Learning Articulated Motions
From Visual Demonstration

Sudeep Pillai, Matthew R. Walter and Seth Teller
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
Email: {spillai, mwalter, teller}@csail.mit.edu

Abstract—**Many functional elements of human homes and workplaces consist of rigid components which are connected through one or more sliding or rotating linkages. Examples include doors and drawers of cabinets and appliances; laptops; and swivel office chairs. A robotic mobile manipulator would benefit from the ability to acquire kinematic models of such objects from observation. This paper describes a method by which a robot can acquire an object model by capturing depth imagery of the object as a human moves it through its range of motion. We envision that in future, a machine newly introduced to an environment could be shown by its human user the articulated objects particular to that environment, inferring from these "visual demonstrations" enough information to actuate each object independently of the user.**

Our method employs sparse (markerless) feature tracking, motion segmentation, component pose estimation, and articulation learning; it does not require prior object models. Using the method, a robot can observe an object being exercised, infer a kinematic model incorporating rigid, prismatic and revolute joints, then use the model to predict the object's motion from a novel vantage point. We evaluate the method's performance, and compare it to that of a previously published technique, for a variety of household objects.

I. INTRODUCTION

A long-standing challenge in robotics is to endow robots with the ability to interact effectively with the diversity of objects common in human-made environments. Existing approaches to manipulation often assume that objects are simple and drawn from a small set. The models are then either predefined or learned from training, for example requiring fiducial markers on object parts, or prior assumptions about object structure. Such requirements may not scale well as the number and variety of objects increases. This paper describes a method with which robots can learn kinematic models for articulated objects in situ, simply by observing a user manipulate the object. Our method learns open kinematic chains that involve rigid linkages, and prismatic and revolute motions, between parts.

There are three primary contributions of our approach that make it effective for articulation learning. First, we propose a feature tracking algorithm designed to perceive articulated motions in unstructured environments, avoiding the need to embed fiducial markers in the scene. Second, we describe a motion segmentation algorithm that uses kernel-based clustering to group feature trajectories arising from each object part. A subsequent optimization step recovers the 6-DOF pose
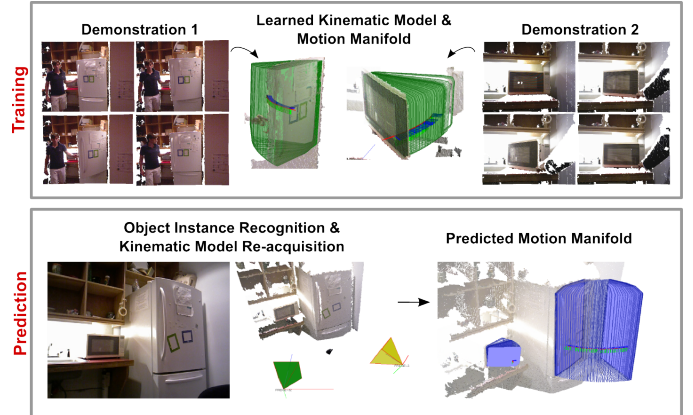

Fig. 1: The proposed framework reliably learns the underlying kinematic model of multiple articulated objects from user-provided visual demonstrations, and subsequently predicts their motions at future encounters.

of each object part. Third, the method enables use of the learned articulation model to predict the object's motion when it is observed from a novel vantage point. Figure 1 illustrates a scenario where our method learns kinematic models for a refrigerator and microwave from separate user-provided demonstrations, then predicts the motion of each object in a subsequent encounter. We present experimental results that demonstrate the use of our method to learn kinematic models for a variety of everyday objects, and compare our method's performance to that of the current state of the art.

II. RELATED WORK

Providing robots with the ability to learn models of articulated objects requires a range of perceptual skills such as object tracking, motion segmentation, pose estimation, and model learning. It is desirable for robots to learn these models from demonstrations provided by ordinary users. This necessitates the ability to deal with unstructured environments and estimate object motion without requiring tracking markers. Traditional tracking algorithms such as KLT [2], or those based on SIFT [15] depend on sufficient object texture and may be susceptible to drift when employed over an object's full range of motion. Alternatives such as large-displacement optical flow [4] or particle video methods [19] tend to be more accurate but require substantially more computation.
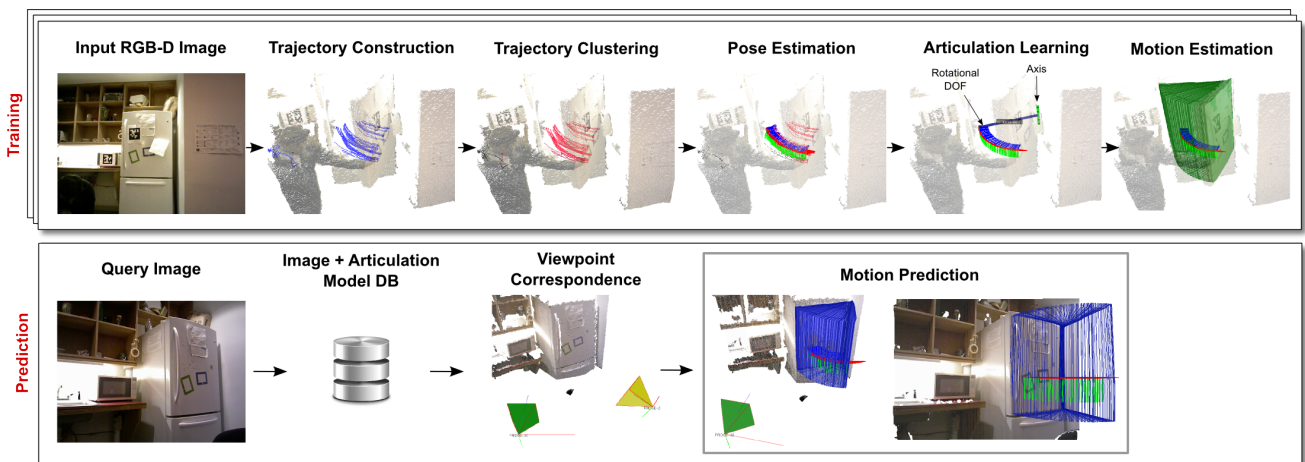
Fig. 2: Articulation learning architecture.

Articulated motion understanding generally requires a combination of motion tracking and segmentation. Existing motion segmentation algorithms use feature based trackers to construct spatio-temporal trajectories from sensor data, and cluster these trajectories based on rigid-body motion constraints. Recent work by Brox and Malik [3] in segmenting feature trajectories has shown promise in analyzing and labeling motion profiles of objects in video sequences in an unsupervised manner. Recent work by Elhamifar and Vidal [5] has proven effective at labeling object points based purely on motion visible in a sequence of standard camera images. Our framework employs similar techniques, and introduce a segmentation approach for features extracted from RGB-D data.

Researchers have studied the problem of learning models from visual demonstration. Yan and Pollefeys [24] and Huang et al. [10] employ structure from motion techniques to segment the articulated parts of an object, then estimate the prismatic and rotational degrees of freedom between these parts. These methods are sensitive to outliers in the feature matching step, resulting in significant errors in pose and model estimates. Closely related to our work, Katz et al. [13] consider the problem of extracting segmentation and kinematic models from interactive manipulation of an articulated object. They take a deterministic approach, first assuming that each object linkage is prismatic and proceed to fit a rotational degree-of-freedom only if the residual is above a specified threshold. Katz et al. learn from observations made in clean, clutter-free environments and primarily consider objects in close proximity to the RGB-D sensor. Recently, Katz et al. [14] propose an improved learning method that has equally good performance with reduced algorithmic complexity. However, the method does not explicitly reason over the complexity of the inferred kinematic models, and tends to over-fit to observed motion. In contrast, our algorithm targets in situ learning in unstructured environments with probabilistic techniques that provide robustness to noise. Our method adopts the work of Sturm et al. [22], which used a probabilistic approach to reason over the likelihood of the observations while simultaneously penalizing complexity in the kinematic model. Their work differs from ours in two main respects: they required that fiducial markers be placed on each object part in order to provide nearly noise-free observations; and they assume that the number of unique object parts is known a priori.

III. ARTICULATION LEARNING FROM VISUAL DEMONSTRATION

This section introduces the algorithmic components of our method. Figure 2 illustrates the steps involved.

Our approach consists of a training phase and a prediction phase. The training phase proceeds as follows: (i) Given RGB-D data, a feature tracker constructs long-range feature trajectories in 3-D. (ii) Using a relative motion similarity metric, clusters of rigidly moving feature trajectories are identified. (iii) The 6-DOF motion of each cluster is then estimated using 3-D pose optimization. (iv) Given a pose estimate for each identified cluster, the most likely kinematic structure and model parameters for the articulated object are determined. Figure 3 illustrates the steps involved in the training phase with inputs and outputs for each component.
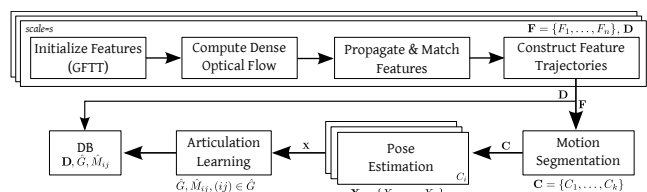

Fig. 3: The training phase.

Once the kinematic model of an articulated object is learned, our system can predict the motion trajectory of the object during future encounters. In the prediction phase: (i) Given RGB-D data, the description of the objects in the scene, $\mathbf{D}_{query}$, is extracted using SURF [1] descriptors. (ii) Given a set of descriptors $\mathbf{D}_{query}$, the best-matching object and its kinematic model, $\hat{G}, \hat{M}_{ij}, (ij) \in \hat{G}$ are retrieved; and (iii) From these correspondences and the kinematic model

parameters of the matching object, the object's articulated motion is predicted. Figure 4 illustrates the steps involved in the prediction phase.
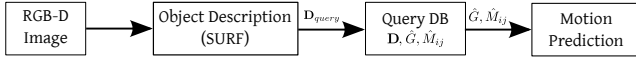

Fig. 4: The prediction phase.

A. Spatio-Temporal Feature Tracking

The first step in articulation learning from visual demonstration involves visually observing and tracking features on the object while it is being manipulated. We focus on unstructured environments without fiducial markers. Our algorithm combines interest-point detectors and feature descriptors with traditional optical flow methods to construct long-range feature trajectories. We employ Good Features To Track (GFTT) [20] to initialize up to 1500 salient features with a quality level of 0.04 or greater, across multiple image scales. Once the features are detected, we populate a mask image that captures regions where interest points are detected at each pyramid scale. We use techniques from previous work on dense optical flow [7] to predict each feature at the next timestep. Our implementation also employs median filtering as suggested by Wang et al. [23] to reduce false positives.

We bootstrap the detection and tracking steps with a feature description step that extracts and learns the description of the feature trajectory. At each image scale, we compute the SURF descriptor [1] over features that were predicted from the previous step, denoted as $\hat{f}^t$, and compare them with the description of the detected features at time t, denoted as f^t. Subsequently, detected features f^t that are sufficiently close to predicted features $\hat{f}^t$ and that successfully meet a desired match score are added to the feature trajectory, while the rest are pruned. To combat drift, we use the detection mask as a guide to reinforce feature predictions with feature detections. Additionally, we incorporate flow failure detection techniques [12] to reduce drift in feature trajectories.

Like other feature-based methods [14] our method requires visual texture. In typical video sequences, some features are continuously tracked, while other features are lost due to occlusion or lack of image saliency. To provide rich trajectory information, we continuously add features to the scene as needed. We maintain a constant number of feature trajectories tracked, by adding newly detected features in regions that are not yet occupied. From RGB-D depth information, image-space feature trajectories can be easily extended to 3-D. As a result, each feature key-point is represented by its normalized image coordinates (u, v), position $\vec{p} \in \mathbb{R}^3$ and surface normal $\vec{n}$, represented as $(\vec{p}, \vec{n}) \in \mathbb{R}^3 \times SO(2)$. We denote $\mathbf{F} = \{F_1, \ldots, F_n\}$ as the resulting set of feature trajectories constructed, where $F_i = \{(\vec{p_1}, \vec{n_1}), \ldots, (\vec{p_t}, \vec{n_t})\}$. To combat noise inherent in our consumer-grade RGB-D sensor, we post-process the point cloud with a fast bilateral filter [18] with parameters $\sigma_s = 20$ px, $\sigma_r = 4$ cm.

B. Motion Segmentation

To identify the kinematic relationships among parts in an articulated object, we first distinguish the trajectory taken by each part. In particular, we analyze the motions of the object parts with respect to each other over time, and infer whether or not pairs of object parts are rigidly attached. To reason over candidate segmentations, we formulate a clustering problem to identify the different motion subspaces in which the object parts lie. After clustering, similar labels imply rigid attachment, while dissimilar labels indicate non-rigid relative motion between parts.

If two features in $\mathbb{R}^3 \times SO(2)$ belong to the same rigid part, the relative displacement and angle between the features will be consistent over the common span of their trajectories. The distribution over the relative change in displacement vectors and angle subtended is modeled as a zero-mean Gaussian, $\mathcal{N}(\mu, \Sigma) = (0, \Sigma)$, where Σ is the expected noise covariance for rigidly-connected feature pairs. The similarity of two feature trajectories can then be defined as:

$$L(i,j) = \frac{1}{T} \sum_{t \in t_i \cap t_j} \exp\left\{ -\gamma \left(d(x_i^t, x_j^t) - \mu_{d_{ij}} \right)^2 \right\} \quad (1)$$

where t_i and t_j are the observed time instances of the feature trajectories i, and j respectively, $T = |t_i \cap t_j|$, and γ is a parameter characterizing the relative motion of the two trajectories. For a pair of 3-D key-point features $\vec{p_i}$, and $\vec{p_j}$, we estimate the mean relative displacement between a pair of points moving rigidly together as:

$$\mu_{d_{ij}} = \frac{1}{T} \sum_{t \in t_i \cap t_j} d(\vec{p_i}^t, \vec{p_j}^t) \quad (2)$$

where $d(\vec{p_i}, \vec{p_j}) = \|\vec{p_i} - \vec{p_j}\|$. For 3-D key-points, we use $\gamma = \frac{1}{2 \text{ cm}}$ in Eqn. 1. Figure 5 illustrates an example of rigid and non-rigid motions of feature trajectory pairs, and their corresponding distribution of relative displacements.

For a pair of surface normals $\vec{n_i}$ and $\vec{n_j}$, we define the mean distance as

$$\mu_{d_{ij}} = \frac{1}{T} \sum_{t \in t_i \cap t_j} d(\vec{n_i}^t, \vec{n_j}^t), \quad (3)$$

where $d(\vec{n_i}, \vec{n_j}) = 1 - \vec{n_i} \cdot \vec{n_j}$. In this case, we use $\gamma = \frac{1}{\cos(15^\circ)}$ in Eqn. 1.

Since the bandwidth parameter γ for a pair of feature trajectories can be intuitively predicted from the expected variance in relative motions of trajectories, we employ DBSCAN [6], a density-based clustering algorithm, to find rigidly associated feature trajectories. The resulting cluster assignments are denoted as $\mathbf{C} = \{C_1, \ldots, C_k\}$, where cluster C_i consists of a set of rigidly-moving feature trajectories.

C. Multi-Rigid-Body Pose Optimization

Given the cluster label assignment for each feature trajectory, we subsequently determine the 6-DOF motion of each cluster. We define Z_i^t as the set of features belonging to cluster C_i at time t. Additionally, we define $\mathbf{X} = X_1, \ldots, X_k$ as the
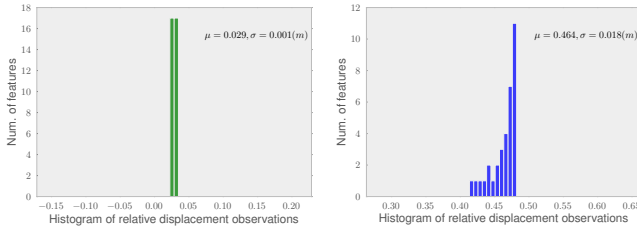
Fig. 5: Histogram of observed distances between a pair of trajectories accumulated over one demonstration. (Left) The distribution of observed distances is centered at $\mu = 0.029$ m with $\sigma = 0.001$ m, indicating rigid-body motion. (Right) Larger variation in observed distances, with $\sigma = 0.018$ m, indicates non-rigid motion.

set of $SE(3)$ poses estimated for each of k clusters considered, and $x_i^t \in X_i$ as the $SE(3)$ pose estimated for the ith cluster at time t.

For each cluster C_i, we consider the synchronized sensor observations of position and surface normals for each of its trajectories, and use the arbitrary pose x_i^0 as the reference frame for the remaining pose estimates of the ith cluster. Subsequently, we compute the relative transformation $\Delta_i^{t-1,t}$ between successive time steps $t-1$ and t for the ith cluster using the known correspondences between Z_i^{t-1} and Z_i^t. Since this step can lead to drift, we add an additional sparse set of relative pose constraints every 10 frames, denoted as $\Delta_i^{t-10,t}$. Our implementation employs a correspondence rejection step that eliminates outliers falling outside the inlier distance threshold of 1 cm, as in RANSAC [8], making the pose estimation routine more robust to sensor noise.

We augment the estimation step with an optimization phase to provide smooth and continuous pose estimates for each cluster by incorporating a motion model. We use the 3-D pose optimizer iSAM [11] to incorporate the relative pose constraints within a factor graph, with node factors derived directly from the pose estimates. A constant-velocity edge factor term is also added to provide continuity in the articulated motion.

D. Articulation Learning

Once the 6-DOF pose estimates of the individual object parts are computed, the kinematic model of the full articulated object is determined using tools developed in Sturm et al. [22]. Given multiple 6-DOF pose observations of object parts, the problem is to estimate the most likely kinematic configuration for the articulated object. Formally, given the observed poses $\mathcal{D}_z$, we estimate the kinematic graph configuration $\hat{G}$ that maximizes the posterior probability

$$\hat{G} = \arg\max_G p(G \mid \mathcal{D}_z) \tag{4}$$

We employ notation similar to that of Sturm et al. [22] to denote the relative transformation between two object parts i and j as $\Delta_{ij} = x_i \ominus x_j$, using standard motion composition operator notation [21]. The kinematic model between part i and j is then defined as M_{ij}, with its associated parameter vector $\theta_{ij} \in \mathbb{R}^{p_{ij}}$, where p_{ij} are the number of parameters

associated with the description of the link. We construct a graph $G = (V_G, E_G)$ consisting of a set of vertices $V_G = 1, \ldots, k$ that denote the object parts involved in the articulated object, and a set of undirected edges $E_G \subset V_G \times V_G$ describing the kinematic linkage between two object parts.

As in Sturm et al. [22], we simplify the problem to recognize only kinematic trees of high posterior probability, in order to reformulate the problem as equation 8 below:

$$\hat{G} = \arg\max_G p(G \mid \mathcal{D}_z) \tag{5}$$

$$= \arg\max_G p(\{(\mathcal{M}_{ij}, \theta_{ij}) \mid (ij) \in E_G\} \mid \mathcal{D}_z) \tag{6}$$

$$= \arg\max_G \prod_{(ij) \in E_G} p(\mathcal{M}_{ij}, \theta_{ij} \mid \mathcal{D}_z) \tag{7}$$

$$= \arg\max_{E_G} \sum_{(ij) \in E_G} \log \ p(\hat{\mathcal{M}}_{ij}, \hat{\theta}_{ij} \mid \mathcal{D}_z) \tag{8}$$

where $\mathcal{D}_z = (\Delta_{ij}^1, \ldots, \Delta_{ij}^t) \ \forall \ (ij) \in E_G$ is the sequence of observed relative transformations between parts i and j.

Since we are particularly interested in household objects, we focus on kinematic models involving rigid, prismatic, and revolute linkages. We then estimate the parameters $\theta \in \mathbb{R}^p$ that maximize the data likelihood of the object pose observations given the kinematic model:

$$\hat{\theta} = \arg\max_\theta p(\mathcal{D}_z \mid \mathcal{M}, \theta) \tag{9}$$

Once we fit each candidate kinematic model to the given observation sequence, we select the kinematic model that best explains the data. Specifically, we compute the posterior probability of each kinematic model, given the data, as:

$$p(\mathcal{M} \mid \mathcal{D}_z) = \int \frac{p(\mathcal{D}_z \mid \mathcal{M}, \theta) \ p(\theta \mid \mathcal{M}) \ p(\mathcal{M})}{p(\mathcal{D}_z)} d\theta \tag{10}$$

Due to the evaluation complexity of this posterior term, the BIC score is computed instead as the approximation:

$$BIC(\mathcal{M}) = -2\log \ p(\mathcal{D}_z \mid \mathcal{M}, \hat{\theta}) + p \log n, \tag{11}$$

where p is the number of parameters involved in the kinematic model, n is the number of observations in the data set, and $\hat{\theta}$ is the maximum likelihood parameter vector. This implies that the model that best explains the observations would correspond to that with the least BIC score.

The kinematic structure selection problem is subsequently reduced to computing the minimum spanning tree of the graph with edges defined by $cost_{ij} = -\log \ p(\mathcal{M}_{ij}, \theta_{ij} \mid \mathcal{D}_{z_{ij}})$. The resulting minimum spanning kinematic tree weighted by BIC scores is the most likely kinematic model for the articulated object given the pose observations. For a more detailed description, we refer the reader to Sturm et al. [22]. Figure 6 shows a few examples of kinematic structures extracted given pose estimates as described in the previous section. Our limitation of linkage types to rigid, prismatic, and rotational does exclude various household objects such as lamps, garage doors, toys etc. with more complex kinematics.
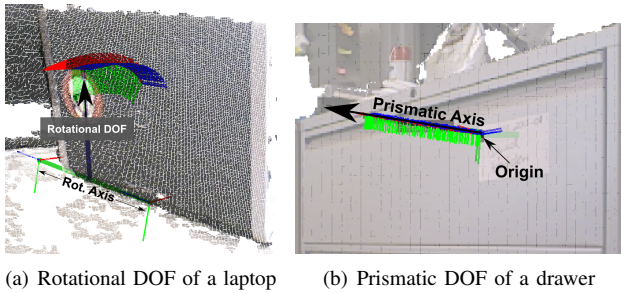
(a) Rotational DOF of a laptop   (b) Prismatic DOF of a drawer

Fig. 6: Examples of correctly estimated kinematic structure from 6-DOF pose estimates of feature trajectories.

E. Learning to Predict Articulated Motion

Our daily environment is filled with articulated objects with which we repeatedly interact. A robot in our environment can identify instances of articulated objects that it has observed in the past, then use a learned model to predict the motion of an object when it is used.

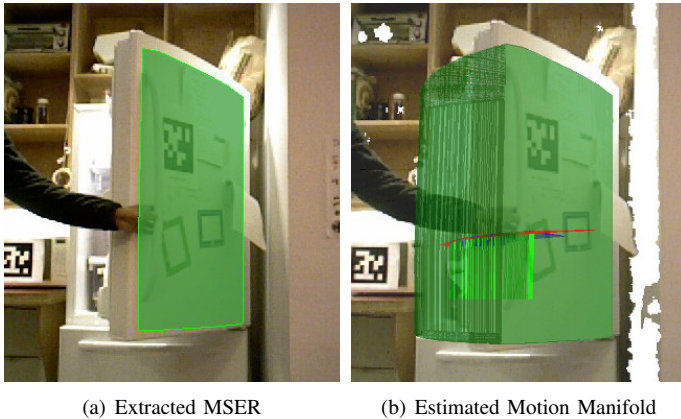

(a) Extracted MSER   (b) Estimated Motion Manifold

Fig. 7: The motion manifold of an articulated object, extracted via MSERs.

Once the kinematic model of an articulated object is learned, the kinematic structure $\hat{G}$ and its model parameters $\hat{\mathcal{M}}_{ij}, (ij) \in \hat{G}$ are stored in a database, along with its appearance model. The feature descriptors extracted (described in Section III-A) for each cluster C_i of the articulated object are also retained for object recognition in future encounters. Demonstrations involving the same instance of the articulated object are represented in a single arbitrarily selected reference frame, and kept consistent across encounters by registering newer demonstrations into the initial object frame. Each of these attributes is stored in the bag-of-words driven database [9] for convenient querying in the future. Thus, on encountering the same object instance in the future, the robot can match the descriptors extracted from the current scene with those extracted from object instances it learned in the past. It then recovers the original demonstration reference frame along with the relevant kinematic structure of the articulated object for prediction purposes. We identify the

surface of the manipulated object by extracting Maximally Stable Extremal Regions (MSER) [16] (Figure 7) for each object part undergoing motion. We use this surface to visualize the motion manifold of the articulated object.

IV. EXPERIMENTS AND ANALYSIS

Our experimental setup consists of a single sensor providing RGB-D depth imagery. Each visual demonstration involved a human manipulating an articulated object and its parts at a normal pace, while avoiding obscuration of the object from the robot's perspective. Demonstrations were performed for multiple robot viewpoints, to capture variability in depth imagery. We performed 43 demonstration sessions by manipulating a variety of household objects: refrigerators, doors, drawers, laptops, chair etc. Each demonstration was recorded for about 30-60 seconds. April tags [17] were used to recover ground truth estimates of each articulated object's motion, which we adopted as a baseline for evaluation. In order to avoid any influence on our method of observations arising from fiducial markers, the RGB-D input was pre-processed to mask out regions containing the tags.

We then compared the pose estimation, model selection and estimation performance of our method to that of an alternative state-of-the-art method (re-implemented by us based on [14]), and to traditional methods using fiducial markers. We incorporated several improvements [12], [18] to Katz's algorithm, as previously described in Section III-A, to enable fair comparison with our proposed method.

A. Qualitative and Overall Performance

Figure 8 shows the method in operation for household objects including a laptop, a microwave, a refrigerator and a drawer. Tables I and II compare the performance of our method in estimating the kinematic model parameters for several articulated objects observed from a variety of viewpoints. Our method recovered a correct model for more objects, and for almost every object tested recovered model parameters more accurately, than Katz's method.

B. Pose Estimation Accuracy

For each visual demonstration, we compared the segmentation and $SE(3)$ pose of each object part estimated by our method with those produced by Katz. We also obtained pose estimates for each object part by tracking attached fiducial markers. Synchronization across pose observations was ensured by evaluating only poses in the set intersection of the timestamps of each pose sequence. For each overlapping time step, we compared the relative pose of the estimated object segment obtained from both algorithms with that obtained via fiducial markers (Figure 9). For consistency in evaluation, the $SE(3)$ poses of individual object parts were initialized identically for both algorithms.

Figure 10 compares the absolute $SE(3)$ poses estimated by the three methods described above, given observations of a chair being moved on the ground plane. Figure 10(a) illustrates a scenario in which both algorithms, ours and Katz's, perform
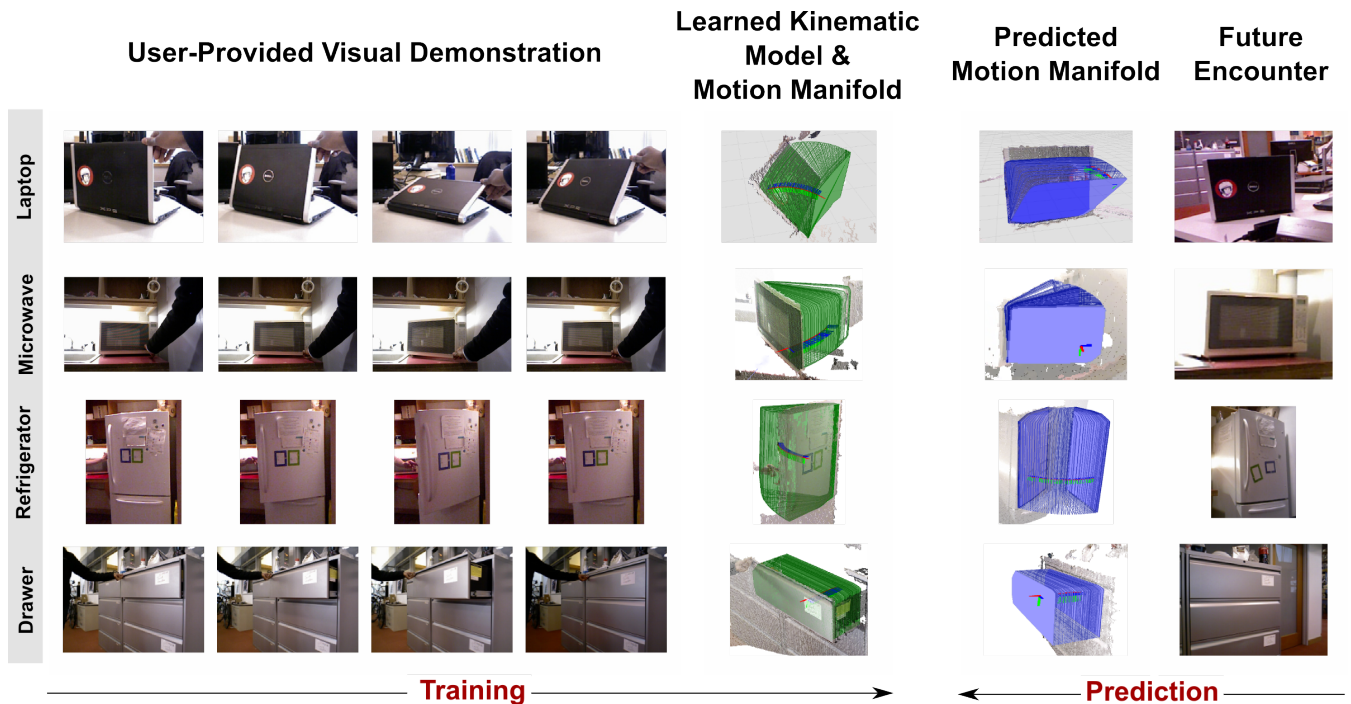
Fig. 8: Articulation learning and motion prediction for various objects.

reliably. Katz's method is within 2.0 cm and $2.6\,°$, on average, of the ground truth pose produced with fiducial markers. Our method achieves comparable average accuracy of 1.7 cm and $2.1\,°$. Using data from another demonstration, Katz's method failed to track the object motion robustly, resulting in drift and incorrect motion estimates (Figure 10(b)). Such failures can be attributed to: *(i)* the KLT tracker that is known to cause drift during feature tracking; *(ii)* SVD least squares minimization in the relative pose estimation stage, without appropriate outlier rejection.

For a variety of articulated objects (Table I), our method achieves average accuracies of 2.4 cm and $4.7\,°$ with respect to ground truth estimated from noisy Kinect RGB-D data. In comparison, Katz's method [14] achieved average accuracies of 3.7 cm and $10.1\,°$ for the same objects. Our method achieved an average error of less than 10 cm and $25\,°$ in 37 of 43 demonstrations, vs. 23 of 43 for Katz.
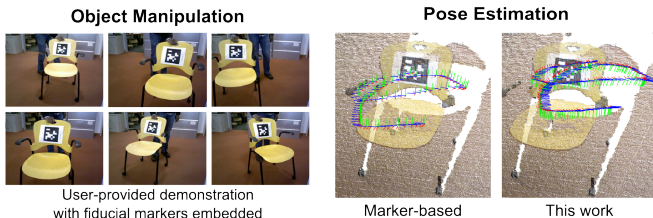

Fig. 9: Pose estimation accuracy of our method, compared to that achieved using fiducial markers.

C. Model Estimation Accuracy

Once the $SE(3)$ poses of the object parts are estimated, we compare the kinematic structure and model parameters of the articulated object estimated by our method with those produced by Katz. As in our other experiments, we use the kinematic structure and model parameters identified from fiducial marker-based solutions as a baseline. Table II summarizes the model estimation and parameter estimation performance achieved with our method and Katz's. The model fit error is defined as the average spatial and orientation error between the $SE(3)$ observations and the estimated articulation manifold (i.e. prismatic or rotational manifold). For the dataset of articulated objects evaluated (Table II), our method achieved an average model fit error of 1.7 cm spatially, and $5.0\,°$ in orientation, an improvement over Katz's method (average model fit errors of 2.0 cm and $5.8\,°$ respectively). Of 43 demonstrations evaluated, our method determined the correct kinematic structure and accurate parameters in 30 cases, whereas Katz did so in only 15 cases.

We also compared the model parameters estimated by our method and Katz's method with ground truth from markers, by transforming poses estimated by both methods into the fiducial marker's reference frame based on the initial configuration of the articulated object. This allows us to directly compare model parameters estimated through our proposed framework, the current state-of-the-art and marker-based solutions. For multi-DOF objects, the model parameter error averaged across each corresponding object part is reported. In each demonstration, the model parameters estimated via our method are closer to the marker-based solution than those obtained by Katz.
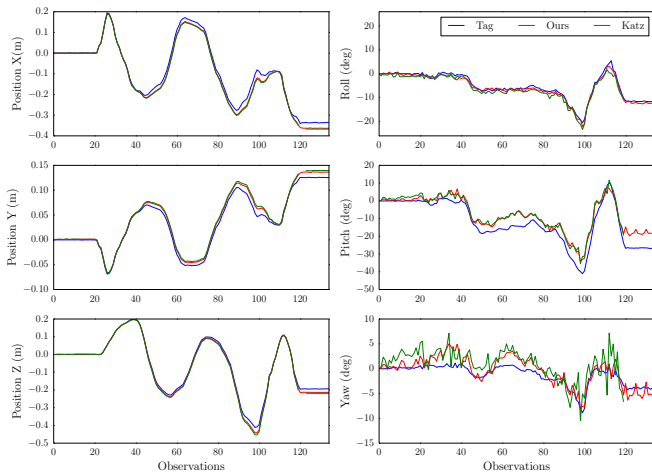
(a) Accurate estimation by current state-of-the-art and our framework
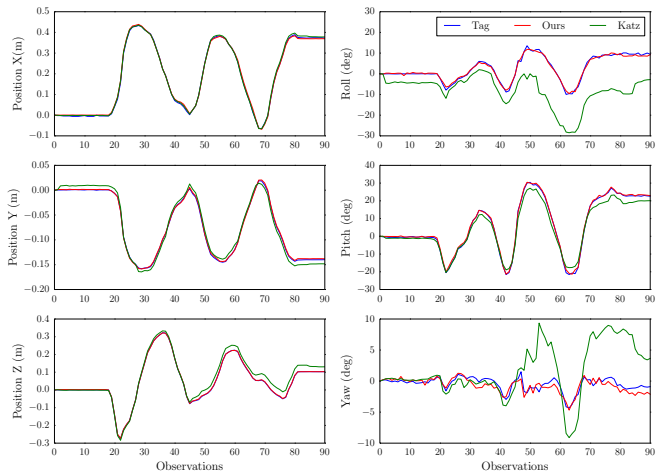
(b) Failed estimation by current state-of-the-art

Fig. 10: Comparison of $SE(3)$ pose for a chair estimated via fiducial markers (Tag), current state-of-the-art (Katz) and our framework (Ours). (a) The figures show the strong performance of our framework, as compared to marker-based solutions and current state-of-the-art algorithms, to robustly track and estimate the $SE(3)$ pose of a chair being manipulated on multiple occasions. (b) Current state-of-the-art, however, fails to robustly estimate the $SE(3)$ pose on certain trials.

Dataset	DOF	Katz et al.			Ours		
		Average Error		Success Rate	Average Error		Success Rate
		Pos.	Orient.		Pos.	Orient.	
Door	1	6.0 cm	6.8°	6/7	**5.0** cm	**5.5°**	**7/7**
Drawer	1	6.1 cm	18.0°	3/7	**3.7** cm	**3.0°**	**6/7**
Fridge	1	2.2 cm	8.1°	4/8	**1.0** cm	**2.9°**	**6/8**
Laptop	1	0.4 cm	2.3°	2/5	**0.3** cm	6.4°	**4/5**
Microwave	1	4.3 cm	14.2°	2/4	**1.9** cm	6.9°	**4/4**
Printer	1	0.7 cm	2.5°	**1/2**	**0.5** cm	**2.3°**	**2/2**
Screen	1	**2.6** cm	24.9°	1/2	3.4 cm	**3.5°**	1/2
Chair	2	3.6 cm	13.2°	2/3	**2.3** cm	**4.5°**	**3/3**
Monitor	2	**0.8** cm	7.2°	1/2	1.8 cm	**2.3°**	**2/2**
Bicycle	3	1.7 cm	10.4°	1/3	**1.1** cm	**9.8°**	**2/3**
Overall		3.7 cm	10.1°	23/43	**2.4** cm	**4.7°**	**37/43**

TABLE I: Comparison of $SE(3)$ pose estimates between our framework and current state-of-the-art (Katz) with marker-based pose estimates considered as ground truth.

Dataset	DOF	Katz et al.				Ours			
		Model Fit Error		Param. Est. Error	Success Rate	Model Fit Error		Param. Est. Error	Success Rate
		Pos.	Orient.			Pos.	Orient.		
Door	1	1.9 cm	6.7°	1.9°	4/7	**0.4** cm	**4.7°**	**1.8°**	**5/7**
Drawer	1	2.0 cm	7.3°	2.5°	2/7	**1.7** cm	**3.1°**	**2.0°**	**6/7**
Fridge	1	0.5 cm	6.5°	5.6°	4/8	**0.4** cm	**5.8°**	**3.5°**	**5/8**
Laptop	1	-	-	-	0/5	**0.2** cm	**6.4°**	**6.1°**	**4/5**
Microwave	1	7.0 cm	**1.2°**	**0.2°**	2/4	**6.5** cm	4.1°	0.3°	**3/4**
Printer	1	**0.9** cm	0.8°	1.5°	1/2	2.1 cm	**0.2°**	**1.4°**	1/2
Screen	1	-	-	-	0/2	**0.9** cm	**0.7°**	**3.2°**	**1/2**
Chair	2	**0.3** cm	11.2°	9.8°	1/3	3.9 cm	**7.9°**	**4.8°**	**2/3**
Monitor	2	-	-	-	0/2	**2.9** cm	**6.4°**	**5.7°**	**1/2**
Bicycle	3	0.9 cm	**5.1°**	**4.2°**	1/3	**0.7** cm	8.5°	7.3°	**2/3**
Overall		2.0 cm	5.8°	3.4°	15/43	**1.7** cm	**5.0°**	**3.3°**	**30/43**

TABLE II: Comparison of kinematic model estimation and parameter estimation capability between our framework and current state-of-the-art (Katz) with marker-based model estimation considered as ground truth.

V. CONCLUSION

We introduced a framework that enables robots to learn kinematic models for everyday objects from RGB-D data acquired during user-provided demonstrations. We combined sparse feature tracking, motion segmentation, object pose estimation and articulation learning to learn the underlying kinematic structure of the observed object. We demonstrated the qualitative and quantitative performance of our method; it recovers the correct structure more often, and more accurately, than its predecessor in the literature, and achieves accuracy similar to that of a marker-based solution. Our framework also enables the robot to predict the motion of articulated objects it has previously learned. Even given our method's limitation to recovering open kinematic chains involving only rigid, prismatic or revolute linkages, its prediction capability may be useful in future robotic encounters requiring manipulation.

REFERENCES

[1] H. Baya, A. Essa, T. Tuytelaarsb, and L. Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[2] J.-Y. Bouguet. Pyramidal implementation of the affine Lucas-Kanade feature tracker description of the algorithm. *Intel Corporation*, 2001.

[3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 282–295, 2010.

[4] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[5] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2797, 2009.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. ACM Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[7] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. 2003.

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

[9] D. Galvez-Lopez and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *Trans. on Robotics*, 28(5):1188–1197, 2012.

[10] X. Huang, I. Walker, and S. Birchfield. Occlusion-aware reconstruction and manipulation of 3d articulated objects. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 1365–1371, 2012.

[11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *Int'l J. of Robotics Research*, 31(2):216–235, 2012.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Proc. Int'l Conf. on Pattern Recognition (ICPR)*, pages 2756–2759, 2010.

[13] D. Katz, A. Orthey, and O. Brock. Interactive perception of articulated objects. In *Proc. Int'l. Symp. on Experimental Robotics (ISER)*, 2010.

[14] D. Katz, M. Kazemi, J. Andrew Bagnell, and A. Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 5003–5010, 2013.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. of Computer Vision*, 60(2): 91–110, 2004.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[17] E. Olson. AprilTag: A robust and flexible visual fiducial system. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 3400–3407, 2011.

[18] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 568–580, 2006.

[19] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *Int'l J. of Computer Vision*, 80(1):72–91, 2008.

[20] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[21] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[22] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *J. of Artificial Intelligence Research*, 41(2):477–526, 2011.

[23] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, 2011.

[24] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Proc. European Conf. on Computer Vision (ECCV)*, pages 94–106, 2006.