2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Factored Pose Estimation of Articulated Objects using Efficient Nonparametric Belief Propagation

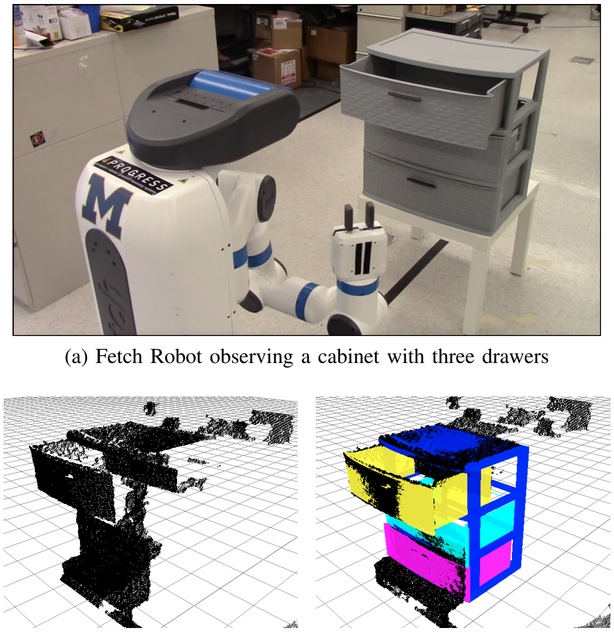
Karthik Desingh[1], Shiyang Lu[1], Anthony Opipari[1], Odest Chadwicke Jenkins[1]

***Abstract*— Robots working in human environments often encounter a wide range of articulated objects, such as tools, cabinets, and other jointed objects. Such articulated objects can take an infinite number of possible poses, as a point in a potentially high-dimensional continuous space. A robot must perceive this continuous pose in order to manipulate the object to a desired pose. This problem of perception and manipulation of articulated objects remains a challenge due to its high dimensionality and multi-modal uncertainty. In this paper, we propose a factored approach to estimate the poses of articulated objects using an efficient nonparametric belief propagation algorithm. We consider inputs as geometrical models with articulation constraints, and observed 3D sensor data. The proposed framework produces object-part pose beliefs iteratively. The problem is formulated as a pairwise Markov Random Field (MRF) where each hidden node (continuous pose variable) models an observed object-part's pose and each edge denotes an articulation constraint between a pair of parts. We propose articulated pose estimation by a Pull Message Passing algorithm for Nonparametric Belief Propagation (PMPNBP) and evaluate its convergence properties over scenes with articulated objects.**

I. Introduction

Robots working in human environments often encounter a wide range of articulated objects, such as tools, cabinets, and other kinematically jointed objects. For example, the cabinet with three drawers shown in Figure 1 functions as a storage container. To accomplish storage and retrieval tasks on this container, a robot would need to perform a sequence of open and close actions on the various drawers. Executing such tasks involves repeated sense-plan-act phases, which occur under uncertainty in the robot's observations and demand a pose estimation framework capable of tracking this uncertainty. The presence of observation uncertainty and environmental occlusions poses a challenge for robots attempting to model cluttered human environments. Additionally, the occurrence of partial sensor observation due to self and environmental occlusions makes the inference problem multi-modal. Further, as the number of object parts in the environment grows, the inference problem becomes high-dimensional.

Pose estimation methods have been proposed that take a generative approach to this problem [1], [2], [3]. These methods aim to explain an observed scene as a collection of object/parts poses using a particle filter formulation to iteratively maintain belief over possible states. Though these approaches hold the power of modeling the world generatively, they have an inherent drawback of scaling inefficiently as the number of rigid bodies being modeled increases. In this paper, we focus on overcoming this drawback by factoring the state as individual object parts constrained by their articulations to create an efficient inference framework for pose estimation.


(a) Fetch Robot observing a cabinet with three drawers

(b) Point cloud observation

(c) Maximum likelihood estimate

Fig. 1: Robot estimating the state of a cabinet with 3 prismatically articulated drawers from a 3D point cloud.

Generative methods exploiting articulation constraints are widely used in human pose estimation problems [4], [5], [6] where human body parts have constrained articulation. We take a similar approach and factor the problem using a Markov Random Field (MRF) formulation where each hidden node in the probabilistic graphical model represents an observed object-part's pose (continuous variable), each observed node indicates the information observed from a particular object-part, and each edge in the graph denotes the articulation constraint between a pair of parts. Inference on the graph is performed using a message passing algorithm that shares information between the parts' pose variables, to produce pose beliefs for each part, collectively giving the estimated state of the articulated object.

Existing message passing approaches [7], [8] represent a message as a mixture of Gaussian components and provide

[1] Department of Electrical Engineering and Computer Science, Robotics Institute, University of Michigan, Ann Arbor {kdesingh,shiyoung,topipari,ocj}@umich.edu

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

Gibbs sampling based techniques to approximate the message product and update operations. Their message representation and message product techniques limit the number of samples used for inference and are not applicable to our application domain that is high-dimensional and multi-modal. In this paper we provide a more efficient "Pull" Message Passing algorithm for Nonparametric Belief Propagation (PMPNBP). The key idea of pull message updating is to evaluate samples taken from the belief of the receiving node with respect to the densities informing the sending node. The mixture product approximation can then be performed individually per sample, and later normalized to form a distribution. This pull updating of message distributions avoids the computational pitfalls of push updating used in [7], [8].

Our system takes a 3D point cloud from sensor measurement and an object geometry model in the form of a URDF (Unified Robot Description Format) as input and outputs belief samples in the continuous pose domain. We use these belief samples to compute a maximum likelihood estimate of an object-part's pose enabling the robot to act on the object. Contributions of this paper include: a) proposal of an efficient belief propagation algorithm (PMPNBP) to estimate articulated object poses, b) articulated object pose estimation experiments and comparisons with a traditional particle filter baseline.

II. Related work

Existing methods in the literature have set out to address the challenge of manipulating articulated objects by robots in complex human environments. Particular focus has been placed on addressing the task of estimating the kinematic models of articulated objects by a robot through interactive perception. Hausman et al. [9] propose a particle filtering approach to estimate articulation models and plan actions that reduce model uncertainty. In [10], Martin et al. suggest an online interactive perception technique for estimating kinematic models by incorporating low-level point tracking and mid-level rigid body tracking with high-level kinematic model estimation over time. Sturm et al. [11], [12] addressed the task of estimating articulation models in a probabilistic fashion by human demonstration of manipulation examples.

All of these approaches discover the articulated object's kinematic model by alternating between action and sensing and are important methods for a robot to reliably interact with novel articulated objects. In this paper we assume that such kinematic models once learned for an object can be reused to localize their articulated pose under real world ambiguous observations. The method proposed in this paper could compliment the existing body of work towards task completion in unstructured human environments.

Existing filtering based articulated object tracking frameworks [13], [14], [15] are initialized with ground truth object poses. Our method could complement these existing tracking frameworks by providing an initial pose estimate. Additionally, belief propagation is applied to articulated pose tracking after initial pose estimation [4], [5]. We consider comparisons with the tracking frameworks as a direction for future work.

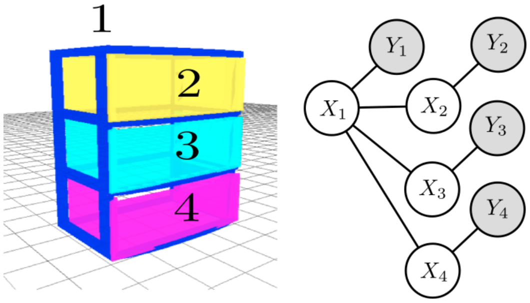


Fig. 2: Cabinet with 3 drawers connect to its frame is converted to a probabilistic graphical model with hidden nodes X_s representing the pose of the object-parts and observed nodes Y_s connected to each of the hidden nodes.

Probabilistic graphical model representations such as Markov random fields (MRF) are widely used in computer vision problems where the variables take discrete labels such as foreground/background. Many algorithms have been proposed to compute the joint probability of the graphical model. Belief propagation algorithms are guaranteed to converge on tree-structured graphs. For graph structures with loops, Loopy Belief Propagation (LBP) [16] is empirically proven to perform well for discrete variables. The problem becomes non-trivial when the variables take continuous values. Sudderth et.al (NBP) [8] and Particle Message Passing (PAMPAS) by Isard et.al [7] provide sampling approaches to perform belief propagation with continuous variables. Both of these approaches approximate a continuous function as a mixture of weighted Gaussians and use local Gibbs sampling to approximate the product of mixtures. NBP has been effectively used in applications such as human pose estimation [4] and hand tracking [5] by modelling the graph as a tree structured particle network. Scene understanding problems where a scene is composed of household objects with articulations demand a large number of sampled hypotheses to infer in the high-dimensional and multi-modal state space. The algorithm proposed in this paper produces promising results and shown to handle such demands. We reported comparisons with an existing NBP algorithm [7] in [17] with 2D examples.

Model based generative methods [18], [19], [20] are increasingly being used to solve scene estimation problems where heuristics from discriminative approaches [21], [22] are used to infer object poses. These approaches do not model object-object interactions or articulations and rely significantly on the effectiveness of the discriminative methods. Our framework doesn't rely on any prior detections but can benefit from them while inherently handling noisy priors [8], [7], [17]. Chua et. al [23] proposed a scene grammar representation and belief propagation over factor graphs, for generating scenes with multiple-objects satisfying the scene grammars. While their objective is similar to ours, we specifically deal with 3D observations along with continuous variables.

III. Problem Statement

We consider an articulated object O to be comprised of N object-parts and $N-1$ points of articulation. Such an object description conforms to the Unified Robot Description Format (URDF) commonly used in the Robot Operating System (ROS) [24]. A kinematic model of this format can be represented as an undirected graph $G=(V,E)$ with nodes V for object-parts and edges E for points of articulation. If G is a Markov Random Field (MRF), it may contain two types of variables X and Y, representing hidden and observed variables respectively. Let $Y=\{Y_s \mid Y_s \in V\}$, where $Y_s = P_s \subseteq P$, with P being a point cloud observed by the robot's 3D sensor. Each object-part has an observed node in the graph G. P_s serves as a region of interest if a trained object detector is used to find the object in the scene, but is optional in our current approach. Each observed node Y_s is connected to a hidden node X_s that represents the pose of the underlying object part. Let $X=\{X_s \mid X_s \in V\}$, where $X_s \in \mathbb{H}_D$ is a dual quaternion pose of an object-part. Dual quaternions [25], [26] are a quaternion equivalent to dual numbers representing a 6D pose $X_s=(x,y,z,q_w,q_x,q_y,q_z)$ as $X_s = q_r + \epsilon q_d$ where q_r is the real component and q_d is the dual component. Alternatively it is represented as $X_s=[q_r][q_d]$. Constructing a dual quaternion X_s is similar to rotation matrices, with a product of dual quaternions representing translation and orientation as $X_s = dq_{pos} * dq_{ori}$, where $*$ is a dual quaternion multiplication. $dq_{ori}=[q_w,q_x,q_y,q_z][0,0,0,0]$ is the dual quaternion representation of pure rotation and $dq_{pos}=[1,0,0,0][0,\frac{x}{2},\frac{y}{2},\frac{z}{2}]$ is the dual quaternion representation of pure translation. This dual quaternion representation is widely used for rigid body kinematics, where the $*$ operation is efficient and elegant compared with matrix multiplication. In addition to representing the hidden variable X_s, dual quaternions can capture the constraints in the edges E and represent articulation types such as prismatic, revolute, and fixed effectively. This will be discussed in detail in Section IV-D.2.

Pose estimation of the articulated object involves inferring the hidden variables X_s that maximize the joint probability of the graph G considering only second order cliques, and is given as:

$$p(X,Y)=\frac{1}{Z}\prod_{(s,t)\in E}\psi_{s,t}(X_s,X_t)\prod_{s\in V}\phi_s(X_s,Y_s) \quad (1)$$

where $\psi_{s,t}(X_s,X_t)$ is the pairwise potential between nodes X_s and X_t, $\phi_s(X_s,Y_s)$ is the unary potential between the hidden node X_s and observed node Y_s, and Z is a normalizing factor. The problem is to infer belief over the possible articulation poses assigned to hidden variables X that are continuous, such that the joint probability is maximized. This inference is generally performed by passing messages between hidden variables X until convergence of their belief distributions over several iterations. After convergence, a maximum likelihood estimate of the marginal belief gives the pose estimate X_s^{est} of an object-part corresponding to the node in the graph G. The collection of all such object-part pose estimates forms the entire object's pose estimate.

IV. Nonparametric Belief Propagation

A. Overview

A message is denoted as $m_{t\to s}$ directed from node t to node s if there is an edge between the nodes in the graph G. The message represents the distribution of what node t thinks node s should take in terms of the hidden variable X_s. Typically, if X_s is in the continuous domain, then $m_{t\to s}(X_s)$ is represented as a Gaussian mixture to approximate the real distribution:

$$m_{t\to s}(X_s)=\sum_{i=1}^{M} w_{ts}^{(i)}\mathcal{N}(X_s;\mu_{ts}^{(i)},\Lambda_{ts}^{(i)}) \quad (2)$$

where $\sum_{i=1}^{M} w_{ts}^{(i)} = 1$, M is the number of Gaussian components, $w_{ts}^{(i)}$ is the weight associated with the i^{th} component, $\mu_{ts}^{(i)}$ and $\Lambda_{ts}^{(i)}$ are the mean and covariance of the i^{th} component, respectively. We use the terms components, particles and samples interchangeably in this paper. Hence, a message can be expressed as M triplets:

$$m_{t\to s}=\{(w_{ts}^{(i)},\mu_{ts}^{(i)},\Lambda_{ts}^{(i)}):1\le i\le M\} \quad (3)$$

Assuming the graph has a tree or loopy structure, computing these message updates is nontrivial computationally. The message update at iteration n in a continuous domain from node $t\to s$ is given by

$$m_{t\to s}^{n}(X_s)\leftarrow \int_{X_t\in\mathbb{H}_D}\left(\psi_{st}(X_s,X_t)\phi_t(X_t,Y_t)\prod_{u\in\rho(t)\backslash s} m_{u\to t}^{n-1}(X_t)\right)dX_t \quad (4)$$

where $\rho(t)$ is the set of neighbor nodes of t. The marginal belief over each hidden node at iteration n is given by

$$\begin{gathered} bel_s^n(X_s)\propto\phi_s(X_s,Y_s)\prod_{t\in\rho(s)} m_{t\to s}^n(X_s) \\ bel_s^n=\{(w_s^{(i)},\mu_s^{(i)},\Lambda_s^{(i)}):1\le i\le T\}\end{gathered} \quad (5)$$

where T is the number of components used to represent the belief.

B. "Push" Message Update

NBP [8] provides a Gibbs sampling approach to compute an approximation of the product $\prod_{u\in\rho(t)\backslash s} m_{u\to t}^{n-1}(X_t)$. Assuming that $\phi_t(X_t,Y_t)$ is pointwise computable, a "pre-message" [27] is defined as

$$M_{t\to s}^{n-1}(X_t)=\phi_t(X_t,Y_t)\prod_{u\in\rho(t)\backslash s} m_{u\to t}^{n-1}(X_t) \quad (6)$$

which can be computed in the Gibbs sampling procedure. This reduces Equation 4 to

$$m_{t\to s}^{n}(X_s)\leftarrow\int_{X_t\in\mathbb{R}^b}\left(\psi_{st}(X_s,X_t)M_{t\to s}^{n-1}(X_t)\right)dX_t \quad (7)$$

Algorithm - Message update

```
Given input messages $m^{n-1}_{u\to t}(X_t) = \{(\mu^{(i)}_{ut}, w^{(i)}_{ut})\}^M_{i=1}$ for each $u \in \rho(t) \setminus s$, and methods to compute functions $\psi_{ts}(X_t, X_s)$ and $\phi_t(X_t, Y_t)$ point-wise, the algorithm computes $m^n_{t\to s}(X_s) = \{(\mu^{(i)}_{ts}, w^{(i)}_{ts})\}^M_{i=1}$
    1. Draw $M$ independent samples $\{\mu^{(i)}_{ts}\}^M_{i=1}$ from $bel^{n-1}_s(X_s)$.
        (a) If $n = 1$ the $bel^0_s(X_s)$ is a uniform distribution or informed by a prior distribution.
        (b) If $n > 1$ the $bel^{n-1}_s(X_s)$ is a belief computed at $(n-1)^{th}$ iteration using importance sampling.
    2 For each $\{\mu^{(i)}_{ts}\}^M_{i=1}$, compute $w^{(i)}_{ts}$
        a Sample $\hat{X}^{(i)}_t \sim \psi_{ts}(X_t, X_s = \mu^{(i)}_{ts})$
        b Unary weight $w^{(i)}_{unary}$ is computed using $\phi_t(X_t = \hat{X}^{(i)}_t, Y_t)$.
        c Neighboring weight $w^{(i)}_{neigh}$ is computed using $m^{n-1}_{u\to t}$.
            (i) For each $u \in \rho(t) \setminus s$ compute $W^{(i)}_u = \sum^M_{j=1} w^{(j)}_{ut} w^{(ij)}_u$ where
                $w^{(ij)}_u = \psi_{ts}(X_s = \mu^{(i)}_{ts}, X_t = \mu^{(j)}_{ut})$.
            (ii) Each neighboring weight is computed by $w^{(i)}_{neigh} = \prod_{u\in\rho(t)\setminus s} W^{(i)}_u$
        d The final weights are computed as $w^{(i)}_{ts} = w^{(i)}_{neigh} \times w^{(i)}_{unary}$.
    3 The weights $\{w^{(i)}_{ts}\}^M_{i=1}$ are associated with the samples $\{\mu^{(i)}_{ts}\}^M_{i=1}$ to represent $m^n_{t\to s}(X_s)$.
```

Algorithm - Belief update

```
Given incoming messages $m^n_{t\to s}(X_t) = \{(w^{(i)}_{ts}, \mu^{(i)}_{ts})\}^M_{i=1}$ for each $t \in \rho(s)$, and methods to compute functions $\phi_s(x_s, y_s)$ point-wise, the algorithm computes $bel^n_s(X_s) \propto \phi_s(X_s, Y_s) \prod_{t\in\rho(s)} m^n_{t\to s}(X_s) = \{(w^{(i)}_s, \mu^{(i)}_s)\}^T_{i=1}$
    1 For each $t \in \rho(s)$
        a Update weights $w^{(i)}_{ts} = w^{(i)}_{ts} \times \phi(X_s = \mu^{(i)}_{ts}, Y_s)$.
        b Normalize the weights such that $\sum^M_{i=1} w^{(i)}_{ts} = 1$.
    2 Combine all the incoming messages to form a single set of samples and their weights $\{(w^{(i)}_s, \mu^{(i)}_s)\}^T_{i=1}$, where $T$ is the sum of all the incoming number of samples.
    3 Normalize the weights such that $\sum^T_{i=1} w^{(i)}_s = 1$.
    4 Perform a resampling step followed by diffusion with Gaussian noise, to sample new set $\{\mu^{(i)}_s\}^T_{i=1}$ that represent the marginal belief of $X_s$.
```

NBP [8] sample $\hat{X}^{(i)}_t$ from the "pre-message" followed by a pairwise sampling where $\psi_{st}(X_s, X_t)$ is acting as $\psi_{st}(X_s | X_t = \hat{X}^{(i)}_t)$ to get a sample $\hat{X}^{(i)}_s$.

The Gibbs sampling procedure is itself an iterative procedure and hence makes the computation of the "pre-message" (as the Foundation function described for PAMPAS) more expensive as M increases.

C. "Pull" Message Update

Given the overview of Nonparametric Belief Propagation above in Section IV-A, we now describe our "pull" message passing algorithm. We represent each message as a set of pairs instead of triplets as in Equation 3, which is

$$m_{t\to s} = \{(w^{(i)}_{ts}, \mu^{(i)}_{ts}) : 1 \leq i \leq M\} \tag{8}$$

Similarly, the marginal belief is summarized as a sample set

$$bel^n_s(X_s) = \{\mu^{(i)}_s : 1 \leq i \leq T\} \tag{9}$$

where T is the number of samples representing the marginal belief. We assume there exists a marginal belief over X_s, as $bel^{n-1}_s(X_s)$, from the previous iteration. To compute the $m^n_{t\to s}(X_s)$ at iteration n, we initially sample $\{\mu^{(i)}_{ts}\}^M_{i=1}$ from the belief $bel^{n-1}_s(X_s)$. We then pass these samples to the neighboring nodes $\rho(t) \setminus s$ and compute weights $\{w^{(i)}_{ts}\}^M_{i=1}$. This step is described in Algorithm - Message update. The computation of $bel^n_s(X_s)$ is described in Algorithm - Belief update. The key difference between the "push" approach of earlier methods (NBP [8] and PAMPAS [7]) and our "pull" approach is the message $m_{t\to s}$ generation procedure. In the "push" approach, incoming messages to t determine the outgoing message $t \to s$. Whereas in the "pull" approach, samples representing s are drawn from its belief bel_s at the previous iteration and weighted by the incoming messages to t. This weighting strategy is computationally efficient. Additionally, the product of incoming messages to compute bel_s is approximated by a resampling step as described in Algorithm - Belief update.

D. Potential Functions

1) Unary potential: Unary potential $\phi_t(X_t, Y_t)$ is used to model the likelihood by measuring how pose X_t explains the point cloud observation P_t. The hypothesized object pose X_t is used to position the given geometric object model and generate a synthetic point cloud P^*_t that can be matched with the observation P_t. The synthetic point cloud is constructed using the object-part's geometric model available *a priori*. The likelihood is calculated as

$$\phi_t(X_t, Y_t) = e^{\lambda_r d(P_t, P^*_t)} \tag{10}$$

where λ_r is the scaling factor, $d(P_t, P^*_t)$ is the sum of 3D Euclidean distance between the observed point $p \in P_t$ and rendered point $p^* \in P^*_t$ at each pixel location in the region of interest.

2) Pairwise Potential and Sampling: Pairwise potential $\psi_{t,s}(X_t|X_s)$ gives information about how compatible two object poses are given their joint articulation constraints captured by the edge between them. As mentioned in Section III, these constraints are captured using dual quaternions. Most often, the joint articulation constraints have minimum and maximum range in either prismatic or revolute types. We capture this information from URDF to get $R_{t|s} = [dq^a_{t|s}, dq^b_{t|s}]$ giving the limits of articulations. For a given X_s and $R_{t|s}$, we find the distance between X_t and the limits as $A = d(X_t, dq^a_{t|s})$ and $B = d(X_t, dq^b_{t|s})$, as well as the

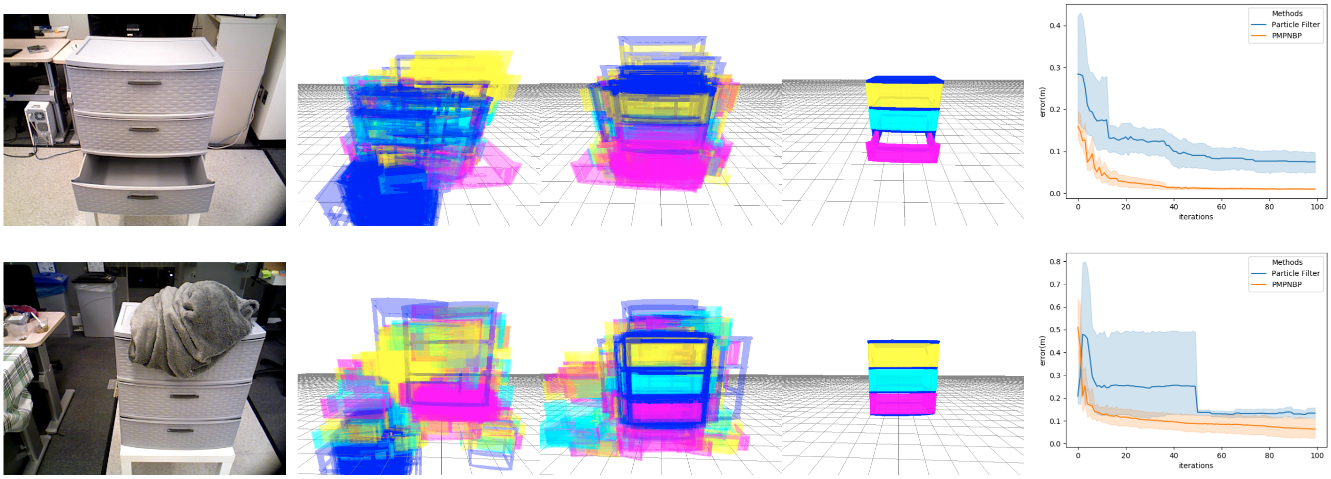


Fig. 3: Convergence of pose estimation on two different scenes: the first column shows the RGB image of each scene, second to fourth columns show the convergence results of PMPNBP. The second column shows randomly initialized belief particles, the third column shows the belief particles after 100 iterations, and the fourth column shows the maximum likelihood estimates of each part. The fifth column shows the estimation error (0.95 confidence interval) using PMPNBP with respect to the baseline particle filter method across 10 runs (400 particles and 100 iterations each). It can be seen that the baseline suffers from local minimas while PMPNBP is able to recover from them effectively.

distance between the limits $C = d(dq^a_{t|s}, dq^b_{t|s})$. Using a joint limit kernel parameterized by $(\sigma_{pos}, \sigma_{ori})$, we evaluate the pairwise potential as:

$$\psi_{t,s}(X_t|X_s) = e^{-\frac{(A_{pos}+B_{pos}-C_{pos})^2}{2(\sigma_{pos})^2} - \frac{(A_{ori}+B_{ori}-C_{ori})^2}{2(\sigma_{ori})^2}} \tag{11}$$

The pairwise sampling uses the same limits $R_{t|s}$ to sample for X_t given an X_s. We uniformly sample a dual quaternion $\bar{X}_t$ that is between $[dq^a_{t|s}, dq^b_{t|s}]$ and transform it back to the X_s's current frame of reference by $X_t = X_s * \bar{X}_t$.

V. Experiments and Results

A. Experimental Setup

We use a Fetch robot, a mobile manipulation platform for our data collection. 3D data is collected using an ASUS Xtion RGBD sensor mounted on the robot. We make use of the intrinsic and extrinsic parameters of the sensor. We use CUDA-OpenGL interoperation to render synthetic scenes on a large set of poses in a single render buffer on a GPU. We render scenes as depth images, then project them back to 3D point clouds via camera intrinsic parameters.

We use a cabinet with three drawers and Fetch robot as our articulated objects to evaluate our method. A CAD model of the objects were obtained from the Internet and annotations of the object's articulations were added manually using Blender to generate a URDF model (Fetch robot comes with URDF model). Obtaining geometrical models and articulation models can either be crowd-sourced [28] or learned using human or robot interactions [10].

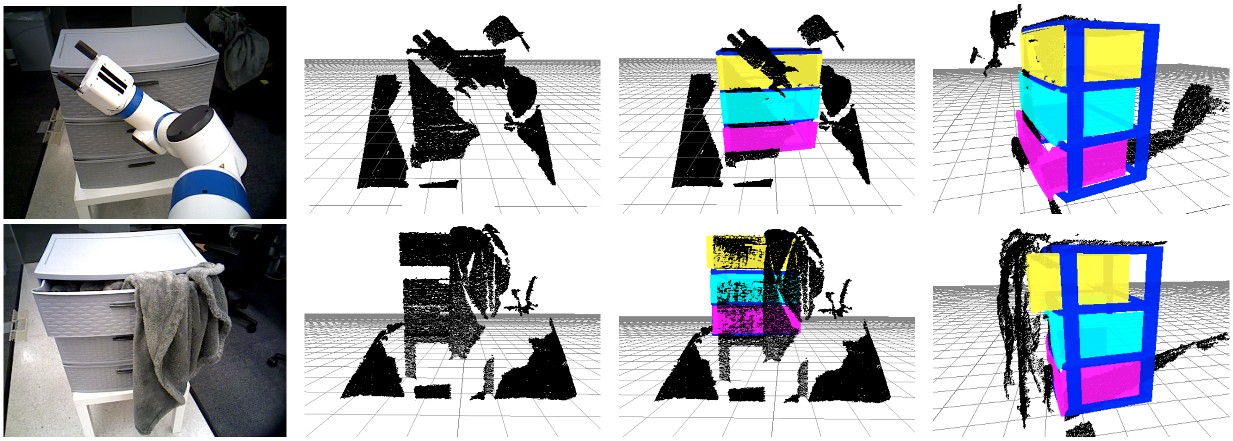
B. Baseline

We implemented a Monte Carlo localization (particle filter) method that includes an object specific state representation. For example, the Cabinet with 3 drawers has a state representation of $(x, y, z, \phi, \psi, \chi, t_a, t_b, t_c)$ where the first 6 elements describe the 6D pose of the object in the world and t_a, t_b, t_c represent the prismatic articulation. The measurement model in the implementation uses the unary potential described in Section IV-D.1. Instead of rendering a point cloud of each object-part, the entire object in the hypothesized pose is rendered for measuring the likelihood in the particle filter. As the observations are static, the action model in the standard particle filter is replaced with a Gaussian diffusion over the object poses.

C. Convergence Results

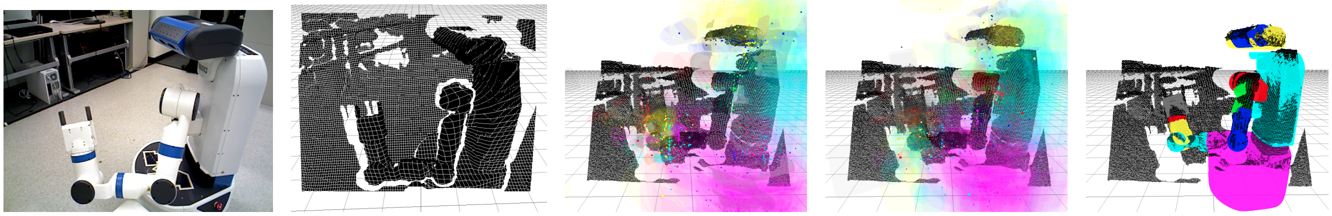
In Figure. 3, we show the convergence of the proposed method visually for two scenes containing different point cloud observations. We collected point cloud observations of the cabinet object in arbitrary poses and performed inference using both the proposed PMPNBP and the baseline Monte Carlo localization. The entire point cloud measurement is used as the observation for all object-parts. The first column shows the scene (RGB not used during inference). The second column shows the uniformly initialized pose samples of the object-parts over the entire point cloud. The third column shows the propagated belief particles for each object-part after 100 iterations. The fourth column shows the Maximum Likelihood Estimate (MLE) of each object-part using the belief particles from the third column.

For the results shown in Figure. 3, we ran our inference for 100 iterations with 400 particles per message. 10 different trials were used to generate the convergence plot that shows the mean and variance in error across the trials. We adopt the average distance metric (ADD) proposed in [29], [20] for comparison between the methods. The point cloud model of the object-part is transformed to its ground truth dual quaternion (dq) and to the estimated pose's dual quaternion



Original Scene Incomplete Observation MLE using PMPNBP MLE from a different view

Fig. 4: Partial and incomplete observations due to self and environmental occlusions are handled by PMPNBP in estimating plausible pose with accuracy



(a) Fetch robot (b) Point cloud observation (c) Belief at iteration 1 (d) Belief at iteration 1000 (e) MLE at iter 1000

Fig. 5: Factored pose estimation using PMPNBP extends to articulated objects such as a Fetch robot (a) which has 12 nodes and 11 edges in the probabilistic graphical model. For a scene (a), which has partial 3D point cloud observation (b), the PMPNBP message passing algorithm, propagates the belief samples from iteration 1 (c) to iteration 1000 (d), that leads to MLE (e). Video with graphical model and iterative covergence - `https://youtu.be/eKdoC8Mq46U`.

($\bar{dq}$). Error is calculated as the pointwise distance of these transformation pairs normalized by the number of points in the model point cloud.

$$ADD = \frac{1}{m} \sum_{p \in \mathcal{M}} \|\bar{dq} * p * \bar{dq_c} - dq * p * dq_c\| \qquad (12)$$

where ($\bar{dq_c}$) and (dq_c) are the conjugates of the dual quaternions [25], [26], m is the number of 3D points in the model set $\mathcal{M}$.

D. Partial and incomplete observations

Articulated models suffer from self-occlusions and often environmental occlusions. By exploiting the articulation constraints of an object in the pose estimation, our inference method is able to produce a physically plausible pose that explains the partial or incomplete observations. In Figure. 4, we show two compelling cases that indicate the strength of our inference method. In the first case, the cabinet is occluded by the robot's arm, while in the second case, a blanket suspended from drawer 1 occludes half of the object. PMPNBP is able to recover from these occlusions and produce a plausible estimate along with belief of possible poses. The factored approach proposed in this paper scales to objects such as a Fetch robot with higher number of links and joints with combinations of articulations compared to a cabinet (see Figure. 5 and [30] for extended results).

VI. Conclusion

We proposed Pull Message Passing algorithm for Nonparametric Belief Propagation (PMPNBP), an efficient algorithm to estimate the poses of articulated objects. This problem was formulated as a graph inference problem for a Markov Random Field (MRF). We showed that the PMPNBP outperforms a baseline Monte Carlo localization method quantitatively. Qualitative results were provided to show the pose estimation accuracy of PMPNBP under a variety of occlusions. We also showed the scalability of the algorithm to articulated objects such as a Fetch robot. The notion of uncertainty in the inference is inevitable in robotic perception. Our proposed PMPNBP algorithm is able to accurately estimate the pose of articulated objects and maintain belief over possible poses that can benefit a robot in performing manipulation tasks.

Acknowledgment

This work was supported in part by NSF award IIS 1638047. The authors also thank Zhen Zeng, Zhiqiang Sui and Adrian Roefer for helpful discussions and feedback.

References

[1] Z. Sui, L. Xiang, O. C. Jenkins, and K. Desingh, "Goal-directed robot manipulation through axiomatic scene estimation," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 86–104, 2017.

[2] K. Desingh, O. C. Jenkins, L. Reveret, and Z. Sui, "Physically plausible scene estimation for manipulation in clutter," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 1073–1080.

[3] Z. Zeng, Z. Zhou, Z. Sui, and O. C. Jenkins, "Semantic robot programming for goal-directed manipulation in cluttered scenes," in *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2018.

[4] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, pp. 421–428.

[5] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using nonparametric belief propagation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, 2004, pp. 189–189.

[6] M. Vondrak, L. Sigal, and O. C. Jenkins, "Dynamical simulation priors for human motion tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 1, pp. 52–65, 2013.

[7] M. Isard, "PAMPAS: Real-valued graphical models for computer vision," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. 613–620.

[8] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, p. 605.

[9] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme, "Active articulation model estimation through interactive perception," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3305–3312.

[10] R. M. Martin and O. Brock, "Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 2494–2501.

[11] J. Sturm, C. Stachniss, and W. Burgard, "A probabilistic framework for learning kinematic models of articulated objects," *Journal of Artificial Intelligence Research*, vol. 41, pp. 477–526, 2011.

[12] J. Sturm, *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots*. Springer, 2013.

[13] J. Brookshire and S. J. Teller, "Articulated pose estimation using tangent space approximations," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 5–29, 2016.

[14] C. G. Cifuentes, J. Issac, M. Wüthrich, S. Schaal, and J. Bohg, "Probabilistic articulated real-time tracking for robot manipulation," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 577–584, 2017.

[15] T. Schmidt, R. A. Newcombe, and D. Fox, "DART: dense articulated real-time tracking," in *Robotics: Science and Systems*, 2014.

[16] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 1999, pp. 467–475.

[17] K. Desingh, A. Opipari, and O. C. Jenkins, "Pull message passing for nonparametric belief propagation," *arXiv preprint arXiv:1807.10487*, 2018. [Online]. Available: http://arxiv.org/abs/1807.10487

[18] V. Narayanan and M. Likhachev, "Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances." in *Robotics: Science and Systems*, 2016.

[19] Z. Sui, Z. Zhou, Z. Zeng, and O. C. Jenkins, "SUM: Sequential scene understanding and manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[20] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 91–99.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[23] J. Chua and P. F. Felzenszwalb, "Scene grammars, factor graphs, and belief propagation," *arXiv preprint arXiv:1606.01307*, 2016. [Online]. Available: https://arxiv.org/abs/1606.01307

[24] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[25] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "A new probability distribution for simultaneous representation of uncertain position and orientation," in *17th IEEE International Conference on Information Fusion (FUSION)*, 2014, pp. 1–7.

[26] B. Kenwright, "A beginners guide to dual-quaternions," *WSCG'2012*, 2012.

[27] A. Ihler and D. McAllester, "Particle belief propagation," in *Artificial Intelligence and Statistics*, 2009, pp. 256–263.

[28] S. R. Gouravajhala, J. Yim, K. Desingh, Y. Huang, O. C. Jenkins, and W. S. Lasecki, "Eureca: Enhanced understanding of real environments via crowd assistance," in *Sixth AAAI Conference on Human Computation and Crowdsourcing*, 2018.

[29] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[30] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, "Factored pose estimation of articulated objects using efficient nonparametric belief propagation," *arXiv preprint arXiv:1812.03647*, 2018. [Online]. Available: https://arxiv.org/abs/1812.03647