Occlusion-Aware Reconstruction and Manipulation of 3D Articulated Objects

Xiaoxia Huang, Ian Walker, and Stan Birchfield
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
{xhuang,iwalker,stb}@clemson.edu

***Abstract*— We present a method to recover complete 3D models of articulated objects. Structure-from-motion techniques are used to capture 3D point cloud models of the object in two different configurations. A novel combination of Procrustes analysis and RANSAC facilitates a straightforward geometric approach to recovering the joint axes, as well as classifying them automatically as either revolute or prismatic. With the resulting articulated model, a robotic system is able to manipulate the object along its joint axes at a specified grasp point in order to exercise its degrees of freedom. Because the models capture all sides of the object, they are *occluded-aware*, enabling the robotic system to plan paths to parts of the object that are not visible in the current view. Our algorithm does not require prior knowledge of the object, nor does it make any assumptions about the planarity of the object or scene. Experiments with a PUMA 500 robotic arm demonstrate the effectiveness of the approach on a variety of objects with both revolute and prismatic joints.**

I. INTRODUCTION

An important trend in robotics is that of autonomous operation in unstructured and dynamic environments. As robots move into unstructured environments such as homes, schools, and workplaces, new approaches to sensing and manipulation will be required to handle the greater variety of objects encountered. For example, rather than expecting the robot to have advanced knowledge of all objects that will be encountered in the physical world, the ability to actively learn about the scene will be crucial.

One problem that has recently caught the attention of researchers is that of reconstructing *articulated* objects [9], [19], [20], [21]. These objects consist of rigid links connected by one or more revolute or prismatic joints. A number of everyday objects, such as laptop computers, staplers, scissors, cabinet drawers, doors, and some cell phones fit such a model. Even a desk or chair sliding on the floor can be modeled, to some degree, by prismatic and revolute joints.

Existing work on articulated objects has been limited, however, to reconstructing surfaces from a single viewpoint. By tracking features in video, clustering those features, and triangulating, the 3D coordinates of a number of points can be estimated, along with the axis of rotation or sliding. However, such results do not yield any information about the back side of the object. Such an approach leaves the robot helpless in any scenario in which it needs to manipulate parts of the object that are not visible in the current view.

In this paper we introduce the term *occlusion aware* to refer to a system that has knowledge about parts of the object that are not currently visible. In grasping systems in which the robot has a complete 3D CAD model of the object, such knowledge has always been assumed. However, in an interactive system in which the robot is learning about the environment as it moves in space, such complete knowledge is not usually available.

We describe an occlusion-aware reconstruction system that is able to recover complete 3D models of articulated objects. Images are captured of the object in two configurations, and structure-from-motion techniques are used to reconstruct two 3D models of the object. Procrustes analysis and RANSAC sampling are then used to automatically segment the points and align the links, from which a straightforward geometric approach enables the estimation of the revolute or prismatic axes. Once the model has been recovered, the robot can align its coordinate system with that of the model and then manipulate the object by exercising the degrees of freedom captured by the model. Our approach makes no planar assumptions, either on the shape of the object surface, the motion of the object, or the viewpoint from the camera. We show the results of the system on a variety of everyday objects, demonstrating the effectiveness of the approach.

II. PREVIOUS WORK

One approach to articulated reconstruction is based upon the factorization method [1], [22]. Exploiting the so-called *rank constraint*, this method reveals the structure of multiple independently moving bodies by examining the block diagonal structure of the measurement matrix consisting of coordinates of tracked feature points. Building on this work, Tresadern and Reid [23] add articulation constraints to the formulation to decrease the rank, enabling the technique to detect the articulated objects, determine their degrees of freedom, and locate the joints. Because the factorization method is limited to affine reconstruction, an additional self-calibration step is needed to convert measurements to a metric coordinate system.

Yan and Pollefeys [28] also investigate the subspace properties of articulated motion in a factorization framework. Their approach segments feature trajectories by local sampling and spectral clustering, after which it builds the kinematic chain as a minimum spanning tree of a graph constructed from the segmented motion subspaces. Focused on recovering human articulated motion, they derive rank constraints for both revolute (1 DOF) and ball (2 DOF) joints, assuming affine projection. Similar work by Ross et al. [14] recovers the kinematic structure of the articulated object as a skeleton using a probabilistic approach. Due to the difficulty of tracking features through occlusion, their video-based approach is also limited to single viewpoints, as well as to affine projection. Other researchers focusing on human motion aim to recover the joint parameters of the human from video or motion capture [4], [7], [10], [13], [16].

In the robotics community, Katz et al. [9] track feature points in video, then perform motion segmentation, and finally 3D reconstruction. Assuming that the plane perpendicular to the axis of rotation, or the plane in which sliding occurs, is known, the revolute or prismatic axis is recovered. Relying on feature points, this approach is also restricted to single-view reconstruction. Other researchers have developed approaches to recover articulated motion using known models [29] or revolute axes using planar surfaces [15].

In other robotics work, Sturm et al. [21] present an approach to learn kinematic models based on observations from a motion capture system that tracks the poses and orientations of rigid parts. A mixture of parameterized and parameter-free (Gaussian process) representations is used to detect the connectivity of the rigid parts of the objects and to find low-dimensional articulation models that best explain the given observation. In related work, Sturm et al. [20] present an approach to learn articulation models of cabinet doors and drawers without using artificial markers. Rectangles in depth images obtained from a self-developed active stereo system are detected using a sampling-based approach, assuming that object surfaces are planar. Then the robot uses generative object models to estimate the type of articulation (revolute or prismatic). Once obtained, these articulated models are used to interact with the environment [19].

III. OCCLUSION-AWARE RECONSTRUCTION

The purpose of this work is to automatically learn the properties of *a priori* unknown articulated objects in unstructured environments in order to facilitate further manipulation of those objects. Figure 1 shows an overview of our system. First, a set of images is captured by a camera of the object from different viewpoints while the object remains stationary. Structure-from-motion techniques are applied to the imagery to build a 3D model of the object. In order to learn the object's kinematic structure, the configuration of the object is interactively changed by exercising its degrees of freedom.

We assume that the capability of performing sufficient exploratory interaction with the object to change its configuration is present. In this way, the approach bears some resemblance to interactive perception [8], [9], [25], [26], [27], except that we allow a human to perform the interaction due to the specific constraints of articulated motion in our objects. Automatically planning the end effector motion path for interactive perception in such situations remains an unsolved problem, because a preliminary model (at least) is needed in order to interact with the object, but the interaction is necessary to estimate the model. Therefore, having the user perform the interaction enables us to escape this difficult chicken-and-egg problem. If progress is made toward developing such autonomous exploratory behavior in the future, the reconstruction method described in this paper would still apply.

Additional images are gathered of the object in the new configuration, and structure-from-motion is again applied to obtain a different 3D reconstruction. These two 3D models are segmented into the object's constituent components (rigid links) using the Procrustes analysis method combined with a RANSAC sampling strategy. From this information, the axis of each joint between neighboring links is found using a geometric method utilizing an axis-angle representation. Based on these models, the robot with eye-in-hand can automatically compute the transformation between the object and robot coordinate systems, enabling it to manipulate the object around the articulation axis with a given grasp point. Due to the complete 3D model, the robot can also interact with occluded, unseen parts of the object, as shown in Figure 2. We now describe these steps in detail.

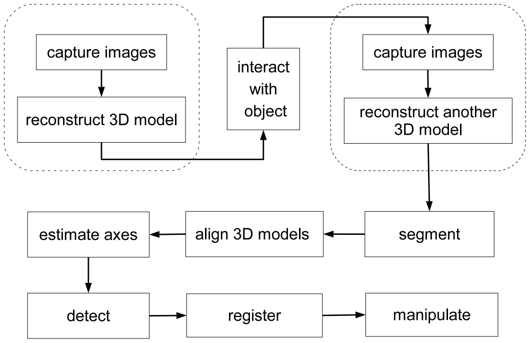

Fig. 1. Overview of our system. Two 3D models of the object are constructed using structure-from-motion techniques applied to two sets of images of the object taken while the object is in two different configurations. After segmenting and aligning the links, the joint axes between neighboring links are found. Manipulation of the object is then enabled by automatically registering the object and robot coordinate frames.

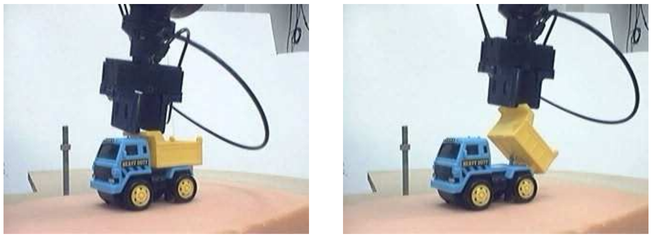

Fig. 2. The PUMA 500 robotic arm manipulates a toy truck using the truck's kinematic model obtained by the occlusion-aware articulated reconstruction procedure.

A. Object Model

We assume that the articulated object is composed of rigid links connected by joints. The object can have any finite number of links and joints, and the joints can be revolute or prismatic. Figure 3 shows the two object models for simplified cases of just two links and one joint. In the case of a revolute joint, the configuration between the two links is represented by the joint angle, while in the case of a prismatic joint, the configuration is represented by the displacement. In both cases, the axis is a ray in 3D space about or along which the movement occurs. The links are represented as 3D point clouds, with a coordinate system attached to each link to enable its position and orientation to be described.

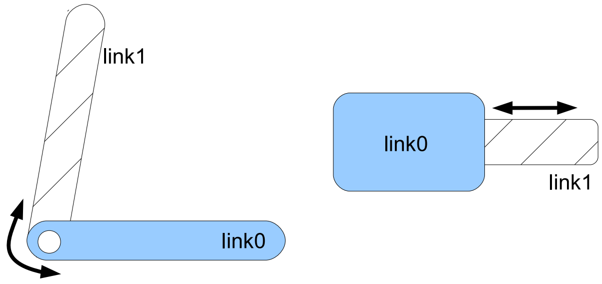

Fig. 3. LEFT: Two rigid links connected by a revolute joint. RIGHT: Two rigid links connected by a prismatic joint.

B. Camera Calibration and Initial 3D Models

Given a set of images from different viewpoints of the object in one configuration, SIFT and patch-based feature points are detected and matched, and structure from motion (SfM) algorithms are used to estimate the camera positions and the 3D coordinates of points on the object. The same procedure is applied to a set of images obtained of the object in a different configuration to yield a second 3D model, where in the second configuration all adjacent links have moved relative to each other. Note that only two configurations are needed, no matter how many links and joints. The Bundler algorithm [17], [18] is used to calibrate the camera and compute the camera locations. Patch-based multi-view stereo (PMVS) [5], [6] is used to reconstruct dense 3D oriented points, where each point has an associated 3D location, surface normal, and a set of visible images. Figures 4 and 5 show the dense 3D reconstruction of a toy truck in two different configurations. Notice that no assumption is made regarding the planarity of the object geometry or the uniformity of the background. There is no constraint on the set of images, except that there must be sufficient overlap in the fields of view in order to facilitate feature matching across different views. In our experience, successive camera viewpoints should differ by no more than about 10 degrees, so that approximately 36 images are needed to capture an accurate 360-degree model; more images are needed to reconstruct the top or bottom of the object.

C. Rigid Link Segmentation

Once the 3D models have been found, the next step is to segment the 3D oriented points into the constituent rigid components of the object. This involves automatically determining the number of links, segmenting the data into the different sets (one set per link), and determining which

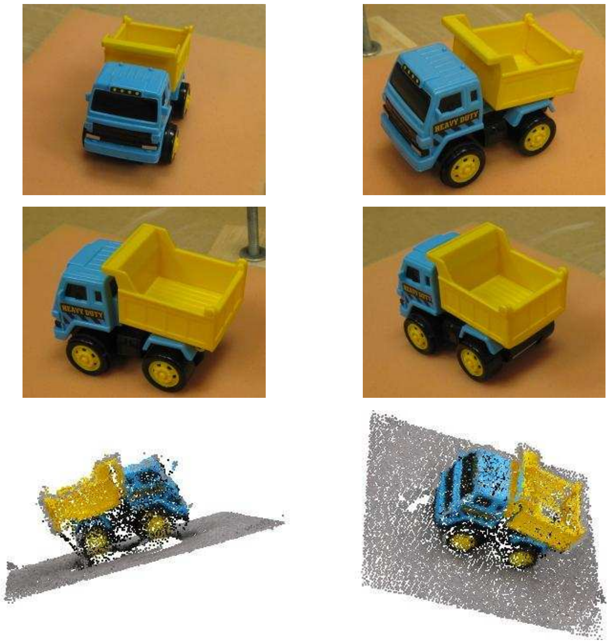

Fig. 4. Four images (out of 112 captured) of a toy truck. The last row shows the 3D reconstruction (two different views) obtained by the structure-from-motion procedure. An occlusion-aware model is produced in which points on the back side, as well as the front, are recovered.

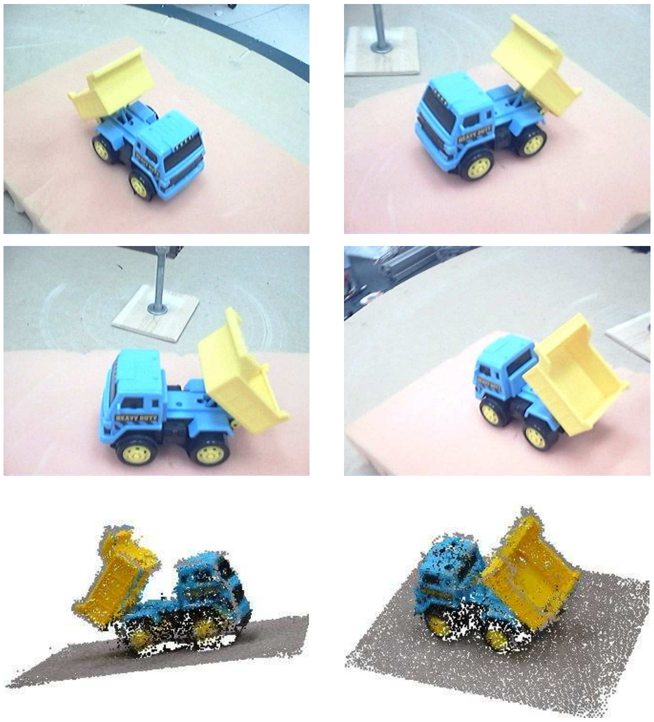

Fig. 5. Four images (out of 132 captured) after interacting with the truck to change its configuration. The last row shows the 3D reconstruction (two different views) by the structure-from-motion procedure. Notice the significant change in lighting conditions.

links are connected. When there are just two links (as in the case of the toy truck, which is composed of a cab and bed), the latter step is trivial, but in more complicated scenarios it is important to determine which links share a joint. Segmentation is performed by clustering feature points according to similarity transformations, using a combination of SIFT, Procrustes, and RANSAC algorithms.

Feature points are found using the SIFT feature detector [11] because of its invariance to image rotation, scaling, and lighting changes. SIFT provides a local descriptor for each feature in addition to the feature's location. For every feature point in the first configuration, a matching feature point in the second configuration is sought. The matching feature point is the one that yields the minimum Euclidean distance of SIFT descriptors, if that distance is below a threshold. If such a match is found, then the same matching algorithm is run in reverse by switching the role of the images, and matches are retained if they agree in both directions. These 2D matches are converted to 3D matches using the 3D model points whose projection lies closest to the corresponding SIFT features.

Given 3D feature correspondences between the two models, our aim is to segment the 3D model points according to their similarity transformations. Similarity transformations include rotation, translation, and scale, the latter being needed to handle the scale ambiguity in images. A RANSAC sampling strategy [3] is adopted in which randomly selected triplets of correspondences are used to compute putative similarity transformations using Procrustes analysis. (For Procrustes, we use the SVD algorithm [2].) Each resulting transformation is used to align the 3D point clouds, yielding an alignment error computed as the mean squared error of the Euclidean coordinates of the corresponding points. Within the RANSAC framework, this process is repeated 1000 times using different randomly selected triplets, and the transformation that results in the smallest alignment error is retained. Using this transformation, all model points that transform to coordinates within some threshold of their match are segmented as a new link. These points are then removed from the models, and the entire process is repeated until no more links can be found (i.e., the smallest alignment error exceeds a threshold).

D. Classifying Joints

We assume that two rigid links are connected by either a revolute joint or a prismatic joint. The type of joint is automatically determined by examining the similarity transformation R, $\mathbf{t}$, and σ between the links determined by Procrustes alignment, where R is the rotation matrix, $\mathbf{t}$ is the translation vector, and σ is the relative scaling between the two models. Although one might be inclined to use the translation vector $\mathbf{t}$ to distinguish between the two types of joints, it is important to note that $\mathbf{t}$ will not in general be zero for a revolute joint. This is because the axis of the coordinate system attached to the link does not necessary (and usually will not) align with the axis of rotation. In other words, although we are interested in rotation about the axis, Procrustes computes the rotation about the origin of the coordinate system, which is somewhat arbitrarily determined by the structure-from-motion technique. While these rotations themselves are identical, a non-zero translation $\mathbf{t}$ is needed to compensate for the misalignment. As a result, we instead determine the type of joint automatically by examining the rotation matrix R: If R is close to the identity matrix, then the joint is determined to be a prismatic joint; otherwise it is a revolute joint. This procedure is repeated for each pair of adjacent links.

E. Finding the Axes of Prismatic Joints

Locating the axis of a prismatic joint is straightforward. Given two links connected by a prismatic axis, we assume that the smaller link moves relative to the larger one, where the size of the link is determined by the number of points in the link. Therefore, the unit vector $\mathbf{t}/\|\mathbf{t}\|$ from the larger link to the smaller link yields the direction of motion along the prismatic joint, while the centroid of the points of the smaller link provides the 3D coordinates of a point on the axis.

F. Finding the Axes of Revolute Joints

Locating the axis of a revolute joint requires a bit more care. In a two-dimensional plane, it is a simple matter to show that any Euclidean transformation (rotation plus translation) can be represented as a rotation applied to translated points. In other words, $R\mathbf{x}+\mathbf{t} = R(\mathbf{x}-\omega)+\omega$, where $\mathbf{x} \in \mathbb{R}^2$ is a point in the plane, and $\omega \in \mathbb{R}^2$ are the coordinates of the axis of rotation. Thus, instead of rotating and then translating a point, this alternate formulation involves shifting the origin of the coordinate system, applying the rotation, then shifting the origin back. As a result, the temporary origin ω about which the rotation is applied specifies the axis of rotation. Figure 6 shows the axis of rotation, indicated by a red dot, estimated by our system for two different objects.

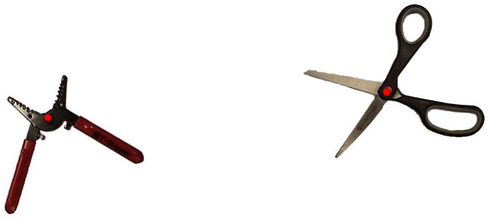

Fig. 6. Revolute axis estimation of pliers (left) and scissors (right) in 2D found by our method. The red dot indicates the estimated axis of rotation.

In 3D, the axis of a revolute joint is defined by a unit vector $\mathbf{u}$ indicating the axis direction and one (somewhat arbitrary) point on the axis. Using the axis-angle representation, the rotation matrix between two corresponding links in the two configurations is parameterized as a unit vector indicating the direction of a free vector parallel to the axis of rotation (the axis direction), and an angle describing the magnitude of the rotation about the axis in the right-hand sense. After aligning the 3D models, and segmenting into links, the transformation between two corresponding links in the two configurations is computed using Procrustes analysis. Given the rotation matrix, the axis direction is found by noticing that any vector parallel to the rotation axis must remain (by definition) unchanged by the rotation, i.e., $R\mathbf{u} = \mathbf{u}$, where **u** is a vector parallel to the rotation axis. From the definition of eigenvalues and eigenvectors, this means that the axis direction is the eigenvector of R corresponding to the eigenvalue of one.

Once the direction of the axis **u** has been found, the 3D rotation about this axis can be thought of as a 2D rotation in the plane perpendicular to **u**. To find the rotation plane, a series of rotations are applied to two configurations such that the z axis of a new coordinate system points in the **u** direction and the xy plane specifies the plane in which the rotation occurs. Then, similar to the 2D axis estimation above, a point in the 2D plane indicating the axis is found, which then yields a 3D point on the axis by appropriate transformations. Together these two parameters (the axis direction and a point on the axis) determine the axis.

IV. MANIPULATING OBJECTS

The resulting occlusion-aware 3D articulated model can be used to enable a robot to manipulate the object. Two capabilities are supported by such a model. First, given a particular point on the object, the robot can move its end effector to that position, even if the point is not visible in the current view. This is one of the main advantages of the occlusion-aware approach, namely, that the robot is not limited only to the side of the object that is currently visible, but rather that a full 3D model is available. Secondly, given a particular grasp point, the robot can grab the object at that point and move in such a way so as to exercise the articulated joint.

The first step for manipulation is to estimate the transformation between the object model and the robot coordinate frame. To make this a Euclidean transformation, we first must overcome the scale ambiguity. The scale of the object can be estimated in one of several ways. If the camera is attached to the robot during capture time, then the known positions of the end effector can be compared with the estimated camera positions to determine the overall scale of the scene. Alternatively, a separate step can compute the projective distance from the camera to the table, which is then compared with the known height of the table. A third alternative is to simply use a known length on the object.

In any case, once the scale is known, the perspective n-point (PNP) problem [12] can be solved for the Euclidean transformation between the object coordinate system and the camera coordinate system. The transformation between the latter and the robot coordinate system can be computed off-line by a standard hand-eye calibration procedure [24], with the camera mounted on the robot. The PNP procedure begins with 3D-2D point correspondences found by extracting and matching SIFT features [11] in the image with 3D points in the model. Using these correspondences, it minimizes the reprojection error that is the sum of squared distances between the observed projections and the projected points, using the camera matrix and lens distortion coefficients obtained from camera calibration.

V. EXPERIMENTAL RESULTS

We evaluated the performance of our approach on several different objects, including scissors, pliers, a toy truck, a Barrett robot hand, and a drawer. For our experiments, we used a PUMA 500 robotic arm and a Logitech Quickcam Pro 5000. Images of the object were gathered at multiple positions, then the configuration of the object was changed, and a new set of images was collected. The sets of images were fed to our procedure, which automatically produced the 3D models, registered the models, segmented the links, and estimated the axis or axes. The collecting of images can be performed either off-line by a person, or on-line by the robot with a scripted path.

We first demonstrate the proposed approach on one-axis revolute objects lying on a table, with the revolute axis perpendicular to the table top, as shown in Figure 7. Similar to the work of Katz and Brock [8], our approach is able to accurately estimate the axes of objects such as pliers or scissors. Unlike [8], however, our approach makes no assumption about the objects lying on the table, and it does not have prior knowledge about the perpendicularity of the rotation axis and the table top. Rather, the axis is automatically estimated in 3D, thereby also estimating the rotation plane for future manipulation of the object from arbitrary positions. In the case of the pliers, the angle between the axis and the table (which was estimated by fitting a plane to 58 points on the table) differed from 90° by just 7.2°.

The next experiment involved a Barrett robot hand, which is a three-fingered gripper. The middle finger ($F3$) is fixed with respect to the central base, while the other two fingers ($F1$ and $F2$) rotate about the base symmetrically, i.e. $\theta_1 = \theta_2$, as illustrated in Figure 8. Sets of images were captured of the hand in the two configurations. Figure 9 shows the two configurations, along with the 3D models reconstructed by our system and the axis of rotation which was correctly estimated. By fitting a plane to 20 points on the table, the angle of the axis with respect to the table was measured to be 89.5°, which is just 0.5° from 90°.

Another articulated object with two links and a single revolute joint is the toy dump truck shown in Figure 10. During the interaction, 112 images were captured of the first configuration, and 132 images were captured of the second configuration. The 3D reconstructions and estimated rotation axis of the bed are shown in the figure. The angle of the axis with respect to the table (obtained by fitting a plane to 20 points) was measured to be 1.4°.

Our approach also works with articulated objects with prismatic joints. In Figure 11, the first row shows a drawer of a cabinet in two configurations. Since drawers of the cabinet are plastic with transparency, we attached a small textured piece of cardboard to the face of the first drawer in order to reconstruct the 3D models. The second row of Figure 11 shows the 3D models corresponding to the first row with the

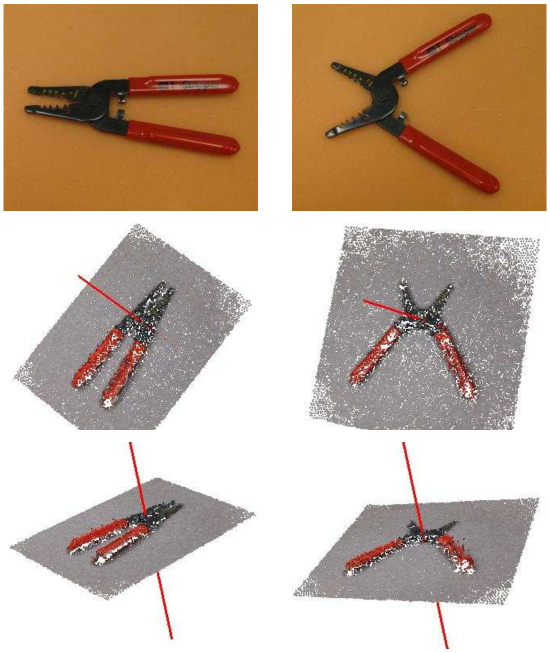

Fig. 7. Revolute axis estimation of pliers in 3D. Shown are images of the pliers (out of 65 and 81, respectively) in two configurations (top row), along with 3D reconstructions from different views with the overlaid red line indicating the estimated axis (bottom two rows). Results for scissors are similar.

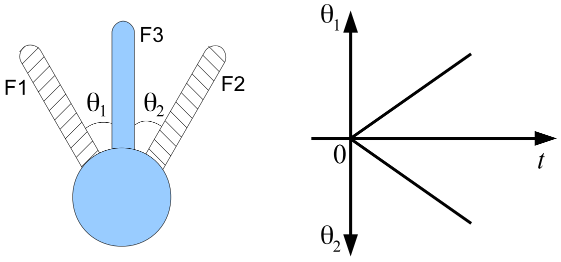

Fig. 8. The model of a Barrett robot hand with three fingers. The finger in the middle ($F3$) is fixed with respect to the central base, while the other two fingers ($F1$ and $F2$) move relative to the base in a symmetric manner ($\theta_1 = \theta_2$).

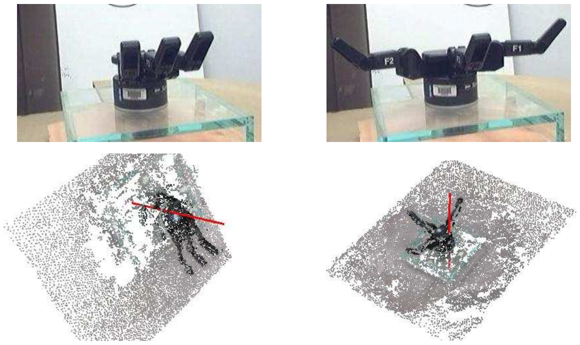

Fig. 9. Two images of the Barrett hand (out of 87 and 87, respectively) in two different configurations (top), and the 3D reconstructions with the estimated axis overlaid (red line, bottom).

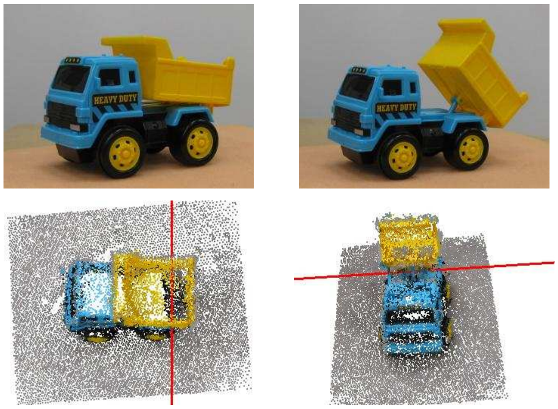

Fig. 10. Axis estimation of the toy dump truck. TOP: images (out of 112 and 132, respectively) of the two configurations of the truck. BOTTOM: 3D reconstructions with the estimated axis (red line).

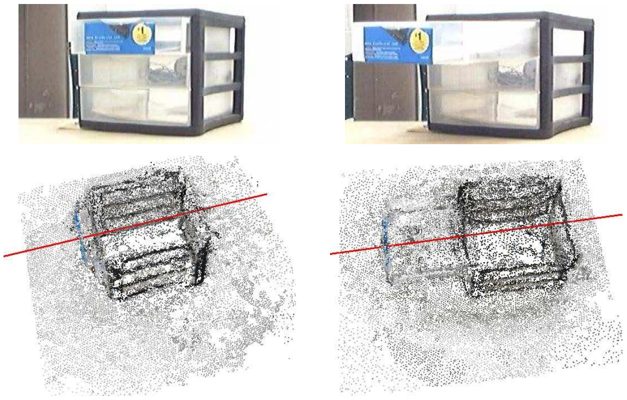

Fig. 11. Axis estimation of the drawer of the cabinet. The first row shows two images (out of 115 and 83, respectively) of the two configurations, and the second row shows 3D reconstructions with the estimated axis (red line).

estimated axis indicated by a red line. We can see that the frame and first drawer of the cabinet were well reconstructed, and that the prismatic axis was found correctly. The angle of the axis with respect to a horizontal bar on the side of the cabinet was found to be 3.4°.

We have also experimented with objects having multiple joints. Figure 12 shows an example of a scraper truck with two revolute joints, showing that the estimated axes are accurate. The angle between the estimated axes was 7.6°, but some of this deviation is due not to error from the algorithm but rather to the pliability of the plastic toy resulting from imprecise manufacturing. Note that the difficulty of automatic segmentation increases significantly as the number of joints increases, particularly with untextured objects. The rest of the algorithm, however, is largely unaffected by the number of joints.

VI. CONCLUSION

In this paper, we have proposed an approach to extract the 3D surface and kinematic structure of articulated objects. Multiple pictures are taken of the object in two different configurations, and 3D models are reconstructed using structure-

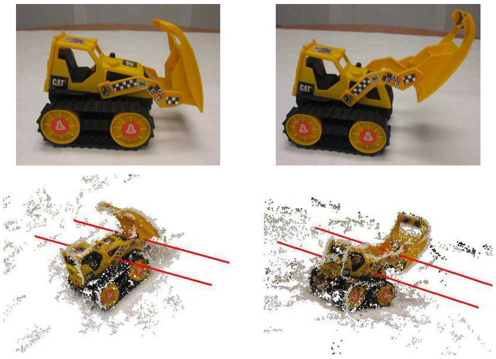

Fig. 12. Occlusion-aware reconstruction of an articulated object with multiple axes. For the two configurations, 111 and 134 images, respectively, were captured.

from-motion techniques based on images captured by a single camera. From these models, the rigid links of the object are segmented and aligned, allowing the joint axes to be estimated. The system supports both revolute and prismatic joints. The learned kinematic structure can then be used to perform purposeful manipulation. The proposed approach does not require prior knowledge of the object nor does it make any assumption regarding planarity. Experiments have shown its effectiveness on a range of environmental conditions and various types of objects.

VII. ACKNOWLEDGMENTS

This work was supported by NSF grant IIS-1017007.

REFERENCES

[1] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Proceedings of the International Conference on Computer Vision*, pages 1071–1076, 1995.

[2] D. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-D rigid body transformations: A comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290, Mar. 1997.

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

[4] D. A. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion: Part 1, Tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, 1(2/3), 2006.

[5] Y. Furukawa and J. Ponce, 2007. PMVS, http://www.cs.washington.edu/homes/furukawa/research/pmvs.

[6] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[7] P. Guan, A. Weiss, A. O. Bălan, and M. J. Black. Estimating human shape and pose from a single image. In *Proceedings of the International Conference on Computer Vision*, 2009.

[8] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *Proceedings of the International Conference on Robotics and Automation*, pages 272–277, May 2008.

[9] D. Katz, A. Orthey, and O. Brock. Interactive perception of articulated objects. In *12th International Symposium on Experimental Robotics (ISER)*, Dec. 2010.

[10] A. G. Kirk, J. F. O'Brien, and D. A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] F. Moreno-Noguer, V. Lepetit, and P. Fua. Accurate non-iterative O(n) solution to the PnP problem. In *Proceedings of the International Conference on Computer Vision*, Oct. 2007.

[13] J. F. O'Brien, J. Robert E. Bodenheimer, G. J. Brostow, and J. K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proceedings of Graphics Interface*, 2000.

[14] D. A. Ross, D. Tarlow, and R. S. Zemel. Learning articulated structure and motion. *International Journal of Computer Vision*, 88(2):214–237, Mar. 2010.

[15] D. Sinclair, L. Paletta, and A. Pinz. Euclidean structure recovery through articulated motion. In *In Proc. 10th Scandinavian Conference on Image Analysis*, 1997.

[16] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

[17] N. Snavely, 2006. Bundler: SfM for unordered image collections, http://phototour.cs.washington.edu/bundler.

[18] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3D. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 25(3):835–846, 2006.

[19] J. Sturm, A. Jain, C. Stachniss, C. Kemp, and W. Burgard. Operating articulated objects based on experience. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[20] 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 International Conference on Robotics and Automation*, 2010.

[21] J. Sturm, C. Stachniss, V. Pradeep, C. Plagemann, K. Konolige, and W. Burgard. Learning kinematic models for articulated objects. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2009.

[22] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[23] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1110–1115, 2005.

[24] R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989.

[25] B. Willimon, S. Birchfield, and I. Walker. Rigid and non-rigid classification using interactive perception. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1728–1733, 2010.

[26] B. Willimon, S. Birchfield, and I. Walker. Classification of clothing using interactive perception. In *International Conf. on Robotics and Automation (ICRA)*, pages 1862–1868, 2011.

[27] B. Willimon, S. Birchfield, and I. Walker. Model for unfolding laundry using interactive perception. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[28] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion, and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):865–877, May 2008.

[29] X. Zhang, Y. Liu, and T. S. Huang. Motion analysis of articulated objects from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):625–636, Apr. 2006.