COMPUTER VISION

Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects

Karthik Desingh[1]*, Shiyang Lu[2], Anthony Opipari[1], Odest Chadwicke Jenkins[1,2]

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 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 multimodal uncertainty. Here, we describe a factored approach to estimate the poses of articulated objects using an efficient approach to nonparametric belief propagation. We consider inputs as geometrical models with articulation constraints and observed RGBD (red, green, blue, and depth) sensor data. The described 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) is an observed object-part's pose and the edges denote the articulation constraints between the parts. We describe articulated pose estimation by a "pull" message passing algorithm for nonparametric belief propagation (PMPNBP) and evaluate its convergence properties over scenes with articulated objects. Robot experiments are provided to demonstrate the necessity of maintaining beliefs to perform goal-driven manipulation tasks.

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 Fig. 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.

In addition, the occurrence of partial sensor observation due to self and environmental occlusions makes the inference problem multimodal. 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 aimed to explain an observed scene as a collection of rigid body poses using a particle filter formulation to iteratively maintain belief over possible states. Although these approaches held the power of modeling the world generatively, they have an inherent drawback of scaling inefficiently when the number of rigid bodies being modeled increases. Here, we focus on overcoming this drawback by factoring the state as individual rigid bodies (object parts) constrained by their articulations to create an efficient inference framework for pose estimation.

In existing literature, particular focus has been placed on addressing the task of estimating the kinematic models of articulated objects by a robot through interactive perception (*4*). Hausman *et al.* (*5*) proposed a particle filtering approach to estimate articulation models and plan actions that reduce model uncertainty. In (*6*), Martin *et al.* suggested 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.* (*7*, *8*) 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. Here, we assumed 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 complement the existing body of work toward task completion in unstructured human environments.

Generative methods exploiting articulation constraints are widely used in human pose estimation problems (*9*, *10*, *11*) where human body parts have constrained articulation. We took a similar approach and factored 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 was performed using a message-passing algorithm that shared information between the parts' pose variables, to produce their pose beliefs, collectively giving the estimated state of the articulated object.

Many algorithms have been proposed to compute the joint probability of an MRF. For tree-structured graphs, belief propagation algorithms are guaranteed to converge. For graph structures with loops, the loopy belief propagation (*12*) has been empirically proven to perform well for discrete variables. The problem becomes nontrivial when the variables take continuous values. Nonparametric belief propagation (NBP) by Sudderth *et al.* (*13*) and particle message passing (PAMPAS) by Isard (*14*) provided sampling approaches to perform belief propagation with continuous variables. Both approaches approximated a continuous function as a Gaussian mixture and used local Gibbs sampling to approximate the product of mixtures. However, these methods can hardly be generalized to three-dimensional (3D) articulated pose

[1]Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48105, USA. [2]Robotics Institute, University of Michigan, Ann Arbor, MI 48105, USA.
*Corresponding author. Email: kdesingh@umich.edu

Fig. 1. A robot needs to estimate the pose of a cabinet to operate on it.

estimation problems because of their high computational expense. Here, we propose a more efficient "pull" message passing algorithm for nonparametric belief propagation (PMPNBP). The key idea of message updating is to evaluate samples taken from the belief of the "pull" receiving node with respect to the densities informing the sending node. The approximation of mixture products can be performed individually per sample and then normalized into a distribution. Our method avoids the computational pitfalls of "push" updating used in NBP (*13*) and PAMPAS (*14*), and we show that our method could be used for 3D articulated pose estimation with compelling examples.

Some recent works have addressed the computational efficiency of NBP. Similar in spirit to our PMPNBP, Ihler and McAllester (*15*) described a conceptual theory of particle belief propagation, where a target node's samples were used to generate a message going from the source to the target. This work emphasized the advantages of using large number of particles to represent incoming messages, along with theoretical analysis. It used an expensive iterative Markov chain Monte Carlo sampling step, mimicking the Gibbs sampling step in NBP (*14*, *13*). PMPNBP is able to avoid this cost through a resampling step with linear computational cost. Specifically, our complexity is $O(DM)$ in computing a message mixture of M components using D incoming mixtures as compared with $O(DKM^2)$ of NBP (*14*, *13*) with K Gibbs iterations.

Kernel-based methods have been proposed to improve the efficiency of NBP. Song *et al.* (*16*) proposed a kernel belief propagation method. Messages in this work were represented as functions in a reproducing Kernel Hilbert space (RKHS), and message updates are linear operations in RKHS. Results presented in this work claimed to be more accurate and faster than NBP with Gibbs sampling (*14*, *13*) and particle belief propagation (*15*) over applications such as image denoising, depth prediction, and angle prediction in a protein-folding problem. Their complexity was $O(D_{\max}L^2)$ with L being the number of basis vectors in kernel space and $D_{\max}$ being maximum degree of a node in the graphical model. We consider comparisons with kernel-based approximators a direction for future work.

Han *et al.* (*17*) introduced sequential density approximation by mode propagation to approximate the slow sampling-based products in NBP (*14*, *13*). The complexity of the sequential density approximation was $O(M^4(D-1))$. However, their approach was limited to non-occluded observations unlike the NBP methods (*14*, *13*). We evaluated our PMPNBP algorithm's convergence characteristics and computational gain over PAMPAS (*14*) on their 2D articulated pattern estimation. We empirically show that PMPNBP enabled faster convergence to comparable convergence characteristics. This greater scaling of message mixture components for improved accuracy makes PMPNBP applicable to 3D articulated object pose estimation for robot manipulation tasks.

For estimating the pose of the 3D articulated objects, our system took 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 used these belief samples to compute a maximum likelihood estimate (MLE) of an object-part's pose enabling the robot to act on the object. The performance of the system was evaluated by articulated object pose estimation experiments and comparisons with a traditional particle filter baseline. This baseline is similar to the iterative generative inference methods (*1*, *2*, *3*) used in object pose estimation. The quantitative evaluation in comparison to particle filter baseline shows that PMPNBP has better convergence characteristics under various occlusion scenarios. The MRF-based factored formulation of PMPNBP let the observed parts of the articulated objects maintain reasonable belief over the parts that were occluded. In addition to this comparison, we qualitatively show the scalability of the PMPNBP to articulated objects with a large number of nodes and edges by estimating the pose of a Fetch robot.

A simple task is illustrated to show how the belief propagation informs a task planner to choose an information gain action and overcome uncertainty in the perceptual estimation. This task shows the benefit of the belief propagation approach toward manipulation tasks.

Contributions of this paper include (i) proposal of an efficient belief propagation algorithm (PMPNBP), (ii) qualitative and quantitative comparisons of PMPNBP with PAMPAS (*14*), (iii) articulated object pose estimation experiments and comparisons with traditional particle filter baseline, and (iv) a belief representation from perception to inform a task planner.

RESULTS

In this section, we show the results on pose estimation of articulated objects, followed by robot experiment illustrating the benefits of belief propagation. We then show the comparison of the proposed PMPNBP algorithm with PAMPAS (*14*) and discuss the computational gain obtained.

Articulated cabinet pose estimation

We first show pose estimation results of the cabinet in Fig. 1 using the PMPNBP algorithm proposed in this paper. Because three drawers are prismatically jointed with the cabinet frame, the whole cabinet could be abstracted using an MRF with four hidden nodes and four observed nodes (Fig. 2). Messages between each pair of connected hidden nodes were represented in 400 particles, where each particle represented the pose of the receiving node. The RGB (red, green, and blue) observation (Fig. 3A) was only used for illustrative purpose here, whereas the corresponding point cloud (Fig. 3B) observed by the sensor was used as input without any segmentation. Belief particles for each part of the cabinet, which were uniformly initialized on the entire point cloud, gradually converged close to the ground truth (Fig. 3C) after 100 iterations. We used the MLE of each part to compose the final pose of the cabinet (Fig. 1D). PMPNBP is an iterative message-passing algorithm, and the belief particles propagate to match the observation while maintaining the articulation constraint.

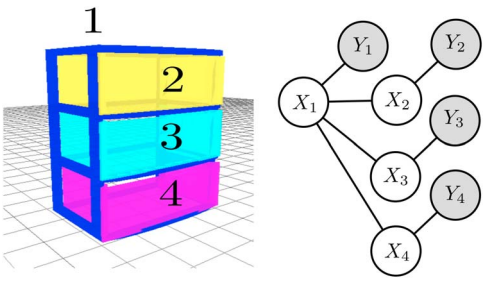


Fig. 2. Probabilistic graphical model of an articulated object. A cabinet with three drawers is converted to a probabilistic graphical model with hidden nodes X_s representing the poses of different parts and observed nodes Y_s connected to each of the hidden nodes. Edges between the hidden nodes capture the articulation constraints between them.

We compared our PMPNBP method with a standard particle filter method, which is a commonly used iterative algorithm for pose estimation problems (*1*, *2*, *3*). We implemented a Monte Carlo localization (particle filter) method that included an object-specific state representation. For example, the cabinet with three drawers has a state representation of $(x, y, z, \phi, \psi, \chi, t_a, t_b, t_c)$, where the first six 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 used the unary potential described in the "Potential functions" section. Instead of rendering a point cloud of each object part, the entire object in the hypothesized pose was rendered for measuring the likelihood in the particle filter. Because the observations were static, the action model in the standard particle filter was replaced with a Gaussian diffusion over the object poses. With the same number of particles (400) for both the proposed PMPNBP and the baseline Monte Carlo localization, we ran the belief propagation for 100 iterations. The entire point cloud measurement was used as the observation for all object parts. With the same point cloud observation and experiment settings, we ran the inference 10 times to generate the convergence plot. Figure 3E shows the mean and variance in error across iterations. The proposed method has better convergence properties with respect to the particle filter baseline. Especially when the cabinet suffered from external occlusion (blanket on the cabinet frame), the particle filter often fell into local minima and hardly recovered. The state representation of the particle filter was specific to the articulated object (cabinet), and the length of this representation grew polynomially with the number of links. This makes the search space of the particle filter grow exponentially. For the cabinet case, the run time of the particle filter was faster comparable to PMPNBP; however, the convergence was often to a local minima compared with PMPNBP. We point out that, when the scene scales with multiple objects with and without articulations, it will not be feasible to use particle filter due to its exponential growth in search space.

To calculate the error in the estimation with respect to the ground truth, we used the average distance metric (ADD) proposed in (*18*, *19*). The point cloud model of the object part was transformed to its ground truth pose [dual quaternion (dq)] and to the estimated pose [dual quaternion $(\overline{dq})$]. The error was calculated as the pointwise distance of these transformation pairs normalized by the number of points in the model point cloud

$$\text{ADD} = \frac{1}{m}\sum_{p\in\mathcal{M}} \|\overline{dq} * p * \overline{dq}_c - dq * p * dq_c\| \quad (1)$$

where $(\overline{dq}_c)$ and (dq_c) are the conjugates of the dual quaternions (*20*, *21*) and m is the number of 3D points in the model set $\mathcal{M}$. Dual quaternions were used to represent the transformations for efficient implementation. We show that our PMPNBP method achieved better convergence than the particle filter method (Fig. 3D).

Pose estimation results on partial and incomplete observations

Articulated models often suffer from self-occlusions and environmental occlusions. By exploiting the articulated constraints, our inference method was able to give a physically plausible estimate that could explain the partial observations. We used two compelling examples that show the strength of our PMPNBP method (Fig. 4). In the first example (Fig. 4, A to D), we used the same cabinet as shown in previous sections. This time, half of the cabinet was occluded by a blanket, which made the observed point cloud noisy. The PMPNBP method was able to handle the occlusion and gave a plausible pose estimation within 100 iterations. Additional qualitative results are shown in fig. S3. Estimating the pose of a Fetch robot (Fig. 4, A to D) is a more complicated task. A cabinet only has prismatic joints, whereas the Fetch robot has many revolute joints as well. The abstracted graphical model of the Fetch robot consists of 12 nodes and 11 edges (see fig. S4). Our factored pose estimation method can scale to objects with a large number of links and joints. Although the robot contains self-occlusion, because many parts of the robot overlap with each other, the PMPNBP method was still able to accurately estimate its pose of the robot by passing messages for 1000 iterations. Furthermore, the algorithm was capable of handling complete occlusions of robot links. This is illustrated in fig. S5, where a robot link was completely not visible due to simulated occlusion and the algorithm was still able to estimate the link's pose.

Benefits of maintaining belief toward planning actions

One advantage of using a belief propagation method instead of a discriminative pose estimation method is that it maintains the beliefs of the estimation. We show how the beliefs aid in robot planning with an example shown in Fig. 5. The robot was performing a task of storing elements in the bottom drawer. Before taking any action to open the drawer, the robot perceived the scene first. With the initial camera setting (Fig. 5A), the robot estimated the pose of the cabinet (Fig. 5B), along with the covariance of the belief particles for each part (Fig. 5C). The covariance represents the uncertainty of pose estimation. A large covariance means a large uncertainty in the estimation. A maximum threshold (2.5 cm) was set on the standard deviation (SD) of (x, y, z) dimensions to decide whether the robot has to change for a better viewpoint.

In this case, the robot was not certain about the estimation because the SD was greater than the threshold. The robot then took an intermediate action (looking down) to provide a new observation of the cabinet (Fig. 3D). With the new observation, the robot estimated the pose again (Fig. 3E). This time, the robot was certain that the bottom drawer was closed because the SD dropped below the threshold (Fig. 3F), and the robot performed an action to open the bottom drawer (Fig. 3, G and H). This is an illustration of how the maintained beliefs can be used in robot planning. In the future work, we plan to characterize the properties of covariance in the estimation with respect to the algorithm parameters (number of particles and iterations).

Comparison with an existing nonparametric belief propagation method

Existing message-passing methods (*7*, *8*) represent message as a mixture of Gaussian components and provide Gibbs sampling–based techniques to approximate message products. We compared our proposed

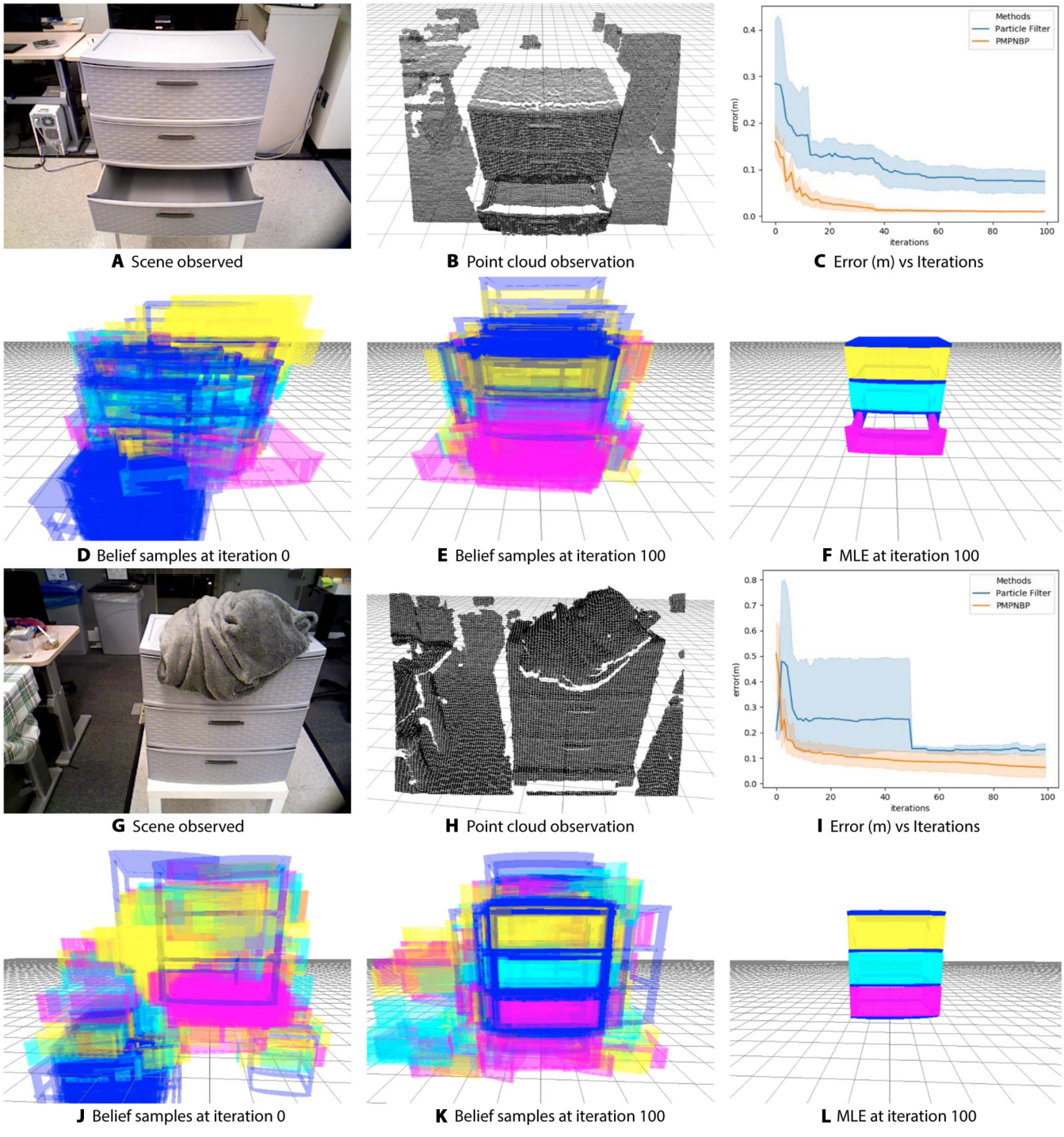


Fig. 3. Pose estimation results of a cabinet in two different scenes using PMPNBP and particle filter. (**A** and **G**) Observed scene with the cabinet. (**B** and **H**) Point cloud observation of the cabinet. (**D** and **J**) Beliefs at iteration 0. (**E** and **K**) Beliefs at iteration 100. (**F** and **L**) MLE of the cabinet pose at iteration 100. (**C** and **I**) Error comparison of PMPNBP and particle filter—400 particles, over 10 runs.

PMPNBP method with PAMPAS (*14*) on their 2D illustrative example (Fig. 4). The pattern has a circle node with state variable $X_1 = (x_1, y_1, r_1)$ denoting its position in the 2D image and the radius of the circle. This circle node has four arms with two links each. These links are nodes in the graph with state variables $X_i(x_i, y_i, \alpha_i, w_i, h_i)$. The links connected to the circle are indexed as $2 \leq i \leq 5$ with their connected outer links as $j = i + 4$.

We show the convergence of the PMPNBP on two examples both qualitatively and quantitatively in Fig. 7. The pattern referred in Fig. 6 was placed in a clutter made of 12 circles and 100 rectangles (Fig. 7A). There are 16 messages, i.e., 4 from circle to inner links, 4 from inner links to circle, 4 from inner links to outer links, and 4 from outer to inner links. The initialization of the messages was done with $M = 75$ particles

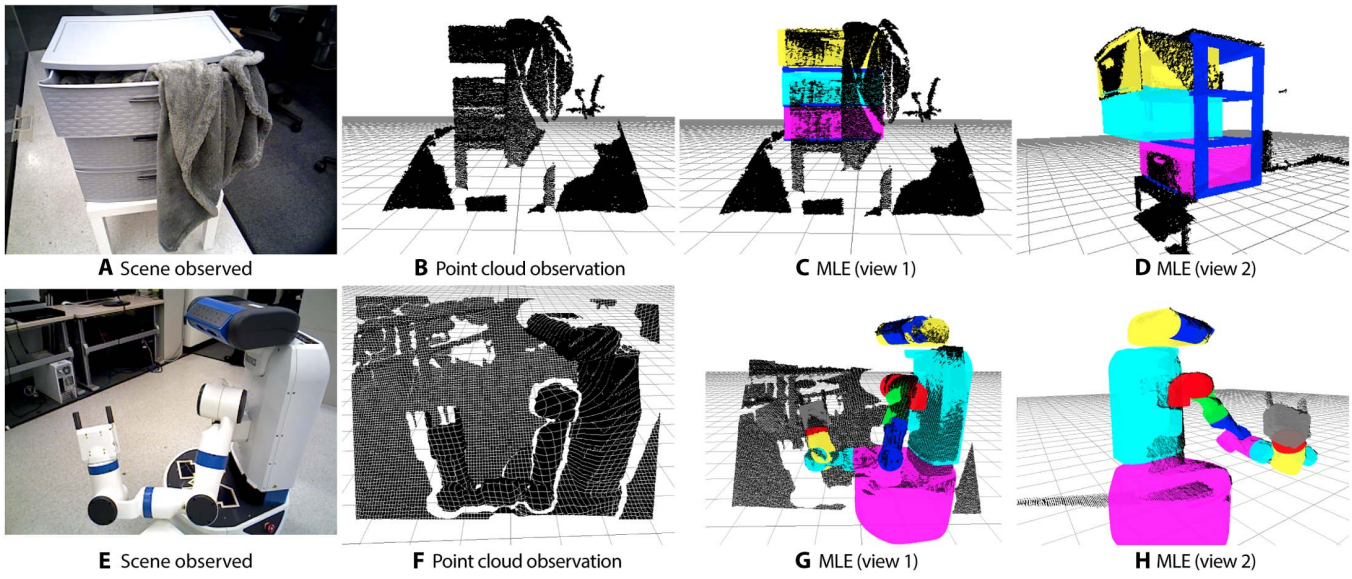


Fig. 4. Pose estimation of articulated objects under occlusions using PMPNBP. (**A** and **E**) RGB observations. (**B** and **F**) Point cloud observations. (**C** and **G**) Pose estimation results from the first viewpoint. (**D** and **H**) Pose estimation results from the second viewpoint.

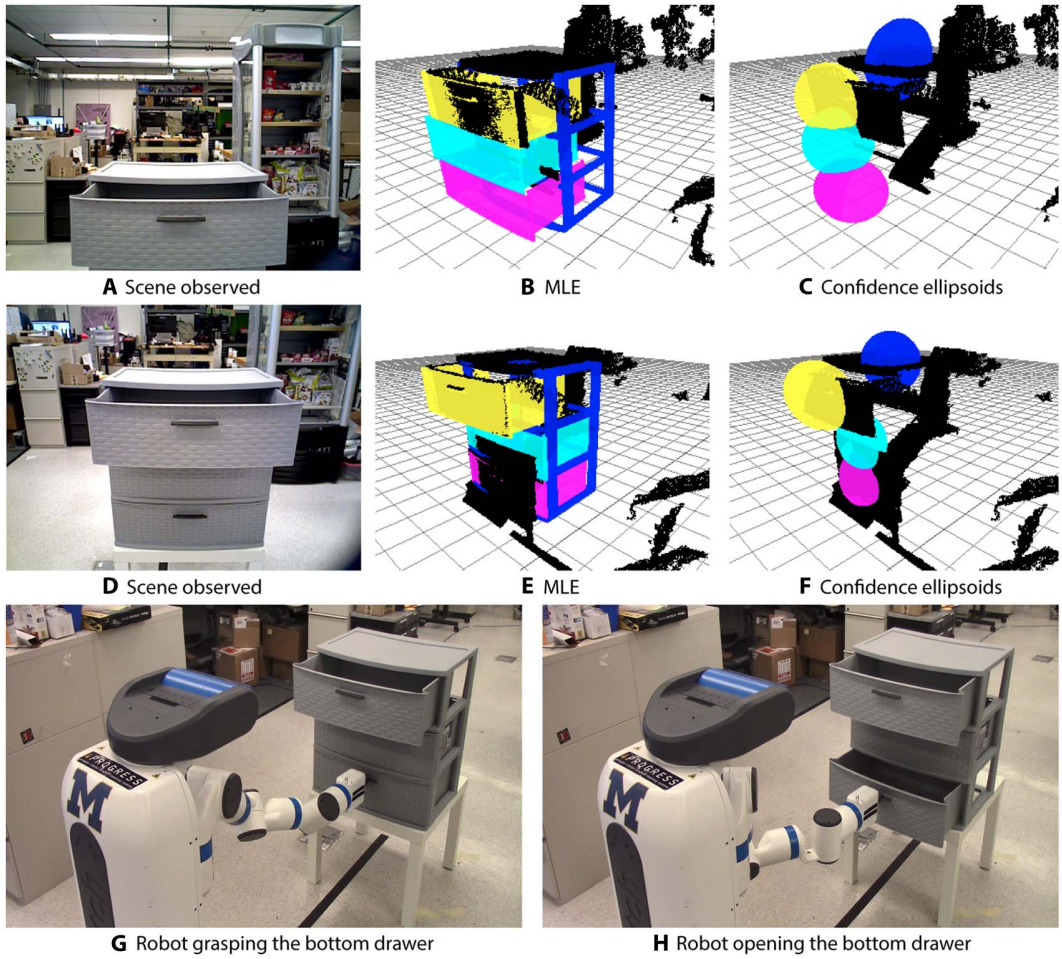
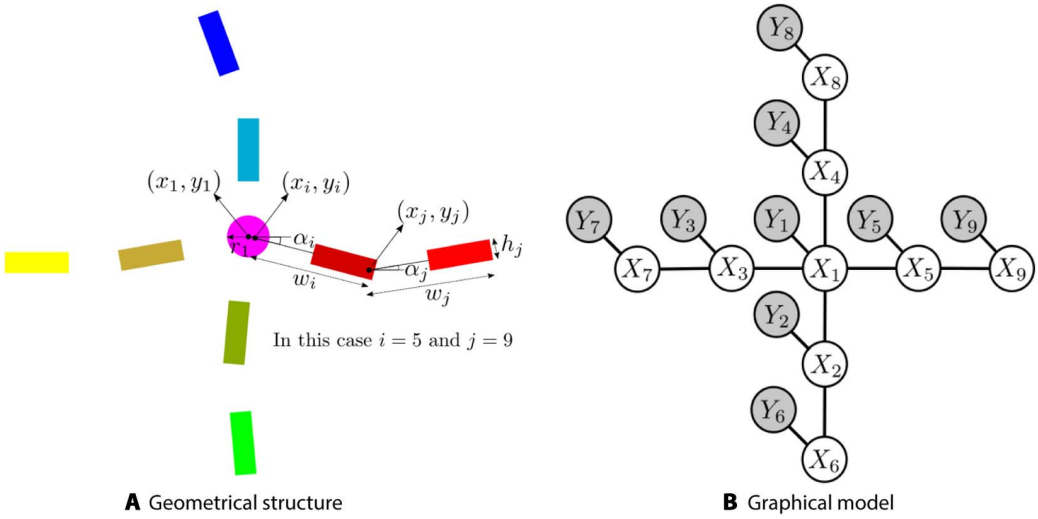


Fig. 5. An example of maintaining beliefs aid in robot planning. The robot's task is to open the bottom drawer. (**A** and **D**) RGB observation of the scene. (**B** and **E**) MLE of the cabinet in the scene. (**C** and **F**) Confidence ellipsoids of the pose estimation generated by the covariance of beliefs. (**G** and **H**) Robot operating the bottom drawer using the MLE from (E), because confidence ellipsoids in (F) are within the provided thresholds.



A Geometrical structure **B** Graphical model

Fig. 6. 2D articulation pattern and its graphical model. The pattern used for the experiments has nine nodes with one circle at the center and four arms with two links each as shown in (**A**). This forms the graphical model shown in (**B**), where hidden nodes X_s are connected to their neighbors and informed by observed nodes Y_s. Geometrically, the circle and links are defined by their location (x_s, y_s), orientation and dimensions as shown in (A).

at (x, y) locations of the image, where the unary potential was greater than 0.4. This was assumed to be the coarse feature detection of the circle and rectangles in the image replicating the initialization in (*14*). In the future iterations, the message had 50% of the samples uniformly sampled in the image to keep exploring, and the other 50% of the samples were sampled from the marginal belief. For the first example with no occlusion (Fig. 7A), the MLE of all the links and circle were close to the pose of the ground truth pattern (Fig. 7D) at the 24th iteration. The second example in Fig. 7E had no circle in the center of the pattern, demonstrating a partial occlusion scenario. The estimation result was similar to the first example but took more iterations to converge (Fig. 7).

In Fig. 7I, we show the convergence of the PMPNBP with respect to PAMPAS (*14*). Convergence here is shown as the average error of the MLE from its ground truth with respect to the number of belief iterations over 10 trials. We plotted this convergence for PMPNBP with M = {50,75,100,200} components versus PAMPAS. The convergence of PMPNBP was better than our implementation of PAMPAS. We note that the PMPNBP has decreasing average errors with increasing numbers of particles. This essentially indicates that, with larger M, the better the inference will be. To evaluate whether PMPNBP accommodates the use of larger M in practice, we plotted the central processing unit (CPU) run time per message update iteration in Fig. 5J. An entire message generation in PAMPAS took $O(KDM^2)$ operations, where D is the number of messages to compute product in the "pre-message," K is the number of iterations for the Gibbs sampler, and M is the number of Gaussian components used to represent a message. In contrast, PMPNBP only took $O(DM)$ operations. The linear time complexity with respect to the number of particles enabled us to use more particles for inference and get better convergence. The implementation details on the unary and pairwise potentials for this 2D experiment are provided in the Supplementary Materials. Figures S1 and S2 show illustrations of the potentials and the sampling involved, respectively. In addition, in fig. S7 and S8, we provide the convergence of pose estimation along with the belief samples. The decrease in the variance of the belief samples can be visualized in these additional qualitative results.

In this section, we discussed the convergence properties of PMPNBP on 3D articulated objects, followed by the comparison with existing NBP method. The background on NBP and PMPNBP algorithms is provided in the "Materials and Methods" section.

DISCUSSION

Here, we discuss some fundamental differences between the methods mentioned in the "Introduction" section and our proposed method. Hausman *et al.* (*5*) focused on estimating articulation models and plan actions, with fiduciary markers (AR tags) to track the pose of the objects. Martin and Brock (*6*) and Katz *et al.* (*22*) focused on estimating kinematic models by incorporating low-level point tracking assuming that objects have visual features. Sturm *et al.* (*7*, *8*) focused on estimating articulation models in a probabilistic fashion by human demonstration of manipulation examples with tracked markers. With the assumptions on the visual features either from markers or textures, these methods primarily focused on kinematic model estimation. In the current study, we focused on estimating the pose of the articulated objects in arbitrary poses with known 3D mesh models of the parts and their articulation constraints. Specifically, the pairwise potential function to be discussed in the "Pairwise potential and sampling" section used limits of articulations from URDF to output compatibility between jointed links. These existing works estimated kinematic models to produce such URDF models of objects. In the process of discovering kinematic models, these existing works tracked the change in the relative poses. However, we focused on estimating the pose of articulated objects at arbitrary poses and challenging configurations, making it a global localization problem as opposed to local localization or tracking problem where the initial configuration was known.

Existing filtering-based articulated object tracking frameworks (*23*, *24*, *25*) were initialized with ground truth object poses. Our method could complement these existing tracking frameworks by providing an initial pose estimate. In addition, belief propagation was applied to articulated pose tracking after initial pose estimation (*9*, *10*), making it suitable for interactive perception. We consider comparisons with the tracking frameworks as a future work.

Limitations

The key problem toward solving a belief propagation problem with continuous variables is a message product that takes $\mathcal{O}(M^D)$. M is the number of Gaussian mixture components used to represent the continuous value, and D is the number of incoming mixtures used to construct an outgoing mixture in the context of message passing. The methods discussed in Introduction proposed approximations to compute this product to make the nonparametric belief propagation tractable in their respective applications domains. Here, we propose another such approximation (PMPNBP) that is much more

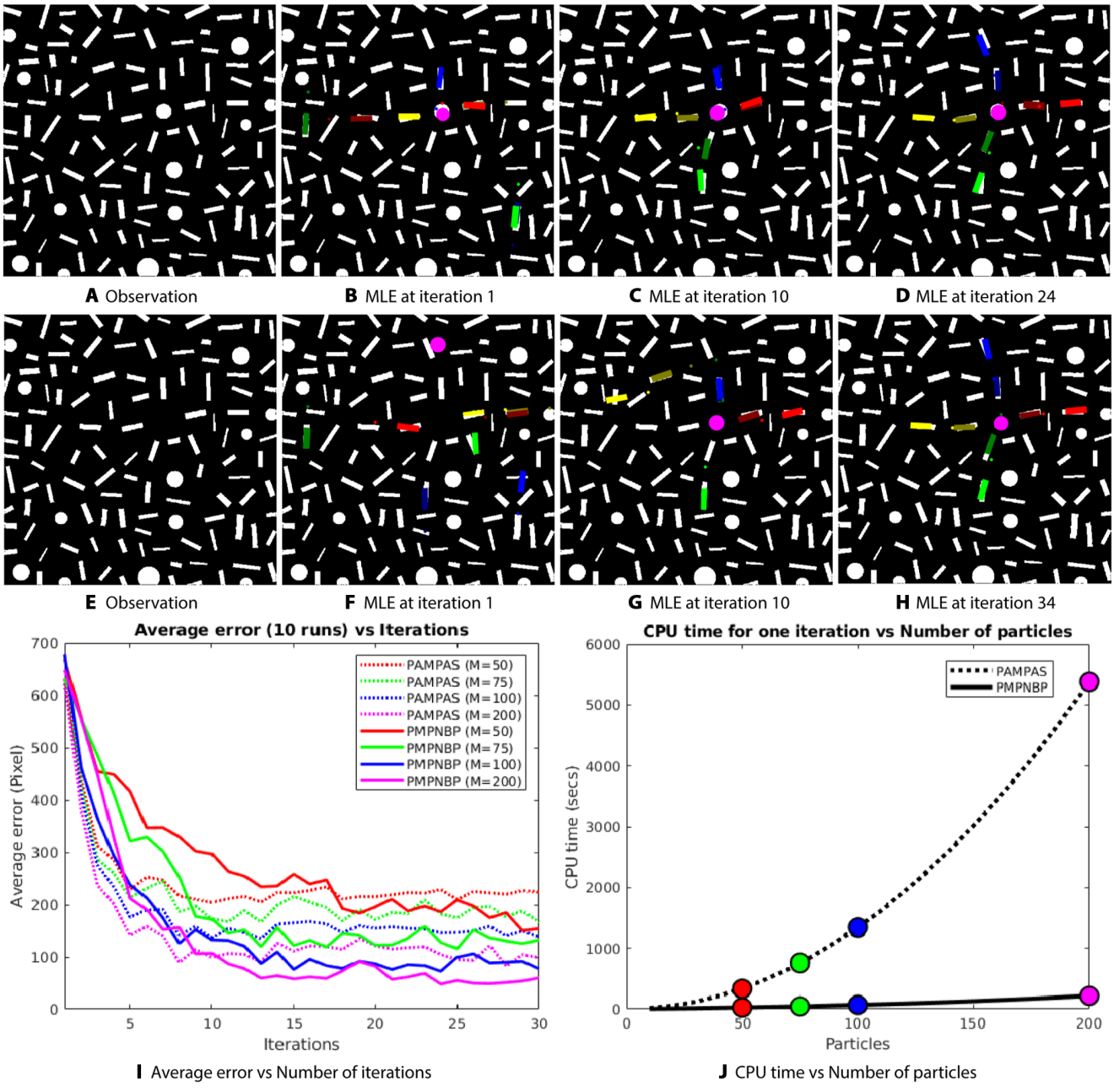


Fig. 7. PMPNBP convergence on 2D patterns and its comparison with PAMPAS. (**A**) 2D observation without occlusion (circle is visible). (**B** to **D**) MLE of the 2D pattern at iterations 1, 10, and 24, respectively. (**E**) 2D observation with occlusion (circle is not visible). (**F** to **H**) MLE of the 2D pattern at iterations 1, 10, and 34, respectively. (**I**) Average error of convergence of PMPNBP and PAMPAS with respect to the number of iterations. (**J**) CPU run time per message update iteration with respect to the number of particles. (Best viewed in color).

efficient and does not grow asymptotically as the other approximations proposed earlier. PMPNBP assumes that the belief of a node generates its incoming message reweighted by the constraints of its neighbors. On the other hand, state-of-the-art methods (*14*, *13*) generated a new message from the source node to the target node by using all the other incoming messages to the source node. When the belief cannot capture samples at the true locations, PMPNBP will fail to generate an incoming message with samples at the true locations. Because PMPNBP can work with large number of samples, it always assumes that samples are available around the true locations that will be exploited in the inference. To avoid this scenario, a percentage of samples from uniform distribution can be used in addition to the samples from the belief. These samples can be considered as exploration samples. We consider this limitation to be reasonable because, computationally, PMPNBP affords to use large number of samples as compared with other methods. In our experiments, for the 2D articulated pattern estimation, we used 50% of the samples to explore, whereas in the 3D cabinet and robot pose estimations, we used only 10% of the samples.

CONCLUSION

We proposed a new message-passing scheme that uses a "pull" approach to update messages in NBP. We represented messages as weighted particles instead of Gaussian mixtures as proposed in earlier algorithms. The proposed message-passing scheme avoided Gibbs sampling–based message products of the earlier methods and provided faster product approximations. We showed the efficiency of the proposed algorithm both in terms of its convergence properties and the computing time with respect to PAMPAS. We applied PMPNBP to estimate the poses of articulated objects. We showed that the PMPNBP outperformed the baseline Monte Carlo localization method quantitatively. Qualitative results are 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 with higher number of nodes and edges in their probabilistic graphical models. In addition, we illustrated how belief propagation can benefit robot manipulation tasks. The notion of uncertainty in the inference is inevitable in robotic perception. Our proposed PMPNBP algorithm was able to accurately estimate the pose of articulated objects and to maintain belief over possible poses that can benefit a robot in performing a task.

MATERIALS AND METHODS

Problem statement

We consider an articulated object O to be composed of N object parts and $N-1$ points of articulation. Such an object description conforms with the URDF commonly used in Robot Operating System (ROS) (*26*). Such a URDF-compliant kinematic model can be represented using an undirected graph $G = (V, E)$ with nodes V for object-part links and edges E for points of articulation. If G is an MRF, then it has two types of variables, X and Y, that are hidden and observed variables, respectively. Let $Y = \{Y_s \mid Y_s \in V\}$, where $Y_s = P_s \subseteq P$, with P being the 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 $X_s \in H_D$ is a dual quaternion pose of an object part. Dual quaternions (*20*, *21*) 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 $X_s = q_r + \varepsilon q_d$, 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 $X_s = dq_{\text{pos}} * dq_{\text{ori}}$, where * is a dual quaternion multiplication. $dq_{\text{ori}} = [q_w, q_x, q_y, q_z][0,0,0,0]$ is the dual quaternion representation of pure rotation, and $dq_{\text{pos}} = [1,0,0,0]\left[0, \frac{x}{2}, \frac{y}{2}, \frac{z}{2}\right]$ 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 when 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 the "Pairwise potential and sampling" section.

Overview of nonparametric belief propagation

Let $G = (V, E)$ be an undirected graph with nodes V and edges E. The nodes in V are each random variables that have dependencies with each other in the graph G through edges E. If G is an MRF, then it has two types of variables, X and Y, denoting the collection of hidden and observed variables, respectively. Each variable is considered to take assignments of continuous-valued vectors. The joint probability of the graph G, considering only second-order cliques, is given as

$$p(X, Y) = \frac{1}{Z} \prod_{(s,t)\in E} \psi_{s,t}(X_s, X_t) \prod_{s\in V} \phi_s(X_s, Y_s) \tag{2}$$

where $\psi_{s,t}(X_s, X_t)$ is the pairwise potential between nodes $X_s \in R^d$ and $X_t \in R^b$ (note that dimensionality remains the same, $d = b$, in the case of estimating 6-DOF object pose), $\phi_s(X_s, Y_s)$ is the unary potential between the hidden node X_s and observed node $Y_s \in R^q$, and Z is a normalizing factor. The problem is to infer belief over possible states assigned to the hidden variables X 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.

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

$$m_{t\to s}(X_s) = \sum_{i=1}^{M} w_{ts}^{(i)} \mathcal{N}\left(X_s; \mu_{ts}^{(i)}, \Lambda_{ts}^{(i)}\right) \tag{3}$$

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 ith component, and $\mu_{ts}^{(i)}$ and $\Lambda_{ts}^{(i)}$ are the mean and covariance of the ith component, respectively. We use the terms "components," "particles," and "samples" interchangeably in this paper. Hence, a message can be expressed as M triplets

$$m_{t\to s} = \left\{\left(w_{ts}^{(i)}, \mu_{ts}^{(i)}, \Lambda_{ts}^{(i)}\right) : 1 \le i \le M\right\} \tag{4}$$

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

$$m_{t\to s}^{n}(X_s) \leftarrow \int_{X_t \in H_D} \left(\psi_{st}(X_s, X_t)\phi_t(X_t, Y_t) \prod_{u\in\rho(t)\setminus s} m_{u\to t}^{n-1}(X_t)\right) dX_t \tag{5}$$

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

$$bel_s^n(X_s) \propto \phi_s(X_s, Y_s) \prod_{t\in\rho(s)} m_{t\to s}^{n}(X_s) \tag{6}$$

$$bel_s^n = \left\{\left(w_s^{(i)}, \mu_s^{(i)}, \Lambda_s^{(i)}\right) : 1 \le i \le T\right\} \tag{7}$$

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

"Push" message update

NBP (*13*) provides a Gibbs sampling approach to compute an approximation of the product $\prod_{u\in\rho(t)\backslash s} m^{n-1}_{u\to t}(X_t)$. Assuming that $\phi_t(X_t, Y_t)$ is pointwise computable, a "pre-message" (*15*) is defined as

$$M^{n-1}_{t\to s}(X_t) = \phi_t(X_t, Y_t) \prod_{u\in\rho(t)\backslash s} m^{n-1}_{u\to t}(X_t) \tag{8}$$

which can be computed in the Gibbs sampling procedure. This reduces Eq. 5 to

$$m^n_{t\to s}(X_s) \leftarrow \int_{X_t\in R^b} \left(\psi_{st}(X_s, X_t) M^{n-1}_{t\to s}(X_t)\right) dX_t \tag{9}$$

NBP (*13*) sample $\hat{X}^{(i)}_t$ from the "pre-message," followed by a pairwise sampling, where $\psi_{st}(X_s, X_t)$ is acting as $\psi_{st}(X_s \mid X_t = \hat{X}^{(i)}_t)$ to get a sample $\hat{X}^{(i)}_s$. The Gibbs sampling procedure in itself is an iterative procedure and hence makes the computation of the "pre-message" (as the Foundation function described for PAMPAS) expensive as M increases.

"Pull" message update

Given the overview of NBP above, we now describe our "pull" message passing algorithm. We represent message as a set of pairs instead of triplets in Eq. 4, which is

$$m_{t\to s} = \left\{ \left(w^{(i)}_{ts}, \mu^{(i)}_{ts} \right) : 1 \le i \le M \right\} \tag{10}$$

Similarly, the marginal belief is summarized as a sample set

$$bel^n_s(X_s) = \left\{ \mu^{(i)}_s : 1 \le i \le T \right\} \tag{11}$$

where T is the number of samples representing the marginal belief. We assume that there is 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 pass these samples over to the neighboring nodes $\rho(t)\backslash s$ and compute the weights $\{w^{(i)}_{ts}\}^M_{i=1}$. This step is described in Algorithm 1. The computation of $bel^n_s(X_s)$ is described in Algorithm 2. The key difference between the "push" approach of the earlier methods [NBP (*13*) and PAMPAS (*14*)] and our "pull" approach is the message $m_{t\to s}$ generation. In the "push" approach, the 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 from previous iteration and weighted by the incoming messages to t. This weighting strategy is computationally efficient. In addition, the product of incoming messages to compute bel_s is approximated by a resampling step as described in Algorithm 2. An illustrative overview of Algorithms 1 and 2 is shown in fig. S6.

Algorithm 1. Message update

Given input messages $m^{n-1}_{u\to t}(X_t) = \{(\mu^{(i)}_{ut}, w^{(i)}_{ut})\}^M_{i=1}$ for each $u \in \rho(t)\backslash s$, and methods to compute functions $\psi_{ts}(X_t, X_s)$ and $\phi_t(X_t, Y_t)$ pointwise, 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 $n = 1$, then the $bel^0_s(X_s)$ is a uniform distribution or informed by a prior distribution.
   b) If $n > 1$, then 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 $w^{(i)}_{\text{unary}}$ is computed using $\phi_t(X_t = \hat{X}^{(i)}_t, Y_t)$.
   c) Neighboring weight $w^{(i)}_{\text{neigh}}$ is computed using $m^{n-1}_{u\to t}$.
      i. For each $u \in \rho(t)\backslash s$ 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 $w^{(i)}_{\text{neigh}} = \prod_{u\in\rho(t)\backslash s} W^{(i)}_u$.
   d) The final weights are computed as $w^{(i)}_{ts} = w^{(i)}_{\text{neigh}} \times w^{(i)}_{\text{unary}}$.
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 2. 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 $\phi_s(x_s, y_s)$ pointwise, 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.

Although URDF models are typically tree structured, our algorithm is not limited to tree graphs and can handle loopy graphs. The NBP algorithms are loop belief propagation methods in continuous domain. Similar to (*9*), which allows additional edges to capture physical collision constraints, our algorithm allows additional edges resulting in loopy graphs.

Potential functions

Unary potential

Unary potential $\phi_t(X_t, Y_t)$ is used to model the likelihood by measuring how a 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

$$\phi_t(X_t, Y_t) = e^{-\lambda_r d(P_t, P^*_t)} \tag{12}$$

where λ_r is the scaling factor and $d(P_t, P^*_t)$ is the sum of 3D Euclidean distance between all the observed points $p \in P_t$ and corresponding rendered point $p^* \in P^*_t$. These point clouds were generated by pixel-wise back projection of the depth image (both observed and rendered) using the intrinsic parameters of the camera, giving us their correspondence. This likelihood calculation is adapted from the methods (*1*, *2*, *3*).

Pairwise potential and sampling

Pairwise potential $\psi_{t,s}(X_t \mid X_s)$ gives information about how compatible two object poses are given their joint articulation constraints captured

by the edge between them. As mentioned in the "Problem statement" section, 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 distance between the limits $C = d(dq^a_{t|s}, dq^b_{t|s})$. Using a joint limit kernel parameterized by (σ_{pos}, σ_{ori}), we evaluate the pairwise potential as

$$\psi_{t,s}(X_t|X_s) = e^{-(A_{pos}+B_{pos}-C_{pos})^2/2(\sigma_{pos})^2-(A_{ori}+B_{ori}-C_{ori})^2/2(\sigma_{ori})^2} \quad (13)$$

The pairwise sampling uses the same limits $R_{t|s}$ to sample for X_t given a 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 $X_t = X_s * \bar{X}_t$.

Experimental setup

We did experiments using the PMPNBP on both artificial 2D patterns and 3D articulated objects. The hardware platform we used for all the experiments was an Ubuntu 14.04 machine with a Core i7-6700HQ CPU, 16 GB random-access memory, and an NVIDIA GTX 1080 graphics processing unit (GPU). For 2D experiments, we used the same artificial pattern as presented in the PAMPAS (*14*) paper. Both PMPNBP and PAMPAS were implemented in MATLAB without any type of parallelization to avoid bias in comparison. For 3D experiments, we used our Fetch robot, a mobile manipulation platform, for data collection and manipulation experiments. RGBD data were collected using an ASUS Xtion RGBD sensor mounted on the robot. Intrinsic and extrinsic parameters were assumed to be given for rendering synthetic scenes with hypothesized object poses. CUDA-OpenGL interoperation was used to render synthetic scenes (depth images) on a GPU.

SUPPLEMENTARY MATERIALS

robotics.sciencemag.org/cgi/content/full/4/30/eaaw4523/DC1
Fig. S1. Unary potential illustration.
Fig. S2. Pairwise potential illustration.
Fig. S3. More results on pose estimation of a cabinet under partial occlusion.
Fig. S4. Pose estimation of a Fetch robot.
Fig. S5. Pose estimation of a Fetch robot with simulated occlusion.
Fig. S6. Illustrative overview of Message and Belief update algorithms.
Fig. S7. PMPNBP results with circle node observed in the 2D articulated pattern estimation.
Fig. S8. PMPNBP results with circle node "occluded" in the 2D articulated pattern estimation.
Movie S1. Research summary.

REFERENCES AND NOTES

1. Z. Sui, L. Xiang, O. C. Jenkins, K. Desingh, Goal-directed robot manipulation through axiomatic scene estimation. *Int. J. Robot. Res.* **36**, 86–104 (2017).
2. K. Desingh, O. C. Jenkins, L. Reveret, Z. Sui, Physically plausible scene estimation for manipulation in clutter, in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2016), pp. 1073–1080.
3. Z. Zeng, Z. Zhou, Z. Sui, O. C. Jenkins, Semantic robot programming for goal-directed manipulation in cluttered scenes, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 7462–7469.
4. J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, G. S. Sukhatme, Interactive perception: Leveraging action in perception and perception in action. *IEEE Trans. Robot.* **33**, 1273–1291 (2017).
5. K. Hausman, S. Niekum, S. Osentoski, G. S. Sukhatme, Active articulation model estimation through interactive perception, in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 3305–3312.
6. R. M. Martin, O. Brock, Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems,* Chicago, IL, 14 to 18 September 2014, pp. 2494–2501.
7. J. Sturm, C. Stachniss, W. Burgard, A probabilistic framework for learning kinematic models of articulated objects. *J. Artif. Intell. Res.* **41**, 477–526 (2011).
8. J. Sturm, Springer Tracts in Advanced Robotics (STAR), in *Approaches to Probabilistic Model Learning for Mobile Manipulation Robots* (Springer, 2013).
9. L. Sigal, S. Bhatia, S. Roth, M. J. Black, M. Isard, Tracking loose-limbed people, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*(IEEE, 2004), pp. 421–428.
10. E. B. Sudderth, M. I. Mandel, W. T. Freeman, A. S. Willsky, Visual hand tracking using nonparametric belief propagation, in *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)* (IEEE, 2004), p. 189.
11. M. Vondrak, L. Sigal, O. C. Jenkins, Dynamical simulation priors for human motion tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 52–65 (2013).
12. K. P. Murphy, Y. Weiss, M. I. Jordan, Loopy belief propagation for approximate inference: An empirical study, in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (Morgan Kaufmann Publishers Inc., 1999) pp. 467–476.
13. E. B. Sudderth, A. T. Ihler, W. T. Freeman, A. S. Willsky, Nonparametric belief propagation, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2003), pp. 605–612
14. M. Isard, PAMPAS: Real-valued graphical models for computer vision, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2003), pp. 613–620.
15. A. Ihler, D. McAllester, Particle belief propagation, in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics* (PMLR, 2009), pp. 256–263.
16. L. Song, A. Gretton, D. Bickson, Y. Low, C. Guestrin, Kernel belief propagation, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (PLMR, 2011), pp. 707–715.
17. T. X. Han, H. Ning, T. S. Huang, Efficient nonparametric belief propagation with application to articulated body tracking, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (IEEE, 2006), pp. 214–221.
18. S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, 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.
19. Y. Xiang, T. Schmidt, V. Narayanan, D. Fox, PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. arXiv:1711.00199 (2017).
20. I. Gilitschenski, G. Kurz, S. J. Julier, U. D. Hanebeck, A new probability distribution for simultaneous representation of uncertain position and orientation, in *Proceedings of the 17th International Conference on Information Fusion (FUSION)* (IEEE, 2014), pp. 1–7.
21. B. Kenwright, A beginners guide to dual-quaternions: What they are, how they work, and how to use them for 3D character hierarchies, in *Proceedings of the 20th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)* (2012), pp. 1–13.
22. D. Katz, A. Orthey, O. Brock, Interactive perception of articulated objects, in *The 12th International Symposium on Experimental Robotics (ISER'10),* New Delhi and Agra, India, 18 to 21 December 2010, pp. 301–315.
23. J. Brookshire, S. Teller, Articulated pose estimation using tangent space approximations. *Int. J. Robot. Res.* **35**, 5–29 (2016).
24. C. Garcia Cifuentes, J. Issac, M. Wüthrich, S. Schaal, J. Bohg, Probabilistic articulated real-time tracking for robot manipulation. *IEEE Robot. Autom. Lett.* **2**, 577–584 (2017).
25. T. Schmidt, R. A. Newcombe, and D. Fox. DART: Dense articulated real-time tracking, in *Robotics: Science and Systems* (2014).
26. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, ROS: An open-source robot operating system. *ICRA Workshop on Open Source Software* **3**, 5 (2009).

Acknowledgments: We thank Z. Zeng, Z. Sui, and A. Röfer from the University of Michigan for helpful discussions and feedback. **Funding:** This work was supported, in part, by NSF award IIS-1638047. **Author contributions:** K.D. and O.C.J. formulated the presented problem statement, computational methods, and experimental design with the help of S.L. and A.O. K.D. wrote the majority of the paper with contributions from S.L. and A.O. and editing revisions by O.C.J. O.C.J. provided advising to K.D., S.L., and A.O. and funding support to K.D. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials.

Submitted 26 January 2019
Accepted 23 April 2019
Published 22 May 2019
10.1126/scirobotics.aaw4523

Citation: K. Desingh, S. Lu, A. Opipari, O. C. Jenkins, Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects. *Sci. Robot.* **4**, eaaw4523 (2019).

Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects

Karthik Desingh, Shiyang Lu, Anthony Opipari and Odest Chadwicke Jenkins

Sci. Robotics **4**, eaaw4523.
DOI: 10.1126/scirobotics.aaw4523