Coupled Recursive Estimation for Online Interactive Perception of Articulated Objects

Journal Title
OnlineFirst(X):1–33
©The Author(s) 2018
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364919848850
www.sagepub.com/
SAGE

Roberto Martín-Martín and Oliver Brock

Abstract

We present onlinie multi-modal perception systems for extracting kinematic and dynamic models of articulated objects from physical interactions with the environment. The systems rely on a RGB-D stream, contact wrenches, and proprioception. The proposed systems share an algorithmic foundation: they are based on an architecture of coupled recursive estimation processes. We present and advocate this architecture as a general, versatile, and robust solution for online interactive perception problems. We validate the architecture in extensive experiments to extract kinematic models interactively, varying the appearance, size, structure, and dynamic properties of objects for different tasks and under different environmental conditions. Additionally, we experimentally show that the information acquired by the online perception systems enables robot manipulation of articulated objects. Furthermore, we discuss the relationship between the proposed architecture for robot perception and insights about biological perception systems.

Keywords

Interactive perception, visual perception, multimodal perception, articulated objects, robot manipulation

1 Introduction

Robots physically interact with their environment to achieve manipulation tasks. These tasks require kinematic configuration of objects to be changed in a task-specific manner. When objects are articulated, i.e., composed of rigid parts connected by joints that restrict relative motion, their manipulation involves changing the configuration of their internal degrees of freedom (DoF). Examples of typical manipulation tasks on articulated objects that a robot operating in a human environment must be able to accomplish include opening a door, closing a drawer, or actuating a valve.

To perform these tasks robustly and successfully, the robot must be able to perceive the kinematic and dynamic properties of objects. Information about the kinematic structure of the object allows the robot to control and plan the interaction, monitor manipulation progress, and detect failure (Bruyninckx and Schutter 1996; Stilman 2007; Pflueger and Sukhatme 2015). Additionally, information about the dynamic properties of the object allows the robot to execute safe and robust interactions (Eppner et al. 2018; Endres et al. 2013). However, for previously unseen objects, this information is unknown and only revealed when the robot has already begun interacting with the objects.

Perception that requires or exploits interaction is called *interactive perception* (IP) (Bohg et al. 2017). We define *online* IP problems as the class of IP problems for which information about the object must be acquired *throughout* that interaction, rather than after the interaction has been completed.

We propose three online IP systems for extracting kinematic and dynamic models of articulated objects from physical interactions: 1) a vision-based system for kinematics, 2) a multimodal system for kinematics, and 3) a multimodal system for kinematics and dynamics. We evaluate these systems in extensive experiments with articulated objects of different size, shape, color, and degrees of freedom. The experiments include different tasks and varying challenging environmental conditions. We show that the robot is able to use information perceived online to support ongoing manipulation, demonstrating that our algorithms are able to solve online IP problems.

The three systems share a common algorithmic architecture that is responsible for the online capabilities and the observed robustness, accuracy, and generality of perception. The architecture consists of a network of coupled recursive estimation processes. Each of the processes encodes a task-specific prior (e.g. physics models, sensor models). Different processes exchange information among each other to assist each other in extracting information from the robot's sensor-action stream. We present and discuss this architecture as a general solution to online IP problems. We analyze the most relevant features of the architecture and identify the structural properties of the online IP problem that it leverages.

The rest of the paper is structured as follows. First, we present a perceptual system that builds online kinematic models of articulated objects from the visual information revealed through interactions (Section 2). The system is a first implementation of our proposed algorithmic architecture for online interactive perception. We will analyze this first system and infer the algorithmic features

Robotics and Biology Laboratory, Technische Universität Berlin, Germany

Corresponding author:
Oliver Brock, Robotics and Biology Laboratory, Technische Universität Berlin, Marchstr. 23, 10587 Berlin, Germany.
Email: oliver.brock@tu-berlin.de

that constitute the algorithmic architecture (Section 3). Then, we present additional instances of the architecture, one for the perception of kinematics from a multimodal sensor stream (Section 4), and one for the perception of dynamic properties (Section 5) of articulated objects. These additional systems demonstrate the generality and extensibility of the proposed architecture. We will conclude in Section 6.1 by analyzing the properties of the algorithmic architecture that make it well-suited for online IP problems. Furthermore, in that section we will compare these properties of the proposed architecture to analogous features of biological perception reported by cognitive scientists, neuroscientists, and psychologists.

The systems that perceive kinematics from visual signals and from multimodal signals were originally presented at conferences (Martín-Martín and Brock 2014; Martín-Martín and Brock 2017). The current paper extends and completes the description of these algorithms; it also presents a more extensive experimental evaluation, including an analysis of their parameter sensitivity and the importance of the communication between processes. The system to perceive dynamic properties of articulated objects is novel to this paper. Most importantly, we present here the three systems as instances of a general algorithmic architecture for online interactive perception. The analysis of these systems and the architecture also are contributions unique to this paper.

2 Online Perception of Kinematics of Articulated Objects from Vision

Articulated objects are composed of rigid parts, i.e. links, connected by joints. The joints restrict the relative motion between links along the dimensions of the degrees of freedom. Doors, drawers, books, laptops, and scissors are some examples of articulated objects that a robot could commonly encounter in a human environment. The kinematic properties of such articulated objects can be represented by the *kinematic structure* and a *kinematic state*. The kinematic structure consists of a set of joints that constrain the motion of adjacent rigid components of the articulated object. The set of all joint states is also called the kinematic state (or configuration) of the articulated object. The task of perceiving kinematics of previously unseen articulated objects comprises two subtasks: the perception of its kinematic structure and the perception of its kinematic state.

2.1 Related Work

The earliest approaches to visually perceive kinematics of articulated objects are based on the seminal work by Costeira and Kanade (1998). These authors reconstructed shape and motion of multiple moving bodies applying spectral clustering to the *measurement matrix* (Tomasi and Kanade 1992), which is a matrix containing the trajectories of a set of point features tracked along an entire video sequence. The results are a (sparse) shape model and the trajectory of the moving bodies in the video sequence. Tresadern and Reid (2005) proposed an integrated approach for segmentation and joint detection based on an analysis of the dependencies in the motion subspaces obtained with spectral clustering. They found structure in these subspaces that indicate constraints on the relative motion between pairs of bodies. Based on these constraints, their method can classify joints into disconnected, universal, and revolute/hinge joints. The method also estimates joint parameters: the axis of rotation of the revolute joint and the point of rotation of the universal joint. However, this method is limited to single-joint objects.

Yan and Pollefeys (2006) extended the idea of Tresadern and Reid (2005) to more complex kinematic structures. They build a fully connected graph where the nodes are moving bodies and the edges are weighed by the motion dependency observed between bodies, which is measured as the minimum principle angle between their motion subspaces (Golub and Van Loan 2012). The pairs of bodies moving with high dependency are connected by joints. The complete kinematic structure of the articulated object is defined as the minimum spanning tree in the motion dependency graph. Their method can deal with multiple articulated objects based on an upper threshold for the minimum principle angle that indicates that two bodies are disconnected, but it cannot deal with closed kinematic chains.

The methods based on spectral clustering demonstrate that it is possible to visually perceive the kinematic constraints between moving bodies and infer the kinematic structure of an object. However, they require accumulating large motion data to correctly estimate the clusters and the motion subspaces of the point features and are, thus, inherently *offline* algorithms.

More recent solutions to perceive kinematic structures have adopted a probabilistic approach. They posed the perceptual problem as the estimation of the model (the kinematic structure and, possibly, also its state) that maximizes the likelihood of the observations. Ross et al. (2008) proposed a generative model as a solution to the underlying multi-body structure from motion (SfM) from point feature trajectories and joint estimation problems. In an iterative process, they first assign point features to links, run SfM, and estimate the points that belong to a pair of links and that do not change their 3D location. These points indicate possible locations of a revolute or a universal joint. The generative model obtained with this method is used to evaluate how well the hypothetical kinematic structure model fits the observed point feature trajectories and then select the most likely one. This method cannot neither cope with prismatic joints nor with multi-joint structures.

Sturm et al. (2009) (Sturm et al. 2010b, 2011) presented a probabilistic approach to joint classification and parameter estimation. Their method uses the 6D pose trajectories of the moving bodies as input to build and maintain four models of possible motion constraints between pairs of bodies. The method estimates the set of parameters maximizing the likelihood of the observed trajectories through optimization. The authors proposed three possible parametric models for joints (rigid, prismatic, or revolute joints) and one non-parametric model, a Gaussian process joint. While the method by Sturm et al. (2009) is elegant, completely defined using probabilistic algebra, and applicable in real-time, it does not tackle the full perceptual problem: the method does not detect and track unknown rigid bodies from raw sensor data. The approach assumes that the number of rigid bodies is known beforehand and their poses are tracked reliably,

thereby delegating the detection and tracking tasks to a visual tracker based on AprilTag-like fiducial markers (Wang and Olson 2016). The exclusion of the "lowest" part of perception (the interpretation of the noisy sensor stream) from the problem is a missed opportunity to link and exploit high level reasoning and low level signal processing.

In two later works, the authors extended their approach to reduce the dependency on the fiducial visual markers. In a first extension, they proposed to obtain the body trajectories from a plane tracker based on depth images (Sturm et al. 2010b). This method is limited to planar objects, and its serial processing procedure does not leverage the information about the kinematic structure that could help the plane tracker. In a second extension, the authors use the robot's end-effector trajectories generated with a model predictive controller as perceptual signal (Sturm et al. 2010a). This approach, while nicely linking perception and action, is limited because it can only perceive single DoF objects rigidly attached to the environment (see Section 4).

Another group of methods perceive the kinematic model from a geometrical analysis of the rigid body trajectories. Huang et al. (2012) presented an offline method to build shape and kinematic models of articulated objects from RGB images of interactions. The authors collected images from multiple views before and after the interaction. Then, they generated two shape models using bundle adjustment (Triggs et al. 2000) and compared them to segment the moving links. Based on the geometrical properties of the trajectory of the moving links, they inferred the type of joints (prismatic or revolute), their parameters, and change in state.

Katz et al. (2014) proposed an RGB-based offline solution to build sparse shape models and kinematic models of articulated objects. The authors applied bundle adjustment to groups of point feature trajectories to estimate their 3D motion and fit the computed 3D trajectories to joint hypotheses based on their geometrical properties. Later on, this method was adapted to analyze RGB-D sequences (Katz et al. 2013). The adapted method does not require the costly bundle adjustment computation and is, therefore, more accurate and computationally more efficient, but it is still offline, as it requires large feature trajectories, and, therefore, it suffers some inherent limitations, for example, when new objects appear (see Section 2.3.3).

Compared to the related work discussed here, the system we present in this section advances the state of the art in interactive perception of kinematics of articulated bodies in three aspects. First, as explained before, most existing IP methods are offline systems and, therefore, the robot cannot use the perceived information to support an ongoing action. Our proposed online system overcomes this limitation and integrates the perception process into the execution of actions. Second, due to their offline setting and batch processing, previous approaches present some failures that are properly addressed with our online incremental system (see Section 2.3.3). Third, existing offline methods are not probabilistic and, hence, do not include an estimate of model uncertainly. We consider that robots in unstructured environments should be able to assess the uncertainty of the information they perceive in order to manipulate and interact safely. One exception is the algorithm by Sturm et al. (2009), but as we mentioned before, this method does not tackle the low level perceptual problem because it relies on fiducial markers for object detection, segmentation, and tracking. Our online IP system addresses the full perceptual problem online and provides uncertainty bounds on the perceived information.

2.2 Method

We propose an online IP system to build kinematic models of articulated objects by factorizing the perceptual problem 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 Fig. 1. Each level exploits level-specific physical priors: motion continuity and projective geometry, rigid body physics (assuming the objects are composed of rigid parts), and kinematics of rigid bodies. These priors are used to interpret the input signals as evidence of unobservable states. The resulting state information is passed as a measurement to the next level (left-to-right arrows). Furthermore, the predicted measurement of each level is fed back as the predicted state to the previous level (right-to-left arrows). The information passed to the next and previous levels is now informed by the prior and improves convergence at the other levels. These design choices (factorization, recursion, use of priors, coupling through communication) are crucial to achieve the effectiveness, robustness, accuracy, and versatility of the proposed online IP system.

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

2.2.1 Estimation of Feature Motion: The first level of recursive processing tracks the motion of a set of salient point features in an RGB-D sensor stream using a recursive procedure. The state of this filter at time t, x_t^{fm} (fm = feature motion) is of the form:

$$x_t^{fm} = \{f_t^n = (x_t^n, y_t^n, z_t^n, l^n)\}_{n\in\{1,\dots,N\}} \quad (1)$$

where $x_t^n, y_t^n, z_t^n \in \mathbb{R}$ are the coordinates of the salient point feature f^n in the 3D Euclidean space relative to the sensor frame at time t, and $l^n \in \mathbb{N}$ is a time-constant label that uniquely identifies the feature. N is the number of tracked points that we maintain constant by detecting new salient point features when necessary as we explain later in this Section.

The measurements for this salient point feature tracking process at time t present the form:

$$z_t^{fm} = \{q_t^n = (u_t^n, v_t^n, l^n)\}_{n\in\{1,\dots,N\}} \quad (2)$$

where $u_t^n, v_t^n \in \mathbb{R}$ are the 2D coordinates of the salient point feature q^n in the image plane at time t, and $l^n \in \mathbb{N}$ is the feature label of the corresponding 3D point. To obtain these measurements, we track points in consecutive RGB images using the Kanade-Lucas-Tomasi (KLT) salient point feature tracker (Tomasi and Kanade 1991).

The KLT estimates the flow of a point feature with an iterative process of linearization (first order Taylor expansion) and minimization of the registration error, i.e. the difference between image intensities in a window around the point features. Being an iterative gradient-based method, the

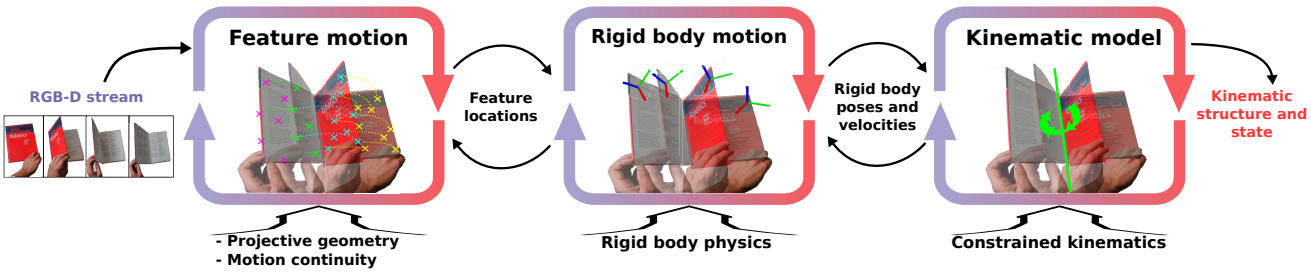


Figure 1. Multi-level recursive estimation of kinematic models from vision: (from left to right) an RGB-D sensor data stream provides information about a scene, from which the feature motion is estimated, from the feature motion, the rigid body motion is estimated, and from the rigid body motion, the kinematic model is estimated; the estimations from each level are passed as measurements to the next level (left to right arrows), and the predicted measurements from one level are passed to the previous level as state predictions (right to left arrows); level-specific physical priors (text on the bottom of each box) are encoded into each estimation process to interpret the inputs as evidence of a latent state; the system is an instantiation of the algorithmic architecture presented in Section 3

solution depends on the initial estimate of the displacement. Different initializations could converge to different local minima of the registration errors. This sensitivity to the initialization is a known problem of the KLT tracker. In our system, we will leverage physical priors and information from other processes to initialize the KLT based on predictions of the motion of the point features (see Fig. 2).

Prediction in Feature Motion Estimation: We propose two forward models to predict the motion of the features. The first model is an internal forward model within the recursive process that assumes that the tracked 3D points do not move from their previous location. This forward model generates a first prediction for the next state (I = first prediction):

$$\hat{x}_t^{fm,I} = \{\hat{f}_t^{n,I} = (x_{t-1}^n, y_{t-1}^n, z_{t-1}^n, l^n)\}_{n\in\{0,\dots,N\}} \quad (3)$$

Therefore, the internal forward model encodes *motion continuity* as physical prior: the current 3D location of a point is close to its previous location.

The second model leverages information from the next level (the estimation of rigid body motion, see Section 2.2.2) as a prior to generate a second prediction for the next state, $\hat{x}_t^{fm,II} = \{\hat{f}_t^{n,II}\}_{n\in\{0,\dots,N\}}$ (II = second prediction). We predict the location of a point feature f^n on a body B that moves with a predicted velocity $\hat{\eta}_t^B$ as*:

$$Loc(\hat{f}_t^{n,II}) = exp\left(\Delta_t \hat{\eta}_t^B\right) Loc(f_{t-1}^n) \quad (4)$$

where Δ_t is the time elapsed between $t-1$ and t, and $exp\,()$ is the matrix exponential map from elements from the Lie algebra $se(3)$ (poses in exponential coordinates) to elements of Lie group $SE(3)$ (homogeneous transformation matrices).

This second forward model leverages physics of rigid bodies as prior: the motion of the point features on a rigid body must be consistent with the motion of that rigid body. This second prior allows us to leverage information determined by the next level, the estimated motion of rigid bodies. The next level effectively acts as a forward model for the recursive estimation of feature motion. These more informed predictions lead to better initialization of the KLT feature tracker in the measurement update, as we will see next.

Measurement Update in Feature Motion Estimation: To predict the measurements, we project the two sets of predicted 3D locations into the image plane and obtain two sets of predicted 2D locations, $\hat{z}_t^{fm,I}$ and $\hat{z}_t^{fm,II}$. In this process, our measurement model leverages projective geometry as physical prior to interpret sensor data as evidence of the state.

We use the predicted 2D locations to initialize the KLT salient point tracking algorithm. The KLT tracker corrects these predicted 2D locations, finding the displacement that minimizes the registration error.

The two sets of predicted 3D locations lead to different initialization values for the KLT feature tracker. The first set, from the internal forward model based on motion continuity, leads to the standard zero initial displacement of the iterative KLT process, $d_0 = (0, 0)$. The KLT tracker then searches for the optimal registration of the intensity window of the point in the first image starting by the window around the same location in the second image. This standard initialization restricts the capabilities of the KLT to track large motions between frames since the initial displacement could lay in the region of convergence of different points. This problem is depicted in Fig. 2.

The second set of predicted 3D locations leads to an initialization of the KLT tracker informed by the motion of the rigid bodies. Predictions about the 3D location of the point features based on the expected motion of the bodies guide the salient point KLT tracker to a different area of the image that is closer to the right location.

We compare the feature tracking residues (final difference in intensity in the surrounding windows) based on the two initializations, $\epsilon^{n,I}$ and $\epsilon^{n,II}$, and select the correction with lowest residue. As explained before, the residue measures the quality of the matching between image patches after applying the estimated optimal flow. The assumption is that the best correction is the one that best aligns the image patches around the salient point. Finally, we update the state of the recursive process, the 3D location of the point features,

*In the following, we will use the operator $Loc(f^n) = (x^n, y^n, z^n)^T$ to build a 3D vector of the location of a feature f^n. Sometimes, we will abuse the terminology and use the same operator to build the vector of homogeneous coordinates of the feature location, $Loc(f^n) = (x^n, y^n, z^n, 1)^T$. The difference will be clear from the context of the operation.

Figure 2. Estimating feature motion; *left*: RGB image input to our perceptual system ; *right top*: detail on the surface of the moving drawer and search window (red) centered around one tracked point feature; *right, middle and bottom*: same area of the drawer in the next processed RGB image and window around the corrected point feature location based on the first initialization (*middle*, green window) and the second initialization (*bottom*, magenta window); the initialization with priors from the next process guides the search to the right location

by querying the value of the depth map at the corrected tracked 2D locations.

Feature Initialization and Maintenance: The above presented procedure recursively estimates the 3D location of a set of N points, associating them to 2D point features in the image. To initialize the recursion, we need to find a set of points that we can track reliably. Additionally, we will need to find new points to maintain a constant number of N tracked points when previous features are lost or actively removed, as we will explain later.

We detect new salient point features to track using the algorithm proposed by Shi and Tomasi (1994). This algorithm finds points where the gradient of the intensity of image changes abruptly in all directions, the so-called corner points. These points correspond to locations where both eigenvalues of the matrix formed by the second order moments of the gradient are large. We detect an initial set of N salient point features where the second order moments of the gradient of the intensity image are maximum.

We compute their corresponding 3D coordinates to initialize the state of the process based on the associated depth value in the fourth channel of the registered RGB-D frame. Often, our approach detects salient points features in sub-optimal locations like depth edges or shadows that do not actually move with the motion of the rigid body. Using our prediction-correction mechanism informed by the motion of the rigid bodies that we explained before, these points can be compensated.

We further improve the reliability of feature tracking by actively rejecting features based on four criteria. First, we remove features when they move out of the field of view because we do not have sensor measurements to update their location. Second, we reject features lying close to depth discontinuities in the RGB-D image. In the presence of sensor noise, these features change their depth drastically, negatively affecting the estimation of rigid body motion. We estimate discontinuities in the depth image using a Canny edge detector (Canny 1986) and use them to reject point features. 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 using the joint angles of the robot's arm and forward kinematics. In this way, we focus the attention of the perceptual system into the degrees of freedom of the unknown articulated objects and not on the known robot arm, although this requires the camera pose in robot's reference frame to be known. And fourth, we reject points if their tracking error residue increases over a maximum value ϵ_{max} or if their saliency (minimum eigenvalue of the matrix of second order moments of the gradient image) falls under a discriminative threshold $\lambda_{min}^{tracking}$.

Due to the mechanisms explained above, features get lost often. To compensate for this loss and to continuously be able to extract useful information from the sensor stream, we increasingly add novel points (based on the approach explained above to detect salient points) to constantly maintain a set of N features.

2.2.2 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 previous estimation level and the kinematic model estimated by the next estimation level (see Section 2.2.3).

The online solution to this problem requires the solving of three interdependent problems. First, we have to continuously associate salient point features to existing or novel rigid bodies. Second, we have to detect when a novel rigid body begins to move. And third, given the association of features to rigid bodies, we have to estimate the motion of each rigid body based on the feature motion.

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 a set of salient point features have been correctly associated to one RBF and describe its prediction and measurement update steps. The RBF to estimate the motion of one rigid body is implemented as an extended Kalman filter (EKF). The detection of rigid bodies and the assignment of features to rigid bodies will be described later in this section.

We represent the kinematic state of a rigid body by its 6D pose and velocity relative to the sensor frame, which we assume to be Gaussian distributed. Representing a Gaussian distribution over 6D poses is not trivial. We adopt the formalism of Barfoot and Furgale (2014) and represent the pose distribution as a mean pose, p (represented in exponential coordinates), that is perturbed with noise in the tangential Lie algebra space, $\Sigma_t^{p_t}$. The resulting state is (*rbm* = rigid body motion):

$$\boldsymbol{x}_t^{rbm} = (\boldsymbol{p}_t, \boldsymbol{\eta}_t) \sim \mathcal{N}((p_t, \eta_t), P_t^{rbm}) \quad (5)$$

$$p_t, \eta_t \in se(3) \quad (6)$$

where the upper-left 6×6 block of the state covariance matrix, $P_t^{rbm} \in \mathbb{R}^{12\times12}$, corresponds to the pose uncertainty in the tangential Lie algebra space as mentioned before[†].

The measurements of this RBF is the set of M point features $f_t^0, \ldots, f_t^{M-1}$ in 3D Euclidean space associated to

[†]Strictly speaking, the vectors of their exponential coordinates p_t and η_t are not elements of the Lie algebra but the matrices $p_t^\times, \eta_t^\times \in se(3)$. Here, we are simplifying the notation.

this rigid body. We stack their 3D locations to compose a measurement vector for the rigid body RBF:

$$z_t^{rbm} = (\text{Loc}(f_t^0), \ldots, \text{Loc}(f_t^{M-1}))^T \in \mathbb{R}^{3M} \quad (7)$$

Prediction in Single Rigid Body Motion Estimation: We use three different process models in parallel to predict the next rigid body state. The first forward model predicts the next pose of the rigid body based on its current pose and velocity and the elapsed time, Δ_t. The second process model handles the special case when a rigid body stops moving abruptly (for example, when closing a door). The third process model uses information from the next estimator (the current kinematic model) to predict an alternative next pose and velocity for the rigid body.

The first forward model is a constant velocity model with random walk in acceleration (I = first prediction):

$$\boldsymbol{x}_t^{rbm,I} = f^{rbm,I}(\boldsymbol{x}_{t-1}^{rbm}) + \boldsymbol{w}_t^{rbm} \sim \mathcal{N}((\hat{p}_t^I, \hat{\eta}_t^I), \hat{P}_t^{rbm,I}) \quad (8)$$

$$\hat{p}_t^I = \Delta_t \eta_{t-1} \oplus p_{t-1} \quad (9)$$

$$\hat{\eta}_t^I = \eta_{t-1} \quad (10)$$

where $\oplus$ is the composition of poses.

The system noise is a zero mean Gaussian distributed random variable defined as:

$$\boldsymbol{w}_t^{rbm} = \begin{pmatrix} \frac{T^2}{2} \\ T \end{pmatrix} \boldsymbol{a}^{rbm} \quad (11)$$

where $T = I_{6x6} \cdot \Delta_t$, I_{6x6} is the identity matrix of size 6×6, and $\boldsymbol{a}^{rbm} \sim \mathcal{N}(0, \Sigma_a)$ is a 6D zero mean Gaussian distributed random variable that represents the unknown rigid body acceleration. The covariance of its distribution is given by:

$$\Sigma_a = \text{diag}(a_x, a_y, a_z, a_{rx}, a_{ry}, a_{rz}) \quad (12)$$

where diag indicates a diagonal square matrix, and the diagonal elements correspond to possible accelerations of the rigid body in the different 6D dimensions. Larger values in the diagonal allow the RBF to adapt to fast motions at the cost of being more sensitive to point feature noise.

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 and the predicted pose to be the current pose (II = second prediction):

$$\boldsymbol{x}_t^{rbm,II} = f^{rbm,II}(\boldsymbol{x}_{t-1}^{rbm}) + \boldsymbol{w}_t^{rbm} \sim \mathcal{N}((\hat{p}_t^{II}, \hat{\eta}_t^{II}), \hat{P}_t^{rbm,II}) \quad (13)$$

$$\hat{p}_t^{II} = p_{t-1} \quad (14)$$

$$\hat{\eta}_t^{II} = 0_{6\times 1} \quad (15)$$

The third process model uses the current kinematic model, estimated by the next estimator (see Section 2.2.3), to predict an alternative next pose and velocity for the rigid body (III = third prediction):

$$\boldsymbol{x}_t^{rbm,III} = f^{rbm,III}(\hat{\boldsymbol{z}}_t^{joint}) + \boldsymbol{w}_t^{rbm} \quad (16)$$

where $\hat{\boldsymbol{z}}_t^{joint}$ is the predicted measurement generated at the next level of our system (Section 2.2.3), and the function $f^{rbm,III}$, independent of the current state, predicts the pose and velocity of the body with respect to the sensor frame based on the relative pose between links predicted by the kinematic model. Therefore, the next level is used as alternative process model of the recursive estimation of rigid body motion.

We will select the prediction among the three alternatives that best predicts the motion of the point features, as we will see later in this section.

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

$$\boldsymbol{z}_t^{rbm,i} = h(\boldsymbol{x}_t^{rbm,i}) + \boldsymbol{v}_t^{rbm} \quad (17)$$

$$z_t^{rbm,i} = \begin{pmatrix} \text{Loc}(\hat{f}_t^0)^i \\ \text{Loc}(\hat{f}_t^1)^i \\ \vdots \\ \text{Loc}(\hat{f}_t^{M-1})^i \end{pmatrix} = \begin{pmatrix} exp\left(\hat{p}_t^{rbm,i}\right) \text{Loc}(f_{init}^0) \\ exp\left(\hat{p}_t^{rbm,i}\right) \text{Loc}(f_{init}^1) \\ \vdots \\ exp\left(\hat{p}_t^{rbm,i}\right) \text{Loc}(f_{init}^M) \end{pmatrix} \quad (18)$$

where $exp\left(\hat{p}_t^{rbm,i}\right) \in SE(3)$ is the homogeneous transformation obtained from the predicted rigid body pose, $\text{Loc}(\hat{f}_t^m)^i$ is the predicted feature location based on that predicted pose, $i \in \{I, II, III\}$ indicates one of the three alternative predictions, and $\text{Loc}(f_{init}^m)$ are the homogeneous coordinates of the 3D 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. Features that are detected after the body starts to move are first transformed back to where they were when the body was initially detected. We use the inverse of the body pose at the feature detection step and increase the uncertainty of the feature based on the uncertainty of the body pose. While the result of the row operations in the matrix of Equation 18 are also homogeneous coordinates, i.e. $(x, y, z, 1)^T$, we remove the constant 1 at the end and stack the remaining elements to obtain a $3M$ dimensional predicted measurement vector.

The noise associated to the measurement, $\boldsymbol{v}_t^{rbm}$, is a zero mean Gaussian distributed variable with $3M \times 3M$ covariance matrix, R_t. We assume that the measured locations are uncorrelated between features and that the uncertainty about the measured location of a feature f^m is defined by the measurement covariance:

$$R_t^m = f_\sigma(\lambda_t^m, z_t^m) \cdot I_{3\times 3} \quad (19)$$

where z_t^m is the z-coordinate of the feature, λ_t^m is the salience of the feature given by the KLT tracker, and $f_\sigma(\lambda_t^m, z_t^m)$ is a function that characterizes the uncertainty of a feature location based on its depth and its salience.

For a measured point feature at the location (x, y, z) with salience value of λ (see Shi and Tomasi (1994) for the definition of the saliency of a point feature), the uncertainty about its location, $f_\sigma(\lambda, z)$, is defined as:

$$f_\sigma(\lambda, z) = \max\left(\sigma_{min}, \frac{\alpha_\lambda}{\lambda - \lambda_{min}^{tracking}}, \alpha_z z^2\right) \quad (20)$$

where σ_{min} is the minimum uncertainty about a point's feature location, $\frac{\alpha_\lambda}{\lambda - \lambda_{min}^{tracking}}$ assigns higher uncertainty to features tracked in visual areas with low texture ($\lambda_{min}^{tracking}$ is the saliency threshold to continue tracking a point, see Section 2.2.1), and $\alpha_z z^2$ represents the quadratic dependency of the measurement uncertainty to the depth of the point in f_σ. This dependency is based on the statistical analysis of RGB-D sensors by Khoshelham and Elberink (2012).

The three alternative process models in our RBF generate different state and measurement predictions, $\text{Loc}(\hat{f}_t^m)^i$. To select the best prediction, the Bayes filter measures the distance between predicted and measured location per feature and adds a vote to the state prediction that leads to the shortest distance. The prediction with most votes is selected as the best state prediction and is used for correction.

Our system exploits in the measurement update the assumption that the environment is composed of rigid bodies and that the motion of a rigid body is governed by known kinematic relationships. The generated predictions for the feature locations (which our system propagates down to the feature motion estimation level) are thus informed by these two priors.

Extended Kalman Filter for Rigid Body Motion: The estimation of rigid body motion as presented above presents non-linearities in the state and measurement updates. We implement an extended Kalman filter for the recursive solution of this state estimation problem. Appendix B summarizes the EKF and its most relevant equations. The EKF linearizes the system using a first-order Taylor expansion of the forward and measurement models around $\boldsymbol{x}_{t-1}^{rbm}$, the previous state estimate. The EKF uses the linearization to estimate the correction and the covariance matrix of the next state of the rigid body.

The linearization of the first forward model corresponds to the following Jacobian matrix:

$$F_t^{rbm,I} = \left.\frac{\partial f^{rbm,I}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,I}} = \begin{pmatrix} \text{Ad}_{\Delta_t \mathrm{v}_{t-1}} & \Delta_t I_{6\times 6} \\ 0_{6\times 6} & I_{6\times 6} \end{pmatrix} \quad (21)$$

where $\text{Ad}_{\Delta_t \mathrm{v}_{t-1}}$ is the adjoint transformation for $se(3)$ corresponding to the kinematic update based on the estimated velocity[‡].

The linearization of the second forward model (assuming an abrupt break event in the motion) corresponds to the following Jacobian matrix:

$$F_t^{rbm,II} = \left.\frac{\partial f^{rbm,II}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,II}} = \begin{pmatrix} I_{6\times 6} & 0_{6\times 6} \\ 0_{6\times 6} & 0_{6\times 6} \end{pmatrix} \quad (22)$$

In practice, what the RBF does if the second forward model generates the best predictions is to consider only the pose of the rigid body, p_t, as the state of the body and correct it.

The third forward model (based on information from the next level) is independent of the state of the filter. To correct the state, we use the linearization of the first forward model, $F_t^{rbm,III} = \left.\frac{\partial f^{rbm,I}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,III}}$, using the prediction from the higher level as a different point for the linearization. If this point is closer to the true mean of the posterior, the result of the EKF correction based on the third prediction generates a better approximation of the true current state.

The linearization of the measurement model with respect to the state yields the following Jacobian matrix:

$$H_t^{rbm} = \left.\frac{\partial h^{rbm}}{\partial \boldsymbol{x}^{rbm}}\right|_{\hat{x}_t^{rbm,i}} = (H_t^{f^0,i}, H_t^{f^1,i}, \ldots, H_t^{f^M,i})^T \quad (23)$$

where $H_t^{f^m,i}$ correspond to the linearization of the model for an individual feature of the form around the best predicted state from the model $i \in \{I, II, II\}$:

$$\begin{pmatrix} & 0 & \text{Loc}_z(\hat{f}_t^m)^i & -\text{Loc}_y(\hat{f}_t^m)^i & \\ I_{3\times 3} & -\text{Loc}_z(\hat{f}_t^m)^i & 0 & \text{Loc}_x(\hat{f}_t^m)^i & 0_{3\times 6} \\ & \text{Loc}_y(\hat{f}_t^m)^i & -\text{Loc}_x(\hat{f}_t^m)^i & 0 & \end{pmatrix} \quad (24)$$

Sequential processing of measurements: The EKF solution involves inverting the covariance of the innovation, $S_t = H_t \hat{P}_t H_t^T + R_t$, to compute the Kalman gain. In our case, this involves the inversion of a $3M \times 3M$ matrix, with M the number of features assigned to the RBF. If M is large (many features are associated to the rigid body), the matrix inversion can be computationally expensive and affect the online capabilities of our system. Therefore, based on the assumption that the measured feature locations are uncorrelated to each other, we process the point features sequentially and avoid the costly inversion of the full matrices (Bar-Shalom et al. 2001).

In the sequential procedure, we initialize the corrected state and its covariance with $\boldsymbol{x}_t^0 = \hat{\boldsymbol{x}}_t$ and $P_t^0 = \hat{P}_t$. We then compute a correction based on each feature m. For the feature m, the corrected state becomes (we dropped some super-indices to make the equations easier to read):

$$x_t^m = x_t^{m-1} + K_t^m(\text{Loc}(f_t^m) - \text{Loc}(\hat{f}_t^m)^i) \quad (25)$$

$$P_t^m = (I - K_t^m H_t^{f^m}) P_t^{m-1} \quad (26)$$

$$K_t^m = P_t^{m-1}(H_t^{f^m})^T [H_t^{f^m} P_t^{m-1} (H_t^{f^m})^T]^{-1} \quad (27)$$

In Equation 25, the first part of the state vector x represents a 6D pose in Euclidean space. To correctly integrate the corrections of this first part of the state vector, we use the composition of poses, $\oplus$, instead of the normal vector sum. Due to the non-linearity of the problem the order of integration could slightly modify the final estimate. However, for the small inter-frame changes of poses, the difference is negligible.

The final correction is then:

$$x_t = x_t^M \quad (28)$$

$$P_t = P_t^M \quad (29)$$

[‡]We can give an intuition of the role of the adjoint transformation in this linearization. The adjoint transformation transforms twist velocities from one reference frame to another. In the linearization of the first forward model, the adjoint plays another role. Here, the adjoint transforms the uncertainty about the pose of the body from the frame of the previous estimate to the frame of the predicted body pose. In general, when we transform a random 6D pose that we assume to be Gaussian distributed with mean p and covariance defined in the tangential Lie space $\Sigma_p \in \mathbb{R}^{6\times 6}$ applying a second pose p', the result will not be Gaussian distributed. However, we can approximate the result to a Gaussian distribution of mean $p_{new} = p \oplus p'$ and covariance $\Sigma_{new} = \text{Ad}_p \Sigma_p \text{Ad}_p^T$, correct to first order (Barfoot and Furgale 2014).

Estimation of Multi-Body Motion: To track the motion of multiple rigid bodies, we have to match point features to the corresponding rigid body RBF and use them to update the state of the filters. This matching process is called data-association. We associate features to those rigid bodies that best predict their motion. We measure the Euclidean distance between the observed and the predicted feature location from the filters and assign the features to the filter with the lowest prediction distance. This hard data association processes each feature independently of the others. Exploring other data association techniques with global score (e.g. multi-model random sample consensus, RANSAC Fenzi et al. (2012)) or with soft assignment (e.g. multi-hypothesis tracking, MHT Reid et al. (1979) or joint probabilistic data assocation, JPDA Fortmann et al. (1980)), while an interesting research direction, is out of the scope of this work.

We assume that the existing filters cannot predict the motion of a feature if all predicted locations are further than d^f_{max} from the measured location. If the motion of a set of features cannot be predicted accuately by any of the existing rigid body Bayes filters, it could be necessary to instantiate a new filter. We instantiate a new filter if a set of non-assigned features move coherently. To evaluate if a set of non-assigned features move coherently, we use RANSAC to find a rigid body transformation describing their motion. If a rigid body transform explains the motion of at least $f_{\min}$ features, a new RBF is created, using this rigid body transform as the initial state. Based on this procedure, the proposed system works for an arbitrary number of moving rigid bodies in the scene, including objects that appear in the scene, as long as $f_{\min}$ visual features can be tracked on each body.

The overall perceptual process begins with a single Bayes filter that represents the static background. We assume that the static background does not move ($\eta_{bg} = 0_{6\times1}$), although it would be easy to integrate an algorithm that provides motion estimates of the camera with respect to the static background (Nistér et al. 2004; Forster et al. 2014). New detected point features are initially assigned to the static rigid body, until they begin to move and their location cannot be predicted by the static background filter. These features are either assigned to another filter (if it can predict their motion) or used to create a new filter for a newly moving body.

2.2.3 Estimation of Kinematic Model: The third level of our system estimates and tracks the kinematic model of the scene based on the motion of rigid bodies obtained on the previous estimation level. We assume a pair of rigid bodies to be related in one of the 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 random state variable $\boldsymbol{x}^{joint}_t$. We instantiate and maintain one RBF of every type for each pair of rigid bodies in the scene.

The measurements $z^{joint}_t \in se(3)$ are obtained from the previous estimation level and correspond to the change in relative pose (in exponential coordinates) between the two rigid bodies attached to the joint, which is defined with respect to one of the bodies:

$$\boldsymbol{z}_t = {}^{parent}_{child}\Delta p_t + \boldsymbol{v}^{joint}_t = {}^{parent}_{child}p_t \ominus {}^{parent}_{child}p_{init} + \boldsymbol{v}^{joint}_t \tag{30}$$

The body that acts as reference is called *parent link*, and the second body is called *child link*. In the previous equation, $\ominus$ represents the subtraction between poses, ${}^{parent}_{child}p_t$ is the current pose of the child link with respect to the parent link (in parent link frame), and ${}^{parent}_{child}p_{init}$ is the pose of the child link with respect to the parent link when the joint starts to be tracked. The terms parent and child link refer to the common terminology for tree structures of kinematic mechanisms in the literature.

The covariance, R^{joint}_t, of the measurement model noise, $\boldsymbol{v}^{joint}_t$ is also obtained from the previous level:

$$\begin{aligned} R^{joint}_t = {}& {}_{parent}\mathrm{Ad}(\Sigma_{{}_{parent}p})\ {}_{parent}\mathrm{Ad}^T \\ & + {}_{parent}\mathrm{Ad}(\Sigma_{{}_{child}p})\ {}_{parent}\mathrm{Ad}^T \end{aligned} \tag{31}$$

We apply the adjoint operator here to transform covariances between reference frames (Barfoot and Furgale 2014). In this case, we transform the covariance of the pose of both parent and child link from the sensor reference frame to the reference frame of the parent link.

In the following, we will explain the state representation, prediction, measurement update, and EKF solution 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 not only enable the estimation and tracking of kinematic models but also the prediction of the next state of the previous 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.

Prismatic Joint Estimation: The state of a prismatic joint is parametrized by the orientation of its axis (azimuth ϕ and elevation θ in spherical coordinates), its joint variable $q^p \in \mathbb{R}$ (translation along the joint axis), and the velocity of the joint variable $\dot{q}^p \in \mathbb{R}$, which we represent with a multidimensional Gaussian distributed random variable. In the prediction step, we use the joint velocity to update the joint state.

To predict the change in the pose of the child link relative to the parent link, we use the following measurement model:

$$\hat{z}^{joint,p}_t = \begin{pmatrix} 0_{3\times1} \\ q^p \cdot o^p \end{pmatrix} \tag{32}$$

where $0_{3\times1}$ is a three dimensional null vector that indicates that the orientation between the links is constrained by the prismatic joint, and $o^p \in \mathbb{R}^3$ is the axis orientation (unit vector) estimated from ϕ and θ as $o^p = (\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta))^T$.

Extended Kalman Filter for Prismatic Joint Estimation: While the forward model of the prismatic joint is linear, the measurement model is non-linear with respect to the joint parameters. Therefore, we implement an EKF to correct the state based on the acquired measurement. The matrix of derivatives of the measurement model with respect to the state, $H^{joint,p}_t$, is detailed in the Appendix (Section B).

Revolute Joint Estimation: The state of a revolute joint is parametrized by the orientation of its axis (azimuth ϕ and elevation θ in spherical coordinates), a point on the axis $p^r \in \mathbb{R}^3$, its joint variable $q^r \in \mathbb{R}$ (rotation about the joint axis), and the velocity of the joint variable $\dot{q}^r \in \mathbb{R}$, which we represent with a multidimensional Gaussian distributed random variable. We use the joint velocity to predict the next joint state as a process model.

To predict the change in pose of the child link relative to the parent link, we use the following measurement model:

$$\hat{z}_t^{joint,r} = \begin{pmatrix} q^r \cdot o^r \\ t^r \end{pmatrix} \tag{33}$$

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

Extended Kalman Filter for Revolute Joint Estimation: As in the case of a prismatic joint, the filter for the revolute joint model contains a non-linear measurement model. We implement an EKF that linearizes this model and corrects the state from the acquired measurement. The linearization of the measurement model, $H_t^{joint,r}$, is detailed in the Appendix (Section C).

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 change in the relative pose between bodies, i.e. $\hat{z}_t^{joint,rigid} = 0_{6\times1}$. Because there are no parameters to estimate, we do not need to implement an EKF for this type of joint.

Estimation of Multi-Type Kinematic Model: After evaluating the RBF of every type for each pair of rigid bodies, we select the RBF that is most consistent with the observed rigid body motion. We base this selection on the likelihood of the measurements given by the estimated models. The likelihood of the observed data is defined as

$$p(z_t^{joint}|M, \boldsymbol{x}_t^{joint,M}) = \mathcal{N}(z_t^{joint}; \hat{z}_t^{joint,M}, \hat{R}_t^{joint,M}) \tag{34}$$

where M are the considered joint models, $M \in \{\text{Prism}, \text{Rev}, \text{Rigid}\}$, $\boldsymbol{x}_t^{joint,M}$ is the current estimate of model M, and $\hat{z}_t^{joint,M}$ and $\hat{R}_t^{joint,M}$ are the predicted measurement mean and covariance, respectively (we assume a constant predicted uncertainty for the predictions from the rigid joint).

Instead of selecting the model that best explains only the latest measurement, we select the one that explains all past measurements. This makes the selection of the most likely joint more stable. We consider that large measured relative motion is more informative to find the most likely joint, as large motions are more difficult to predict randomly. Therefore, we assign a weight to the estimated likelihood at each step proportional to the amount of change in relative motion between links, which is measured as the norm of the vector of exponential coordinates $\|z_t^{joint}\|$, and compute the mean of these weighted likelihood values over the trajectory.

We select the model with the maximum accumulated weighted likelihood as the joint that best explains the motion between a pair of bodies. We consider that none of the models can explain the motion with sufficient reliability if any of their accumulated weighted likelihoods is over a minimum threshold, L_{disc}. In this case, 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. As joints are always determined considering only pairs of rigid bodies, our system can naturally determine the kinematic model of branching mechanisms and closed kinematic chains.

2.3 Experiments

We conducted four sets of experiments. In the first set, we study the sensitivity of the system to the number of tracked features, N. Additionally, we evaluate if the computation time, the accuracy, and the robustness depend on N. To measure the accuracy, we compute the error between the estimated rigid body poses and the ground truth obtained with a motion capture system (Motion Analysis 2017). Furthermore, we evaluate the contribution of the predictions from higher levels in the performance of the system.

In the second set of experiments, we evaluate the performance of the online IP system with different articulated objects. We measure the robustness, quality, and convergence of the kinematic model estimation by comparing to the ground truth. To obtain the ground truth of the joint parameters, we placed artificial markers that are not used by the system to estimate the kinematic model. We then manually measured the joint parameters in the RGB-D stream using the markers.

In the third set of experiments, we test our system in scenarios where offline systems fail. Additionally, in the fourth set, we make use of the online abilities of the system to control the motion of a robot, closing the loop between perception and action. This demonstrates that the perceived information is relevant for the robot manipulation of DoF.

In all experiments, the input is an RGB-D stream, provided either by a Kinect or a Carmine RGB-D sensor, delivering images of 640×480 pixels at 30 Hz. The articulated objects are of different size, color, texture, and have different kinematic structures (number and type of joints). The only constraint for the objects is that they have some visible texture. Moreover, we vary lighting conditions and the relative pose between the objects and the sensor. In these experiments, we use N between 150 and 250. Our system computes at a frame rate of 30 frames per second, executing real-time on an Intel Xeon E5520 PC at 2.27 GHz. Table 4 in Section 7 contains the value of the most relevant parameters used in the experimental evaluation. The multimedia attachment 1 shows experiments of this section.

2.3.1 Parameter Sensitivity Analysis:

The computational complexity of our system increases with the number of features N. This parameter is involved in the feature motion and the rigid body motion estimation. Therefore, we first evaluate if the number of features influences the computation time at these levels.

The first row in Fig. 3 shows the computation time spent on feature motion for different values of N. We observe that the computation at this level is independent of the number of tracked features. In most of the iterations, our system spends approximately 20 ms in the computation of the motion of the features. This time includes the tracking

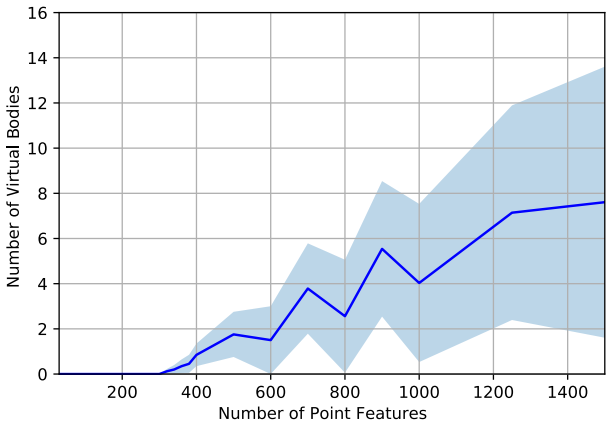


Figure 4. Generation of virtual (wrong) rigid body hypotheses as a function of the number of tracked features N; imposing a large number of features to track increases the amount of noisy trajectories and the probability of creating a virtual body hypothesis

of the features with the KLT algorithm (Tomasi and Kanade 1991) and the detection of new features to maintain their number N, using the approach by Shi and Tomasi (1994). If the detection process does not generate a sufficient number of new features, our system repeats feature detection. Each detection increases the time by approximately 10 ms. In some iterations, our system successfully tracks all features and does not need to detect new features. These iterations last approximately 10 ms. The computation time of this level allows us to estimate at 30 fps (approximately 33.33 ms between frames).

The second row in Fig. 3 depicts the computation time of the rigid body motion for different values of assigned features. The computation time increases from 2 ms to 4 ms when the associated features increase from 30 to 320. These times do not restrict the performance of the system.

So far, we did not find any strong limitation on the number of tracked features N from the analysis of the computation times. However, if the number of tracked features is high, some of them are placed in less distinctive locations. Features in non-distinctive locations will produce noisy trajectories. When the proportion of noisy trajectories grows, subsets of features will randomly move in a coherent manner and create virtual rigid bodies. Even though the "life-span" of these virtual bodies is short, we consider them harmful for the perceptual process and we will try to reduce their appearance. Fig. 4 depicts the number of virtual (error) rigid bodies created as function of the tracked features N. When N is under 300, there are almost no virtual bodies. Over this value, the number of virtual bodies increases.

While in theory, these noisy features should be rejected based on the minimum required quality for tracking, $\lambda_{min}^{tracking}$, in practice, deciding this threshold is difficult: if the value is high, we only track features in points with very strong contrast changes, and if the value is low, we accept features in non-pure distinctive points (e.g. shadows). We decided to set up a relatively low value from an experimental evaluation on different surfaces to prime perceiving motion. We relying on the predictions from the higher level (based on rigid body tracking) to "clean up" the noisy tracked features, even in the case of virtual bodies.

In the last experiment, we evaluate the contribution of the predictions from other levels to the overall performance of the system. To evaluate this contribution, we compare the accuracy in the estimated rigid body motion in the fully integrated system (with predictions) versus the system without predictions from kinematics to rigid body motion estimation, or without predictions from rigid body motion to feature motion estimation. In the experiment, the system perceives the motion of a drawer from human interactions during 6.2 s. The ground truth of this motion is obtained with a motion capture system Motion Analysis (2017). Fig. 5 shows the results of the experiment.

The combined system using all predictions outperforms the other two variants. Significantly, the predictions about next feature locations help to reject noise, especially when the number of tracked features is large. In this case, many of the features are noisy and of low quality: the predictions from the rigid body level helps to reject them and track them in a more stable manner.

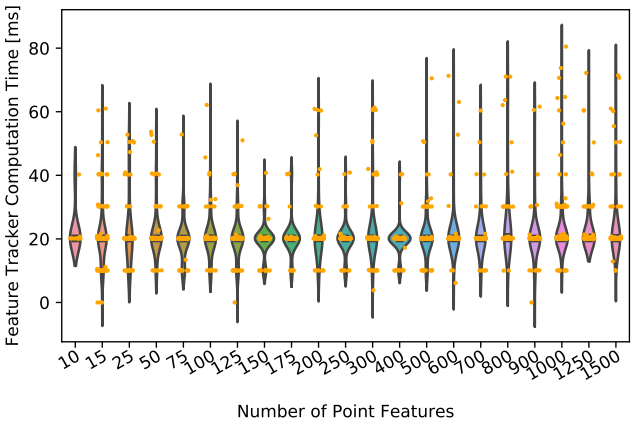
2.3.2 Experimental Evaluation: We measured the accuracy and convergence of our online IP system for kinematic properties on four articulated objects. Fig. 6 shows initial, intermediate (after 1 s), and final frames of these experiments. Additionally, the figure includes graphs of the estimation error 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. We recorded all interactions and made them publicly available[§]. In the following, we discuss each of the experiments from Fig. 6.

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. The 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.

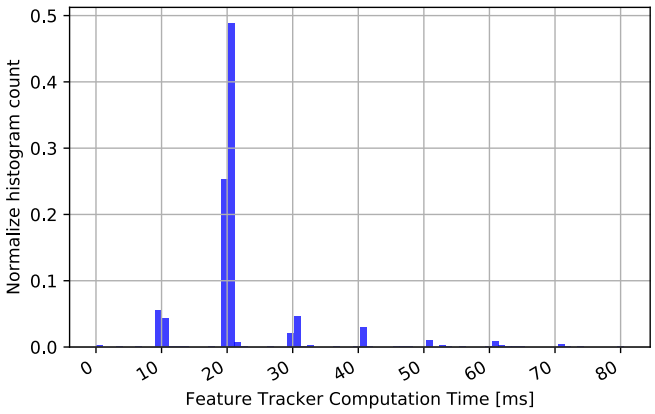
Umbrella Experiment: The umbrella is extended by 40 cm in a motion lasting 10 s. The joint is continuously estimated correctly as prismatic. The features on the umbrella are correctly assigned. Some features on the hand are also assigned to the umbrella as they move coherently with it. The error of the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

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 system 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 system 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

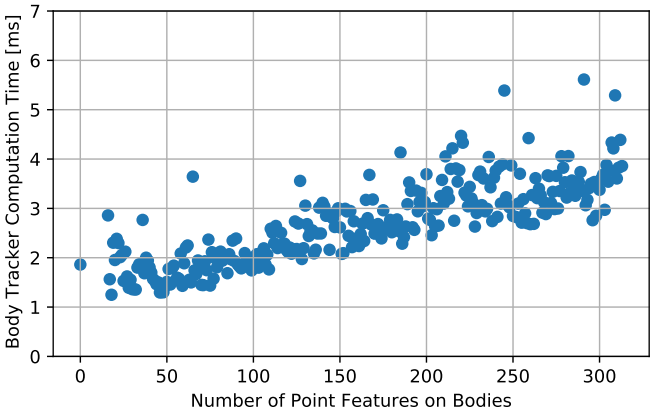
[§] https://tinyurl.com/onlineIPdata



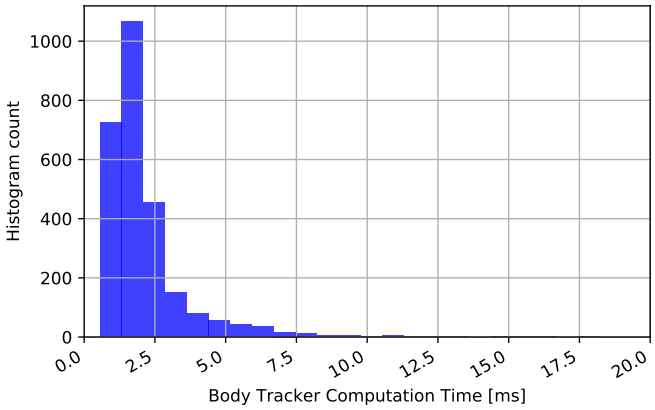
(a) Distribution of computation time as function of the number of tracked features, N; orange dots indicate the computation time of each iteration; the computation time is independent of the number of features N



(b) Histogram of computation time per frame; most iterations require around $20\,\mathrm{ms}$ to track and detect features to maintain N; the detection of new features requires around $10\,\mathrm{ms}$ that cause the periodically spaced peaks

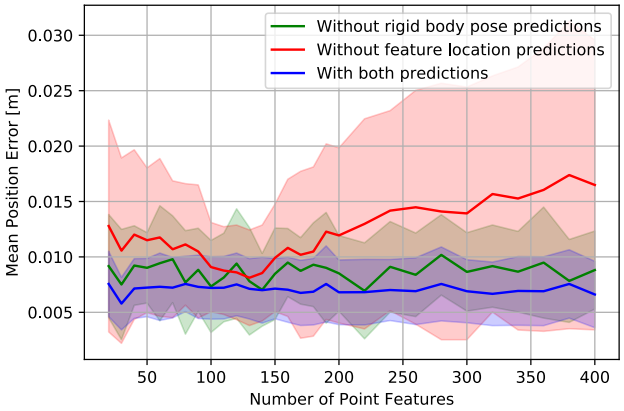


(c) Computation time for the rigid body motion level at each iteration and associated number of features; the time slowly increases with the number of features assigned to a rigid body

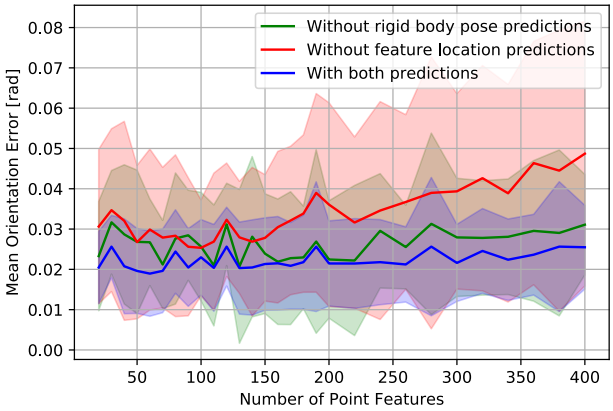


(d) Histogram of computation times at the rigid body level; most of the iterations consume approximately $2\,\mathrm{ms}$

Figure 3. Evaluation of the computation time at the feature motion level (first row) and rigid body motion level (second row)



(a) Mean error in position over the entire video sequence $(6.2\,\mathrm{s})$ for different number of tracked features



(b) Mean error in orientation over the entire video sequence $(6.2\,\mathrm{s})$ for different number of tracked features

Figure 5. Error in the estimated rigid body pose; the nominal system (with predictions from kinematics to rigid motion estimation and from rigid motion to feature motion estimation) outperforms the two variants without one of the two predictions: without predictions about the next rigid body pose or without predictions about the next feature location

accurate value. The joint between the base and the forearm is initially classified as revolute, but the system quickly detects that there is no direct connection (disconnected joint) between them. The estimation error of the first revolute axis

(shown in the graph) remains under 6° for orientation and $5\,\text{cm}$ for position; for the second joint, the error remains under 8° and $8\,\text{cm}$ 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. To solve this problem, we attached checkerboards to it. In this experiment, the RGB-D sensor points parallel to the joint axes of the robot to simplify ground truth estimation. The experiments demonstrate the system's ability to determine multiple DoF of a kinematic chain at the same time.

Human Head Experiment: The system estimates the neck joint of a human shaking his head. The human rotates his head 100° in $5\,\text{s}$. 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 $4\,\text{cm}$. The RGB-D sensor points 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 system on large semi-rigid articulated bodies.

2.3.3 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 and incremental IP system. Existing offline non-incremental methods will fail in the following scenarios.

Disappearing Features: The motion of the object may cause all features obtained at the beginning of the motion to disappear by moving out of the visual field or simply due to any 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 Fig. 7) to demonstrate that the incremental nature of our online IP system aims to overcome this problem. We rotate the globe 300° in $31\,\text{s}$ and open the poster hanger $70\,\text{cm}$ in $13\,\text{s}$. Our online system 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).

Appearing Objects: The articulated object may not be visible at the beginning of the analysis. To demonstrate how our online IP system can address this, we use a book in a cabinet and a Pioneer mobile base (see Fig. 7). The cabinet has to be opened to perceive the book. We then open the book 30° in $3\,\text{s}$. 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 $82\,\text{cm}$ in $22\,\text{s}$ 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.

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 our online IP system overcomes this problem of some offline IP methods, we experiment with a cabinet door and a drawer. The drawer is opened and closed ($50\,\text{cm}$ of accumulated motion) in $6\,\text{s}$, and the door is opened and closed (80° of accumulated motion) in $7\,\text{s}$. The proposed online IP system accurately estimates the kinematic model. The model remains converged after the object returns to its initial configuration.

2.3.4 Monitoring Interaction With Online IP: One of the main advantages of an online IP system is the ability to use the kinematic model to control the robot's interaction with the environment. By using the information for an ongoing interaction, we demonstrate that the perceived information is relevant for the mechanical manipulation of DoF. In this section, we demonstrate the utility of online perception 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 $5\,\text{cm}$ 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. Fig. 7 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 system 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 $5.07\,\text{cm}$. 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 $7.03\,\text{cm}$ with a maximum of $49.71\,\text{cm}$ for a failed trial. Without this value the mean position error is $2.28\,\text{cm}$ (under the $5\,\text{cm}$ threshold).

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 $15\,\text{cm}$. 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 $15.55\,\text{cm}$ and the maximum and minimum are $15.9\,\text{cm}$ and $15.2\,\text{cm}$, 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 our online IP algorithms 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. Additionally, we showed that

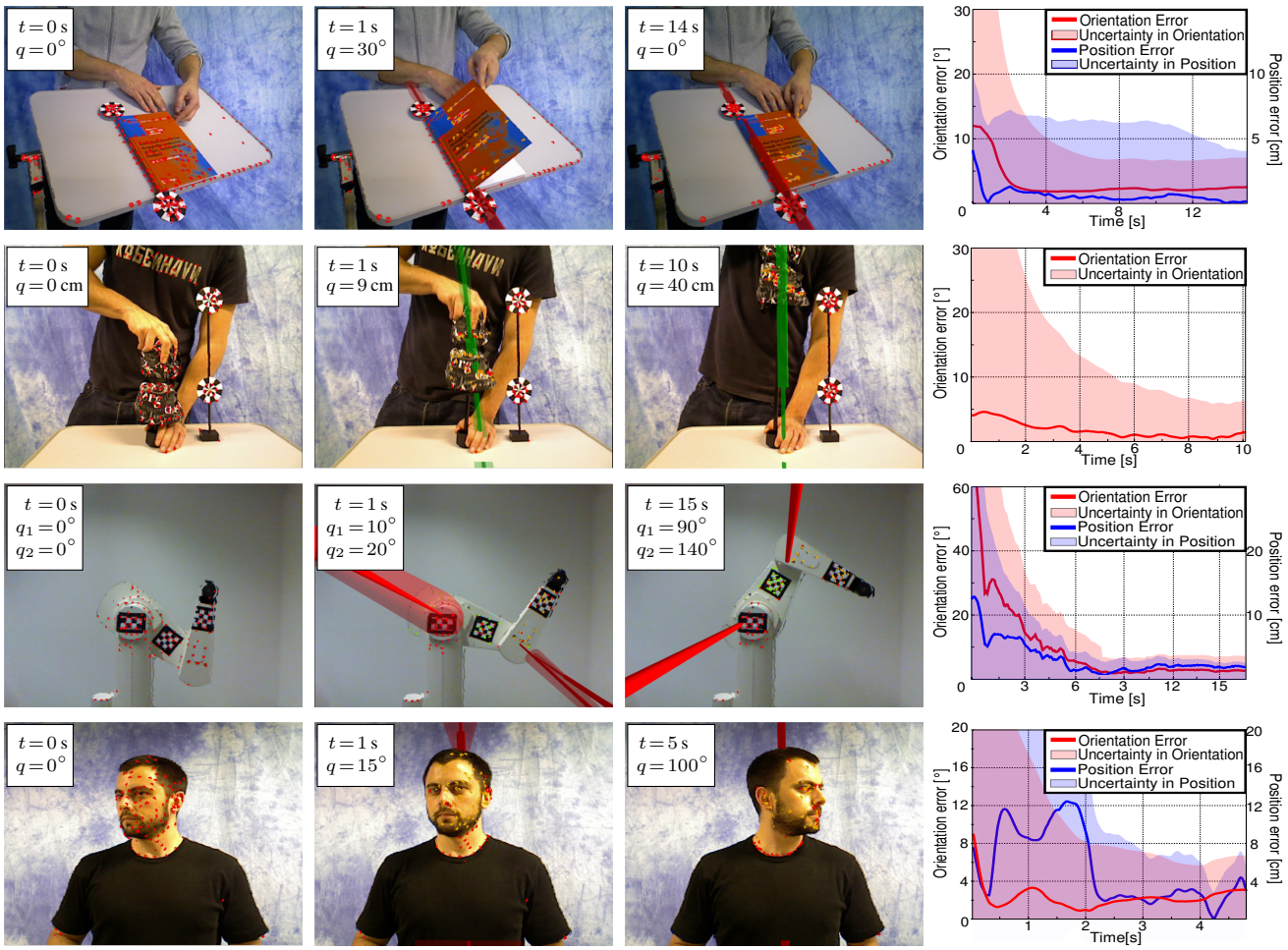


Figure 6. Experiments with online IP (each row represents a different experiment): initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model, including error plot (fourth column) of joint configuration estimation, relative to ground 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

the online estimation of joint values can be used to monitor and attain manipulation goals, expressed in terms of specific joint configurations.

2.4 Limitations

The presented system necessitates motion in the scene to perceive kinematics. This motion can be produced by robot interaction, by others in the scene, or by the object itself, but there must be motion to create information-rich visual signals.

In our system, motion is perceivable if the objects have sufficient trackable texture. This is a consequence of the use of an LKT salient point feature tracker in the first recursive process. Additional limitations inherited from the LKT tracker are the need of relatively stable lighting conditions, bounded object acceleration, and clear visibility (no occlusions). We will show how to alleviate these limitations of the visual system by incorporating other sensor modalities in Section 4.

The presented system handles four types of pairwise kinematic relationships: revolute and prismatic joints, rigid connections, and the state of being disconnected. These are the most common kinematic constraints in human environments. Other types of joints can also be composed from these. The system could easily be extended to include additional joint filters.

3 An Algorithmic Architecture for Online Interactive Perception

In the previous section, we presented an online system to visually perceive the kinematics of articulated objects. In this section, we will discuss the algorithmic architecture underlying the previous system and analyze its most relevant features. Our hypothesis is that this algorithmic architecture is a general solution that can be applied to other online interactive perception problems. To validate this hypothesis, we will present and evaluate in the next sections (Section 4 and 5) two new perceptual systems implementing the proposed algorithmic architecture.

Structurally, the previously presented system is *a network of coupled recursive estimation processes*. Each process of our system estimates recursively an unobservable state based on the previous state, information from other processes, and the most recent observation. Based on this recursive procedure, our system extracts online information from

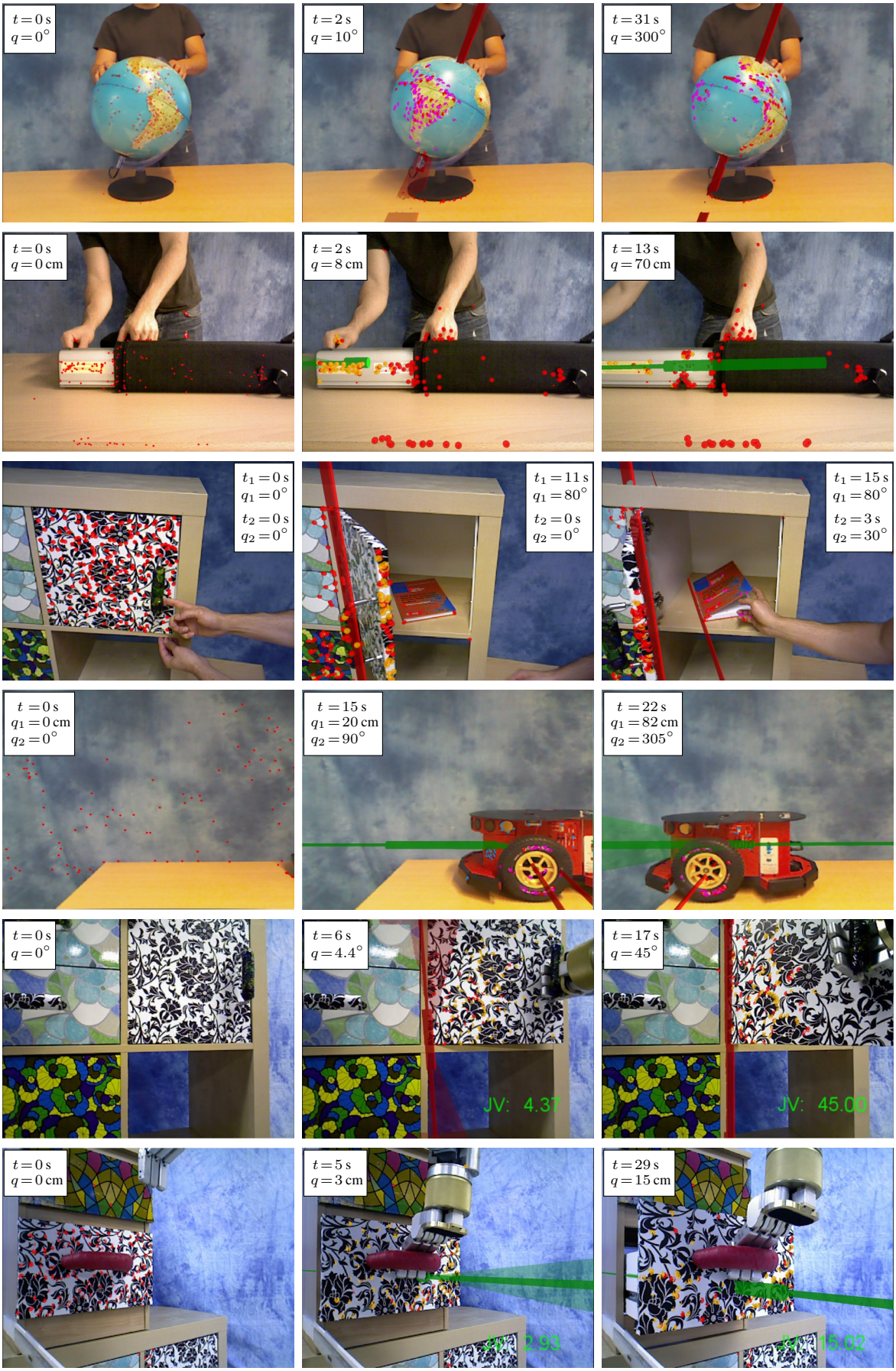
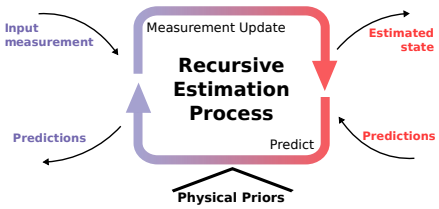
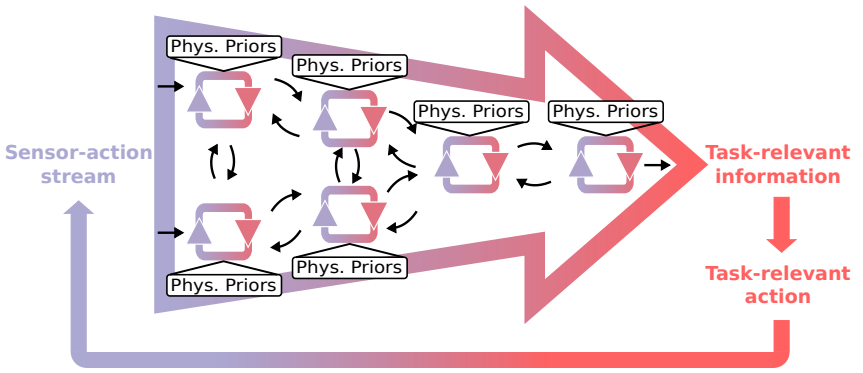


Figure 7. Experiments with online IP (each row represents a different experiment): initial (first column), intermediate (second column), and final frame (third column) of the estimation of the kinematic model; the insets show the time t and the estimated joint variable q; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation



(a) One coupled recursive estimation process; inputs: measurements and predictions from other processes; outputs: estimated states and predictions to other processes; the process "enriches" the measurements with prior information: physical priors, temporal priors (from recursion) and priors from other processes



(b) Our proposed algorithmic architecture for online interactive perception problems: a network of coupled recursive estimation processes; the coupled processes interpret the sensor-action stream based on task-specific physical priors, temporal priors (recursion), and information from other processes; the sensor-action signals (left) are "enriched" with the priors and transformed into a stream of task-relevant information (right); the stream of perceived information can be used to monitor, steer, generate, and plan task-relevant actions that affect the input sensor-action stream

Figure 8. Schema of one coupled recursive estimation process (*left*) and the proposed algorithmic architecture for online interactive perception (*right*) resulting from coupling multiple recursive processes

changing sensor signals. We will discuss in the following the most notable features of this algorithmic architecture.

First, the architecture encodes known regularities about the physical processes in the environment, i.e. physical priors. These priors (e.g. kinematics, rigid body assumption, projective geometry) are encoded in the models that the recursive estimation processes use to predict and update the perceived information online. The physical priors "enrich" the sensor signals and allow the recursive process to interpret them as evidence of the modeled physical processes while being general enough to apply to many unstructured environments.

Second, the architecture promotes the coupling between recursive processes to solve the original perceptual problem. To couple the processes, the architecture intercommunicates them. Estimated states from one process are passed to other processes as virtual measurements. Predictions from one process are passed to others as prior-informed guesses of next states; these predictions are based on additional information and are, in general, closer to the true unobservable state.

Finally, our algorithmic architecture integrates interactions as part of the perceptual process with two main purposes (Bohg et al. 2017). First, the architecture relies on interactions (from the robot or from another agent) to create information-rich signals and reveal hidden information, e.g. the kinematics of an articulated object. Second, the architecture leverages information about the interaction to interpret changes in the sensor signals. We will see in the following sections perceptual systems based on the proposed architecture that use knowledge about the robot interaction (in the form of proprioceptive signals) to interpret the changes in the environment and perceive kinematic and dynamic properties.

Methodologically, to apply the proposed architecture, we need to factorize the original perceptual task into simpler subtasks that we can solve using coupled recursive processes. This imposes a constraint in the original perceptual task: it must be *weakly decomposable* (Simon 1996). We will discuss this constraint further in Section 6.

Fig. 8 illustrates the proposed algorithmic architecture for online interactive perception problems. We depict the recursive processes that tackle perceptual subtasks as loops. The processes integrate sensor signals and information about the interaction (sensor-action stream), physical priors, previous estimates (recursion), and information from other processes to continuously extract patterns of information. The output can act again as an input to another recursive estimation process. The outputs can, therefore, be seen as signals of a virtual sensor. The combination, integration, and sequencing of these estimation processes lead to a general flow of information from sensor-action data (blue, on the left) to task-relevant information (red, on the right). The idea is to *incrementally process and interpret the sensor streams through cascades of coupled estimation processes until the necessary perceptual information has been extracted* robustly and efficiently. The coupling between these processes indicates that different but correlated information can be leveraged to further increase the robustness and efficiency. Based on the perceived information, the robot can monitor, control, and plan actions, thus, closing the loop and affecting the input to the processes.

3.1 Related Work

Other authors have previously studied related algorithmic features for artificial perception. In their book, Clark and Yuille (1990) discussed many of the elements that we propose in our algorithmic architecture: prior-based interpretation of sensor signals, hierarchical structures, and inter-process coupling. Our proposed architecture further develops these ideas and combines them with recursive estimation to be applied to online interactive perception problems.

Closely related to our proposed approach are methods based on dynamic Bayesian networks, DBN (Dagum et al. 1992). DBN are probabilistic graphical models for inference

problems with temporal recursion. Variables in the model could be conditionally dependent to variables in the current or previous time steps. In the context of artificial perception, Liao et al. (2007) proposed a DBN with a hierarchical structure to perceive the goal of moving agents tracked with GPS signals, and Kwok and Fox (2004) proposed a related method to track moving objects. These approaches relate to ours in that they factorize the original problem into subproblems that can be solved recursively using information from other coupled subprocesses. However, in DBN, the information from other processes can only be used as Bayesian prior as enforced by the graphical model. Our solution relaxes this constraint and allows for other types of dependencies between subprocesses (e.g. search space initialization) at the cost of having to define this dependency for each coupling.

Another difference between DBN and our architecture is in the inference step. Inferring the state of all nodes of the DBN is a single computationally expensive inference step. Our architecture allows for the state of each subprocess to be computed separately, enabling higher parallelization. This separate computation could lead to a divergence in the estimations at different levels. We avoid this divergence with a tight coupling between levels based on the intercommunication between them at each estimation step.

Another related line of work is the approach by Rao and Ballard (Rao and Ballard 1999, 1997). They presented an architecture for object recognition based on coupled hierarchical recursive processes implemented as artificial neural networks. Their solution exploits information flowing bottom-up (from image to label) and top-down (from label to image). The neural networks of their architecture perform a prediction-update procedure similar to a Bayesian filter. While their method learns about the models (measurement, system) from the labeled data, we encode known physical laws to avoid having to learn the known regularities. The architecture by Rao and Ballard (1997, 1999) has only been successfully applied to simple object recognition problems, but not to online perceptual problems.

A similar idea was presented by Lee and Mumford (2003) to explain biological neural activity in the visual cortex. Additionally, the authors suggested that the bidirectional communication in the visual cortex is the result of a coupled hierarchical recursive architecture that can be modeled as particle filters in a belief propagation network (a non-parametric solution for a DBN). However, the authors used the model only to *explain* measurements in biological visual systems (e.g. latency and strength of neural responses) not to implement artificial ones.

In the previous section, we presented an instance of this algorithmic architecture for the visual perception of kinematics of articulated objects. In the following sections, we will present several instances of the architecture applied to the perception of kinematics from a multimodal sensor stream (Section 4) and dynamics from multimodal signals (Section 5). Apart from validating the generality of our algorithmic architecture, the perceptual systems of the following sections obtain task-relevant information about articulated objects that robots can use to manipulate the objects, as we show in the experimental evaluation.

4 Online Perception of Kinematics from a Multimodal Sensor Stream

In Section 2, we presented a system to perceive kinematics of articulated objects from visual signals. This system fails when the visibility of the objects is not ideal, e.g. due to bad lighting conditions, occlusions, or not enough trackable texture on the objects' surface. To overcome these limitations, we will present in this section a perceptual system for kinematics based on proprioceptive signals and integrate it with the visual system to generate a more robust and versatile multimodal system.

Proprioception refers to sensory information about the configuration of the robot's own body (kinesthetics) and the forces it exerts (haptics). Our robot (depicted in Fig. 9, left) obtains proprioceptive signals from a force-torque sensor on its wrist, from the air-pressure sensors monitoring the chambers of its pneumatic soft-hand, and from the joint encoders on its arm. In this section, we will combine these signals with visual information in an interactive perception system to build kinematic models of articulated objects. This multimodal system is based on the algorithmic architecture presented in the previous section.

4.1 Related Work

As, in this section, we present 1) a novel interactive perception system to build kinematic models of articulated objects based on proprioceptive signals and 2) a multimodal interactive perception system integrating proprioception and vision, we will cover these two areas of related work.

4.1.1 Perceiving Kinematic Models From Proprioception: Previous approaches have shown that the kinematic properties of an articulated object can be perceived from end-effector trajectories (Sturm et al. 2010a) or from the wrenches the robot applies during interaction (Karayiannidis et al. 2016). These methods are based on two assumptions that restrict their applicability: 1) The robot interaction causes motion only on one body connected through a joint to the static environment and 2) there is no translation between the end-effector and the moving body during the interaction. We will overcome these limitations using prior information from vision to estimate a model of the grasp and to perceive more complex kinematic structures.

4.1.2 Multimodal Interactive Perception: Recursive estimation has been previously applied to overcome limitations of unimodal robotic perceptual systems by integrating other sensor modalities. The common methodology is to update the state by fusing the multimodal signal into a single estimate (Ilonen et al. 2014; Hebert et al. 2012). With this methodology, the system cannot leverage information from one modality to help interpret the others. In contrast, our algorithmic architecture, while also fusing modalities, allows the perceptual system to exploit the results from one recursive filter as priors to extract more information in the others.

A similar cross-modal exploitation of the information was successfully applied by Garcia Cifuentes et al. (2017) to track a robot arm and an object from a multimodal stream. However, their method requires models of the object and

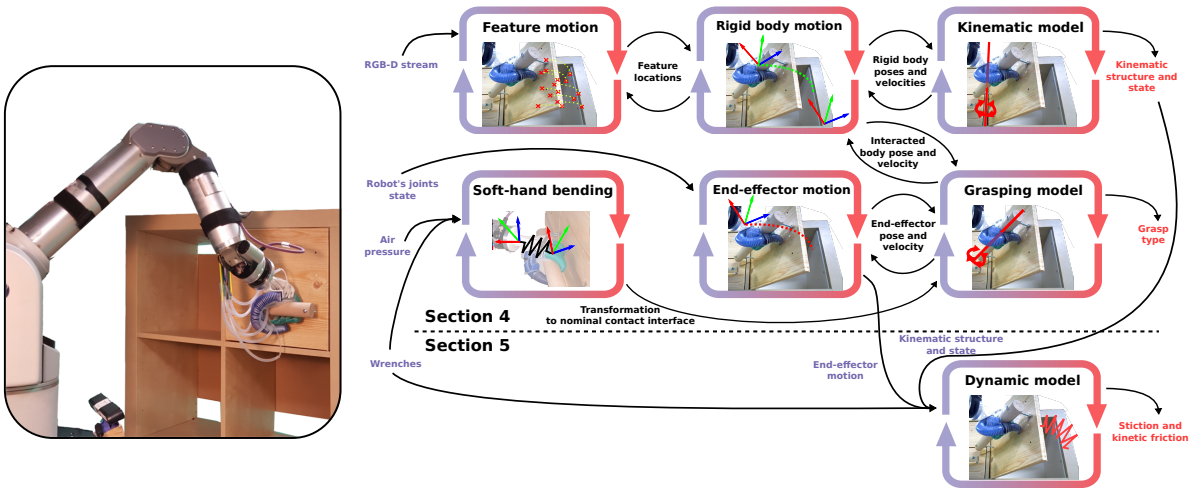


Figure 9. Our proposed system for interactive perception of kinematic and dynamic properties of articulated objects based on information flow between coupled recursive filters; *left*: our robot platform, a Barrett WAM 7 DoF arm and a RBO soft-hand 2 (Deimel and Brock 2016); *right*: perceptual systems to perceive kinematic (*top*, Section 4) and dynamic properties (*bottom*, Section 5), including the most important recursive estimation processes and their information flow (coupling)

cannot be applied to perceive previously unseen articulated objects.

Previous interactive perception methods applied to object segmentation and recognition (van Hoof et al. 2012; Sinapov et al. 2011; Schiebener et al. 2014), shape reconstruction (Xu et al. 2015), and the perception of dynamic (Endres et al. 2013) and kinematic properties (Hausman et al. 2015) of articulated objects are based on a single modality, or they use multiple sensor modalities but apply each one independently to one perceptual subtask. This neglects the benefits of a tighter integration and exploitation of the interdependencies between subtasks. The multimodal interactive perception system that we will present in this section improves robustness and versatility over previous approaches and over the system presented in Section 2 by using cross-modal information: processes in one modality acquire priors for the interpretation of the other modality. The cross-modal information flow is a result of applying our algorithmic architecture for online interactive perception.

4.2 Method: Online Perception of Kinematics from Proprioception

Our goal is to overcome some of the limitations of the vision-based system (Section 2) by integrating vision and proprioception. The multimodal system will be a second instance of the general algorithmic architecture for online IP problems that we presented in Section 3. In this section, we will first present a novel perceptual system for kinematic models based only on proprioceptive signals. Then, we will explain the way to integrate vision (Section 2) and proprioception exploiting our algorithmic architecture. The integrated system with its most relevant recursive processes is depicted in Fig. 9; the top row depicts the visual system of Section 2, and the bottom row shows the recursive processes novel to this section.

We assume that the robot has grasped one of the bodies that is part of the articulated object and interacts with it. We call this body the *interacted body*. The motion of the interacted body and the robot's end-effector are coupled, as their relative motion is constrained by their contact interface i.e. the grasp. As our robot uses a soft-hand for the interaction, the relative motion between the hand and the interacted body depends not only on the degrees of freedom of the contact interface but also on the deformation of the hand.

Based on the previous analysis, we factorize the proprioception-based perception of kinematics of articulated bodies into the following five subproblems: The estimation of 1) the motion of the end-effector, 2) the bending state of the soft-hand, 3) the kinematic model of the grasp, 4) the motion of the interacted body, and 5) the constraints and DoFs in the motion of the interacted body (kinematic model). Fig. 9 depicts the recursive processes addressing these subproblems and the processes of the vision-based system. The estimation of motion of the interacted body is subsumed with the estimation of other bodies from vision in the box "Rigid body motion."

As in our vision-based system, estimated states are passed as measurements from one process to the other. Thus, the originating process acts as a virtual sensor for the second process. We used this communication pattern to "enrich" with priors the sensor signals until we solve the original perceptual problem, the estimation of kinematics.

Predictions (e.g. measurements and states) are passed as state predictions from one process to the other. By exploiting this communication pattern, we restrict the search space of possible solutions of one subproblem using the other processes (and their priors) as alternative forward and measurement models. We will apply these intercommunication patterns in the perceptual system based on proprioception, and in the final system by combining vision and proprioception.

In the following, we will explain the ways we address the subproblems of the proprioception-based system using coupled recursion estimation. The last subproblem is solved the same way as the estimation of kinematic models from rigid body motion as explained in Section 2, and, thus, we do not repeat the explanation here.

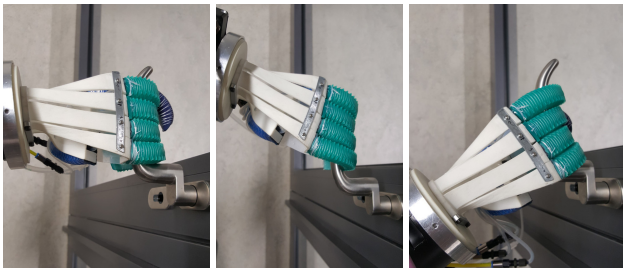


Figure 10. Effect of the deformation of the soft-hand; left: hand in the nominal state; middle and right: hand in the bent state after a motion of the end-effector without motion of the interacted body (a door handle)

4.2.1 Estimation of End-Effector Motion: The first recursive filter estimates the motion of the end-effector. The state of the end-effector is represented by the end-effector's pose and velocity, $\boldsymbol{x}_t^{ee} \sim \mathcal{N}((_{ee}p_t, _{ee}\eta_t), P_t^{ee})$. To predict the next state based on the previous estimate, we use a velocity-based kinematic update:

$$_{ee}\hat{p}_t = \Delta_t \, _{ee}\eta_{t-1} \oplus _{ee}p_{t-1} \tag{35}$$

$$_{ee}\hat{\eta}_t = _{ee}\eta_{t-1} \tag{36}$$

The measurements for the estimation of the end-effector motion are the pose and velocity of each robot joint provided by the robot's joint encoders, $z^{ee} = (q_j, \dot{q}_j)$, $j \in \{0, \ldots, J-1\}$, where J = robot's number of joints. Predicting this measurement based on the state would require solving an inverse kinematics problem. Instead, we combine the measurements on the robot's joints' poses and velocities with prior knowledge about the robot's embodiment and forward kinematics. This way, we obtain a direct measurement of the end-effector's pose and velocity, which we integrate recursively:

$$z'^{ee} = (_{ee}p_z, _{ee}\eta_z) \tag{37}$$

where the sub-index z indicates that they are measurements (direct observations of the state).

With this measurement model, the estimation of end-effector motion corresponds to a filtering of the proprioceptive measurements, weighted by the uncertainty of the observations, $R_t'^{ee}$, which we set proportional to the velocity (fast end-effector motion corresponds to highly uncertain pose measurements):

$$R_t'^{ee} = \mathrm{diag}(\alpha_p \, _{ee}\eta_z, \alpha_v \, _{ee}\eta_z) \tag{38}$$

4.2.2 Estimation of Hand Bending: When the robot interacts with an object, the soft-hand deforms (bends). This changes the relative pose between the hand and the object (see Fig. 10). In the second recursive filter, we estimate the consequences of this bending effect.

We represent the bending state of the soft-hand as the relative transformation between the *nominal* end-effector pose (estimated by the filter described above) and the pose of a virtual body we call *bent* end-effector (defining the hand's physical pose),

$$\boldsymbol{x}_t^{bent} \sim \mathcal{N}(_{bee}^{ee}p_t, P_t^{bent}) \tag{39}$$

$$_{bee}^{ee}p = {}^{bee}p \ominus {}_{ee}p \tag{40}$$

where $\ominus$ is the inverse composition of poses, and $_{bee}$ indicates bent end-effector. We assume that the bending state remains constant between consecutive time steps, $\hat{x}_t^{bent} = x_{t-1}^{bent}$.

We use as measurements the signals of the proprioceptive stream that correlate to the bending of the hand. These are the wrenches measured at the robot's wrist and the pressure values in the four air chambers of the soft-hand:

$$z^{bent} = (w, a) \tag{41}$$

where the wrenches are $w \in \mathbb{R}^6$, $w = (f, \tau)^T$, and the air pressure signals $a \in \mathbb{R}^4$.

Defining an analytic measurement model relating bending and proprioceptive signals for a complex soft-manipulator as the RBO Hand 2 (Deimel and Brock 2016) is a difficult problem (Smoljkic et al. 2015). We will adopt a data-driven approach and learn from experiences a model that transforms the proprioceptive signals into direct observations of the bending state:

$$f(w, a) = \boldsymbol{z}'^{bent} \sim \mathcal{N}(_{bee}^{ee}p_z, R_z^{ee-bee}) \tag{42}$$

where the sub-index z indicates that they are measurements (direct observations of the state).

We approximate the model f using an artificial neural network. To obtain labeled data to train the model, we execute 15 interactions of the robot grasping an object that is rigidly attached to the environment. We record the wrenches and the pressure signals at different relative poses of the bent soft-hand with respect to the nominal pose during these interactions. We then train a multi-layered perceptron regressor (MLPR1) to map from wrenches and pressure signals to the 6D relative pose observations.

To integrate the observations recursively, we also need to learn their uncertainty, R_z^{ee-bee}. Following the approach proposed by Rojas (1996), we train several partial MLPRs, leaving out groups of two trials, and computing the standard deviation between predictions from these partial MLPRs and the fully trained MLPR. We then train a second MLPR (MLPR2), mapping wrenches and pressure signals to the standard deviation of the regressor. With this procedure, the second MLPR learns the difficulty of the transformation problem for each input signal and allows us to filter proprioceptive signals into a robust estimate of the hand bending state.

4.2.3 Estimation of Interaction-Grasp Model: In the third recursive filter, we estimate a kinematic model of the grasp. The grasp model explains the kinematic constraints between the motion of the bent end-effector and the interacted body. We maintain and estimate independently the parameters of four filters for the grasp models, one for each type of grasp that our anthropomorphic soft-hand can perform: *(i)* perfect grasp (no relative motion), *(ii)* revolute grasp (allowing rotation around the grasping axis), *(iii)* cylindrical grasp (allowing rotation around and translation along the grasping axis), and *(iv)* failed grasp (no motion constraint).

For revolute and cylindrical grasps, the state of the filter is parametrized by the orientation of the axis (azimuth $\phi^{gr,r}$ or $\phi^{gr,c}$, and elevation $\theta^{gr,r}$ or $\theta^{gr,c}$ in spherical coordinates), and by a point on the axis ($p^{gr,r} \in \mathbb{R}^3$ or $p^{gr,c} \in \mathbb{R}^3$). For the perfect grasp, the state is parametrized by a fixed 6D pose

between the bent end-effector and the interacted body (${}^{bee}_{ib}p$). The failed grasp does not impose any motion constraints and therefore does not have any parameters to estimate, $x^{gr,f} = \emptyset$. We initialize these parameters based on the morphology of the hand and an initial low uncertainty, indicating that this initial estimate for the parameters of the grasping models should be trusted.

The estimation of the grasp model leverages the coupling between filters to obtain measurements. The estimates of the pose of the bent hand (from the previous two filters) and the interacted body (from the next filter) are combined to generate a measurement:

$$z^{gr} = f(x^{ee}, x^{bent}, x^{ib}) = {}_{ib}p \ominus ({}_{ee}p \oplus {}^{ee}_{bee}p) = {}^{bee}_{ib}p \quad (43)$$

The estimation of the parameters and the most likely type are performed similarly to the estimation of joint parameters of a kinematic model in Section 2. A difference with respect to our approach to estimate joints of kinematic models of articulated objects is that the grasping model estimation does not include the estimation of the joint state. The predicted measurements (relative poses) are a function of this joint state. For each measurement, we compute the current joint state of each model that minimizes the difference between the predicted relative pose (a function of the joint state) and the measured relative pose. We will use this minimum difference to evaluate the most likely model. Also, given the low uncertainty of the initial estimates of the grasping parameters, the method presented here can be seen as a model-selection approach (among a set of predefined models).

4.2.4 Estimation of Interacted Body Motion: The fourth recursive filter estimates the motion of the body the robot interacts with. The state of the interacted body is represented by its pose, $x^{ib} = {}_{ib}p$. The prediction of its next state also leverages the coupling between filters: the change in pose depends on the motion of the end-effector, corrected with the bending effect and propagated through the grasping model,

$${}_{ib}p_t = (\bar{x}^{gr}\, {}^{ee}_{bee}\mathrm{Ad}_{\,ee}\eta\Delta_t) \oplus {}_{ib}p_{t-1} \quad (44)$$

where $\bar{x}^{gr}$ is a 6×6 matrix representation of the kinematic constraints of the grasping model and ${}^{ee}_{bee}\mathrm{Ad}$ is the adjoint transformation associated with the bending effect.

None of the proprioceptive signals can be used as observations of the motion of the interacted body, and thus the predicted distribution over the next state becomes the current belief.

Limitations of Proprioception-Based estimation of Kinematic Models: The first limitation of the system based only on proprioception is due to the mutual dependency between the estimation of the interacted body motion and the grasp model. The motion of the interacted body is estimated based on the current belief over the grasp model. In turn, the grasp model is updated based on the estimated motion of the interacted body. This mutual dependency effectively reaffirms the initial prior distribution over the grasp model. The accuracy of the estimated interacted body motion depends thus on the accuracy of this grasp model prior.

The second limitation is that the proprioceptive signals, because of their limited range, only provide measurements about the state of the robot and the responses from the interacted body. The system can only perceive a single body connected by a joint to the environment, defining the kinematic model. Overcoming both limitations will require additional prior knowledge that our integrated system will obtain from vision coupling recursive processes across modalities.

4.3 Method: Online Perception of Kinematics from Multimodal Signals

We have explained how to extract information from each modality individually (from vision in Section 2 and from proprioception directly above). In the following, we will present a method to integrate both unimodal systems into a unified multimodal system. We will see that the mechanisms to couple recursive processes that our proposed algorithmic architecture offers can be exploited to leverage information from one modality and enable the correct interpretation of the other. The proposed multimodal system, thus, will exploit *cross-modal information*, i.e. information flow across different modalities, to overcome the limitations of unimodal perception system.

In our proposed multimodal system, predictions about the motion of the interacted body from proprioception are leveraged to correctly assign visual point features to the body, even under challenging visual conditions, e.g. with very low lighting or large occlusions. The features can be used as observations to correct the proprioceptive predictions, $z^{ib} = x^{fm|ib}$, where $x^{fm|ib}$ are the visual point features assigned to the interacted body. The cross-modal predictions from proprioception to vision and the corrections from vision to proprioception lead to a new estimate that breaks the mutual dependency of the proprioception-only system, $x^{ib} = {}_{ib}p$.

Using the interacted body motion perceived from cross-modal information, our system can correctly interpret the constraints in the bent end-effector motion perceived from proprioception, and retrieve the kinematic grasp model, $\boldsymbol{x}^{gr}$. The type and parameters of the grasp model are inferred from the relative motion between the bent end-effector and cross-modal estimates of the interacted body motion (Section 4.2.3):$z^{gr} = {}_{bee}p \ominus {}_{ib}p$.

The system can use grasp model estimates from cross-modal information as a prior to further interpret proprioceptive signals when the visual modality degenerates (e.g. the object goes out of the field of view, or is occluded, or due to extremely bad lighting conditions or not enough visual texture). The prior obtained from cross-modal information is sufficient to estimate the kinematic model of the interacted body using only proprioceptive signals.

The integrated multimodal system correctly interprets the constraints in the motion of the interacted body perceived from proprioception, leveraging information from vision. The system perceives from vision the motion of other bodies apart from the directly interacted one and uses this prior to analyze the motion constraints of the interacted body from proprioception. The integrated system based on cross-modal information can perceive complex kinematic models with

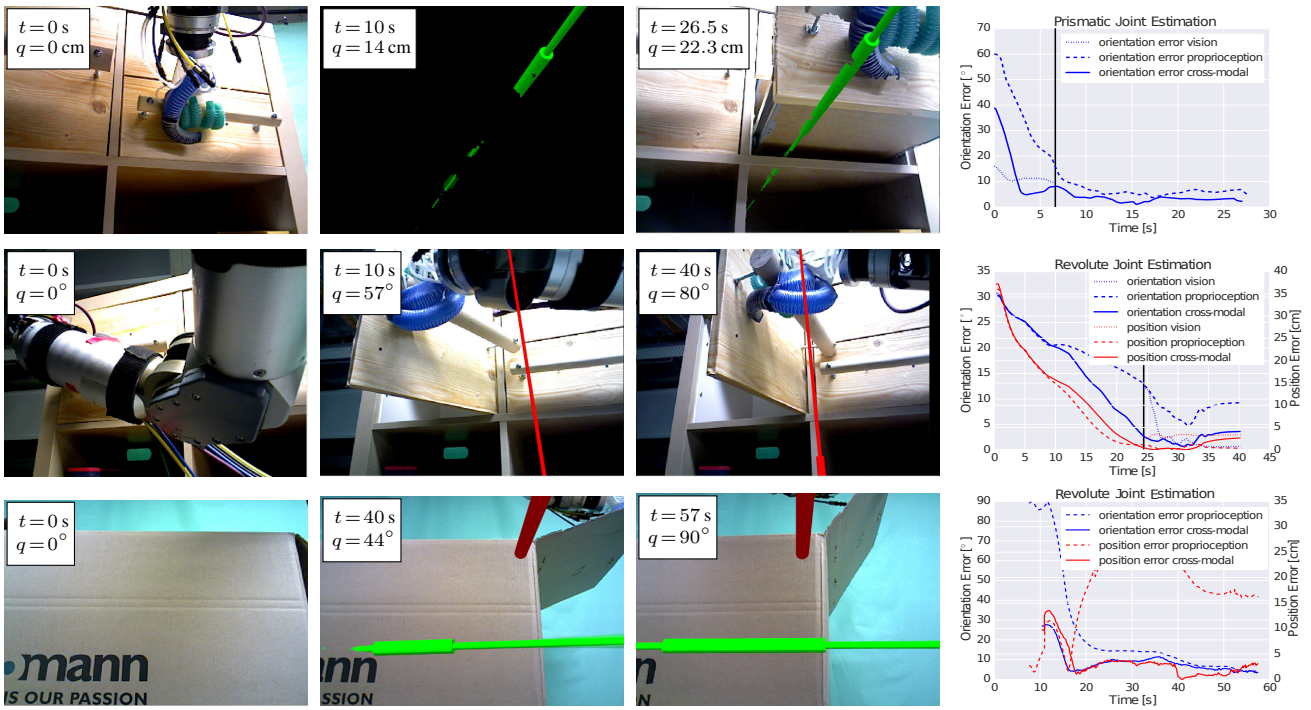


Figure 11. Experiments of the estimation of kinematic models (each row represents a different object): initial (first column), intermediate (second column), and final frame (third column) of the estimation, including error plot (fourth column) of estimated joint parameters relative to ground truth; the insets in the three images show the time t and the estimated joint configuration q using the multimodal variant; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders

multiple joints or when the interacted body is not connected to the static environment, $\boldsymbol{x}^{joint}$.

4.4 Experiments

We conducted two sets of experiments to evaluate our multimodal system. In the first set, we measure quantitatively the performance of our system when perceiving different articulated objects, and compared the use of 1) only vision, 2) proprioception, or 3) the multimodal stream exploiting cross-modal information between vision and proprioception. We measure the robustness, accuracy, and convergence of the kinematic model estimation by comparing the estimates to ground truth.

In the second set, we make use of the kinematics perceived online with the multimodal system to control the robot's motion and fulfill a manipulation task. The robot first explores an articulated object with a pulling interaction until it discovers its kinematic structure and perceives that the object reaches a queried kinematic state. For this first exploratory phase the robot relies on a velocity impedance controller that maintains a constant pulling velocity while being compliant in the other end-effector dimensions based on the measured wrenches.

When the robot perceives that the object has reached the desired state, it exploits the perceived information to plan a new end-effector trajectory to return the object to its initial configuration. To do that the robot interpolates the object's joint configuration towards the desired state and computes the motion of the interacted body necessary to obtain the desired object's joint configuration. The robot can then compute the operational space trajectory of its end-effector that will generate the necessary interacted body motion. We measure the accuracy of the execution (final joint state) of both the explorative and the exploitative interactions.

4.4.1 Experimental Setup: In our robot experiments, we use a robot platform composed of a Barrett WAM arm and a RBO soft-hand 2 (Deimel and Brock 2016) (see Fig. 9, left). The joint configurations of the arm are measured at 200 Hz by encoders placed at the motors controlling the cables. The stretching of the cables introduces uncertainty about the end-effector's pose that we model with a standard deviation of 1 cm and 3° in the end-effector pose measurements, resulting from an offset calibration. The visual input is an RGB-D stream provided by a Carmine sensor rigidly attached and registered to the base of the robot arm. The force-torque signals are provided by an ATI 6DoF sensor mounted on robot's wrist delivering signals at 100 Hz. Air pressure in the chambers of the soft-hand are delivered at 100 Hz. To compensate for the disparity in sensor frequencies we accumulate proprioceptive signals and process them at 30 Hz. This estimation rate can be maintained on an Intel Xeon E5520 PC at 2.27 GHz.

We implement the recursive estimation processes as Kalman filters or extended Kalman filters. The neural network regressors (MLPR) have a topology of three layers with 10-10-10 fully connected neurons. This topology was selected in a hyperparameter search by a leave-one-out cross-validation process, selecting between 1 and 100 neurons per layer in networks of one, two, or three layers. The vision-based system tracks $N = 200$ point features. The value of other significant system parameters are summarized in Table 5 in Section 7.

We evaluate our system on articulated objects with different types of joints, size, color, and surface properties.

We did not add artificial visual markers that could facilitate the visual perception. The objects are placed at different poses with respect to the robot and the sensors. In some experiments we also abruptly change the lighting conditions to evaluate the robustness of the perceptual systems. To obtain the ground truth of the kinematic properties, we manually measured the joint parameters and the final joint state.

4.4.2 Experimental Evaluation:

Unimodal vs. Multimodal Perception: We evaluate the accuracy and convergence of the kinematic model estimates from the three perceptual systems: 1) only vision, 2) only proprioception and 3) the multimodal integration of vision and proprioception based on cross-modal information flow. Fig. 11 shows three images from the RGB-D sensor (initial, middle, and final steps) and graphs of the estimation error to ground truth over time.

In the first experiment, the robot interacts with a drawer. After 6.5 s (indicated with a vertical line in the plot in Fig. 11, first row, most right) we change abruptly the lighting conditions by switching off the lights. The vision-only system stops perceiving the object while the proprioception-only and the multimodal system continue the estimation. The final joint state estimated by the multimodal system is the most accurate (22.3 cm, ground truth 22.5 cm), followed by the proprioception-only (23 cm). The vision-only system stops tracking at (9.8 cm). The multimodal system achieves the best performance because it combines vision and proprioception to estimate an accurate grasping model, and uses this model to estimate correctly the motion of the grasped body and to interpret the proprioceptive signals when the lights go off.

In the second experiment, the robot interacts with a door that rotates around a revolute joint. The robot almost completely occludes the object during the first 25 s of interaction (end of the occlusion indicated with a vertical bar in the plot in Fig. 11, second row, most right). The proprioception-only and the multimodal systems perceive the kinematics of the object during the entire interaction. The final joint state estimation from the multimodal system (80°, ground truth 85°) is the most accurate, followed by the proprioception-only (78°). The vision-only system first perceives the interacted object and its kinematics when it becomes clearly visible. Therefore, the final estimation of the visual system (43°) is affected by the delayed start. The multimodal system achieves the best performance because it uses the proprioceptive signals to interpret the visible motion in the small non-occluded parts of the object.

In the third experiment, the robot interacts with a cardboard box and closes one of its lids. As a result from this explorative interaction, the entire box translates. We focus the analysis on the estimation of the relative revolute joint between the box and the lid since both unimodal systems fail to detect this joint. The vision system only perceives the lower part of the box. The proprioception system perceives only the motion of the lid and interprets it as a revolute joint with respect to the rest of environment. The multimodal system correctly perceives the relative joint between the box and the lid because it uses the motion of the box perceived from vision to correctly interpret the motion constraints of the lid perceived from proprioception. The final joint state estimate from multimodality is 90° (ground truth 100°).

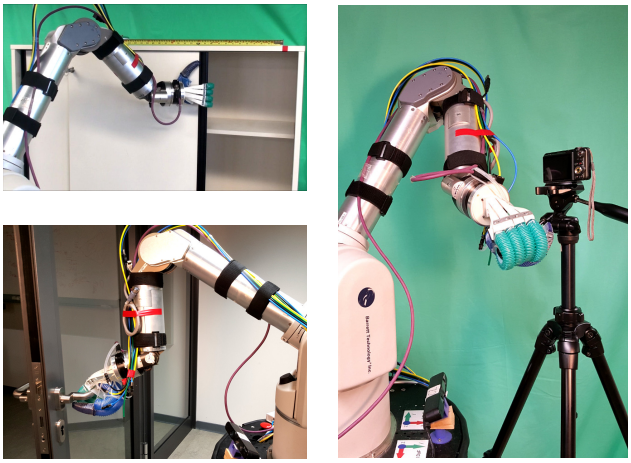


Figure 12. Our robot manipulating three articulated objects (a cupboard door, a glass door, and a camera tripod) and perceiving their kinematic structure; the robot uses a RBO soft-hand (Deimel and Brock 2016) for safe interactions; the exploratory interaction is steered using our velocity-impedance controller; our online perceptual system integrating vision (RGB-D stream) and proprioception (joint encoders, force-torque and air-pressure signals) acquires information from the exploration and generates robot trajectories for new manipulation tasks

Table 1. Error of exploration and exploitation phases

Object	Exploration Error	Exploitation Error
Sliding Door	2.2 cm±1.6 cm	1.8 cm±1.6 cm
Camera Tripod	7.8°±2.3°	2.6°±2.24°
Glass Door	1.3°±0.73°	0.6°±0.5°

Controlling the Interaction with Online Interactive Perception: We tested our multimodal perceptual system for the manipulation of three previously unseen objects (see objects in Fig. 12): opening a glass door (GD) 20°, turning camera tripod (CT) 45° and opening a sliding door (SD) 30 cm. These objects are challenging because they do not present strong textured surfaces and because the hand cannot grasp them perfectly.

We repeated the interactions five times on each object with different initial robot-object pose. The results (mean and standard deviation on the error to joint state ground truth) are depicted in Table 1. The largest error is in the exploration of the camera tripod (7.8°); the small distance between the end-effector and the axis of rotation leads to a wrongly estimated location of the joint and an associated error in the state. This error is corrected in the exploitation phase after more signals have been integrated.

We consider a successful interaction if the error in the final kinematic state is under 5% of the total desired joint actuation. The interaction succeeded in the 15 trials (see multimedia attachment 2). This indicates that the robot can use the multimodal system to monitor manipulations and to obtain information to generate new successful trajectories.

4.5 Limitations

Our proposed multimodal system can only be used when the robot interacts with the articulated object. When the robot observes the interaction of another agent there are no proprioceptive signals, and, thus, the perceptual system reduces to the vision-based system presented in Section 2.

Our method to estimate the deformation of the hand can only be applied to pneumatic hands with air-pressure readings from the hand's chambers. While this is a limitation, most existing pneumatic hands provide this type of sensor readings. Clearly, for each new hand one would have to repeat the data collection and neural network offline training procedures.

5 Online Perception of Dynamics

Previous sections were focused on perceiving the constraints in motion and degrees of freedom between the components of an articulated object: the kinematic properties. In this section, we turn to the problem of perceiving the dynamic properties of the articulation. In general, dynamics is concerned with the relationship between forces and motion. In an articulated object the dynamic properties of the articulation describe the relationship between applied wrenches and changes in the kinematic state of the object's degrees of freedom.

5.1 Related Work

Previous approaches have shown that it is possible to perceive dynamic properties from robot interactions but have mainly focused on the estimation of the inertia properties of a single non-articulated grasped object. For example, Atkeson et al. (1986) inferred the inertia parameters of an object from interactions using the information of the robot's joint encoders. However, when it comes to the estimation of dynamic properties of articulated objects, most existing approaches have tackled the problem of estimating the dynamics of a robot arm based also on the robot's joint encoder signals (Xinjilefu et al. 2014; Ma and Hollerbach 1996).

Few methods have addressed the estimation of dynamics of external articulated objects for which internal joint sensors are not available. Endres et al. (2013) presented an approach to learn dynamic models of doors with a force/torque sensor on robot's wrist. They model dynamics with a parametric component that represents the moment of inertia and a non-parametric component (a Gaussian process) that represents the deceleration of the mechanism due to friction (some sort of approximate viscous friction model). To obtain the kinematic information necessary to estimate the dynamics (e.g. the velocity of the actuation of the joint), the authors employed the method by Sturm et al. (2011). Endres et al. showed that the learned dynamic model is useful for manipulation with an experiment where the robot uses the model to plan and execute swing interactions on a door with the goal of bringing it to a predefined configuration. On the contrary, we are interested on controlled interactions with a grasped articulated object rather than dynamic swinging manipulations and, therefore, our system models dynamics differently. In our manipulation scenario, the inertia and viscous friction effects are negligible while the force necessary to initiate and to maintain the joint actuation are relevant to control the interaction.

In a different type of work, Jain et al. (2010) presented a study of doors and drawers from human interactions. They estimated kinematic and dynamic properties of several everyday objects in human environments. The mechanisms of their study contained springs that create dynamic effects depending on the configuration of the joint. They also observed that the highest forces are required to initiate the actuation. This supports the simple model of the joint dynamics we will present in this section that also acknowledges the importance of the force to initiate the actuation (the force to overcome stiction).

5.2 Method

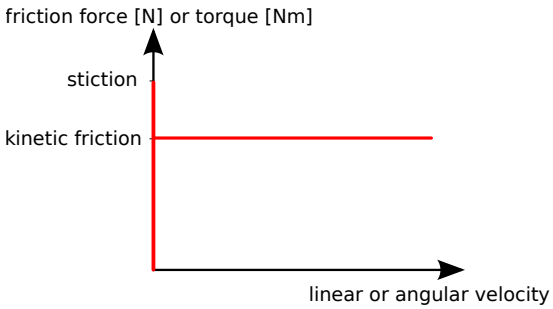
The combination of vision and proprioception allows the robot to infer new information about an actuated articulated object: its dynamic properties. The dynamic properties relate the forces and torques applied to an object with their kinematic effects.

Because in this work we are interested in controlled and safe robot interactions with constrained mechanisms, the wrenches the robot applies on the objects are bounded, and so are the joint accelerations they generate. In these conditions, we can neglect the inertia effects from our analysis of the dynamics, and we apply a *quasi-static* analysis, where the dominating term is the friction. We also deem the effect of other dynamic processes (e.g. damping and viscous friction) to be negligible for the type of objects and safe contact interactions we consider.

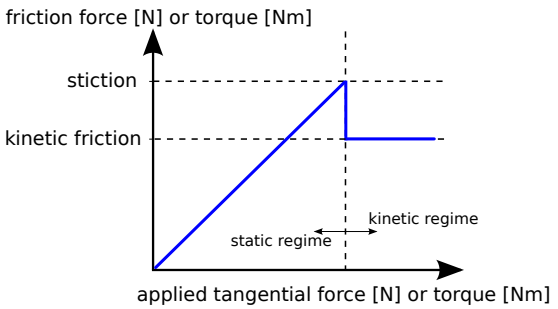
Roboticists have developed multiple models to explain the friction effects in articulated mechanisms. These models vary in complexity and in the number of parameters. In our estimation method, we use the Coulomb friction model and estimate two parameters: *stiction* and *constant kinetic friction* (Dupont 1990). Both values and their relationship to the actuation of the mechanism are depicted in Fig. 13, and explained below.

The contact surface between two bodies (e.g. within a kinematic joint) creates friction forces/torques[¶]. We can distinguish between two dynamic regimes with different friction effects, depending on the relative motion between the surfaces: If the bodies do not move with respect to each other, the equilibrium of forces/torques is in the static regime, and the force/torque opposing the applied force/torque is called *static friction*. Static friction is a force/torque equal in magnitude and opposite in direction to the applied force/torque in the allowed dimension by the joint, the so-called *tangential force/torque*. Forces/torques in the dimensions constrained by the joint, the so-called *normal forces/torques*, do not generate motion and are always counteracted by the mechanism (until breakage). Therefore, we do not consider the normal component of the applied force/torque in our dynamic analysis.

[¶]In the rest of the text we will use the terms *force/torque* (instead of wrench) and *linear/angular velocity* (instead of twist) to keep the explanation general for any type of kinematic constraint; however, note that for prismatic joints, we only consider forces and linear velocities, and for revolute joints, torques and angular velocities.



(a) Friction as a function of the velocity; without motion (static regime), friction ranges between zero and *stiction*; with motion (kinetic regime), friction is independent of the velocity and equal to the *constant kinetic friction*; if in motion the applied tangential force/torque decays under the *constant kinetic friction* the actuation stops



(b) Friction as function of the applied tangential force/torque; without motion (static regime), friction is equal to the applied tangential force/torque; with motion (kinetic regime), friction is constant and equal to the *constant kinetic friction*; if in motion the applied tangential force/torque decays under the *constant kinetic friction* the actuation stops

Figure 13. Coulomb model of friction; motion of a joint begins when the applied tangential force (for prismatic joints) or torque (for revolute joints) overcomes *stiction*; friction during motion is constant and equal to *kinetic friction*; if the applied tangential force/torque decreases under the *constant kinetic friction*, the motion of the joint stops

When the applied tangential force/torque overcomes a threshold, the two bodies begin to move with respect to each other. This threshold is called *stiction* (S) and is one of the parameters we estimate in our model since it is relevant information to control the manipulation.

During motion, the equilibrium of forces/torques is in the kinetic regime, and the force/torque opposing the motion is called *kinetic friction*. We assume this force/torque to be approximately constant and independent of the relative velocity (Coulomb model). We call this value *(constant) kinetic friction* (KF) and it is the second parameter we estimate in our model. If the applied tangential force/torque decays under the (constant) kinetic friction, the motion decelerates and stops quickly, and the equilibrium of forces/torques returns to the static regime. Knowledge about the force to overcome stiction and kinetic friction allows to plan and execute safe interaction (Endres et al. 2013; Eppner et al. 2018).

Given the previous definitions, we propose to estimate the parameters of the friction model recursively using a particle filter. The state of the filter is represented by a set of particles sampled following the distribution over dynamic parameters of the joint: $\boldsymbol{x}^{dyn} = \{p^{dyn,i}\}, i \in \{1 \dots N_{dyn}\}$. Each particle represents a hypothesis about the dynamic parameters, $p^{dyn,i} = (S^i, KF^i)$.

The observations to update the state of the filter, z^{dyn}, are joint velocities, $\dot{q}$, and the magnitude of applied tangential force/torque, $\|ft^{tan}\|$ (Fig. 9, bottom). As explained before, our dynamics (friction) model is independent of the magnitude of the joint velocity. Therefore, we simplify the measurements and consider $\dot{q}$ a binary variable, indicating whether the joint moves or not, $\dot{q} \in \{0, 1\}$.

In the following, we will first assume that the measurements for the estimation, z^{dyn}, are given and explain how to update the state of the filter. Then, we will explain how we obtain these measurements within an integrated multimodal system for kinematics and dynamics. We will leverage cross-modal information from other subprocesses of the system to obtain the measurements for the dynamics estimation.

The way we use the measured tangential force/torque to update the filter state (the measurement model) depends on the current dynamics regime: static or kinetic, or boundary state. To evaluate the current dynamics regime, we compare the current and previous observations of the joint motion, and distinguish four cases:

- The joint was not moving before and is not moving now: This case indicates that in both the previous and the current steps, the tangential force/torque is not enough to overcome stiction. The current tangential force/torque is *under* stiction.
- The joint was not moving before and is moving now: The tangential force/torque now is enough to overcome stiction and initiate motion. The current tangential force/torque is *over* stiction.
- The joint was moving before and is moving now: The tangential force/torque is enough to maintain motion. The current tangential force/torque is *over* kinetic friction.
- The joint was moving before and is not moving now: Kinetic friction dominates the quasi-static scenario and impedes the motion now. This effect indicates that the current tangential force/torque is *under* the kinetic friction.

These four cases lead to four measurement updates, with different importance functions for the particles.

In the first case, the particles predicting motion should receive a lower importance factor than the particles correctly predicting no motion. The importance should be even lower for particles that hypothesize that the threshold to initiate motion (stiction) was largely overcome. For a given measured tangential force/torque, $\|ft_{tan}\|$, we define the importance factor of a particle $p^{dyn,i} = (S^i, KF^i)$ with the function

$$p_1(z^{dyn}|p^{dyn,i}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{S^i - \|ft_{tan}\|}{\sigma_{ft}\sqrt{2}}\right)\right] \quad (45)$$

The equation above is the accumulative density function of a Gaussian distribution with mean $\|ft_{tan}\|$ and covariance σ_{ft}

evaluated at the stiction value of the particle, S^i. We will see later how to obtain these mean and covariance values that represent the measured tangential force/torque and its uncertainty. [‖] Using the accumulative density function we represent the approximate nature of our tangential force/torque measurements (which we assume Gaussian distributed), caused by the probabilistic nature of the joint axis estimation in our system.

The previous importance factor function is depicted in Fig. 14a. While this is not a well-defined probability density function (its integral over the entire space is not equal to one), the renormalization of the particles before the resampling step assures that the filter is probabilistically consistent. The function penalizes the particles where the stiction was largely surpassed by the applied tangential force/torque.

In the second case, the particles that predict correctly that the motion starts in the current step should receive a higher weight 1) than any other, i.e. a higher weight than particles predicting no motion because the tangential force/torque is under stiction, and 2) than particles predicting that the threshold to initiate motion (stiction) was largely overcome. For a given magnitude of the tangential force/torque, $\|ft_{tan}\|$, we define the importance factor of a particle $p^{dyn,i} = (S^i, KF^i)$ as

$$p_2(z^{dyn}|p^{dyn,i}) = \frac{1}{\sqrt{2\pi\sigma_{ft}^2}} exp\left(-\frac{S^i - \|ft_{tan}\|}{2\sigma_{ft}^2}\right) \quad (46)$$

These importance factor function is depicted in Fig. 14b. The function benefits the particles that correctly predicted that the motion of the joint should begin now (values of stiction close to the measured $\|ft_{tan}\|$).

The importance factor of the particles in the third and fourth cases are analogous to the first and second cases, but based on the particle's kinetic friction value:

$$p_3(z^{dyn}|p^{dyn,i}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{KF^i - \|ft_{tan}\|}{\sigma_{ft}\sqrt{2}}\right)\right] \quad (47)$$

and

$$p_4(z^{dyn}|p^{dyn,i}) = \frac{1}{\sqrt{2\pi\sigma_{ft}^2}} exp\left(-\frac{KF^i - \|ft_{tan}\|}{2\sigma_{ft}^2}\right) \quad (48)$$

The importance factor functions are depicted in Fig. 14c and Fig. 14d.

At each step we use the previous and current observations of the motion of the joints, $\dot{q}_{t-1}$ and $\dot{q}_t$, to identify the dynamics regime and to select the right importance factor function to update the filter state. These observations are provided by an existing subprocess of the perceptual system, the estimation of the kinematic model from visual data (Section 2). In an example of cross-modal integration, information from the estimation of kinematics is used as a prior to correctly interpret the measured tangential force/torque, $\|ft_{tan}\|$. The subprocess estimating kinematics also provides prior information to decompose the applied force/torque into the tangential and the normal components: the joint axis definition. In the rest of this section, we will explain how to obtain these measurements of the tangential force/torque relative to a given kinematic joint.

We compute the magnitude of the tangential force/torque applied by the robot in a two-step process. First, we need to compute the force/torque the robot applies on the object, and second, we need to decompose the applied force into the tangential and normal components.

To compute the applied force/torque we use measurements from a sensor attached to the robot's end-effector. We account for the effect of the gravity on the end-effector by subtracting the end-effector's weight from the raw force/torque readings, w, using the end-effectors mass, center of mass, and pose. The remaining force/torque signal is the applied force/torque by the robot on the object, $w_{app} = (f_{app}, \tau_{app})^T$.

To obtain the tangential force/torque, we geometrically compute the decomposition of the applied force/torque given the joint axis definition. Because in our perceptual system the joint axes are defined by probability distributions over joint parameters, we sample multiple joint axes from these distributions and project the applied force/torque onto the hypothesized axes. Then, we collect the vectors of the tangential projections, compute their norm and fit a Gaussian to the resulting norm samples. The result of this process, $\|ft_{tan}\|$ and σ_{ft}, is a probability distribution over the applied tangential force/torque, grounded in the uncertainty about the kinematic model. In the following, we will explain in detail the geometric decomposition for a prismatic and revolute joint.

For a prismatic joint, the tangential component corresponds directly to the projection of the applied force onto the direction of the axis, f_{tan} (see Fig. 15, left).

For a revolute joint, we first compute the applied force/torque at the sampled axis, $w'_{app} = (f'_{app}, \tau'_{app})^T$, from the applied force/torque at the point of contact, $w_{app} = (f_{app}, \tau_{app})^T$. For this transformation we assume that both locations are on the same rigid body, i.e. the robot applies forces/torques on one of the two links connected to the joint. The transformation of the force/torque to another point on the same rigid body is given by

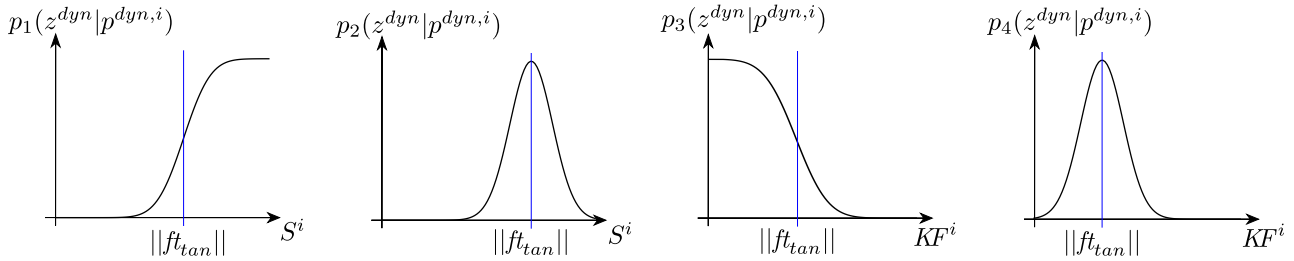
$$(f'_{app}, \tau'_{app}) = (f_{app}, \tau_{app} + \bar{r} \times f_{app}) \quad (49)$$

where $\bar{r}$ is the vector connecting the point of application of the force/torque and one point on the sampled axis. Finally, we project the transformed torque, τ'_{app}, onto the direction of the sampled revolute axis and obtain the tangential torque, τ_{tan} (see Fig. 15, right).

To obtain measurements of the tangential applied force/torque our method integrates information from two other subprocesses in the perceptual system: the estimation of kinematic models (to project the applied force/torque into the joint axis) and the estimation of end-effector motion (to transform the measured wrenches from the reference frame of the robot's wrist to the spatial configuration of the joint).

The particle filter method we propose reduces continuously the uncertainty over the dynamic parameters as more haptic measurements are acquired, especially when the joint starts or stops moving. An example of the evolution of the

[‖] $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x exp\left(-t^2\right) dt$ is the *error function*, the probability of a random variable normally distributed with mean 0 and variance $1/2$ being in the range $[-x, +x]$.



(a) Case 1: $\dot{q}_{t-1} = 0$ and $\dot{q}_t = 0$

(b) Case 2: $\dot{q}_{t-1} = 0$ and $\dot{q}_t = 1$

(c) Case 3: $\dot{q}_{t-1} = 1$ and $\dot{q}_t = 1$

(d) Case 4: $\dot{q}_{t-1} = 1$ and $\dot{q}_t = 0$

Figure 14. Four different importance factor (likelihood) functions for the four cases, depending on whether the joint was actuated or not in the previous and current steps; the functions are centered at the mean applied tangential force/torque, μ_{FT} and "spread" accordingly to its covariance, σ_{FT}; the functions are applied to the set of particles, $\{p^i_{dyn}\}$, based on their hypothesis about the stiction (S) or the kinetic friction (KF)

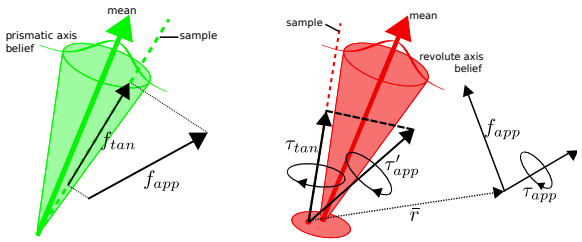


Figure 15. Geometric projection of the applied wrench onto a sample of a prismatic **(left)** or a revolute **(right)** joint; translucent cones indicate one standard deviation to the mean of the axis orientation; the translucent sphere indicates one standard deviation to the mean position of the axis; the projection of the applied wrench decompose it into a tangential components (f_{tan} and τ_{tan}) and normal components (not shown)

estimated dynamic parameters from continuously arriving haptic measurements is depicted in Fig. 16.

5.3 Experiments

5.3.1 Experimental Setup: We apply our approach for the estimation of dynamic properties to six different articulated objects. We analyze the properties of our proposed system, first in isolation, using kinematic information from a motion capture system, and second, in integration with the rest of the online IP system for articulated objects based on multimodal sensor signals (Section 4).

For the experiments where we evaluate the estimation of dynamic properties in isolation, we use a motion capture system to obtain ground truth about the kinematic information (Motion Analysis 2017). We use the motion capture system to obtain ground truth poses of the links of the mechanisms and the joint parameters. The interactions are performed with a force-torque sensor attached to a stick, whose pose is also tracked using the motion capture system. This data is part of a dataset of articulated objects and sensor data of interactions that we have released and made public for other researchers**.

The parameters used in the experimental evaluation are summarized in Table 6. We impose in the particles the additional physical constraint that stiction must be higher or equal to the kinetic friction. We use the same computers and robot platforms as in the experimental evaluation of the previous section (see Section 4.4.1).

To obtain the ground truth of the dynamic properties we measured with a force gauge the minimum tangential force/torque to initiate and maintain joint actuation. Even a careful manual ground truth measurement leads to slightly different readings. We average over three ground truth measurements to obtain a more reliable ground truth value, and report in the Table 2 and Table 3 the mean and standard deviation of these three measurements.

5.3.2 Experimental Evaluation: Table 2 summarizes the results of the first set of experiments, where we use ground truth poses and kinematic models and evaluate the accuracy of the estimation of dynamic properties in isolation. The estimation values depicted in the table are the result of averaging five estimation processes from different interaction sequences. We fit a Gaussian to the particles resulting from the interaction and obtain five mean and five standard deviation values. In the table, the first value (before $\pm$) depicts the mean of the means of these five fitted Gaussian distributions. The second value (after $\pm$) depicts the standard deviation of the means of the five fitted Gaussian distributions (do not confuse them with the standard deviation of each of the fitted Gaussian distributions from an estimation, which is not shown in the Table). In these sequences we vary the pose of the object, the pose of the force/torque sensor, and the contact point between the hand and the link of the articulated object.

The method correctly estimates both the threshold in force to initiate motion (stiction) and the minimum force to maintain the motion (kinetic friction). After fitting a Gaussian distribution to the final set of particles, the error between the estimated values and ground truth is within the uncertainty bound (under 0.9 standard deviations in the fitted Gaussian distribution of the particles of each estimation).

Only in two cases the estimated values diverge strongly from the ground truth: the cabinet drawer and the microwave door. The cabinet drawer is quite heavy and possesses high quality bearings. This means that while a high force is necessary to initiate motion (stiction), once the drawer is moving the inertia of the body (an effect that we don't model) is significant, and keeps the drawer moving even without additional effort. In this conditions, measuring the

**Our dataset is publicly available under https://tu-rbo.github.io/articulated-objects/.

Table 2. Estimation of stiction and kinetic friction from human interaction and ground truth kinematics

Object (joint type)	Parameter	Estimation	Ground Truth
Ikea (prismatic)	Stiction	2.94±0.756 N	2.9±0.12 N
	Kinetic Friction	0.94 N±0.45 N	1.0 N±0.3 N
Ikea (revolute)	Stiction	1.73±0.54 N m	1.8±0.4 N m
	Kinetic Friction	1.69±0.63 N m	1.7±0.6 N m
Cabinet (prismatic)	Stiction	8.56 N±0.85 N	8.31 N±1.2 N
	Kinetic Friction	0.62 N±0.41 N	1.83 N±0.52 N
Microwave (revolute)	Stiction	1.34±0.71 N m	1.2±0.35 N m
	Kinetic Friction	0.48±0.82 N m	0.65±0.36 N m
Ikea small (prismatic)	Stiction	3.23 N±0.47 N	3.12 N±0.65 N
	Kinetic Friction	1.78 N±0.42 N	1.98 N±0.82 N
Laptop (revolute)	Stiction	9.38±2.31 N m	9.58±0.84 N m
	Kinetic Friction	7.94±0.83 N m	8.40±0.63 N m

Table 3. Estimation of stiction and kinetic friction from robot interaction in the integrated multimodal perceptual system

Object (joint type)	Parameter	Estimation	Ground Truth
Ikea (prismatic)	Stiction	3.4±0.77 N	2.9±0.12 N
	Kinetic Friction	1.35±0.73 N	1.0±0.3 N
Ikea (revolute)	Stiction	2.07±0.61 N m	1.8±0.4 N m
	Kinetic Friction	1.93±0.79 N m	1.7±0.6 N m
Ikea small (prismatic)	Stiction	3.83±0.88 N	3.12±0.65 N
	Kinetic Friction	2.51±1.36 N	1.98±0.82 N

force to maintain the actuation (constant kinetic friction) is hard and highly dependent on the acceleration of the drawer. The microwave door is very light and presents low friction. In these conditions, the noise in the measured wrenches strongly affect the estimation of the parameters.

In the second set of experiments, we evaluate the integration of our dynamics estimation system with the multimodal system for the online perception of kinematics (Section 4). We evaluate the estimation of frictional properties on a cabinet door and two drawers. The results are depicted in Table 3, and the temporal evolution of the estimation from continuously arriving signals in one of the experiments is depicted in Fig. 16. While the integration in the system increases the uncertainty about the working wrench, the estimated dynamic parameters are still close to the ground truth values. However, the robot consistently overestimates the dynamic parameters because the proposed system does not discount the part of the tangential force that the soft-hand absorbs and transforms into deformation.

5.4 Limitations

Our system is restricted to articulated objects with one joint. In the case of several partially aligned DoFs the system would not be able to associate the applied wrench to the observed actuation of each joint. While theoretically this association would be possible for orthogonal DoFs, we haven't proved it experimentally.

The dynamics model of our system is simple but it explains with sufficient accuracy the dynamic effects that dominate the interaction of the most common articulated objects in human environments like doors and drawers. However, the model does not represent dynamic effects that depend on the joint state, e.g. the different effort at different configurations necessary to initiate the actuation of some doors with spring mechanisms. Our system would continuously change the estimated parameters within the range of possible values for such a mechanism. The model also neglects the inertia effects that should be considered for heavy objects as we observed for the metal drawer in the experimental evaluation.

6 Discussion

In Sections 2, 4, and 5 we presented interactive perception systems for building kinematic and dynamic models of articulated objects from visual and proprioceptive signals. These systems are implementations of our proposed algorithmic architecture for online interactive perception (Section 3): a network of coupled recursive estimation processes. Based on the analysis of the systems presented above, we discuss in this section the most important features of the algorithmic architecture (indicated with **bold** text) and the structural properties of the perceptual problem that the features exploit (indicated with *italic* text). We also point to similarities between the architecture and our understanding of biological perception systems. We conclude with a discussion on the architecture's most severe current limitations.

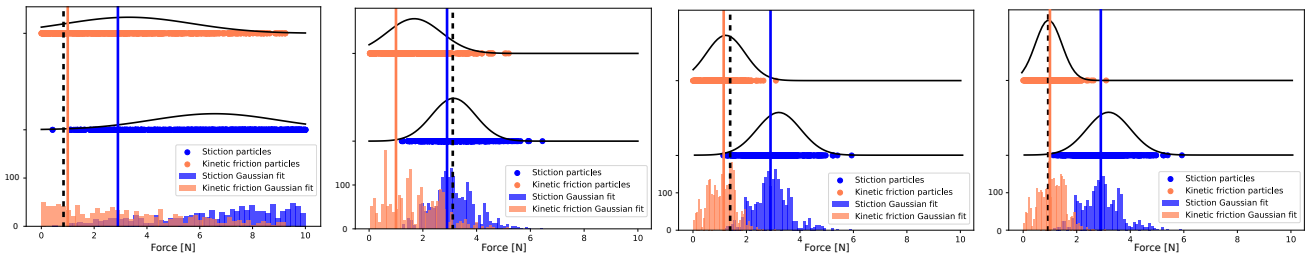
6.1 Features of Our Proposed Architecture for Online IP

The presented systems perceive online. They extract task-relevant information from the visual and proprioceptive sensor streams at 30 Hz. The robot uses this information to support the ongoing manipulation as well as to obtain a kinematic and dynamic model of the environment, the original goal of online interactive perception.

To achieve these online capabilities, our architecture resorts to **recursive estimation**. The instantiated systems refine the previously perceived state using the latest measurement instead of performing a time-consuming batch process of a set of measurements. Online processing is possible because the physical processes involved in the online perceptual tasks (e.g. motion, actuation of DoFs) contain a clearly perceivable *temporal structure*.

The recursive processes rely on models that encode regularities in the signal, for example derived from physics. The aforementioned temporal structure, as well as the correlation between changes in the state of the systems and in sensor signals, are encoded in the forward and measurement models of the recursive process. These models thus rely on **task-related physical priors**, e.g. kinematics, rigid body assumption, and projective geometry. The physical priors "enrich" and filter the sensor signals; they allow the recursive process to interpret signals as evidence of the state of the physical processes. Since the physical priors represent *regularities* of all unstructured environments and articulated objects, they are important features contributing to the observed versatility of the perceptual systems.

In our systems, online perception is supported by a factorization of the original perception problem into simpler



(a) Estimation after several measurements without joint motion ($t = 0.2\,\mathrm{s}$)

(b) Estimation when the joint starts to move ($t = 1.7\,\mathrm{s}$)

(c) Estimation after several measurements with joint motion ($t = 1.75\,\mathrm{s}$)

(d) Estimation when the joint stops moving ($t = 4.2\,\mathrm{s}$)

Figure 16. Four steps of the estimation of the dynamic properties in a prismatic joint (Ikea); blue vertical line: stiction ground truth ($2.9\,\mathrm{N}$); orange vertical line: minimum kinetic friction ground truth ($1.0\,\mathrm{N}$); red vertical line: current tangential applied force; each plot depicts the histogram of particles (bottom axis), the particles and Gaussian fit for the stiction parameter (middle axis) and the particles and Gaussian fit for the kinetic friction parameter (top axis); the multimedia attachment 3 shows the robot interaction of this run

subtasks that can be solved using recursive processes. The key to solving each subproblem and to addressing the original problem is to promote the **coupling between processes**: the processes communicate estimations and predictions to help each other. Solving the original online perception tasks with factorization and coupling indicates that the tasks are *weakly decomposable*: the problem can be decomposed into correlated subproblems. Our proposed architecture leverages these correlations.

Finally, the architecture we propose allows integrating the robot's interactions as part of the perceptual process. The interactions are used to create information-rich signals, e.g. motion cues that reveal kinematic constraints. But also knowledge about the interaction itself can be used to interpret changes in the signals, for example, the proprioceptive signals explain the perceived motion as a consequence of the dynamic properties of the joints. By using interactions to enable perception—the central idea of **interactive perception**—our architecture exploits the *strong correlations between the robot's actions and changes in the sensor signals*.

6.2 Similarities with Biological Perception

In this section, we want to discuss similarities between the proposed architecture for online interactive perception and knowledge about biological perception systems. We are compelled to report these similarities because we find them striking and also to contribute to bridging the studies of artificial and biological perceptual systems at the algorithmic level. However, we certainly do not want to make any claims as to how biological perceptual systems function.

The observed similarities between our architecture and biological perception span high-level concepts, architectural principles, as well as implementation-level details, such as the types of priors being used. The wealth of similarities we encountered indicate that the proposed algorithmic architecture could be used to generate and validate targeted experimental hypothesis for the study of biological perception. Doing so can ensure that the generated hypotheses are in principle consistent with other findings also reflected in the architecture. Furthermore, these similarities might facilitate the transfer of insights about biological perception to robotic systems.

6.2.1 *Conceptual similarity: Interactive Perception*

At the conceptual level, both our proposed architecture and biological perception systems rely on a close coupling between action and perception (Bohg et al. 2017)—a view that is not traditionally espoused in robotics and computer vision, which to some extend proceed in separation.

In psychology, J.J. Gibson probably was the first to strongly advocate perception as an interactive process (Gibson 1966, 1979). In one of his experiments, subjects were asked to recognize artificially-created pebbles of varying, complex shapes. Depending on the level of interaction allowed, the subjects performed very differently: no interaction (49% accuracy), observing the pebbles rotating but without control over their motion (72% accuracy), and full control of the pebble motion and therefore of the obtained visual signals (99% accuracy). This provides evidence for the importance of linking perception and perception in biological perception.

Animal experiments by Held and Hein (1963) also support the idea that perception is intrinsically linked to actions. The authors repeatedly placed two kittens, otherwise raised in complete darkness, in a carousel. This carousel was constructed such that both kittens would experience the exact same visual stimuli but only one kitten was controlling the motion with its walking movements, while the other was passively moved based on the motion of the first kitten. Only the kitten that controlled the motion learned to understand the correlation between its actions and changing visual signals. The second kitten, even though it had been exposed the identical visual signals, was unable to avoid obstacles or to follow a path. These experiments provide further support for the fact that biological perception depends on interactions.

6.2.2 *Architectural similarities: Interconnected hierarchies*

At the level of architectural features of information processing, we are also able to find similarities between our proposal for online IP and biological perception.

There is evidence from neuroscience, cognitive science, and psychology supporting the view that the human brain processes sensor signals in parallel subprocessing units, and that these units share information at multiple levels to help each other. For example, Livingstone et al. (1988) discovered that the human perceptual system contains two

parallel functional and anatomical subprocesses that share information at various levels.

The *reverse hierarchy theory* of biological perception (Hochstein and Ahissar 2002; Ahissar and Hochstein 2004) suggests that information processing is based on hierarchically structured components, where information is passed up and down within this hierarchy. Information of lower-level components serves as the input to higher-level components. The output of higher-level components is passed to lower-level components, arguably to enable or improve the perceptual performance of these components. Reverse hierarchy theory therefore exactly describes our proposed architecture.

Functional MRI studies of the brain support the hypothesis that information in the primate visual cortex does not only flow bottom-up from sensors to higher-level, more abstract visual processing units, but also laterally, and top-down. This exchange of information even happens across different sensor modalities (McGurk and MacDonald 1976). Top-down feedback serves as prior to interpret noisy sensor signals (Mumford 1992; Kersten and Yuille 2014). These findings are consistent with the information flow pathways and their uses in our proposed architecture.

Concrete examples of this include information passing between V2 (handling object and pattern recognition tasks) and V1 cortical areas (in charge of the perception of fine high frequency visual structures). The experiments by He et al. (2012) showed that low-level aftereffects, such as the tilt aftereffect, are reduced if the visual stimulus is recognized as part of an object. This indicates that feedback from V2 to V1 helps disambiguate low-level information. There are analogous findings for a number of different visual tasks. This feature of passing feedback between different perceptual processes is also present in our architecture. However, the visual system of primates seems to be able to modulate the importance of the feedback depending on the contextual need (Qiu et al. 2016; Fang et al. 2008). It might be interesting to examine, based on our algorithms, whether this is simply an energy-saving provision or whether it has perceptual benefits.

6.2.3 Low-level similarities: Priors

Given the similarities in the kind of information flow and the use of information within hierarchical architectures, one might also wonder if there are similarities regarding the components of those information processing hierarchies. Are they of similar type and do they exploit similar priors?

In implementations of the proposed architecture, the lowest level leverages spatial and temporal coherence for tracking of image features. There is evidence that this prior is also used in humans. Subjects perform better on perceptual tasks that present a strong temporal structure, i.e. temporal consistency and correlation (Kristjánsson et al. 2010; Maljkovic and Nakayama 1994; Maljkovic and Martini 2005; Niemi and Näätänen 1981).

There is extensive experimental support for the importance of perceptual grouping in biological perception (Watt and Phillips 2000; Qiu et al. 2016; Herzog 2018). Grouping is very naturally informed by the information flow in hierarchies described above. Information from a higher level of abstraction can, for example, be used to group lower-level features, facilitating their processing. In the presented instances of the proposed architecture, the concept of grouping also plays a crucial role, for example, when informing feature tracking by information about the knowledge of the motion of rigid bodies, or when informing the latter by knowledge of the kinematic structure.

In addition to these very general priors (temporal and spatial coherency, grouping), the human perceptual system exploits task-specific priors, similar to what we proposed for online IP. Many of these priors are even hard-wired into the physiology of the senses, starting with the retina in the human eye (Gollisch and Meister 2010). The retina reduces the raw visual signal by two orders of magnitude before passing the information to the visual nerve (Jonas et al. 1992). To do so without affecting task performance, it must perform this reduction in a way that maintains task-relevant information, i.e. based on priors. One of these priors is the geometric distribution of the receptors across the retina.

Other priors seem encoded in later parts of the biological information processing pipeline, in the form of intuitive physics (Spelke et al. 1995; Hespos and vanMarle 2012). Experiments show that infants are surprised by experimental illusions that contradict their intuitive concepts of solidity, occlusions, object permanence, or containment. The human perceptual system exploits these invariant physical rules to help in the interpretation of the noisy signals from the senses.

6.2.4 Interpretation

One the one hand, one might find these similarities between robotics and biology striking. On the other hand, they have a very natural explanation. Both the proposed robotic IP solution and biological perception must exploit the structure of the high-dimensional perceptual problem to facilitate a computationally efficient and robust solution. Because the structure leveraged by our system matches the structure exploited in biology, we believe that online interactive perception provides progress towards robust and relatively general perceptual systems.

6.3 Limitations

We have shown that we can create interactive perception systems for articulated objects based on our proposed algorithmic architecture. However, the current architecture has some limitations. The most severe limitation is the high design effort required to apply our architecture to an online interactive perception problem. Designing the presented systems requires finding a suitable factorization of the original problem, encoding the physics-based models of the recursive processes, and defining coupling between processes. This engineering process should be complemented by data-driven methods to propose stable factors, and to detect correlations between perceptual subtasks (Tatavarty et al. 2007).

Data-driven approaches can also replace or support the specification of recursive estimation processes. An initial example of this idea is the learned measurement model to estimate soft-hand bending using a neural network in Section 4.2.2. For that recursive process, we learned the measurement model from interactions. Similar data-driven approaches can be used to learn measurement and forward models that would interact with each other within our

proposed architecture (Jonschkowski et al. 2018; Haarnoja et al. 2016).

Another limitation is the persistence of the perceived information. In its current form, the architecture does not provide long-term memory nor generalization from previous experiences. The systems we presented forget about the perceived objects after the interaction, and all new objects need to be interacted. A promising extension of this work would use our method to label visual data and train a data-driven approach to predict kinematic structures before an actual interaction. Such an experience-based method could learn the visual invariant features that indicate different kinematic relationships, generalizing to previously unseen objects.

7 Conclusion

We presented systems for multimodal online interactive perception, extracting kinematic and dynamic models from objects in the environment. Our systems rely on visual information, depth data, contact wrenches, and proprioception. The systems are capable of perceiving object kinematics and dynamics from multimodal signals. In an extensive experimental evaluation we demonstrated the ability to successfully and robustly perceive articulated objects of different sizes, colors, and shapes with varying number of joints under challenging and varying environmental and lighting conditions.

We generalized shared features of the presented systems into a unifying algorithmic architecture for online interactive perception. This architecture consists of interconnected recursive estimation loops. Each loop leverages a task prior for the robust extraction of a task-relevant perceptual feature. Loops communicate information about the extracted features among each other, supporting each other in their perceptual task, including the possibility of compensating for varying degrees of uncertainties. This information flow occurs across different levels of abstraction of perceptual features. The IP systems described in this paper are specific instances of this architecture and demonstrate the benefit of the underlying architectural principles.

We finally discussed the most relevant features of the architecture in the context of robot manipulation. We also identified the underlying problem structure that each architectural feature leverages. We discussed limitations and generality of the architecture and its similarities to features of biological perception systems.

Funding

We gratefully acknowledge the funding provided by the German Research Foundation (DFG, Exploration Challenge, BR 2248/3-1) and the Alexander von Humboldt foundation through an Alexander von Humboldt professorship (funded by the German Federal Ministry of Education and Research).

Bibliography

Ahissar M and Hochstein S (2004) The reverse hierarchy theory of visual perceptual learning. *Trends in Cognitive Science* 8(10): 457–464.

Atkeson CG, An CH and Hollerbach JM (1986) Estimation of Inertial Parameters of Manipulator Loads and Links. *The International Journal of Robotics Research* 5(3): 101–119.

Bar-Shalom Y, Li X and Kirubarajan T (2001) *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc.

Barfoot TD and Furgale PT (2014) Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics* 30(3): 679–693.

Bohg J, Hausman K, Sankaran B, Brock O, Kragic D, Schaal S and Sukhatme GS (2017) Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics* 33(6): 1273–1291.

Bruyninckx H and Schutter JD (1996) Specification of force-controlled actions in the "task frame formalism" - a synthesis. *IEEE Transactions on Robotics* 12(4): 581–589.

Canny J (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6): 679–698.

Clark JJ and Yuille AL (1990) *Data Fusion for Sensory Information Processing Systems*. Norwell, MA, USA: Kluwer Academic Publishers.

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

Dagum P, Galper A and Horvitz E (1992) Dynamic network models for forecasting. In: *Proceedings of the Eighth International Conference on Uncertainty in Artificial Intelligence*, UAI'92. San Francisco, CA, USA, pp. 41–48.

Deimel R and Brock O (2016) A novel type of compliant and underactuated robotic hand for dexterous grasping. *The International Journal of Robotics Research* 35(1-3): 161–185.

Dupont PE (1990) Friction modeling in dynamic robot simulation. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 1370–1376.

Endres F, Trinkle J and Burgard W (2013) Learning the dynamics of doors for robotic manipulation. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 3543–3549.

Eppner C, Martín-Martín R and Brock O (2018) Physics-based selection of informative actions for interactive perception. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. p. accepted.

Fang F, Kersten D and Murray SO (2008) Perceptual grouping and inverse fmri activity patterns in human visual cortex. *Journal of Vision* 8(7): 2–2.

Fenzi M, Dragon R, Leal-Taixé L, Rosenhahn B and Ostermann J (2012) 3d object recognition and pose estimation for multiple objects using multi-prioritized ransac and model updating. In: Pinz A, Pock T, Bischof H and Leberl F (eds.) *Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 123–133.

Forster C, Pizzoli M and Scaramuzza D (2014) Svo: Fast semi-direct monocular visual odometry. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 15–22.

Fortmann TE, Bar-Shalom Y and Scheffe M (1980) Multi-target tracking using joint probabilistic data association. In: *Proceedings of the IEEE International Conference on Decision and Control*. IEEE, pp. 807–812.

Garcia Cifuentes C, Issac J, Wüthrich M, Schaal S and Bohg J (2017) Probabilistic articulated real-time tracking for robot manipulation. *IEEE Robotics and Automation Letters (RA-L)* 2.

Gibson JJ (1966) *The Senses Considered as Perceptual Systems*. 1 edition. Greenwood Press Reprint. ISBN 0-313-23961-4.

Gibson JJ (1979) *The ecological approach to visual perception*. Routledge. ISBN 978-0-89859-959-6.

Gollisch T and Meister M (2010) Eye smarter than scientists believed: Neural computations in circuits of the retina. *Neuron* 65(2): 150–164.

Golub GH and Van Loan CF (2012) *Matrix computations*, volume 3. JHU Press.

Haarnoja T, Ajay A, Levine S and Abbeel P (2016) Backprop kf: Learning discriminative deterministic state estimators. In: *Advances in Neural Information Processing Systems (NIPS)*. pp. 4376–4384.

Hausman K, Niekum S, Osentoski S and Sukhatme GS (2015) Active articulation model estimation through interactive perception. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 3305–3312.

He D, Kersten D and Fang F (2012) Opposite modulation of high-and low-level visual aftereffects by perceptual grouping. *Current Biology* 22(11): 1040–1045.

Hebert P, Hudson N, Ma J, Howard T, Fuchs T, Bajracharya M and Burdick J (2012) Combined shape, appearance and silhouette for simultaneous manipulator and object tracking. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 2405–2412.

Held R and Hein A (1963) Movement-produced stimulation in the development of visually guided behavior. *Journal of Comparative and Physiological Psychology* 56(5): 872–876.

Herzog MH (2018) Perceptual grouping. *Current Biology* 28(12): R687–R688.

Hespos SJ and vanMarle K (2012) Physics for infants: Characterizing the origins of knowledge about objects, substances, and number. *Wiley Interdisciplinary Reviews: Cognitive Science* 3(1): 19–27.

Hochstein S and Ahissar M (2002) View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron* 36: 791–804.

Huang X, Walker I and Birchfield S (2012) Occlusion-aware reconstruction and manipulation of 3D articulated objects. *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)* : 1365–1371.

Ilonen J, Bohg J and Kyrki V (2014) Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing. *The International Journal of Robotics Research* 33(2): 321–341.

Jain A, Nguyen H, Rath M, Okerman J and Kemp CC (2010) The complex structure of simple devices: A survey of trajectories and forces that open doors and drawers. In: *Proceedings of the IEEE, RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. pp. 184–190.

Jonas JB, Schmidt AM, Müller-Bergh J, Schlötzer-Schrehardt U and Naumann G (1992) Human optic nerve fiber count and optic disc size. *Investigative ophthalmology & visual science* 33(6): 2012–2018.

Jonschkowski R, Rastogi D and Brock O (2018) Differentiable particle filters: End-to-end learning with algorithmic priors. In: *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania. DOI:10.15607/RSS.2018.XIV.001.

Karayiannidis Y, Smith C, Barrientos FEV, Ögren P and Kragic D (2016) An adaptive control approach for opening doors and drawers under uncertainties. *IEEE Transactions on Robotics* 32.

Katz D, Kazemi M, Bagnell JA and Stentz A (2013) Interactive segmentation, tracking, and kinematic modeling of unknown articulated objects. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 5003–5010.

Katz D, Orthey A and Brock O (2014) Interactive perception of articulated objects. In: Khatib O, Kumar V and Sukhatme G (eds.) *Experimental Robotics*, *Springer Tracts in Advanced Robotics*, volume 79. Springer Berlin Heidelberg, pp. 301–315.

Kersten D and Yuille A (2014) Inferential models of the visual cortical hierarchy. *The Cognitive Neurosciences* : 389.

Khoshelham K and Elberink SO (2012) Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* 12(2): 1437–1454.

Kristjánsson Á, Eyjólfsdóttir KÓ, Jónsdóttir A and Arnkelsson G (2010) Temporal consistency is currency in shifts of transient visual attention. *PloS one* 5(10).

Kwok CCT and Fox D (2004) Map-based multiple model tracking of a moving object. In: *RoboCup 2004: Robot Soccer World Cup VIII*. pp. 18–33.

Lee TS and Mumford D (2003) Hierarchical bayesian inference in the visual cortex. *Journal of the Optical Society of America* 20(7): 1434–1448.

Liao L, Patterson DJ, Fox D and Kautz H (2007) Learning and inferring transportation routines. *Artificial Intelligence* 171(5): 311 – 331.

Livingstone M, Hubel D et al. (1988) Segregation of form, color, movement, and depth- anatomy, physiology, and perception. *Science* 240(4853): 740–749.

Ma D and Hollerbach JM (1996) Identifying mass parameters for gravity compensation and automatic torque sensor calibration. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*, volume 1. pp. 661–666.

Maljkovic V and Martini P (2005) Implicit short-term memory and event frequency effects in visual search. *Vision Research* 45(21): 2831–2846.

Maljkovic V and Nakayama K (1994) Priming of pop-out: I. role of features. *Memory & cognition* 22(6): 657–672.

Martín-Martín R and Brock O (2014) Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 2494–2501.

Martín-Martín R and Brock O (2017) Cross-modal interpretation of multi-modal sensor streams in interactive perception based on coupled recursion. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 3289–3295.

McGurk H and MacDonald J (1976) Hearing lips and seeing voices. *Nature* 264(5588): 746–748.

Motion Analysis (2017) Motion analysis corporation. http://ftp.motionanalysis.com, Accessed: 2017-11-01. URL https://ftp.motionanalysis.com.

Mumford D (1992) On the computational architecture of the neocortex. *Biological Cybernetics* 66(3): 241–251.

Niemi P and Näätänen R (1981) Foreperiod and simple reaction time. *Psychological bulletin* 89(1): 133.

Nistér D, Naroditsky O and Bergen JR (2004) Visual odometry. In: *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1. pp. I–I.

Pflueger M and Sukhatme GS (2015) Multi-step planning for robotic manipulation. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 2496–2501.

Qiu C, Burton PC, Kersten D and Olman CA (2016) Responses in early visual areas to contour integration are context dependent. *Journal of vision* 16(8): 19–19.

Rao RP and Ballard DH (1999) Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience* 2(1): 79–87.

Rao RPN and Ballard DH (1997) Dynamic model of visual recognition predicts neural response properties in the visual cortex. *Neural Computation* 9(4): 721–763.

Reid D et al. (1979) An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control* 24(6): 843–854.

Rojas R (1996) *Neural Networks: A Systematic Introduction*. New York, NY, USA: Springer-Verlag New York, Inc. ISBN 3-540-60505-3.

Ross D, Tarlow D and Zemel R (2008) Unsupervised learning of skeletons from motion. In: *Proceedings of the European Conference on Computer Vision*. pp. 560–573.

Schiebener D, Ude A and Asfour T (2014) Physical interaction for segmentation of unknown textured and non-textured rigid objects. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 4959–4966.

Shi J and Tomasi C (1994) Good features to track. In: *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*. pp. 593–600.

Simon HA (1996) *The sciences of the artificial*. MIT press.

Sinapov J, Bergquist T, Schenck C, Ohiri U, Griffith S and Stoytchev A (2011) Interactive object recognition using proprioceptive and auditory feedback. *The International Journal of Robotics Research* 30(10): 1250–1262.

Smoljkic G, Borghesan G, Reynaerts D, Schutter JD, Sloten JV and Poorten EV (2015) Constraint-based interaction control of robots featuring large compliance and deformation. *IEEE Transactions on Robotics* 31(5): 1252–1260.

Spelke ES, Vishton P and Von Hofsten C (1995) Object perception, object-directed action, and physical knowledge in infancy. In: *The cognitive neurosciences*. Cambridge, MA, US: The MIT Press. ISBN 978-0-262-07157-4, pp. 165–179.

Stilman M (2007) Task constrained motion planning in robot joint space. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 3074–3081. DOI:10.1109/IROS.2007.4399305.

Sturm J, Jain A, Stachniss C, Kemp CC and Burgard W (2010a) Operating articulated objects based on experience. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 2739–2744.

Sturm J, Konolige K, Stachniss C and Burgard W (2010b) Vision-based detection for learning articulation models of cabinet doors and drawers in household environments. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 362–368.

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

Sturm J, Stachniss C, Pradeep V, Plagemann C, Konolige K and Burgard W (2009) Learning kinematic models for articulated objects. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. pp. 1851–1856.

Tatavarty G, Bhatnagar R and Young B (2007) Discovery of temporal dependencies between frequent patterns in multivariate time series. In: *IEEE Symposium on Computational Intelligence and Data Mining*. pp. 688–696.

Tomasi C and Kanade T (1991) Detection and tracking of point features. Technical report, Carnegie Mellon University.

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

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

Triggs B, McLauchlan PF, Hartley RI and Fitzgibbon AW (2000) Bundle adjustment — a modern synthesis. In: Triggs B, Zisserman A and Szeliski R (eds.) *Vision Algorithms: Theory and Practice*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 298–372.

van Hoof H, Kroemer O, Ben Amor H and Peters J (2012) Maximally informative interaction learning for scene exploration. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 5152–5158.

Wang J and Olson E (2016) Apriltag 2: Efficient and robust fiducial detection. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 4193–4198.

Watt RJ and Phillips WA (2000) The function of dynamic grouping in vision. *Trends in Cognitive Science* 4(12): 447–454.

Xinjilefu X, Feng S, Huang W and Atkeson CG (2014) Decoupled state estimation for humanoids using full-body dynamics. In: *Proceedings of the IEEE International Conference on Robotics and Automotion (ICRA)*. pp. 195–201.

Xu K, Huang H, Shi Y, Li H, Long P, Caichen J, Sun W and Chen B (2015) Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Transactions on Graphics (TOG)* 34(6): 177.

Yan J and Pollefeys M (2006) Automatic kinematic chain building from feature trajectories of articulated objects. In: *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1. pp. 712–719.

Appendices

A The Extended Kalman Filter

The goal in Bayesian recursive state estimation is to determine the probability distribution over the space of

possible current states of a dynamical system conditioned on the signals acquired so far (measurements and control signals), $p(x_t|z_{t:1}, u_{t:1})$, based on the estimated distribution in the previous step, $p(x_{t-1}|z_{t-1:1}, u_{t-1:1})$, and the latest acquired signals, z_t and u_t.

To tackle the recursive estimation of the state of the dynamical system, we need to model the temporal evolution of its state and the relationship between states and measurements. Let's assume we can model them by the following discrete time model with additive Gaussian noise:

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, u_t) + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \sim \mathcal{N}(0, Q_t) \tag{50}$$

$$\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim \mathcal{N}(0, R_t) \tag{51}$$

where $\boldsymbol{x}_t$ is the state at previous time step, u_t is a control input, f and g are possibly non-linear but linearizable functions that represent the system and measurement models, and $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are zero-mean Gaussian distributed process and measurement noises. Let's also assume that the state can be represented by a Gaussian distribution.

The extended Kalman filter (EKF) linearizes the possibly non-linear models using a first order Taylor expansion around the expected state to find a recursive solution:

$$f(\boldsymbol{x}_{t-1}, u_t) \approx f(x_{t-1}, u_t) + f'(x_{t-1}, u_t)(\boldsymbol{x}_{t-1} - x_{t-1}) \tag{52}$$

$$f'(x_{t-1}, u_t) = \left.\frac{\partial f(\boldsymbol{x}, u_t)}{\partial \boldsymbol{x}}\right|_{x_{t-1}} = F_t \tag{53}$$

$$h(\boldsymbol{x}_t) \approx h(x_t) + h'(x_t)(\boldsymbol{x}_t - x_t) \tag{54}$$

$$h'(x_t) = \left.\frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x_t} = H_t \tag{55}$$

Here we use the notation $\left.\frac{\partial f(x)}{\partial x}\right|_{\bar{x}}$ to indicate that we compute the derivative of function f with respect to its variable x, and substitute in the result x by a specific value $\bar{x}$.

Based on the linearization, the EKF predicts the distribution of the next state as:

$$p(x_t|z_{t-1:1}, u_{t:1}) = \mathcal{N}(\hat{x}_t, \hat{P}_t) \tag{56}$$

$$\hat{x}_t = f(x_{t-1}, u_t) \tag{57}$$

$$\hat{P}_t = F_t P_{t-1} F_t^T + Q_t \tag{58}$$

The EKF predicts the distribution of next measurement as:

$$p(z_t|x_t) = \mathcal{N}(\hat{z}_t, \hat{S}_t) \tag{59}$$

$$\hat{z}_t = h(\hat{z}_t) \tag{60}$$

$$\hat{R}_t = H_t \hat{P}_t H_t^T \tag{61}$$

$$\hat{S}_t = \hat{R}_t + R_t \tag{62}$$

where $\hat{R}_t$ is the covariance matrix of the measurement noise and $\hat{S}_t$ is the covariance of the innovation.

Finally, based on the previous definitions and the linearizations, the EKF computes the posterior distribution as:

$$p(x_t|z_{t:1}, u_{t:1}) = \mathcal{N}(x_t, P_t) \tag{63}$$

$$x_t = \hat{x}_t + K_t(z_t - h(\hat{x}_t)) \tag{64}$$

$$P_t = (I - K_t H_t)\hat{P}_t \tag{65}$$

$$K_t = \hat{P}_t H_t^T (H_t \hat{P}_t H_t^T + R_t)^{-1} \tag{66}$$

B Prismatic Joint Measurement Model Linearization

The matrix of derivatives of the measurement model with respect to the state, $H_t^{joint,p}$, is defined by:

$$H_t^{joint,p} = \begin{pmatrix} -q^p s_\phi s_\theta & q^p c_\phi c_\theta & c_\phi s_\theta & 0 \\ q^p c_\phi s_\theta & q^p s_\phi c_\theta & s_\phi s_\theta & 0 \\ 0 & -q^p s_\theta & c_\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{67}$$

where we have made use of the equivalence between spherical (ϕ and θ) and Cartesian (o^p) representations of the orientation vector, and $s_\phi = \sin\phi$, $c_\phi = \cos\phi$, $s_\theta = \sin\theta$, $c_\theta = \cos\theta$.

C Revolute Joint Measurement Model Linearization

The linearization of the measurement model, $H_t^{joint,r}$, is defined as:

$$H_t^{joint,r} = \begin{pmatrix} -q^r c_\phi s_\theta p_z^r & -q^r(s_\theta p_y^r + s_\phi c_\theta p_z^r) & \dots \\ -q^r s_\phi s_\theta p_z^r & q^r(c_\phi c_\theta p_z^r + s_\theta p_x^r) & \dots \\ q^r s_\theta(c_\phi p_x^r + s_\phi p_y^r) & q^r c_\theta(s_\phi p_x^r - c_\phi p_y^r) & \dots \\ -q^r s_\phi s_\theta & q^r c_\phi c_\theta & \dots \\ q^r c_\phi s_\theta & q^r s_\phi c_\theta & \dots \\ 0 & -q^r s_\theta & \dots \end{pmatrix}$$

$$\begin{pmatrix} \dots & 0 & q^r c_\theta & -q^r s_\phi s_\theta & c_\theta p_y^r - s_\phi s_\theta p_z^r & 0 \\ \dots & 0 & 0 & q^r c_\phi s_\theta & c_\phi s_\theta p_z^r - c_\theta p_x^r & 0 \\ \dots & -q^r c_\theta & -q^r c_\theta & -q^r c_\phi s_\theta & s_\theta(s_\phi p_x^r - c_\phi p_y^r) & 0 \\ \dots & q^r s_\phi s_\theta & 0 & 0 & c_\phi s_\theta & 0 \\ \dots & 0 & 0 & 0 & s_\phi s_\theta & 0 \\ \dots & 0 & 0 & 0 & c_\theta & 0 \end{pmatrix} \tag{68}$$

where $s_\phi = \sin\phi$, $c_\phi = \cos\phi$, $s_\theta = \sin\theta$, and $c_\theta = \cos\theta$, and $p^r = (p_x^r, p_y^r, p_z^r)^T$.

D Parameters

Tables 4, 5 and 6 list the most relevant parameters of the perceptual systems of sections 2, 4 and 5 as used in the corresponding experimental sections.

Table 4. Parameters for the visual perception of kinematic properties

Parameter	Description	Value(s)
N	Number of tracked features	150–250*
λ_{min}	Threshold for the smallest eigenvalue of the second order moments of the gradient around a point feature	0.005
σ^2_{min}	Minimum standard deviation of the feature location measurements	1 cm
α_λ	Constant factor for the uncertainty of a feature due to its saliency	5×10^{-4}
α_z	Constant factor for the uncertainty of a feature due to its depth	2.58 mm/m^2
a^2_x, a^2_y, a^2_z	Linear acceleration noise in rigid body motion estimation	0.02 m s^{-1}
$a^2_{rx}, a^2_{ry}, a^2_{rz}$	Angular acceleration noise in rigid body motion estimation	0.2 rad s^{-1}
d^f_{max}	Maximum prediction error for the feature-to-body data association	1.5 cm
L_{disc}	Minimum likelihood for joint models	0.1

Table 5. Parameters for the multimodal perception of kinematic properties

Parameter	Description	Value(s)
α_p	Multiplying factor for the uncertainty of the end-effector pose measurements	0.2
α_v	Multiplying factor for the uncertainty of the end-effector velocity measurements	0.1
Q^{ee}	Covariance of the additive system noise in the estimation of end-effector motion	diag$(0.01, 0.01, 0.01, 0.05, 0.05, 0.05, 0.001, 0.001, 0.001, 0.005, 0.005, 0.005)$
Q^{bent}	Covariance of the additive system noise in the estimation of hand bending	diag$(0.01, 0.01, 0.01, 0.05, 0.05, 0.05)$

Table 6. Parameters for the multimodal perception of dynamic properties

Parameter	Description	Value(s)
N_p	Number of particles	1000
U^p_{low}	Lower bound of the initial uniform distribution for dynamic property particles in a prismatic joint	0 N
U^p_{up}	Upper bound of the initial uniform distribution for dynamic property particles in a prismatic joint	10 N
U^r_{low}	Lower bound of the initial uniform distribution for dynamic property particles in a revolute joint	0 N m
U^r_{up}	Upper bound of the initial uniform distribution for dynamic property particles in a revolute joint	3 N m

E Index to Multimedia Extensions

The table 7 lists the multimedia files accompanying this paper.

Table 7. Index to multimedia extensions

Extension	Multimedia Type	Description
1	Video	Experiments from Section 2.3, evaluating the online interactive perception of kinematics from visual signals: The videos show the visual input to the robot with superimposed results.
2	Video	Experiments from Section 4.4, evaluating the online interactive perception of kinematics from multimodal signals: The videos show the visual input to our robot with superimposed results as well as external and 3D visualizations.
3	Video	An experiment from Section 5.3, evaluating the online interactive perception of dynamic properties: The video shows an interaction of the robot and different steps in the estimation of the stiction and kinetic friction.