Online Interactive Perception of Articulated Objects with Multi-Level Recursive Estimation Based on Task-Specific Priors

Roberto Martín Martín[1] Oliver Brock[1]

***Abstract*— To successfully manipulate in unknown environments, a robot must be able to perceive degrees of freedom of objects in its environment. Based on the resulting kinematic model and joint configurations, the robot is able to select and adapt actions, recognize their successful completion and detect failure. We present an RGB-D-based online algorithm for the interactive perception of articulated objects. The algorithm decomposes the perception problem into three interconnected levels of recursive estimation. The estimation problems at each level are much simpler than the original problem and their robustness is improved by level-specific priors that help reject noise in the measurements. These three estimators mutually inform each other to further improve the convergence properties of the three estimation solutions. We demonstrate that the resulting algorithm is robust, accurate, and versatile in real-world experiments. We also show how the perceptual skill can be used online to control the robot's behavior in real-world manipulation tasks.**

I. Introduction

Robots achieve tasks by manipulating their environment. This manipulation is the deliberate change of the configuration of objects. To perform such manipulation successfully, the robot must be able to detect and track degrees of freedom (DOF) in the environment. Detection includes the characterization of DOF based on joint type and joint axis. Tracking implies the continuous perception of DOF state in order to monitor manipulation progress, recognize completion, or detect failure. These perceptual capabilities are a fundamental prerequisite for successful manipulation in unstructured environments with unknown objects (Fig. 1).

In this paper, we present a novel *online* interactive perception (IP) algorithm to estimate parametrized kinematic models of unknown objects from streaming RGB-D data. The proposed online IP algorithm consists of three interconnected levels of recursive state estimation: *1)* the estimation of 3-D feature motion based on the 2-D motion of tracked RGB features, *2)* the estimation of rigid body motion based on the estimated feature motion, and *3)* the estimation of the kinematic model based on the rigid body motion (Fig. 2). The probabilistic representations used for estimation yield a kinematic model with uncertainty estimates. We demonstrate the robustness, accuracy, and generality of the proposed algorithm in extensive real-world experiments as well as the usefulness of these uncertainty estimates.

We gratefully acknowledge the funding provided by the Alexander von Humboldt foundation and the Federal Ministry of Education and Research (BMBF) and through the First-MM project (European Commission,FP7-ICT-248258). We thank SimLab for their support.

[1]R. Martín Martín and O. Brock are with the Robotics and Biology Laboratory, Technische Universität Berlin, Germany

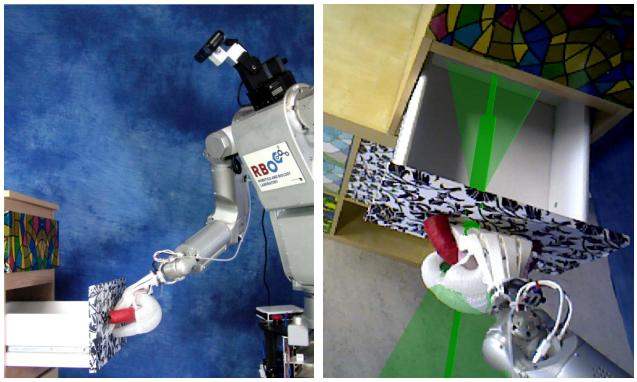

Fig. 1. Example of online interactive perception: The robot pulls on the drawer using an anthropomorphic soft hand built in our lab [1] and perceives the prismatic joint (joint axis shown as narrow green cylinder, joint value shown as wider green cylinder), including an estimate of the uncertainty (transparent green cone)

Two key features of the algorithm lead to the observed robustness, accuracy, and generality. First, the factorization of the overall perceptual problem into three levels enables the use of highly relevant, level-specific priors, namely motion continuity, rigid body physics and kinematics of rigid bodies. These physical priors effectively improve the quality of data at each level. Second, the three levels of the recursive estimation problem are interconnected. The information improved by the level-specific priors is passed to other levels, thereby also improving the effectiveness of the estimation process on other levels. The overall effect is that the combined estimation process is informed not only by sensor data but also by three specific process models, each containing task-relevant information to help interpret the uncertain data.

The proposed algorithm advances the state of the art in interactive perception of (rigid) articulated bodies in three respects. First, existing IP methods are offline algorithms and can therefore not inform the ongoing action of the robot, originally the goal of interactive perception. The proposed online method now overcomes this and integrates the perception process into the execution of actions. Second, the offline setting leads to failure cases that are properly addressed with our online method. Third, existing offline methods are not probabilistic and hence do not include an estimate of model uncertainly. We deem it to be important to reason about uncertainty when manipulating in unstructured environments.

II. Related Work

Interactive perception (IP) captures the idea that perception and interaction are inextricably linked. Manipulation of objects reveals perceptual information about the world which in turn informs manipulation. This concept has been applied to image segmentation [2, 3], object recognition [4, 5], and object singulation in cluttered environments [6, 7, 8]. In our review of related work, however, we will focus on work about perceiving kinematic models of articulated objects from visual data.

Yan and Pollefeys [9] propose an integrated approach for segmentation and joint detection. They use structure from motion to estimate 3-D feature trajectories and apply spectral clustering to identify rigid bodies. Ross et al. [10] improve this work by using maximum likelihood estimation instead of spectral clustering. These methods require to accumulate large motion data to estimate the depth of point features and are thus inherently offline algorithms.

Sturm et al. [11, 12] present a probabilistic approach to joint classification and characterization. Although this method incorporates uncertainty estimates about the joint types and is applicable in real-time, it omits the important part of detecting and tracking unknown rigid bodies. It requires that the rigid bodies are known beforehand and that their poses are tracked reliably. In contrast, our method incrementally detects the moving rigid bodies and tracks them, addressing the complete perceptual problem.

Huang et al. [13] present an offline method to extract 3-D models of articulated rigid objects using IP. However, this method requires multiple object views to first generate a full point cloud of an object, which is then used to estimate the kinematic state by matching the configurations before and after the interaction.

Katz et al. [14] propose an RGB-based, offline solution for the perception of three-dimensional, rigid kinematic structures. Subsequently, this method was adapted for RGB-D sensors [15]. The use of RGB-D sensing avoids the costly structure from motion computation and is therefore more accurate and computationally more efficient, but still offline, and thus suffering some inherent limitations, for example for newly appearing objects (see Section V-B).

The IP method described in this paper differs fundamentally from prior work in this area. The step from offline to online algorithms changes the nature of the perception problem, which can now be formulated as probabilistic recursive state estimation.

III. Recursive State Estimation

Recursive state estimation refers to a family of filters that estimate the current state x_t of a time-varying system based on the previous state x_{t-1}, a last observation z_t, and a control input u_t. When state and observation are stochastic processes, recursive estimation can be solved by using recursive Bayesian filtering. In this case, the filter estimates the posterior $p(x_t|z_{1:t}, u_{1:t})$ over the state, based on $p(x_{t-1}|z_{1:t-1}, u_{1:t-1})$ [16]. In a first step, called *prediction*, recursive Bayes filters use a *process model* to predict the distribution of the next state based on the current state distribution and the control input. In a second step, called *measurement update*, recursive Bayes filters predict the next measurement using a *measurement model*, and use the difference between predicted and the subsequently acquired new measurement to generate a new probabilistic estimate of the state.

If process and measurement model are linear functions with Gaussian noise and the probability of the initial state $p(x_0)$ is normally distributed, Bayes filters can be optimally implemented using Kalman filters [16]. If process or measurement model are not linear but linearizable, extended Kalman filters (EKF) are a suitable implementation for the Bayes filters. In the latter case, the process model becomes $x_t = g(x_{t-1}, u_t) + \varepsilon_t$ and the measurement model $z_t = h(x_t) + \delta_t$, where g and h represent the possibly nonlinear functions that have to be linearized, and ε and δ are process and measurement additive Gaussian noise.

In the following, we will use this structure of the recursive estimator and the terminology introduced here to describe the three interconnected recursive state estimators that make up the multi-level recursive estimator for online IP of articulated objects.

IV. Multilevel Recursive State Estimation for IP

Our proposed online algorithm factorizes the interactive perception of articulated objects into three recursive state estimation levels: estimating feature motion, rigid body motion, and the overall kinematic model. The structure and interactions of these levels is depicted in Figure 2. Each level exploits a level-specific prior: motion continuity, rigid body physics, and the kinematics of rigid bodies. These priors improve convergence of the state estimate. The resulting state information is passed as a measurement to the next-higher level (blue arrows). The predicted measurement of each level is also fed back as the predicted state to the next-lower level (red arrows). The information passed to the next-higher and next-lower levels is now informed by the prior and improves convergence at the other levels. Both of these design choices (use of priors, feedback to lower levels) are crucial to achieve the effectiveness, robustness, accuracy, and versatility of the proposed online IP algorithm.

We will now explain in detail the three recursive state estimation levels that constitute our online IP algorithm.

A. Recursive Estimation of Feature Motion

The first level of recursive state estimation tracks the motion of a set of point features in an RGB-D sensor stream using a recursive filter. The state of this filter is the vector $\boldsymbol{x}_t^{fm} \in \mathbb{R}^{3N}$ (fm = feature motion) containing the 3-D coordinates of N tracked point features. The measurements for this feature tracking filter $\boldsymbol{z}_t^{fm} \in \mathbb{R}^{2N}$ are the 2-D coordinates of the features in the image plane. We obtain these measurements by tracking the features in the RGB sensor stream using a point feature tracker. The estimation of feature motion relies on two priors: motion continuity (the location of a feature is close to its previous location) and physics of rigid bodies

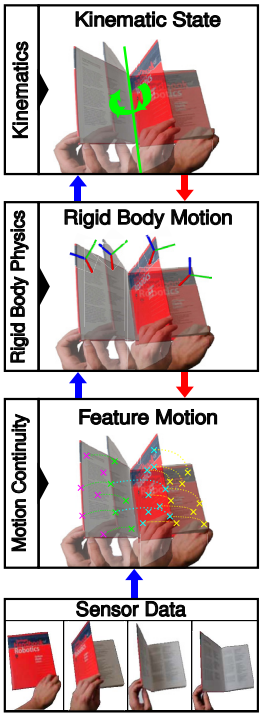

Fig. 2. Multi-level recursive estimation for IP: (from bottom to top) an RGB-D sensor data stream provides information about a scene, feature motion is estimated, from the feature motion rigid body motion is estimated, from the rigid body motion the kinematic model is estimated; the estimated state of each level is passed as measurement to the next-higher level (blue arrows) and the predicted measurements from one levels are passed to the next-lower level as state predictions (red arrows); level-specific priors to help the estimation process are a key feature of the proposed algorithm (vertical text on the left side of the boxes)

(the motion of the features on a rigid body must be consistent with the motion of the rigid body). This second prior allows us to leverage information determined by the next-higher level, the estimated motion of rigid bodies.

1) Prediction in Feature Motion Estimation: We predict the next feature locations from their current location and the estimated rigid body velocities. These velocities are estimated on the next-higher estimation level, the recursive Bayesian estimation of rigid body motion (see Section IV-B). The next-higher level effectively acts as the process model of the recursive estimation of feature motion.

2) Measurement Update in Feature Motion Estimation: We project the predicted 3-D feature locations into the image plane to obtain estimates of the next feature location as input to the Kanade-Lucas-Tomasi (KLT) tracking algorithm [17]. The KLT algorithm refines these 2-D locations, which we subsequently use to update the filter state. This recursive estimation schema (prediction and measurement update) in feature motion improves the tracking accuracy and robustness of the KLT algorithm.

3) Feature Initialization and Maintenance: The initial set of N features is selected based on Kanade-Tomasi corner point features in the first RGB image and their corresponding 3-D coordinates. Often, features get lost. To compensate for this loss and to continuously be able to extract useful information from the sensor stream, we increasingly add novel features to constantly maintain a set of N features.

To improve the reliability of feature tracking, we actively reject features based on three criteria. First, we reject features whose SURF descriptor [18] changes more than a given threshold (in our experiments: 0.1) over the course of 15 frames. Second, we reject features lying close to depth discontinuities in the RGB-D image. In the presence of tracking noise, these features change their depth drastically, negatively affecting the estimation of rigid body motion. Third, when the robot arm enters the field of view, we reject features tracked on its surface. We determine these features by projecting a geometric model of the robot into the image plane.

B. Recursive Bayesian Estimation of Rigid Body Motion

The second level of recursive state estimation is responsible for detecting and tracking the motion of rigid bodies, based on the feature motion estimated by the next-lower estimation level and the kinematic model estimated by the next-higher estimation level (see Section IV-C).

The motion of one single rigid body is estimated with a recursive Bayesian filter (RBF). We instantiate and maintain one independent RBF for each moving rigid body. In the following, we first suppose that some features have been correctly assigned to one RBF and describe its prediction and measurement update steps. (The detection of rigid bodies and the assignment of features to rigid bodies will be described later in this section.) This RBF is implemented as an extended Kalman filter (EKF). The state of the EKF is given by a 6-DOF rigid body pose and velocity represented as 6-D twists: $\boldsymbol{x}_t^{rbm} = (\boldsymbol{p}, \boldsymbol{v})^T \in \mathbb{R}^{12}$ (rbm = rigid body motion). The measurement $\boldsymbol{z}_t^{rbm} \in \mathbb{R}^{3M}$ is a vector containing the 3-D locations of the M features assigned to this rigid body.

1) Prediction in Single Rigid Body Motion Estimation: We use three different process models in parallel to predict the next rigid body state. The first model predicts the next pose of the rigid body based on its current pose and velocity and the elapsed time. The second process model handles the special case when a rigid body stops moving abruptly (for example, when closing a door), setting the current velocity to zero. The third process model uses the current kinematic model, estimated by the next-higher estimator, to predict an alternative next pose and velocity for the rigid body. The next-higher level is used therefore as process model of the recursive estimation of rigid body motion.

2) Measurement Update in Single Rigid Body Motion Estimation: The measurement input consists of the 3-D feature locations estimated on the next-lower level. We predict the future locations of features based on the predicted state of the rigid body and the following observation model:

$$\boldsymbol{z}_t^{rbm} = h(\boldsymbol{x}_t^{rbm}) + \boldsymbol{\delta}_t^{rbm} = \begin{pmatrix} T(\boldsymbol{p})\boldsymbol{f}_{init}^1 \\ T(\boldsymbol{p})\boldsymbol{f}_{init}^2 \\ \vdots \\ T(\boldsymbol{p})\boldsymbol{f}_{init}^M \end{pmatrix} + \boldsymbol{\delta}_t^{rbm}, \qquad (1)$$

where $T(\mathbf{p}) \in SE(3)$ is the homogeneous transform obtained from the rigid body pose and $\boldsymbol{f}^l_{init}$ is the 3-D location of the features when the body was initially detected. We predict the new observations relative to this reference to take advantage of increased precision with larger feature motion. The covariance of the measurement uncertainty δ^{rbm}_t is set proportional to the square of the feature depth (based on the RGB-D sensor properties) and inversely proportional to the tracking score given by the feature tracking algorithm. The EKF linearizes this measurement model using a first-order Taylor expansion of $h(\boldsymbol{x})$ around $\boldsymbol{x}^{rbm}_t$ and uses it to estimate the covariance matrix of the next state of the rigid body.

The state predictions obtained by the three process models generate different measurement predictions (next feature locations). The Bayes filter compares predicted and acquired measurements and uses the most likely state prediction given the actual measurement. As each of the predictions for the feature locations is generated under the assumption of rigid body motion, the predicted feature locations are informed by this prior.

3) Recursive Bayesian Estimation of Multi-body Motion: To track the motion of multiple rigid bodies, we have to group features accordingly and use a separate Bayes filter for each one. We assign features to those rigid bodies that best predict their motion. If the motion of a set of features cannot be accurately predicted (error under 2 cm) by any of the existing rigid body Bayes filters, we use RANSAC to find a rigid body transform that describes their motion. If a rigid body transform explains the motion of at least $f_{\min} = 15$ features, a new RBF is created, using this rigid body transform as the initial state. Using this procedure, the proposed algorithm works for an arbitrary number of moving rigid bodies in the scene, as long as $f_{\min}$ visual features can be tracked on each body.

C. Recursive Bayesian Estimation of Kinematic Model

The third level of our algorithm estimates and tracks the kinematic model of the scene, based on the motion of rigid bodies obtained on the next-lower estimation level. We assume a pair of rigid bodies to be related in one of four possible ways: *(i)* prismatic joint, *(ii)* revolute joint, *(iii)* rigid connection, or *(iv)* disconnected, the latter being a special case defined as the absence of relationships *(i)–(iii)*. We model these relationships with different types of RBF, each type modeling the necessary parameters for that relationship (joint axis, joint variable, etc.) in the state $\boldsymbol{x}^{joint}_t$. The measurements $\boldsymbol{z}^{joint}_t \in \mathbb{R}^6$ are obtained from the next-lower estimation level as the twist of relative motion between the two rigid bodies. The covariance of the measurement model noise δ^{joint}_t is also obtained from the next-lower level. We instantiate and maintain one RBF of every type for each pair of rigid bodies in the scene.

In the following we explain the state representation, prediction and measurement update of the three different RBF types. Each RBF type uses a different kinematic prior which defines its state and measurement model. As before, the priors enable the estimation and tracking of kinematic models, but also the prediction of the next state of the next-lower level (feedback). Following this, we explain how we estimate the most likely joint type between two bodies and the overall kinematic structure, which completes the description of this estimation level.

1) Prismatic Joint Estimation: The state of a prismatic joint is parametrized by the orientation of its axis (azimuth ϕ and elevation θ), its joint variable $q^p \in \mathbb{R}$ (translation along the joint axis), and the velocity of the joint variable. In the prediction step, we use the joint velocity to update the joint state. To predict the pose of one rigid body relative to the other, we use the following measurement model:

$$\boldsymbol{z}^{joint,p}_t = \begin{pmatrix} q^p \cdot \hat{\boldsymbol{o}}^p \\ \mathbf{0}_3 \end{pmatrix} + \delta^{joint}_t \tag{2}$$

where $\hat{\boldsymbol{o}}^p \in \mathbb{R}^3$ is the axis orientation (unit vector) estimated from ϕ and θ, and $\mathbf{0}_3$ is a three dimensional null vector.

2) Revolute Joint Estimation: The state of a revolute joint is parametrized by the orientation of its axis (azimuth ϕ and elevation θ), a point on the axis $\boldsymbol{p}^r \in \mathbb{R}^3$, its joint variable $q^r \in \mathbb{R}$ (rotation about the joint axis), and the velocity of the joint variable. Again, we use the joint velocity to predict the next joint state as process model. To predict the pose of one rigid body relative to the other, we use the following measurement model:

$$\boldsymbol{z}^{joint,r}_t = \begin{pmatrix} \boldsymbol{t}^r \\ q^r \cdot \hat{\boldsymbol{o}}^r \end{pmatrix} + \delta^{joint}_t \tag{3}$$

where $\hat{\boldsymbol{o}}^r \in \mathbb{R}^3$ is the axis orientation (unit vector) estimated from ϕ and θ and $\boldsymbol{t}^r = (-q^r \cdot \hat{\boldsymbol{o}}^r) \times \boldsymbol{p}^r$ is the linear relative motion between rigid bodies.

3) Rigid Joint Estimation: A rigid joint does not allow for relative motion between rigid bodies. Therefore, it has no parameters nor variables to estimate. The measurement model of a rigid joint predicts that there is no relative motion between bodies, i.e. $\boldsymbol{z}^{joint,rigid}_t = \mathbf{0}_6 + \delta^{joint}_t$.

4) Recursive Bayesian Estimation of Multi-type Kinematic Model: After evaluating the RBF of every type for each pair of rigid bodies, we select the one most consistent with the observed rigid body motion. If none of the RBF can explain the motion between a pair of rigid bodies, we declare this pair of rigid bodies to be disconnected. From all pairwise selected joint types and parameters, we build the kinematic model of the scene. Because joints are always determined considering only pairs of rigid bodies, our algorithm can naturally determine the kinematic model of branching mechanisms and closed kinematic chains.

V. Experiments

We conducted three sets of experiments. In the first set we evaluate the performance of the online IP algorithm with different articulated objects. We measure the robustness, quality, and convergence of the kinematic model estimation by comparing to ground truth. To obtain the ground truth for the joint parameters, we placed artificial markers that are not used by the algorithm to estimate the kinematic model. We then manually measured the joint parameters in

the RGB-D stream. In the second set of experiments, we test our algorithm in scenarios were offline algorithms fail. And in the third set, we make use of the online abilities of the algorithm to control the motion of a robot.

In all experiments, the input is an RGB-D stream, provided either by a Kinect or a Carmine RGB-D sensor. The articulated objects are of different size, color, texture, and with different kinematic structures (number and type of joints). The only constraint for the objects is that they have some visible texture. We also vary lighting conditions and the relative pose between the objects and the sensor. The algorithm tracks $N = 150$ features at a frame rate of 15 frames per second, running on real-time on an Intel Xeon E5520 PC at 2.27 GHz.

A. Experimental Evaluation

We measured the accuracy and convergence of our online IP algorithm on four articulated objects. Figure 3 shows initial, intermediate (after 1 s), and final frames of these experiments. The figure also includes graphs of the estimation error including estimated uncertainty over time. In some of the experiments, the observed motion was produced by human interaction, in some by a robot interacting with the environment, and in some the environment moved autonomously. In the following, we discuss each of the experiments from Figure 3.

1) Book experiment: The book is opened 60° and closed again (120° of accumulated motion) in 14 s. The joint is correctly classified from the first frame and converges within 1 s to a stable set of parameters. Point features are correctly assigned to the moving book cover. The error remains under 4° for the orientation and under 2 cm for the position of the revolute axis. We used artificial markers to obtain the ground truth of the revolute axis.

2) Umbrella experiment: The umbrella is extended by 40 cm in a motion lasting 10 s. The joint is continuously estimated correctly as prismatic. The features on the umbrella are correctly assigned. Some features on the hand are also assigned to the umbrella since they move coherently with it. The error of the estimated joint axis remains under 5° during the entire experiment. We used artificial markers to obtain the ground truth of the prismatic axis.

3) PUMA 560 experiment: In a motion lasting 15 s, the shoulder joint of the PUMA 560 robot moves 90° and the elbow joint moves 140°. Initially, our algorithm detects both links as a single moving rigid body. When the motion of the two links of the robot arm is different enough (0.7 s), the algorithm succeeds at separating them. Once both moving rigid bodies are detected, the features are correctly assigned. The revolute axis between base and upper arm and the revolute axis between upper arm and forearm are quickly classified as revolute, and their parameters converge fast to a stable accurate value. The joint between the base and the forearm is initially classified as revolute, but the algorithm quickly detects that there is no direct connection (disconnected joint). The estimation error of the first revolute axis (shown in the graph) remains under 6° for orientation and 5 cm for position; for the second joint the error remains under 8° and 8 cm after convergence. The estimates of joints connecting two moving bodies are usually less accurate, as the errors in motion estimation for both bodies add up. The robot does not have sufficient texture to reliably track features at this distance; we attached checkerboards to it to remedy this problem. In this experiment the RGB-D sensor is pointing parallel to the joint axes of the robot to simplify ground truth estimation. The experiments demonstrates the algorithm's ability to determine multiple DOF of a kinematic chain at the same time.

4) Human head experiment: The algorithm estimates the neck joint of a human shaking his head. The human rotates his head 100° in 5 s. The joint is correctly classified from the beginning of the motion, all features are correctly assigned, and the error of the axis after convergence remains under 5° and 4 cm. The RGB-D sensor is pointing perpendicular to the orientation of the joint to simplify ground truth estimation. The joint position is manually measured in the point cloud. This experiment demonstrates the performance of the algorithm on large semi-rigid articulated bodies.

B. Failure Cases of Previous Offline Algorithms Solved with Online IP

In this section, we show three situations that can only be handled by an online incremental IP algorithm. Existing offline methods would fail in the following scenarios.

1) Disappearing features: The motion of the object may cause all features obtained at the beginning of the motion to disappear by moving out of visual field or simply due to tracking error. Offline IP methods would fail, as they cannot find matching features between the initial and the final frame. We use a rotating globe and a portable projection screen with casing (see Figure 4, first and second rows) to demonstrate that the incremental nature of our online IP aims to overcome this problem. We rotate the globe 300° in 31 s and open the poster hanger 70 cm in 13 s. Our online algorithm quickly detects the moving bodies and incrementally assigns new features to them as they appear. This allows us to successfully track the motion of the rigid body, even when the initially visible parts of the object get obstructed (globe) or leave the field of view (projection screen).

2) Appearing objects: The articulated object may not be visible at the beginning of the analysis. To demonstrate how online IP can address this, we use a book in a cabinet and a Pioneer mobile base (see Figure 4, third and fourth rows). The cabinet has to be opened to perceive the book. We then open the book 30° in 3 s. Once the book is visible, new features are detected on its surface, and the joint can be perceived when the book is opened. The Pioneer base enters the field of view from the right. The base moves 82 cm in 22 s after entering the scene. The revolute joint connecting the wheel to the base as well as the prismatic joint between the robot base and the background are correctly estimated. At the end of the experiment the uncertainty about the prismatic joint increases because the robot base slightly changes its orientation.

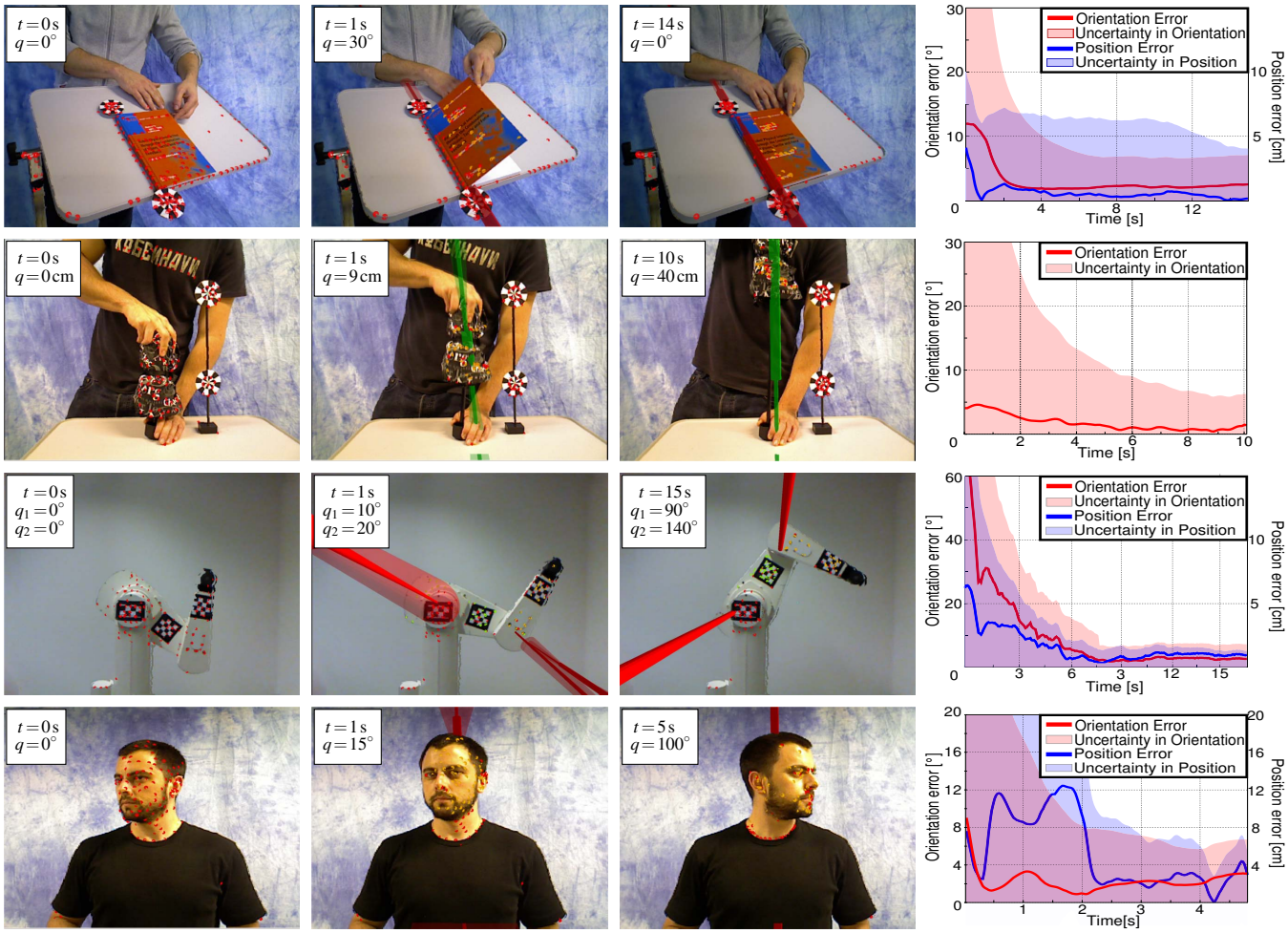

Fig. 3. Experiments with online IP (each row represents a different experiment): initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model, including error plot (fourth column) of joint configuration estimation, relative to ground truth, including uncertainty (shaded areas); the insets in the three images show the time t and the estimated joint variable q; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation; red dots are features assigned to the static background; dots of other colors are features assigned to moving rigid bodies

3) Identical initial and final configuration: When the initial and final configuration of the object performing the motion are identical, a comparison of these poses will not reveal information about the kinematic model. To show that online IP overcomes this problem of some offline IP methods, we experiment with a cabinet door and a drawer. The drawer is opened and closed (50 cm of accumulated motion) in 6 s, and the door is opened and closed (80° of accumulated motion) in 7 s. The proposed online IP algorithm accurately estimates the kinematic model. The model remains converged after the object returns to its initial configuration.

C. Controlling Interaction with Online IP

One of the main advantages of an online IP algorithm is the ability to use the kinematic model to control the robot's interaction with the environment. We demonstrate this in two experiments with two objects each (door and drawer). The goal of the first experiment is to obtain a kinematic model with a specified uncertainty bound (5° in orientation and 5 cm in position of the joint axis). The goal of the second experiment is to move one of the joints to a specific configuration. Each experiment is repeated ten times. Figure 4, rows five and six, shows initial, intermediate, and final frames of two trials of these experiments, with the online estimated joint variable in the bottom right corner.

In the first experiment we measure the amount of interaction necessary for the algorithm to reduce the uncertainty below a required level, and the deviation of the estimated kinematic model to ground truth (manually measured in the point clouds). In the case of the drawer, our controller stops, due to the attained uncertainty bounds, after a mean amount of motion of 5.07 cm. The mean error of the estimated axis is 4° with a single value above 5° (5.08°). In the case of the door our controller stops due to the attained uncertainty bounds after a mean amount of motion is 8.4°, with a maximum value of 26°. The mean error of the estimated axis is 2.95° with a maximum of 4.47°. The mean error in the estimated joint axis position is 7.03 cm with a maximum of 49.71 cm for a failed trial. Without this value the mean position error is 2.28 cm (under the 5 cm threshold).

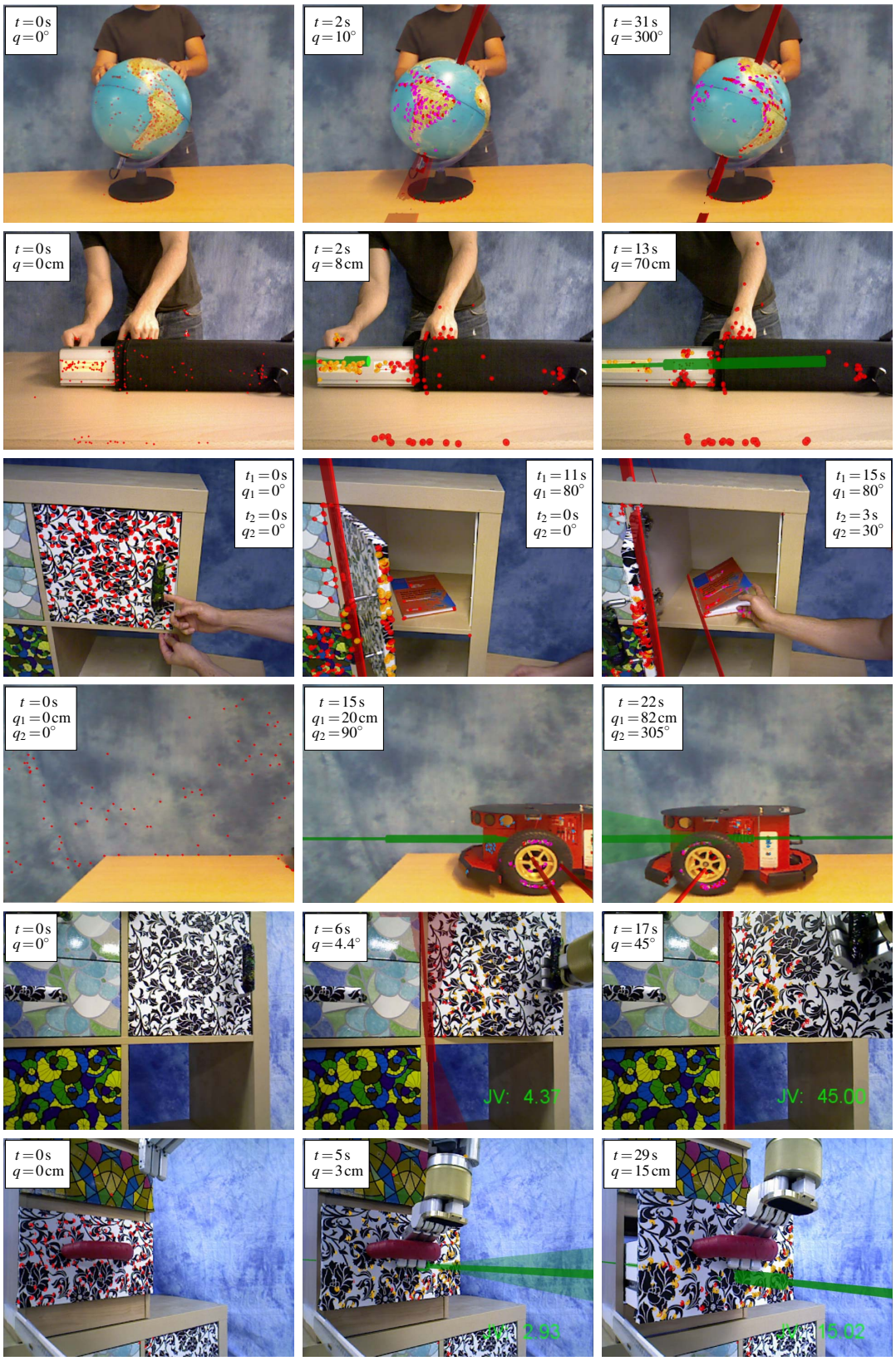

Fig. 4. Experiments with online IP (each row represents a different experiment): initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model; the insets show the time t and the estimated joint variable q; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation

In the second experiment, the robot manipulates the same objects as before so as to attain a certain value of a joint variable. In the case of the drawer, this value is 15 cm. The robot stops, when its model indicates this amount of motion. We measure the ground truth motion manually. The mean value of the measured joint value is 15.55 cm and the maximum and minimum are 15.9 cm and 15.2 cm, respectively. In the case of the door, the desired joint configuration is 45°. The mean value of the measured rotation is 44.8° with a minimum of 44° and a maximum of 46°.

The results of these experiments demonstrate that online IP can be used to monitor and control interactions with articulated objects in the environment. We showed that it is possible to adjust the robot's action based on a desired uncertainty bound for the accuracy during the estimation of a kinematic model. This demonstrates that the estimated uncertainty reflects the correctness of the estimated kinematic structure. We also showed that the online estimation of joint values can be used to monitor and attain manipulation goals, expressed in terms of specific joint configurations.

VI. Limitations

The proposed method inherently depends on motion in the scene. This motion can be produced by robot interaction, by others in the scene, or by the object itself, but there must be motion for our method to work. In a separate line of our research, we investigate how robots can generate such motion so as to explore efficiently [19, 20].

Only objects with sufficient trackable texture can be perceived. As a result, our method inherits the limitations of feature tracking, including the requirement of "good" features, relatively stable lighting conditions and bounded object acceleration. Note that we explicitly address the case of high deceleration to zero velocity (see Section IV-B.1).

There are computational limitations, but we do not deem these severe. To be able to integrate into the robot's action loop, our algorithm must perform at reasonably high frame rates. The frame rate is mostly affected by the number of tracked features. In all our experiments, we track 150 features at 15 Hz, independent of the number of moving rigid bodies. Increasing this number to 250 reduces the frame rate to 9 Hz.

Currently, our method only handles four kinematic relationships (revolute and prismatic joint, rigidly connected, and disconnected). We are extending the estimation of kinematic structures to other types of kinematic relationships [11, 12].

VII. Conclusion

We presented an online algorithm for the interactive perception of articulated bodies. It receives as input an RGB-D stream and at interactive frame rates outputs a kinematic model of the observed scene, including joint configuration values. This perceptual capability is a prerequisite for successful manipulation in unstructured environments. We formulated the perception problem as three interconnected recursive estimation filters, successively estimating feature motion, rigid body motion, and kinematic model of moving objects in the scene. The composition of these three filters and the bidirectional flow of information between them result in a highly robust algorithm. This robustness is a result of level-specific priors that help to interpret the data and to reject measurement noise. The connectivity between the levels passes valuable information from simpler sub-problems among the levels, further improving the convergence of the overall estimator. We demonstrate the effectiveness of our method in twelve real-world experiments in which the algorithms successfully estimates the degrees of freedom of humans, robots, and objects in the scene.

References

[1] R. Deimel and O. Brock, "A novel type of compliant, underactuated robotic hand for dexterous grasping," in *Robotics: Science and Systems*, 2014, submitted.

[2] P. Fitzpatrick, "First contact: an active vision approach to segmentation," in *International Conference on Intelligent Robots and Systems*, vol. 3, 2003, pp. 2161–2166.

[3] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *International Conference on Robotics and Automation*, 2009, pp. 1377–1382.

[4] B. Willimon, S. Birchfield, and I. Walker, "Rigid and non-rigid classification using interactive perception," in *International Conference on Intelligent Robots and Systems*, 2010, pp. 1728–1733.

[5] K. Hausman, F. Balint-Benczedi, D. Pangercic, Z.-C. Marton, R. Ueda, K. Okada, and M. Beetz, "Tracking-based interactive segmentation of textureless objects," in *International Conference on Robotics and Automation*, 2013, pp. 1122–1129.

[6] B. Willimon, S. Birchfield, and I. Walker, "Classification of clothing using interactive perception," *International Conference on Robotics and Automation*, pp. 1862–1868, 2011.

[7] L. Chang, J. R. Smith, and D. Fox, "Interactive singulation of objects from a pile," *IEEE International Conference on Robotics and Automation*, pp. 3875–3882, 2012.

[8] T. Hermans, J. M. Rehg, and A. Bobick, "Guided pushing for object singulation," in *International Conference on Intelligent Robots and Systems*, 2012, pp. 4783–4790.

[9] J. Yan and M. Pollefeys, "Automatic kinematic chain building from feature trajectories of articulated objects," in *Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 712–719.

[10] D. Ross, D. Tarlow, and R. Zemel, "Unsupervised learning of skeletons from motion," *Computer Vision–ECCV*, pp. 560–573, 2008.

[11] J. Sturm, C. Stachniss, V. Pradeep, C. Plagemann, K. Konolige, and W. Burgard, "Learning kinematic models for articulated objects," in *International Joint Conference on Artificial Intelligence*, 2009.

[12] 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 *International Conference on Robotics and Automation*, 2010, pp. 362–368.

[13] X. Huang, I. Walker, and S. Birchfield, "Occlusion-aware reconstruction and manipulation of 3D articulated objects," *International Conference on Robotics and Automation*, pp. 1365–1371, 2012.

[14] D. Katz, A. Orthey, and O. Brock, "Interactive perception of articulated objects," in *International Symposium on Experimental Robotics*, 2010.

[15] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz, "Interactive segmentation, tracking, and kinematic modeling of unknown articulated objects," in *International Conference on Robotics and Automation*, 2013, pp. 5003–5010.

[16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[17] C. Tomasi and T. Kanade, "Detection and tracking of point features," Tech. Rep., 1991.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[19] D. Katz, Y. Pyoro, and O. Brock, "Learning to manipulate articulated objects in unstructured environments using a grounded relational representation," *Robotics: Science and Systems IV*, p. 254, 2009.

[20] S. Höfer, T. Lang, and O. Brock, "Extracting kinematic background knowledge from interactions using task-sensitive relational learning," in *International Conference on Robotics and Automation*, 2014, accepted.