(t), f_i(t)) - \min_t \delta(f_i(t), f_j(t))|$, where $\delta(\cdot,\cdot)$ is the distance between two features in pixels, is below a noise threshold of 5 pixels, we conclude that f_i and f_j are likely to belong to the same rigid body ($w_{i,j} = 1$). This is illustrated in Figure 3. The figure shows a cabinet door in two configurations, open and closed. Tracked features are marked by pink circles. As the motion is approximately parallel to the image plane, the relative distance between pairs of features changes little over time. The predictor would therefore increase our belief that the pink features belong to the same rigid body. Note, however, that this is *only* true for features undergoing motions parallel to the image plane; the existence of relative motion among features therefore does not imply that they are on different rigid bodies.

The **Short Distance** predictor predicts two features to be on the same rigid body if they are close to each other in the image. It computes a belief value as a function of the

Fig. 3. Illustration of the Relative Motion predictor (for details see text above)

Fig. 5. Illustration of the Color Segmentation predictor (for details see text above)

feature distance $\delta(f_i, f_j)$. If the distance is smaller than 10 pixels, it sets $w_{i,j} = 1$, indicating our belief that the two features are on the same rigid body. Otherwise, it sets $w_{i,j}$ to $\frac{1}{2}$, indicating that the evidence is not conclusive. The **Long Distance** predictor is similar to the previous one. Here, large distances indicate that f_i and f_j belong to different rigid bodies. If $\delta(f_i, f_j)$ is larger than 160 pixels, we set $w_{i,j}$ to 0. If it is smaller than 30 pixels, we set $w_{i,j}$ to $\frac{1}{2}$ (no decision). Otherwise, the belief is a linear function of the distance: $w_{i,j} = 1 - (\frac{1}{2} + \frac{\delta(f_i,f_j)-30}{2\cdot(160-30)})$. Figure 4 shows two clusters of features (pink and blue circles) obtained using only these two predictors.

Fig. 4. Illustration of the Short Distance and Long Distance predictors (for details see text above)

The **Color Segmentation** predictor uses the assumption that rigid bodies have similar visual appearance. It uses color and texture information to segment an image into color-consistent regions (see Figure 5). Segmentation is based on the implementation provided by [7]. Point features that are in the same region are more likely to belong to the same body than points that are in neighboring regions. The more color regions separating between a pair of features f_i and f_j, the weaker is the predictor's belief that they are on the same rigid body. If a pair of features are separated by more than $n = 5$ regions, we set $w_{i,j} = \frac{1}{2}$, indicating neutral belief. Otherwise, the predictor sets $w_{i,j} = 1 - \frac{n-2}{4}$.

The **Triangulation** predictor relies on the insight that features on the same rigid body generally maintain neighborhood relationships throughout short motions of the object: if a feature f_i is to the left of f_j at time t, it is expected to be to the left of that feature at time $t+1$. Do determine this neighborhood relationship efficiently, the predictor relies of the Delaunay triangulation. Figure 6 illustrates this process for the case that a features moves in a way that is not consistent with a single rigid body hypothesis. The left image shows the Delaunay triangulation for a set of features at time t. The right image corresponds to the adjacency relationship at time t for feature locations at time $t+1$. In this example, only one feature (blue circle) has moved, and therefore is not consistent with the other features (orange circles). Features inconsistent with a single rigid body hypothesis can be detected by searching for edge intersections (red circles). Edges between vertices that violate the triangulation are assigned weight of zero, whereas edges that do not violate it are assigned weight of one. This predictor differs from the first four predictors as it formulates a hypothesis encompassing all features, instead of just one pair of features at a time.

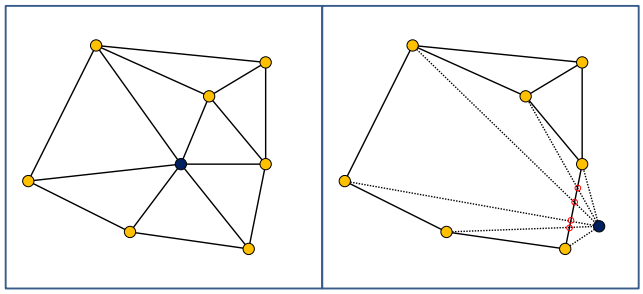

Fig. 6. Illustration of the Triangulation predictor (for details see text above

The **Fundamental Matrix** predictor formulates hypotheses about plausible real-world 3D motion of a subset of the features that could have given rise to their observed 2D trajectories in the image plane. Hypotheses are computed for sets of eight features selected using RANSAC [8] based on the Fundamental Matrix algorithm [13]. This algorithm takes the image plane locations of eight features at times t and $t+1$ and determines a rigid body transform under the assumption that the features are indeed on a single rigid body. The predictor now scores the degree to which the hypothesis explains the trajectories of all other features. Finally, the predictor uses the J-Linkage algorithm [24] to cluster the features into groups of features that most closely match a common hypothesis. We set $w_{i,j}$ to 1 if the motion of features f_i and f_j can be explained by the same fundamental matrix hypothesis; the weight is zero otherwise. Here too, we utilize global information to formulate hypotheses about the motion

of a group of features. Figure 7 shows two views of the same scene. Using eight features (yellow circles) that are matched between the images, we compute the fundamental matrix to explain the motion of the camera between frames. If the eight features are on a single rigid body, we can explain the motion regardless of whether the objects or the camera are moving, or both of them at the same time.

Fig. 7. Illustration of the Fundamental Matrix predictor (for details see text above)

Using these six predictors, the algorithm weights the edges of G; edges with weight zero are removed. Our experimental results in Section IV will show that each predictor adds valuable information that improves the resulting segmentation. Each strongly connected components of the graph now represents the hypothesis that the corresponding features are on the same rigid body.

To break the graph into highly connected subgraphs, we use the weighted min-cut algorithm [4]. We invoke min-cut recursively [14] until cutting a graph requires removing more than half of its edges. Our min-cut algorithm has worst case complexity of $O(nm)$, where n represents the number of nodes in the graph and m represents the number of clusters [18]. In practice, $m \ll n$, as the robot's field of view typically contains only a few objects. We can therefore conclude that for practical purposes clustering is linear in the number of tracked features.

It should be noted that the robustness of our algorithm is to a large extent a consequence of this graph labeling and cut procedure. Noisy features perform motion that is inconsistent with the motion of other features. The corresponding vertices will therefore be disconnected from other vertices. Noisy features can thus be removed, leading to highly robust segmentation.

C. Segmenting Articulated Objects

The previous components of the algorithm have provided us with clusters of features, each hypothesized to be on a different rigid body. The last component of our algorithm uses the object segmentation algorithm by [20] to perform fixation-based object segmentation for each of the clusters.

Ultimately, we would like to compute one segmentation for each object at every time t. Given a cluster of features, we use each feature $f_i(t)$ in that cluster as a fixation point for fixation-based segmentation at time t. We then combine the results of segmentation for each feature in the cluster to achieve the overall object segmentation at time t.

D. Putting Things Together

By combining the three components described in Sections III-A–III-C, we obtain a robust perceptual skill. This skill only makes some general assumptions. One assumption is that the scene contains sufficient texture to identify visual features. The second assumption is that the robot is able to make contact with the environment. Turning this second assumption into a specific manipulation will be the subject of future research. Here, we provide the perceptual capabilities to enable that manipulation.

IV. EXPERIMENTAL VALIDATION

We validate the proposed method for 3D object segmentation in real-world experiments. The experiments were conducted with our robotic platform for autonomous mobile manipulation (see Figure 1). Our robot consists of a holonomic mobile base with three degrees of freedom, a seven degree-of-freedom manipulator arm, and a three-fingered hand. The robot interacts with various articulated objects (see Figure 2). These objects—door, box, wooden-train toy, fridge, laptop, elevator doors, and tricycle—vary in scale, shape, color, and texture. An off-the-shelf web camera with a resolution of 640 by 480 pixels provides a video stream of the scene throughout the interaction. Experiments were conducted without active control of lighting conditions.

In our experiments, a robot interacts with an articulated object to acquire and track a segmentation of its rigid parts over time. The robot tracks the 500 most prominent Lucas-Kanade features in the scene. During the interaction, which was performed using pre-recorded motions, about half of these features are lost. Among the remaining ones, about half are noisy and are discarded by our algorithm.

A. Identifying Rigid Bodies

The task of the first two components of our algorithm is to identify rigid bodies in the scene. Figure 8 shows the results of analyzing the same scene to identify rigid bodies, each time using a different subset of the six predictors described in the previous section. The hypothesis behind our work is that with more information, identifying rigid bodies becomes easier and more reliable. The experimental results in Figure 8 support this hypothesis.

Figure 9 shows seven experiments, one in each row, with different real-world objects, illustrating the performance of our algorithm in identifying rigid bodies in an unstructured scene. The leftmost column shows each object before the robot interacts with it. The second column shows an instance of the interaction itself. The third column shows the final pose of the object (after the interaction). The rightmost column shows the results of clustering the tracked features into rigid bodies. The obtained graph clusters are shown in white. Each cluster corresponds to a hypothesized rigid body. We will now describe each experiment.

In the first experiment (top row), the robot interacts with a box by pushing it and closing the flap. Three clusters are identified by the algorithm: one is associated with the box, the second with the flap, and the third with a static object

Fig. 9. Experimental results showing the process of identifying rigid bodies in a scene using interactive segmentation. Left to Right: The first column shows the object before the interaction; the second column shows the interaction itself; the third column shows the object after the interaction; and the fourth column shows the results of segmenting the graph of tracked features into clusters of features on the same rigid body.

(picture cube). Here, the long distance predictor helps us separate the picture cube from the box. The fundamental matrix predictor identifies the difference in the motion of the three bodies. And the short distance predictor reinforces the connectivity between close features.

In the second experiment, the robot interacts with a door. The fundamental matrix and triangulation predictors differentiate between the door and the frame. The long distance predictor further supports this distinction. Color-based segmentation, in contrast, encourages us to cluster all features together because most of the scene has a uniform yellow texture. If our algorithm relied solely on color, it would fail here. The strong signals provided by the other predictors enable the robot to correctly separate the door from the frame.

The third experiment is similar to the second, except that here the robot interacts with a smaller object—a fridge door. The texture, lighting conditions, and viewing angle are all different. The result, however, remains similar, and the robot differentiates between the door, the frame, and two other distant static objects.

In the fourth experiment the robot interacts with a laptop by pushing it and by opening the lid. As a result, three clusters associated with the three rigid bodies in the scene

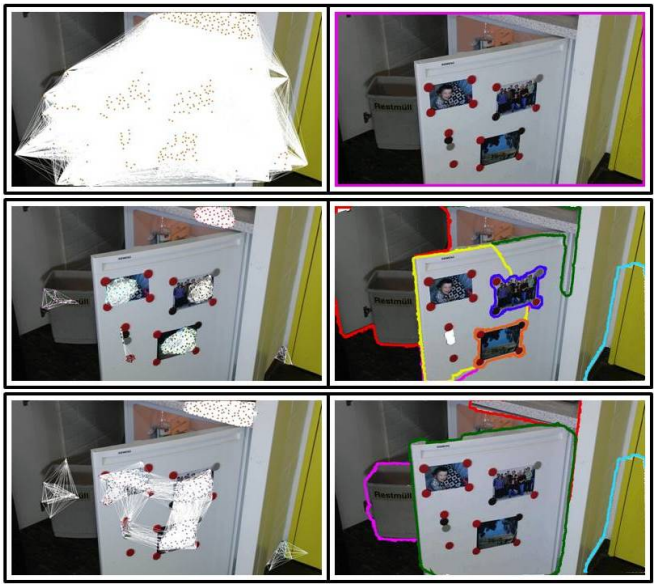

Fig. 8. This figure shows the clusters generated using different mixes of the predictors. Top: the relative distance, short distance and color segmentation predictors were used. They cannot leverage motion or structure violation cues, and therefore cluster all features together. The result is a segmentation of the entire image as one object. Middle: all predictors except for the fundamental matrix were applied. The results are good, but lack the ability to recognize that clusters on the fridge's door move together and therefore belong to the same rigid body. Bottom: here all predictors were used, and indeed the result is a segmentation of the image into the moving body (the fridge door), and some static objects that are spatially far from each other.

are identified: the static power supply, keyboard and screen.

In the fifth and sixth experiment the robot interacts with a prismatic joint: opening a cabinet door, and translating a wheeled table. In both cases the algorithm separates static from moving bodies. It also separates between two static bodies because of the large distance between them.

The last experiment, in row seven, shows the result of interacting with a train toy. Here, color segmentation alone would do very badly because each rigid part is composed of multiple brightly colored blocks, whereas the base of the engine and car have identical wooden texture. The algorithm relies here on the strong motion signal to distinguish between the static object (game tokens) and the two parts of the train.

These experiments show that we can reliably generate accurate hypotheses about which features belong to the same rigid object. Throughout the paper, we show only a selection of the experimental data used to test the algorithm, but all of our ten experiments were successful and show similar results.

B. Computing Object Segmentation

We now demonstrate that the clusters of features obtained by our algorithm serve as good fixation points for fixation-based object segmentation. In our experiments, we use a fixation-based segmentation algorithm developed and implemented by Mishra, Aloimonos, and Fah [20]. To segment a rigid body at time t, we use all features $f_i(t)$ associated with a single rigid body as fixation points. Each fixation point results in a segmentation candidate. We combine all segmentation candidates by including every pixel that appears in any one of the candidates, to produce the body's final segmentation.

Figure 10, shows the results of fixation-based object segmentation for four objects (drawers, laptop, train toy, and a fridge), in two different configurations. Each rigid body is indicated by a surrounding color strip. The resulting segmentations match well with the physical extents of each rigid body, providing information that is vital for manipulating these objects. Because we use a segmentation-based algorithm from the literature, we only show a selection of our experimental results. For a more detailed evaluation of the algorithm, see [20]. In other experiments, we obtained segmentations of comparable quality.

Fig. 10. Image segmentation results for drawers cabinet, a laptop, a train toy, and a fridge (see text for details)

C. Discussion

In all experiments, the proposed algorithm detected, segmented, and tracked the segmentation of all rigid bodies containing a sufficient number of visual features. This robustness is achieved using a low-quality, low-resolution web

camera, without tuning any parameters. Experiments were performed under uncontrolled lighting conditions, different camera positions and orientations, and for different initial poses of the objects. The demonstrated robustness and effectiveness provides evidence that the presented perception skill is suitable for manipulation in unstructured environments.

V. CONCLUSIONS

We presented a perceptual skill in support of manipulation in unstructured environments. This skill identifies and segments articulated rigid objects in unstructured scenes. It tracks the segmentation as the objects perform arbitrary motions in three dimensions. The proposed perceptual skill does not require knowledge of the objects and is computationally efficient.

We believe that this perceptual skill is a necessary capability for reliable manipulation of articulated objects in unstructured environments. To successfully plan and execute a manipulation task, the robot has to identify the boundary of an object so as to know where and how to exert forces onto it. Further, the robot has to track the object's motion to determine the effects of the exerted forces. The presented perceptual skill provides these capabilities in a robust and general way.

There are two key insights that lead to the robustness and generality of the described perceptual skill. The first insight is that segmentation should be focused on individual rigid bodies, rather than the entire image. The second insight is that to achieve robustness in perception, multiple sources of information must be combined. Our perceptual skill computes segmentation using multiple visual cues. The most important cue it uses is motion of the object. An important contribution of this work is the demonstration that purposeful interactions should be considered as part of the robot's perceptual toolbox. We show that deliberate interactions with the environment reveal a perceptual signal that enables robust segmentation in unstructured scenes.

VI. ACKNOWLEDGMENTS

We gratefully acknowledge financial support by the Alexander von Humboldt foundation through an Alexander von Humboldt professorship (funded by the German Federal Ministry of Education and Research) and by the National Science Foundation (NSF) under award number CSE IIS 0545934 funded in the CAREER program.

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] A. Blake and A. Yuille, editors. *Active Vision*. MIT Press, 1992.

[3] N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Automatic 3D object segmentation in multiple views using volumetric graph-cuts. In *British Machine Vision Conference*, pages 530–539, 2007.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

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

[6] 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.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[9] 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.

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

[11] K. G., N. Bergstrm, and K. D. Using symmetry to select fixation points for segmentation. In *International Conference on Pattern Recognition*, 2010.

[12] 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.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[14] E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach, and R. Shamir. An algorithm for clustering cdnas for gene expression analysis. In *RECOMB '99: Proceedings of the third annual international conference on Computational molecular biology*, pages 188–197, New York, NY, USA, 1999. ACM.

[15] D. Katz, Y. Pyuro, and O. Brock. Learning to manipulate articulated objects in unstructured environments using a grounded relational representation. In *Proceedings of Robotics: Science and Systems IV*, pages 254–261, Zurich, Switzerland, June 2008.

[16] J. Kenney, T. Buckley, and O. Brock. Interactive segmentation for manipulation in unstructured environments. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1343–1348, Kobe, Japan, May 12-17 2009. IEEE Press.

[17] 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.

[18] D. W. Matula. Determining edge connectivity in O(mn). *Proceedings of the 28th Symp. on Foundations of Computer Science*, pages 249–251, 1987.

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

[20] A. Mishra, Y. Aloimonos, and C. L. Fah. Active segmentation with fixation. In *Proceedings of the International Conference on Computer Vision*, 2009.

[21] 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.

[22] 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.

[23] 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.

[24] R. Toldo and A. Fusiello. Robust multiple structures estimation with j-linkage. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 537–547, Berlin, Heidelberg, 2008. Springer-Verlag.

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

[26] J. Yan and M. Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 712–719, Washington, DC, USA, 2006. IEEE Computer Society.

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

[28] 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.