# Interactive Perception of Articulated Objects

Dov Katz and Andreas Orthey and Oliver Brock

**Abstract** We present a skill for the perception of three-dimensional kinematic structures of rigid articulated bodies with revolute and prismatic joints. The ability to acquire such models autonomously is required for general manipulation in unstructured environments. Experiments on a mobile manipulation platform with real-world objects under varying lighting conditions demonstrate the robustness of the proposed method. This robustness is achieved by integrating perception and manipulation capabilities: the manipulator interacts with the environment to move an unknown object, thereby creating a perceptual signal that reveals the kinematic properties of the object. For good performance, the perceptual skill requires the presence of trackable visual features in the scene.

## 1 Introduction

We present a perceptual skill for the interactive and autonomous acquisition of complete kinematic models of three-dimensional (3D) rigid objects with prismatic and revolute joints. Such objects are common in everyday environments: doors, drawers, pliers, scissors, tools, etc. Their intrinsic degrees of freedom are coupled with their intended use; knowledge of the kinematic structure of objects is thus a prerequisite for their manipulation. We rely on visual perception to address this challenge as cameras are ubiquitous and cheap. Our task, therefore, is to extract the shapes and kinematic relationships of rigid bodies from two-dimensional (2D) trajectories of image features on these bodies, an inherently ill-defined task.

To successfully devise a skill for perceiving 3D kinematic models, we must find a factorization [11] of the perceptual problem. A factorization is a decomposition of a problem into subproblems, each of which can be solved reliably and whose

Dov Katz and Andreas Orthey and Oliver Brock
Robotics and Biology Laboratory, School of Electrical Engineering and Computer Science, Technische Universität Berlin, Germany

sub-solutions can be composed into a solution to the original problem. In our experience, such a factorization rarely aligns with the established boundaries between the subdisciplines of robotics, such as vision, manipulation, control, or planning. Instead, robust solutions must cross these boundaries and require a close integration of the subdisciplines.

The proposed factorization of our perceptual task has two distinguishing features. First, it eliminates the traditional decomposition of manipulation into separate components for perception and manipulation. This seems to make the problem harder at first—but the opposite is the case: by interacting with objects, the robot excites their degrees of freedom and reveals a pertinent visual signal that would otherwise remain hidden (see Figure 1). Second, we decompose the problem of going from 2D feature trajectories in the image plane to 3D motion estimates into three factors. This represents a different decomposition from the one traditionally used in structure from motion. There, 3D shape estimates are obtained directly from the feature trajectories in the image plane. In contrast, our factorization first segments all feature trajectories in the image plane into groups hypothesized to belong to a single rigid body. This grouping is already influenced by characteristics that result from the 3D motion—but it does not lead yet to any hypotheses about this motion. The second step (factor) then uses the grouped feature trajectories to infer shape and motion information. And finally, the third factor identifies kinematic relationships. This division into three factors will prove to be critical for the robustness of our approach.

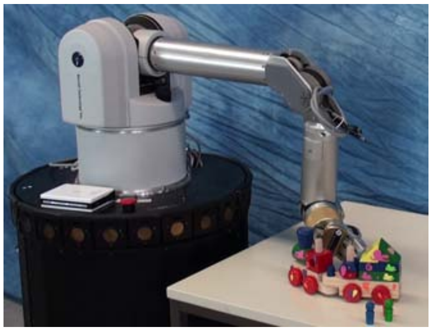


**Fig. 1** Our Mobile Manipulator interacts with a toy train. To extract a kinematic model of the train, it first segments a set of rigid bodies from the background: train engine, train car, and wooden figures. This segmentation is then used to estimate the 3D shape and motion of the rigid bodies. Based on these estimates, it determines the kinematic relationships between the bodies and thus the kinematic model of the train.

We evaluated the proposed perceptual skill with 10 real-world articulated objects. A mobile manipulation platform, equipped with a regular web-cam as the vision sensor, interacted with the various objects. Our approach reliably and robustly identified the kinematic structure of all ten articulated objects, given the assumption that objects contain a sufficient number of trackable features. These results show that our interactive perception approach can robustly acquire kinematic models for previously unseen articulated objects.

The proposed perceptual skill provides an important component for the development of autonomous manipulation capabilities. Traditionally, approaches to robotic manipulation rely on a priori information about the environment. Given sufficient information, these approaches manipulate successfully in controlled factory settings and staged research experiments. They will fail, however, in unstructured environments, i.e. environments for which it is impossible to provide sufficient information a priori. We must therefore develop manipulation skills that do not depend on detailed a priori information. To succeed at general manipulation, robots must be able to acquire information about their environment autonomously, using perceptual skills like the one described here.

## 2 Related Work

To find a factorization for our perceptual problem, we integrate concepts from multiple research areas. We briefly review the methods and techniques that are most relevant to each of the three steps (factors) of our algorithm.

### *2.1 Image Segmentation*

Image segmentation identifies boundaries around image regions with consistent properties [7]. Here, we are concerned with object segmentation, i.e. segmentation of an image region showing a single object. Our perceptual skill must identify the boundaries of objects in the visual stream to compute the kinematic model.

Segmentation methods for single images commonly identify discontinuities in color, texture, brightness, or depth [7, 2]. Regions are determined using thresholding, edge detection, clustering, or region growing to group pixel according to these image properties [7]. The resulting boundaries, however, do not necessarily correspond to object boundaries, as rigid object can be multi-colored, for example, or possess internal depth discontinuities.

In the context of manipulation, an appropriate image segmentation should group together those image regions that belong to a physically connected object. To overcome the problem of single-image segmentation, interactive segmentation has been proposed [6, 14]. These methods rely on the robot's body to induce motion in the observed scene, thereby revealing a visual signal of "objectness". The resulting sequence of images is then analyzed to identify object boundaries. Our work also relies on interaction to solve the segmentation problem but combines it with the techniques of single-image segmentation.

### *2.2 Depth Reconstruction*

Our perceptual skill must identify the shape and relative motion of objects. To achieve this, it must extract three-dimensional scene information from two-dimensional image data. This is exactly what depth reconstruction methods accomplish.

Depth reconstruction can be divided into methods that extract depth from a single image and those that rely on sequences of images. Methods in the first category must make strong assumptions about the settings such as known objects (e.g. hands and faces [20]), known structure (e.g. walls and floors [4]), uniform color [16], or uniform texture [17]. Other methods require training, and employ machine learning techniques [22]. All methods for recovering depth from a single image do not consider motion, which, without assuming prior experience, is the single most conclusive evidence for identifying the three-dimensional structure of a scene.

A second category of methods uses motion information to recover depth. Methods in this category typically assume a single-body rigid world and a moving camera. This category can be further divided into global optimization methods (e.g. bundle adjustment [9, 15]) and recursive estimation methods (e.g. Extended Kalman Filter [3, 24]). Some methods for recovering 3D shape from motion remove the single rigid body assumption [8]. These algorithms, however, are computationally complex and require a long sequence of images. Most existing methods for recovering structure from motion assume that the world is static, i.e. they can only handle a single object that is the entire world.

In this paper, we propose a different decomposition of the depth reconstruction problem, specifically to overcome the assumption that the world is static. We first compute a segmentation of the scene into rigid bodies by using deliberate interactions with the environment, solely based on two-dimensional image information. We then use this segmentation of the camera image into regions belonging to individual rigid bodies to apply classical structure from motion techniques.

### 2.3 Kinematic Modeling

The robotics and computer vision communities have recently begun investigating the problem of autonomously acquiring kinematic models from sensor data. Anguelov et al. [1] obtain kinematic models of doors by tailoring perception to the kinematics of doors. Yan and Pollefeys [25] rely on structure from motion [3] to obtain 3D feature trajectories, and then use spectral clustering to identify rigid bodies and their kinematic relationship. This work assumes affine geometry of the scene and only considers revolute joints. Ross et al. [21] follow the same principle, using maximum likelihood estimation instead of spectral clustering. The strength of the two latter algorithms is that they can handle bodies that undergo slight deformation during the motion. All of the aforementioned approaches only handle revolute joints and make strong assumptions to simplify the perception problem.

Sturm et al. [23] learn models of kinematic joints from three-dimensional feature trajectories, generated from deliberate interactions with the environment. This approach does not attempt to identify rigid bodies in the scene; it assumes that only one object is moving at a time.

In prior work [10, 13], we presented a perceptual skill to extract kinematic models of planar articulated objects. That work considered revolute and prismatic joints. In this paper, we extend this perceptual capability to three dimensions.

## 3 Acquiring 3D Kinematic Structures

We believe that the concurrent estimation of object segmentation, 3D structure, and kinematic structure of a scene with multiple moving bodies is too difficult to solve robustly in a single step. We therefore decompose the problem of perceiving kinematic models into three components—we call them factors. The decomposition is chosen so that each factor can make extensive use of the structure inherent to the problem. We now describe the three factors—segmentation, reconstruction, and joint detection, of our perceptual skill.

### *3.1 Rigid Body Segmentation*

The first factor of our algorithm addresses the task of object segmentation. It leverages structure inherent to the problem by interacting with the world to cause object motion. This gives rise to the visual signal that exposes objectness and thus greatly facilitates object segmentation.

There are two challenges associated with motion-based segmentation. First, object motion must be present. Second, we must decode objectness from the ambiguous and noisy 2D projection of 3D feature motion.

To address the first challenge, we use the robot's manipulation capabilities to induce object motion. By physically causing objects to move, the robot generates a strong perceptual signal for object segmentation. In the current work, the robot's motion is scripted. This restriction will be removed in future work.

To address the second challenge, we leverage another type of structure inherent to the problem: features associated with a single rigid body will have similar spatial, temporal, and appearance characteristics. Color and texture consistency over a spatially contiguous region can imply similarity of material. The distance between features can be indicative of spatial proximity of the corresponding points in the scene. The structural integrity expected of a rigid object imposes structure on the relationship between the projected trajectories of features on the same body. And finally, the observed 2D feature trajectories must be explainable by a possibly ambiguous 3D motion. All of these clues exploit the structure of the problem but by themselves are insufficient to determine an object segmentation. Our algorithm therefore integrates all these clues to generate a combined object segmentation hypothesis.

Segmentation hypotheses are captured in a fully connected multi-graph $G = (V,E)$. A vertex $v \in V$ corresponds to an LK-feature $f_i$ in the image and contains the feature observations $f_i(t) = \{u,v,c\}$, where $(u,v)$ are the feature's image coordinates and $c$ is its color at time $t$. The weight $w(e_{i,j})$ of an edge $e_{i,j} \in E$ indicates the confidence that $f_i(t)$ and $f_j(t)$ belong to the same rigid body. The weight is $w(e_{i,j}) = \prod_k P_k(f_i, f_j)$, where $P_k(f_i, f_j)$ indicates the likelihood of $f_i$ and $f_j$ being on the same rigid object as estimated by one of our six predictors, described below (see also [12]).

**Relative Motion Predictor:** The distance between two features $f_i$ and $f_j$ that belong to the same rigid body and are moving approximately parallel to the image plane changes little over time. The relative motion predictor leverages this heuristic. It computes the maximum change in distance between $f_i$ and $f_j$ over time. If this maximum change is below a noise threshold of five pixels, we conclude that $f_i$ and $f_j$ are likely to belong to the same rigid body, and set $w_{i,j}$ to 1.

**Short and Long Distance Predictors:** If the distance between two features $f_i$ and $f_j$ is small, they are more likely to belong to the same rigid body than to lie on different ones. Conversely, if this distance is large, the features more often belong to different bodies. The short and long distance predictors leverage these two complementary heuristics. They compute a confidence value as a function of the distance $\delta(f_i, f_j)$ between the features before and after the interaction.

**Color Segmentation Predictor:** The color segmentation predictor exploits the fact that image regions sharing similar color and texture are more likely to be part of the same rigid body. It uses color and texture information to segment an image into color-consistent regions [5] (see Figure 2) and then measures the number of color regions separating a pair of features. Point features that are in the same color region are more likely to belong to the same rigid body than points that are in neighboring regions. The more color regions separate a pair of features $f_i$ and $f_j$, the weaker the predictor's confidence is that they lie on the same rigid body.

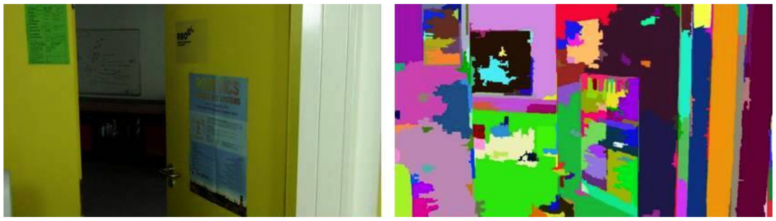


**Fig. 2** Illustration of the Color Segmentation predictor

**Triangulation Predictor:** Features on the same rigid body are likely to maintain neighborhood relationship throughout the motion of the object. The triangulation predictor exploits this structural integrity. It computes a Delaunay triangulation of the features and then updates the position of each feature, maintaining the graph connectivity. If an edge $e_{i,j}$ intersects with another edge, the predictor assigns $w_{i,j} = 0$. If the neighborhood relationships of features remain consistent after the motion, edges are assigned $w_{i,j} = 1$.

Figure 3 shows feature motion that violates structural integrity. 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).

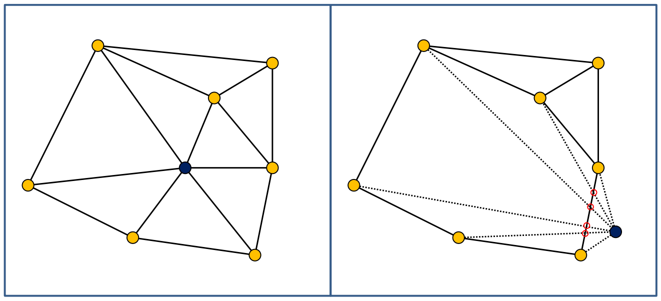


**Fig. 3** Illustration of the Triangulation predictor

**Fundamental Matrix Predictor:** Features on the same rigid body undergo a consistent 3D motion. The recorded trajectories $f_i(t)$ only provide the 2D projection of the true 3D trajectories. The Fundamental Matrix predictor searches for plausible 3D motions that best explain large subsets of the features. It computes motion hypotheses in the form of a fundamental matrix [9] for sets of eight features. It then clusters features into groups whose motion can be explained accurately by the same 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 set to zero otherwise.

To extract an object segmentation hypothesis from the resulting weighted graph, we first discard edges with weight zero. We then use recursive weighted min-cut [18] to decompose the graph into its strongly connected components. Each of these components represents a rigid body hypothesis.

## *3.2 Three-Dimensional Reconstruction*

In the previous section we described how to segment image feature trajectories into rigid body hypotheses. This now enables us to apply standard methods for the extraction of 3D motion from 2D features.

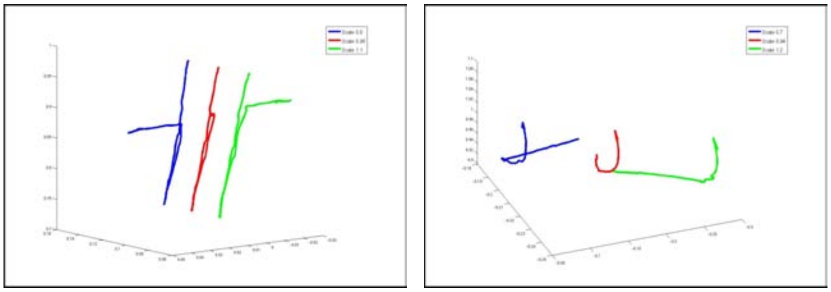
The problem of recovering structure from motion is well studied [24, 3, 9]. Our instance of the problem, however, is more challenging than the standard case. Rather than using features distributed across the entire image, we can only rely on features in small image regions. Furthermore, structure from motion is usually applied to scenes with large depth discontinuities, which aids reconstruction. In our case, manipulable objects often only exhibit small depth discontinuities. Both of these properties make recursive estimation of 3D structure from 2D feature trajectories much more difficult.

The key challenges in our scenario are the initialization and convergence of recursive estimation [19]. We use bundle adjustment [15] to estimate the depth values of all features in the first frame. We then initialize an Extended Kalman Filter [3] with these depth values and let it estimate the 3D motion of the object over time. Using these techniques we get reliable results. However, we believe that this component of our algorithm can be greatly improved by tailoring it to the specific requirements that differ from general structure from motion applications.

### *3.3 Joint Detection*

The last component (factor) of the algorithm computes the kinematic structure of an object. Now that we have performed object segmentation and 3D reconstruction, this step becomes straightforward. The last factor reasons about discrete entities (rigid links) to classify their relative motion into revolute motion, prismatic motion, rigid connection, and disconnected motion.

Joint detection has to overcome one challenge: scale ambiguity. The 3D trajectories obtained by the previous step of the algorithm are only correct up to a scaling factor. Knowing the correct scale is important when comparing the trajectories of two bodies, possibly reconstructed at scales $s_1$ and $s_2$. If $\frac{s_1}{s_2} \neq 1$, the estimated relative motion between the two bodies will be wrong, making it impossible to determine their correct kinematic relationship.



(a) Relative motion between two drawers (prismatic joint)

(b) Relative motion between a laptop's screen and keyboard (revolute joint)

**Fig. 4** The relative motion between two rigid bodies at three different relative scales. At the correct relative scale (red), it is easy to infer the joint between the bodies.

Our algorithm searches for a scaling factor $\lambda$. This factor is used to re-scale the first body so its new scale is $\lambda \cdot s_1$. For every joint type, the algorithm selects the $\lambda$ at which the relative motion between the bodies is best explained by that type. The range in which we need to search for $\lambda$ is limited, as object parts typically have a similar size. The joint type that best explains the data is incorporated into the kinematic model of the object.

Figures 4(a) and 4(b) illustrate the importance of correcting the relative scale before determining the kinematic structure. The left image shows the relative motion between two drawers at three different relative scales. The right image shows the relative motion between a laptop's monitor and keyboard. In both cases, the red curve shows the correct scale (as detected by the algorithm). At this relative scale, the algorithm correctly detects a prismatic joint between the drawers and a revolute joint between the laptop's screen and keyboard.

Once all joints have been identified, the kinematic structure of the articulated object can be computed up to a single scaling factor. Scale ambiguity can now be resolved using the robot's proprioception to compute the true depth of one feature.

We now describe how the different joint types are detected, given the relative motion of two rigid bodies and a hypothesis $\lambda$ about their relative scale:

**Revolute Joint:** If two rigid bodies are connected by a revolute joint, features on these bodies perform a relative motion in the form of parallel arcs centered on the rotational axis. To compute whether a set of relative motion trajectories can be explained by a revolute joint, we project the trajectories onto a plane we compute from the feature trajectories; this plane is orthogonal to the hypothesized rotational axis. We then scale the projected trajectories and compute how well they match a unit circle. Based on this match, the algorithm computes a confidence value, indicating the degree to which it is certain that the two bodies are connected by a revolute joint.

Figure 5 illustrates the identification of revolute joints. The top row of images shows the relative motion between two rigid bodies. The bottom row shows the relative motion projected onto the plane, and scaled to best fit inside the unit circle. In the left column, the relative motion trajectories have the form of parallel arcs, and indeed, the projected trajectories match the unit circle well. The relative motion in the right column belongs to two disconnected rigid bodies.

**Prismatic Joint:** If two rigid bodies are connected by a prismatic joint, features on these bodies perform a relative motion in the form of straight, parallel lines of equal lengths. The algorithm translates the relative motion trajectories to the origin of an arbitrary coordinate frame. We then fit a cylinder around the projected trajectories. Based on the diameter of this cylinder and the variance of trajectory lengths, we determine a confidence value for the presence of a prismatic joint.

**Rigid Joint:** To detect a rigid joint, we simply search for a scale correction $\lambda$ at which the relative motion between the bodies is almost zero. If such a relative scale exists, we declare the bodies to be rigidly connected. This type of joint is important, as it can correct over-segmentation in the previous steps of the algorithm.

**Disconnected Joint:** In all cases, the algorithms computes a value indicating its confidence in the detected joint type. If the algorithm's confidence values for the rigid, prismatic, and revolute cases are low, we declare the two bodies to be disconnected (or connected by six degree-of-freedom joint).

## 4 Experimental Validation

We validate the proposed method for perceiving 3D rigid articulated objects in real-world experiments. The experiments are conducted with our robotic platform for

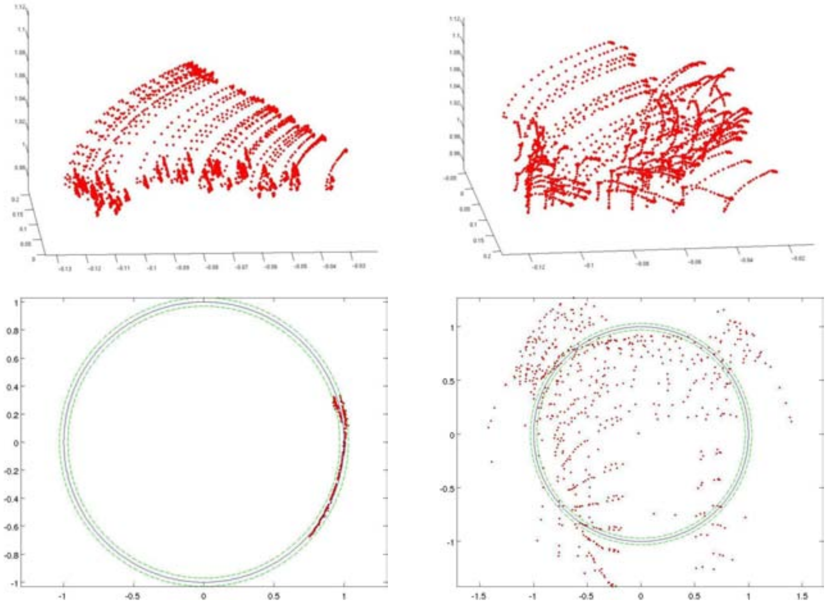


**Fig. 5** Detecting a revolute joint (see text for details)

autonomous mobile manipulation (see Figure 1). The robot interacts with various articulated objects (Figures 6 and 7). These objects vary in scale, shape, color, texture, and kinematic structure. An off-the-shelf web camera with a resolution of $640 \times 480$ pixels provides a 30 frames-per-second video stream of the scene throughout the interaction.

Figure 6 shows two experiments in detail, illustrating the performance of our algorithm in identifying rigid bodies in an unstructured scene. The top row shows two objects, a toy train and a cupboard, before the robot interacts with them. The second row shows a snapshot of the interaction, the third shows the results of clustering the tracked features into rigid bodies (edges of the graph are shown in white). The fourth row illustrates the joints detected between the rigid bodies (purple lines) and the clustered features (one color per cluster).

In the first experiment (left column of Figure 6), the robot interacts with a prismatic joint by sliding the cupboard's door. The algorithm separates static from moving bodies. It also segments the static picture cube and the non-moving door as separate objects. The detection of the prismatic joint between the moving door and the two static objects is done with a very high confidence value of 99.7%. The rigid joint between the static door and the picture cube is detected with 100% confidence.

The second experiment (right column of Figure 6) shows an interaction with a train toy by pushing its parts. Here, color segmentation alone would fail as each rigid part is composed of multiple brightly colored blocks, and the base of the engine and car have identical wooden texture. Instead, our algorithm relies on the strong motion signal to distinguish between the static wooden figures and the two parts of the train.

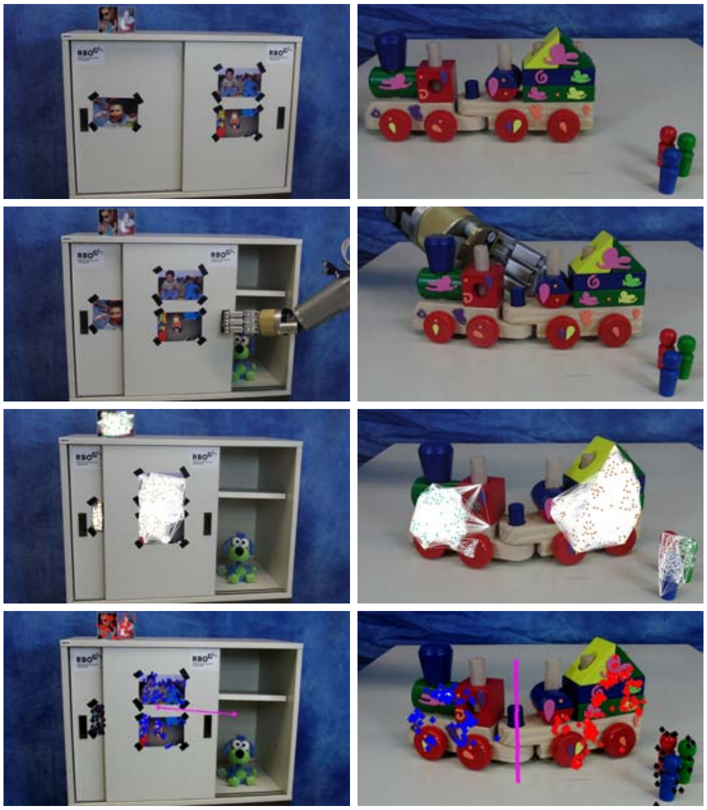


**Fig. 6** Experimental results showing the process of perceiving the shape and kinematic structure of articulated objects (see text for details).

The train's revolute joint is detected with a confidence value of 97%. The train is also segmented from the background by a disconnected joint with 95% confidence.

Figure 7 shows the results of eight additional experiments. Each image shows an articulated object, overlayed with the joints detected between its rigid parts (pink lines). We now describe these experiments (left-to-right and top-to-bottom).

In the first experiment, the robot pushes a tricycle, making the wheel spin. Joint detection assigns high confidence values to a disconnected joint between the background and both the wheel and frame. It also discovers the revolute joint between the wheel and the frame of the tricycle (99% confidence). Our algorithm works in this case because it can detect and analyze any number of rigid bodies, even when moving simultaneously.

The second experiment shows an interaction with an elevator. Here, the robot actuated the elevator by pressing a button. The algorithm correctly discovers three prismatic joints between the two doors and the frame of the elevator (all three confidence values are above 95%). Two joints are between the the frame and the two doors; the

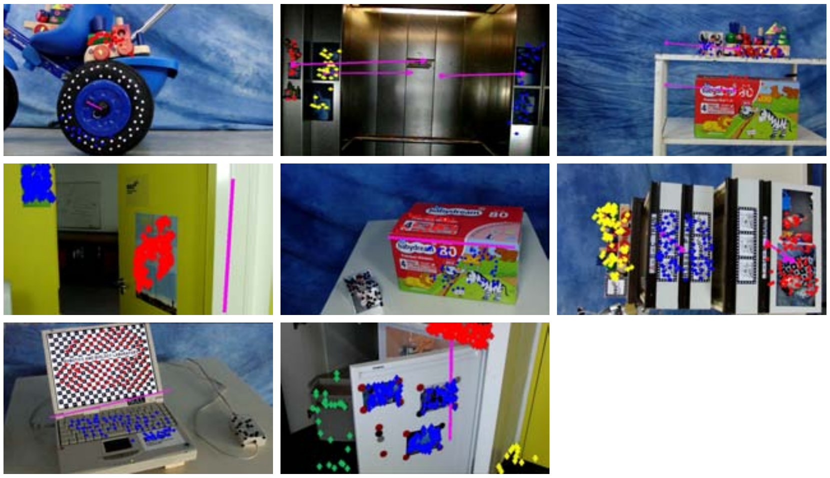


**Fig. 7** Experimental results showing the perception of 3D articulated objects (see text for details).

third joint—without physical manifestation—exists between the two doors and is indicated by the longer purple line.

In the third experiment, the robot pushes a shelf on wheels. The algorithm separates the moving bodies from the static background. The objects on the top and bottom shelves are identified as different rigid bodies, due to the large relative distance between them. However, the algorithm detects that the two bodies are rigidly connected, effectively correcting for over-segmentation. Should the robot interact with one of the objects by itself, of course, it would discover additional degrees of freedom and update the kinematic model accordingly.

In the fourth experiment, the robot opens a door. The algorithm correctly separates the door from its frame, and detects a revolute joint with 100% confidence.

The fifth experiment shows the result of interacting with a box. The algorithm identifies three clusters: box, flap, and picture cube (static). It determines, with high confidence, a revolute joint between the two box parts (97%) and that the flap is disconnected from the background (93%). Joint detection is less certain about the connectivity between the body of the box and the background (80% disconnected, 20% revolute). The lower confidence is due to the fact that the robot has actually pushed the box along an arc. Another interesting property of our method is the ability to adapt to new evidence. Further interaction with the box is likely to generate trajectories that are more difficult to explain as revolute, resulting in increased confidence that the parts are disconnected.

In the sixth experiment, the robot interacts with a drawers cabinet (the image is rotated 90 degrees counter-clockwise). The algorithm identifies the two drawers and the frame. It detects the kinematic structure (the prismatic joints) with high confidence (above 97% in all cases).

The seventh experiment shows the result of interacting with a laptop by pushing it and opening the lid. Three clusters are identified: a static power supply, keyboard, and screen. The revolute joint between the keyboard and screen is detected correctly (95%). The algorithm also detected that the screen is disconnected from the static power supply (80%). The robot's interaction with the laptop was such that it moved along an arc. As a result, a revolute joint is detected between the keyboard and the background (power supply), with a low confidence of 70%. Further interactions would reveal that the keyboard is disconnected from the background.

In last experiment the robot opens a refrigerator door. The algorithm detects one cluster associated with the refrigerator door and three static background clusters. It detects a rigid joint between the static clusters and a revolute joint between the door and the background with 100% confidence. Because the observed motion was small, the position of the axis is not very precise.

In all experiments, the proposed algorithm detected, segmented, and tracked all rigid bodies containing a sufficient number of visual features. The algorithm successfully obtained the kinematic structure in 10 out of 10 experiments. It detected the position and orientation of the joint correctly in 30 out of 31 cases. The one exception is the refrigerator experiment, in which the type of joint and the orientation of the axis of rotation are correct, but the position of the axis is offset. 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.

For our experiments, we do not have available the ground truth of the kinematic models in the scene. We therefore rely on visual inspection to judge the effectiveness of our method. Ultimately, we will combine the proposed perceptual skill with manipulation skills. Then we will be able to determine if the accuracy of the extracted kinematic models is sufficient to guide manipulation planning and execution. Given the results presented here, however, we are confident that this is the case.

The greatest limitation of the described perceptual skill is its dependency on the presence of trackable visual features on each of the rigid bodies. Given our low-resolution camera, we had to add artificial features in seven of the ten experiments (e.g. place a poster on a feature-less door). A foveated or higher-resolution vision system would reduce the need for artificial features.

The runtime of all three steps of the algorithm depends on the number of tracked features as well as the number of rigid bodies in the scene. In our experiments, scenes were composed of 2-5 rigid bodies, and between 200 and 400 features were tracked. The runtime of the algorithm varied from 5 to 10 minutes. We believe that a significant improvement will be achieved by optimizing the implementation. Further improvement could be achieved by parallelizing parts of the code.

## 5 Conclusion

We presented a perceptual skill for manipulation in unstructured environments. This skill enables the autonomous acquisition of 3D kinematic models for articulated rigid objects in the world. This ability is a prerequisite for determining appropriate motion plans for manipulation, monitoring a plan's execution, detecting its completion, and identifying failures. To achieve this, the presented perceptual skill interacts with the environment to cause the actuation of degrees of freedom, analyzes its observation of this interaction, and finally determines a fully instantiated kinematic model of the observed objects.

Our experiments showed the successful acquisition of 3D kinematic models of ten real-world objects. No prior knowledge of the objects was assumed. The kinematic models were accurate, even in the presence of substantial noise due to the use of a low-quality camera. The proposed skill does require, however, visual evidence of the kinematic degrees of freedom in the form of motion and a sufficient number of trackable features on each of the rigid objects in the scene.

The skill's robustness is a consequence of the careful decomposition into three components. The first component identifies rigid objects in the scene, based on 2D feature trajectories. The second component applies structure from motion techniques to identify the motion of features associated with a single object. The last component determines the kinematic relationship between pairs of rigid bodies. Each of the components is able to leverage structure inherent to the sub-problem, thus allowing for a robust solution of the overall problem.

## Acknowledgments

We gratefully acknowledge the generous support of the Alexander von Humboldt Foundation through the Alexander von Humboldt Professorship, financed by the German Federal Ministry of Education and Research (BMBF). We are equally grateful for support by the National Science Foundation (NSF) under grants CNS-0454074 and IIS-0545934 and by the EC within the 7th Framework Programme under grant agreement no FP7-248258-First-MM.

## References

1. D. Anguelov, D. Koller, E. Parker, and S. Thrun. Detecting and modeling doors with mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3777–3784, 2004.
2. A. Blake and A. Yuille, editors. *Active Vision*. MIT Press, 1992.
3. A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):523–535, 2002.
4. E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2418–2428, Washington, DC, USA, 2006. IEEE Computer Society.
5. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

6. 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.
7. D. A. Forsyth and J. Ponce. *Computer Vision – A Modern Approach*. Prentice Hall, 2002.
8. 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.
9. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
10. 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.
11. D. Katz and O. Brock. A factorization approach to manipulation in unstructured environments. In *14th International Symposium of Robotics Research*, pages 1–16, Lucerne, Switzerland, August 31-September 3 2009. Springer Verlag.
12. D. Katz and O. Brock. Interactive segmentation of articulated objects in 3D (under review). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
13. 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.
14. 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.
15. M. A. Lourakis and A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.
16. A. Maki, M. Watanabe, and C. Wiles. Geotensity: Combining motion and lighting for 3D surface reconstruction. *International Journal of Computer Vision*, 48(2):75–90, 2002.
17. J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces, 1997.
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. J. Montiel, J. Civera, and A. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*, August 2006. This is a prestigious new single-track international robotics conference with online-only proceedings.
20. T. Nagai, T. Naruse, M. Ikehara, and A. Kurematsu. HMM-based surface reconstruction from single images. In *International Conference on Image Processing*, volume 2, pages II–561 – II–564 vol.2, 2002.
21. D. A. Ross, D. Tarlow, and R. S. Zemel. Unsupervised learning of skeletons from motion. In *Proceedings of the European Conference on Computer Vision*, pages 560–573, Berlin, Heidelberg, 2008. Springer-Verlag.
22. A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 76(1):53–69, 2008.
23. J. Sturm, K. Konolige, C. Stachniss, and W. Burgard. Vision-based detection for learning articulation models of cabinet doors and drawers in household environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010.
24. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
25. J. Yan and M. Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.