Articulated Pose Estimation via Over-parametrization and Noise Projection

Jonathan Brookshire and Seth Teller
MIT Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA 02139
{jbrooksh, teller}@csail.mit.edu

***Abstract*—We describe an algorithm to estimate the pose of a generic articulated object. Our algorithm takes as input a description of the object and a potentially incomplete series of observations; it outputs an on-line estimate of the object's configuration. This task is challenging because: (1) the distribution of object states is often multi-modal; (2) the object is not assumed to be under our control, limiting our ability to predict its motion; and (3) rotational joints make the state space highly non-linear.**

The proposed method represents three principal contributions to address these challenges. First, we use a particle filter implementation which is unique in that it does not require a reliable state transition model. Instead, the method relies primarily on observations during particle proposal, using the state transition model only at singularities. Second, our particle filter formulation explicitly handles missing observations via a novel proposal mechanism. Although existing particle filters can handle missing observations, they do so only by relying on good state transition models. Finally, our method evaluates noise in the observation space, rather than state space. This reduces the variability in performance due to choice of parametrization, and effectively handles non-linearities caused by rotational joints.

We compare our method to a baseline implementation without these techniques and demonstrate, for a fixed error, more than an order-of-magnitude reduction in the number of required particles, an increase in the number of effective particles, and an increase in frame rate. Source code for the method is available at http://rvsn.csail.mit.edu/articulated.

I. Introduction

To be useful in human-constructed environments, robots must interact with objects which have internal degrees of freedom (DOFs) or are constrained with respect to the environment. Such articulated mechanisms are found in homes (e.g., cabinet drawers, appliance doors, faucet handles, toaster levers, corkscrews, lidded boxes) and workplaces (e.g., construction equipment, pliers, screwdrivers, clamps, swivel chairs). Accurate estimates of the internal DOFs of such objects would facilitate manipulating or avoiding them. Knowledge of the state of an excavator, for example, would aid a robot navigating alongside it.

Our method takes as input:

1) A kinematic model of the articulated object, including: a function mapping joint values to observations (the "forward kinematics"); a function mapping small changes in configuration space to small changes in the observation space (the "Jacobian"); and any joint limits. In our application, this model is specified via a URDF file [17].

Fig. 1: The goal is to estimate the joint values and base pose of an articulated object, e.g. an excavator. In this case, links are tracked in an image stream (dots at left); recovering the joint values enables 3D reconstruction (right).

2) A series of observations, sufficient for system observability (i.e. with sufficient information to estimate the object's configuration with finite variance). Observations may be noisy or incomplete, i.e., not included in each sensor update.

The filter estimates the pose of one link (the "base pose") and the value of each joint parameter. In the excavator example (see Fig. 1), we might have independent feature detectors recovering pixel locations of the tracks, operator cab, boom, stick, and bucket. Of course, since these pixel measurements are not independent, the kinematic model can be used to improve the estimates.

In § II, we discuss related work and several alternative solutions to this tracking task. Of these alternative solutions, a "typical" implementation of a particle filter is discussed, along with several of its shortcomings, in § III. § IV details our proposed solution, and § V demonstrates it on household dishwasher, PR2 robot, and construction-site excavator examples.

II. Background

A. Model estimation

We assume that a kinematic model of the object is available. Sources could include an object database, user input, or some prior estimation process. Sturm [13] and Katz [11] demonstrate a prior estimation process, recovering the kinematic model of an unknown object while observing and/or manipulating it. These methods might be used to estimate the kinematic model of an object and provide input to our system.

B. Articulated object tracking

Much of the research in articulated object tracking is focused on tracking the human body: Deutscher [3], Urtasun [15], Ziegler [18], Cabido [1], and Gall [5]. These methods handle a specific kind of input (e.g., video only) or attempt to learn motion models to improve performance. By contrast, our technique is not specific to a particular kinematic model or kind of input. Additionally, we handle settings in which motion is not easily predicted.

To address the multi-modal nature of articulated tracking, Hauberg [7] also uses a particle filter for tracking humans. Similar to our approach, they consider a manifold of valid kinematic configurations. Proposing from the state transition model in this manifold yielded good results in their application; lacking a good state transition model, we propose in the manifold using observations. They also neglect manifold singularities, which we specifically address.

The only articulated tracking work validated on several different kinematic objects, of which we are aware, is by Comport [2]. Comport tracks an articulated object by optimizing a single hypothesis at each time step. Their technique is specific to image detections and, as it maintains a single greedy hypothesis, will be subject to local minima. Additionally, their method assumes all links are observed at all times.

C. Alternate solutions

To motivate our method, we consider several alternative solutions and discuss their shortcomings. Given that we have observations and a kinematic model, one possible solution is to simply optimize over configurations at each time step. This simple technique exhibits several shortcomings: (1) the system may have multiple solutions which cannot be modeled by optimization; (2) no estimate of uncertainty is provided; and (3) joint limits are supported only by adding constraints. The multi-modal nature of the problem is particularly problematic. Multiple solutions arise from redundancies in the kinematics and incomplete observations which do not fully constrain each link.

The Unscented Kalman Filter (UKF) [9] can be used to address some of these issues. As a filter, it can leverage historical data to avoid some local minima and provides an estimate of uncertainty. The UKF can also be augmented [12] to handle joint limits. However, because the UKF can only model a unimodal distribution, its applicability in our domain is limited.

Many other particle filter formulations exist, in addition to the baseline method discussed later. Doucet [4] and Merwe [16], for example, apply an Extended Kalman Filter (EKF) and UKF (respectively) to each particle in the filter. Their formulations expand the state space for each particle, maintaining additional covariance parameters. The EKF approach is similar in spirit to our Jacobian tangent space; on the other hand, our formulation does not require a Gaussian approximation or require that a separate covariance be maintained for each particle.

D. Particle filters

A particle filter [8] maintains a discrete approximation to a probability distribution using a collection of state hypotheses, or particles. During each cycle of the filter, particles are proposed, weighted, and possibly duplicated/eliminated. During the proposal phase, the ideal proposal distribution, or optimal importance function (OIF) [4], is $p(x_k \left| x_{k-1}^{(i)}, z_k \right.)$. In other words, the probability of a new state, x_k, depends on the i-th previous particle's state, $x_{k-1}^{(i)}$, and the new observation, z_k. Using Bayes rule, it can be shown that:

$$p\left(x_k \left| x_{k-1}^{(i)}, z_k \right.\right) = \frac{p\left(z_k|x_k\right) \cdot p\left(x_k|x_{k-1}^{(i)}\right)}{p\left(z_k|x_{k-1}^{(i)}\right)} \tag{1}$$

where $p\left(z_k|x_k\right)$ corresponds to the observation model and $p(x_k|x_{k-1}^{(i)})$ corresponds to the state transition model. The denominator, $p(z_k|x_{k-1}^{(i)})$, is the probability of the observation, given the previous state. In general we will not know this quantity, but can marginalize over the current state:

$$p\left(z_k|x_{k-1}^{(i)}\right) = \int p\left(z_k|x\right) \cdot p\left(x|x_{k-1}^{(i)}\right) dx \tag{2}$$

Note that the two terms in the integral are analogous to the two terms in the numerator of Eq. 1 and can be readily calculated. Doucet [4] also notes that the weight update for the OIF is:

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot p\left(z_k|x_{k-1}^{(i)}\right) \tag{3}$$

It is often impossible to sample directly from the OIF. As a result, the particle filter samples from an approximation and uses an importance weight to compensate. It is important that the approximation yield particles from the high-probability regions of the OIF. To achieve this, one of two approximations is often made:

1) When the state transition model is more accurate than the observations, we can approximate that $x_k \perp z_k$, i.e., that for the purposes of proposal, the state is independent of the observations. The OIF then conveniently becomes the familiar $p(x_k|x_{k-1}^{(i)})$, i.e., the state transition model. The weight update then becomes $w_k^{(i)} = w_{k-1}^{(i)} \cdot p\left(z_k|x_k\right)$ [14].
2) When the observation model is more accurate than the state transition model, we can approximate that $x_k \perp x_{k-1}^{(i)}$, and propose particles based only on observations. In this case, the OIF becomes $p\left(x_k|z_k\right)$. This proposal function is often complex, requiring estimation of a state from observations (e.g., via inverse kinematics). However, this choice enables the incorporation of observations during particle proposal.

The first approximation (1) is most typical, as it produces simple equations. However, because it relies only on the state transition model to propose particles, the next generation of particles is only as good as that model. In our case, a state transition model specific to the articulated object is

unavailable, and we rely on generic zero-velocity and constant-velocity models. These models are relatively poor and result in particles with low observational probability.

Instead, our solution incorporates observations during particle proposal (2). Grisetti [6] successfully demonstrated such an approach for a mapping task. The state transition model for the robot's motion was often noisier than the sensor data, so the method used LIDAR observations, processed through a scan-matcher, to propose new particles. The result was that generated particles had higher probability and, thus, fewer particles were required. We adopt the same approach here, but do not require a function to convert directly from observations to an object configuration (i.e., we do not require a scan-matcher or equivalent).

III. Baseline Implementation

As a basis for comparison, we develop a "baseline" method which uses the typical particle filter formulation (approximation (1) above). We expect that this technique will have difficulty – motivating our method – because its state transition model is not well-tailored to the object's motion.

When limited to AGN models, as in our case for the zero- and constant-velocity models, the baseline method exhibits an additional problem: the rotational joints result in a highly non-linear state space which, if ignored, can produce unlikely particles and result in wasted computation.

To see this, suppose we wish to track the simple 2D kinematic linkage shown in Fig. 2. Consider adding a small noise value to each joint, represented in the state. It is clear that adding a small rotational value to the joint at p_2 will result in a relatively large movement of the link at p_4.

Such an experiment is repeated many times and the resulting link positions are shown in Fig. 3. Here, p_1 (blue) was chosen as the base link, and rotational noise accumulates, causing a wide banana-like dispersion of the fourth link (magenta). This is problematic because the fourth link's position is unlikely according to the observations (the observation density is represented by the underlying contours). In practice, this means that a particle's distal link may be far from the observations, receive a low weight, and result in wasted computation on an unlikely hypothesis.

In the next section, we will address this issue by proposing particles with noise in the observation space, rather than the state space. This will produce configurations which correspond more closely to the observational contours.

IV. Our Particle Proposal Method

During particle proposal, the algorithm considers the valid configurations of the articulated object as a manifold $\mathcal{M}$ in a higher-dimensional space. We chose this high-dimensional space to be the observation space so that proposed particles correspond to observation noise and because this space is fixed in our application. The manifold is defined by the forward kinematics and observation function, $f(x)$, which converts from an $M \times 1$ state vector, x, to a point in the higher dimensional space. A complete observation z is an N-dimensional point in the observation space. (We assume our system is observable, i.e., $N \geq M$.)

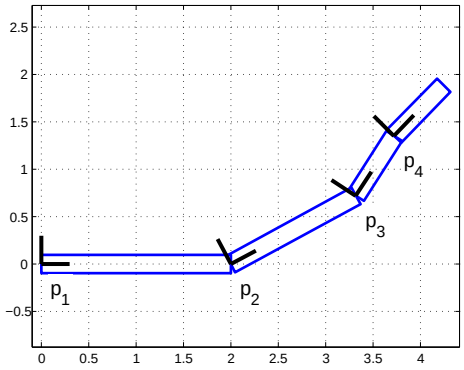


Fig. 2: An example 2D articulated system with four rigid links (with origins at p_1-p_4) and three revolute joints.

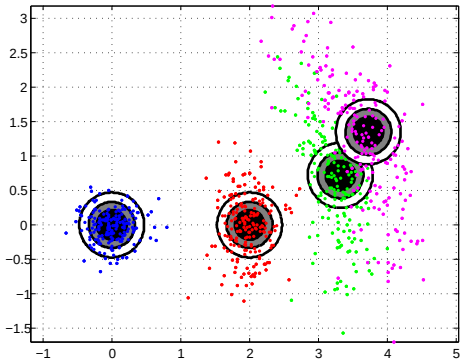


Fig. 3: The dots show the link positions of the particles proposed from $p(x_k|x_{k-1}^{(i)})$ for links 1-4 (respectively blue, red, green, magenta). The underlying contours illustrate the Gaussian $p\left(z_k|x_k\right)$. Notice that for link 4, for example, many particles are unlikely.

Our method proceeds by:

1) Using available observations, z_k, to find the nearest valid configuration/state (i.e., the closest point on $\mathcal{M}$).
2) Approximating the OIF (Eq. 1) using a set $\mathcal{X}^{(i)}$ of discrete samples, generated by:
   a) Adding noise in the observation (not state) space, then using a first-order Taylor approximation and projection to map the noise to the state space,
   b) Adding noise in the state space only when the Jacobian is singular, and
   c) Rejection sampling to satisfy joint limits.

As a result of (1), we can exploit observations during the proposal stage without requiring an inverse kinematic function. As we will see, we can extract information from observations which are either redundant ($\dim(z_k) > M$) or incomplete ($\dim(z_k) < M$). As a result of (2a), the particle proposals are less dependent on the state parametrization, because particle perturbations are created in the fixed observation space. Attribute (2b) handles degenerate observations by relying on the

state transition model. In (2c), we ensure that joint limits are satisfied. Because rejection sampling can be expensive, we sample directly from the discrete approximation $\mathcal{X}^{(i)}$ when choosing $x_k^{(i)}$ (rather than re-approximating with a Gaussian, as in [6]).

Because the algorithm is not provided with a state transition model, we select a model which will generalize to many different kinematic chains. In all our examples, we found a zero-velocity model sufficient. That is, we use the state transition model $P(x_k|x_{k-1}) \sim N(x_{k-1}, \Sigma_x)$. If another transition model were more appropriate, e.g., a constant-velocity model, the following techniques could be easily modified by adding the constant velocity, in addition to diffusion noise, in the null space.

A. Updating with the current observations

The algorithm begins by considering each particle from the previous time step, $x_{k-1}^{(i)}$. For notational simplicity, let $x = x_{k-1}^{(i)}$. We wish to update this particle using whatever observations are available. Thus, we find a new point m_k which is both near the previous particle and near the observations: $m_k = x + \widehat{dx}$. Here, $\widehat{dx}$ is a small change in the state space and we wish to find it by minimizing the Mahalanobis distance between the observation and the configuration manifold. Further, the observations, z_k, may be incomplete; let Q be a $N \times N$ diagonal matrix whose diagonal entries are 1 if the observation is made at time k and zero otherwise (i.e. Q selects the valid observations). Then

$$\begin{aligned}\widehat{dx} = \underset{dx}{\operatorname{argmin}} &[Qz_k - Qf(x+dx)]^T \Sigma_{obs}^{-1} \\ &\cdot [Qz_k - Qf(x+dx)]\end{aligned} \tag{4}$$

where Σ_{obs} is the covariance of the observations. Let $L^T L = \Sigma_{obs}^{-1}$ be the Cholesky factorization of Σ_{obs}^{-1}. Then

$$\begin{aligned}\widehat{dx} &= \underset{dx}{\operatorname{argmin}} [Qz_k - Qf(x+dx)]^T L^T L \\ &\qquad \cdot [Qz_k - Qf(x+dx)] \qquad (5) \\ &= \underset{dx}{\operatorname{argmin}} \; |LQz_k - LQf(x+dx)|^2 \qquad (6)\end{aligned}$$

The algorithm then finds the nearest point on the manifold by projecting onto a tangent plane (Fig. 4). We assume that $\mathcal{M}$ is well-approximated by a first-order Taylor series in a neighborhood corresponding to the observation noise:

$$f(x+dx) = f(x) + J_x \cdot dx \tag{7}$$

We also note that $z_k = f(x) + dz$, where dz is the vector between the previous particle and the current observation.

$$\begin{aligned}\widehat{dx} &= \underset{dx}{\operatorname{argmin}} \, |LQf(x) + LQdz - LQf(x) - LQJ_x \; dx|^2 \\ &= \underset{dx}{\operatorname{argmin}} \, |LQdz - LQ \; J_x \; dx|^2 \qquad (8)\end{aligned}$$

The pseudo-inverse, $(\cdot)^\dagger$, solves this problem:

$$\widehat{dx} = (LQJ_x)^\dagger \; LQ \; dz \tag{9}$$

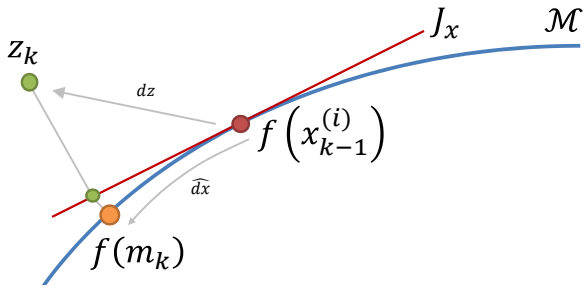


Fig. 4: The previous particle $x_{k-1}^{(i)}$ is updated with the observations z_k using a Taylor approximation and projection.

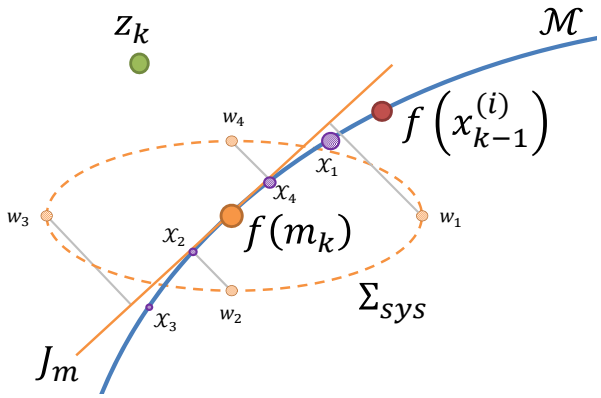


Fig. 5: A Taylor approximation and projection relate noise in observation space to noise in state space.

Substituting back into the equation for m_k:

$$m_k = x + (LQJ_x)^\dagger \; LQ \; dz \tag{10}$$

$$= x_{k-1}^{(i)} + (LQJ_x)^\dagger \; LQ \; \left(z_k - f\left(x_{k-1}^{(i)}\right)\right) \tag{11}$$

The matrix product (LQJ_x) may be singular due to (1) a specific configuration or (2) missing observations. In either situation, there are dimensions of $x_{k-1}^{(i)}$ which are unaltered in m_k (the null space). On the other hand, the dimensions that lie in the range are updated and "pulled" as close as possible to the observation. This has the desirable result that all information is extracted from the observations, while the unobserved dimensions remain centered at $x_{k-1}^{(i)}$. When z_k has redundant observations and over-constrains $x_{k-1}^{(i)}$, Eq. 11 computes $\widehat{dx}$ by projecting onto J_x.

B. Particle proposal

With information from the observations incorporated, the algorithm creates a set $\mathcal{X}^{(i)} = \{ \; \mathcal{X}_1 \quad \cdots \quad \mathcal{X}_P \; \}$ of points centered around m_k. These P points discretely approximate the OIF (and thus avoid a Gaussian assumption as in [4, 16]) and are sampled to select $x_k^{(i)}$, the next generation particle.

Our algorithm proposes noise in the observation space, rather than the state space. States proposed in this way will lie closer to the actual observations. Along some dimensions, however, it may not be possible to convert uncertainty in the

observation space to uncertainty in the state space. This can occur when the Jacobian is singular due to a specific joint configuration or missing observations. When this occurs, the algorithm proposes using state uncertainty in this null space.

As shown in Fig. 5, we sample AGN perturbations from $\mathsf{w}_j \sim N(0, \Sigma_{sys})$. Σ_{sys} is an $M \times M$ matrix which expresses the uncertainty of the state in the observation space. Again, using a Taylor approximation:

$$\underbrace{\mathcal{X}_j}_{M\times 1} = \underbrace{m_k}_{M\times 1} + \underbrace{J_m^\dagger}_{M\times N}\ \underbrace{\mathsf{w}_j}_{N\times 1} \tag{12}$$

This equation uses the pseudo-inverse of the Jacobian, J_m, evaluated at m_k, to convert uncertainty in the observation space to uncertainty in the state space. Then w_j is simply projected onto $\mathcal{M}$ to produce the points $\mathcal{X}_j$.

J_m may be singular due to a specific configuration of the articulated object. In such a case, $\mathcal{X}_j$ will equal m_k along the null dimensions of J_m. This is a problem because no noise has been added along these null dimensions. In other words, w_j does not define a unique change in m_k and the pseudo-inverse extinguishes perturbations along those dimensions.

To see this, consider again the four-link chain shown in Fig. 6a with all links aligned along the horizontal axis. If a bearing-only sensor is positioned at $(-1, 0)$ looking along the horizontal axis, the Jacobian is degenerate; the entire linkage can slide along the horizontal axis without changing the bearing observations. Using Eq. 12, the particles shown in Fig. 6b are generated. Note that along the (horizontal) null dimension, there is no perturbation in the particles for the leftmost link. Increasingly to the right, there is some displacement in the horizontal dimension due only to motion in the constraint manifold. Thus Eq. 12 does not meaningfully distribute the particles, which is undesirable.

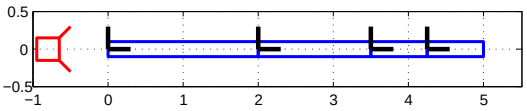
The solution is to use the state transition model to propose particles in the null space of J_m. In so doing, our method effectively prioritizes observations over the previous state, but the previous state can be used when the observations provide no information. Revising Eq. 12:

$$\mathcal{X}_j = m_k + J_m^\dagger\ \mathsf{w}_j + \underbrace{\mathcal{N}(J_m)}_{M\times M}\ \underbrace{v_j}_{M\times 1} \tag{13}$$

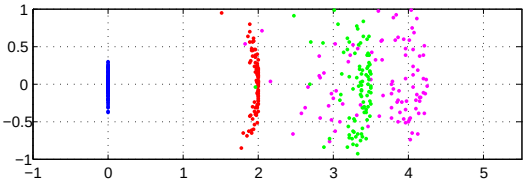
where $v_j \sim N(0, \Sigma_x)$ and Σ_x is the $M \times M$ covariance matrix for the state (as used by the baseline method). The $\mathcal{N}(J_m) = I - J_m^\dagger\ J_m$ matrix is the null space projector matrix. When using Eq. 13, particles are well distributed even when singularities exist in the Jacobian.

Our method draws samples from the high-dimensional observation space and projects them onto a (typically lower-dimensional) plane tangent to the configuration manifold. The samples are then transferred onto the manifold itself according to the Taylor approximation. An alternative could be to sample directly on the tangent plane, for example from a distribution suitably constructed from the sigma points [9] of Σ_{sys} (Fig. 5).

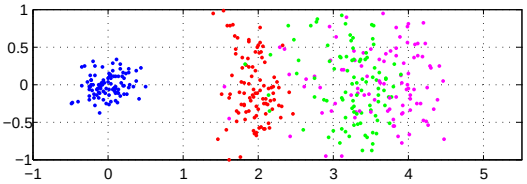
Although the samples produced by Eq. 13 are guaranteed to satisfy the kinematic equations (i.e., lie in the manifold),



(a) A bearing-only sensor (red) observes the four link kinematic chain. In this configuration, the system is singular because the chain can be moved along the horizontal axis without affecting the observations.



(b) Particles proposed with Eq. 12 have little motion along the singular dimension.



(c) Particles proposed with Eq. 13 use the state transition model to propose along the singular dimension, improving the sampling when the system is degenerate.

Fig. 6: Sampling at singularities

they may not satisfy the joint limits. As a result, the algorithm iterates Eq. 13, generating a sample and evaluating it against the joint limits. Finally, the OIF is evaluated using Eq. 1 and a discrete version of Eq. 2. Since all $\mathcal{X}_j$'s satisfy the joint limits and kinematic constraints, we can select $x_k^{(i)}$ by drawing from $\mathcal{X}$ according to the OIF probabilities.

V. Experiments

We compared a baseline particle filter and our approach on a household dishwasher (planar), PR2 robot (3D), and a construction-site excavator (3D) example. For each system, we (1) varied the number of particles, (2) performed 100 Monte Carlo simulations, and (3) evaluated the root mean squared error (RMSE) of the tracker. The excavator system is multi-modal, and we also compare with the performance of a UKF.

1) Dishwasher: Approximately 1300 frames of RGB-D data were collected of a dishwasher (see Fig. 7) being opened and closed. Ground truth and the 6-DOF kinematic model were established via manual annotation. Three independent TLD trackers were manually initialized on the door and drawers, providing pixel observations. (For details on TLD, see [10]; a tracker could have also been trained offline.) There are periods of missing observations when the door obscures the drawers or the TLD loses track. Our algorithm is still able to use these partial observations during particle proposal, c.f. Eq. 11.

Since only the positions of the dishwasher door and drawer are available, a singularity exists when the door is closed (vertical). In this configuration, observation movement in the

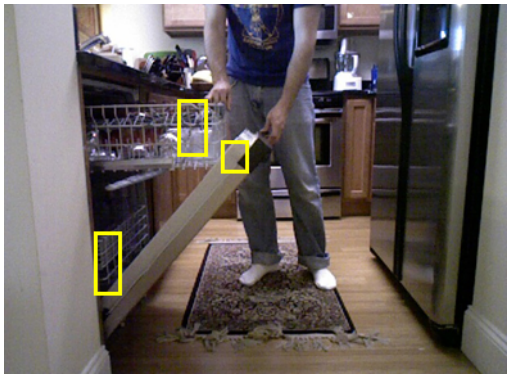


Fig. 7: A TLD tracker provided positions of the dishwasher's articulated links as input.

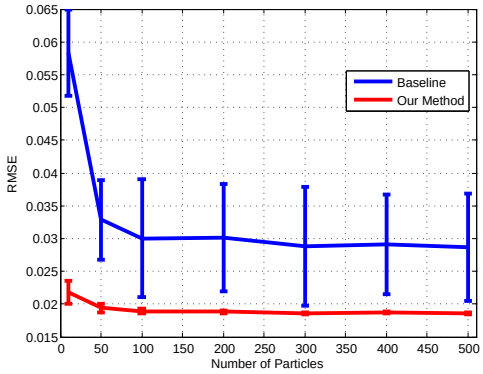


Fig. 8: In addition to lower RMSE, our method demonstrated less variation in accuracy while tracking the dishwasher, since it quickly recovered when missing observations resumed.

horizontal direction can be explained both by movement of the base pose or by a slight opening of the door and movement of the drawers. The null space term in Eq. 13 is crucial so that meaningful particles are proposed during the initial frames.

Fig. 8 shows results for the dishwasher (error bars show one standard deviation) and a video is available at http://rvsn.csail.mit.edu/articulated. In addition to higher accuracy, our method also exhibits substantially less variation. This is primarily due to periods when the observations were insufficient to make the system observable. Both methods would "wander" during these singularities, each proposing unconstrained particles in the null space. However, when observations became available again, our method was able to quickly recover and beginmo proposing particles near those observations. Our Matlab implementation executed at ~23 FPS, exceeding the 10 FPS rate of the RGB-D data. Finally, our method maintained ~50% effective particles, while the baseline had only about 5%.

2) PR2: In this experiment, we consider a PR2 robot holding a 60cm PVC pipe, and the goal is to estimate the pose of the pipe. This task is interesting because PR2 cannot grip the pipe rigidly due to its weight and length. As a result, the pipe tends to slip in the gripper; however, this slip occurs only along certain directions. As suggested in Fig. 9, the pipe may translate in the gripper along the pipe's long axis or along

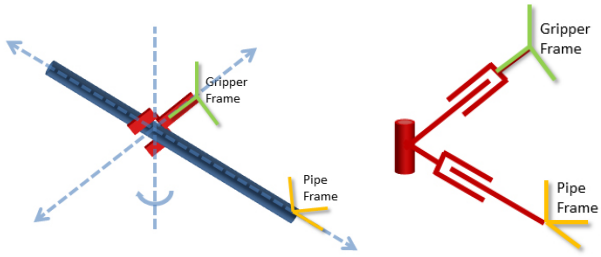


Fig. 9: The PR2's grip on the pipe is not rigid, but still constrains some movement. The system can be modeled as a 3-DOF kinematic chain.

the grip direction. The pipe can also rotate about the suggested axis.

We might wish to know the pose of the pipe so that it can be accurately manipulated — when, for example, inserting into a mating coupling. An obvious solution might be to track the pipe in RGB-D data using a RANSAC cylinder fit. A complication with this approach, however, is that the absolute pose of the pipe is not observable. Rotation about the pipe's long axis cannot be observed due to its rotational symmetry. Thus, even recovering a relative pose of the pipe is not possible by observing only the pipe itself.

Although the gripper does not hold the pipe rigidly along all six DOFs, it does provide rigid support along three DOFs. Thus, 6-DOF information from the pose of the gripper (as provided by the PR2's telemetry and forward kinematics) and 4-DOF information about the partial pose of the pipe (as provided by a RANSAC cylinder fit on RGB-D data) can be combined to track the pipe in 6-DOF. (We do not track the absolute orientation of the pipe about the long axis; this rotation is tracked relative to the initial orientation.) It is clear that the RANSAC estimate of the pipe's location will have errors; but the pose of the gripper is also subject to errors. Mechanical slop, imperfect PR2 models, and errors in the Kinect calibration all lead to errors in the Kinect-to-gripper transform.

We achieve the pipe tracking by explicitly modeling the DOFs between the gripper and the pipe. As shown in Fig. 9, the system can be modeled as a 6-DOF pose of the gripper and a kinematic chain with three joints (two prismatic and one rotational). The scenario in Fig. 10 demonstrates significant pipe rotation in the gripper as the gripper rotates. Velocity spikes correspond to events when the pipe passed a vertical orientation and "fell" (or, more accurately, slipped) in the gripper. During these two events, the pipe moved over 90 degrees in just 2-3 frames, during which time the state transition model fit the data particularly badly.

Fig. 11 shows the RMSE and number of effective particles for the PR2 experiment. Using only around 20 particles, our method achieves error levels lower than those achieved by the baseline method with up to 500 particles. Our method also achieves a higher N_{eff} indicating that the particles it does produce lie in higher-probability regions.

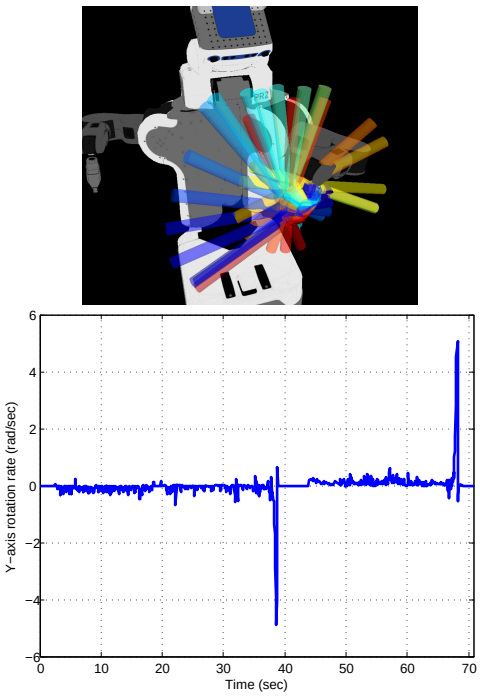


Fig. 10: In this sequence, the PR2 rotates the pipe. The top image shows the pipe poses color coded by time; the gripper rotates the pipe, moving it through red-yellow-blue poses. The bottom plot shows rotational velocity; two large velocity spikes correspond to times when the pipe underwent significant slip in the gripper.

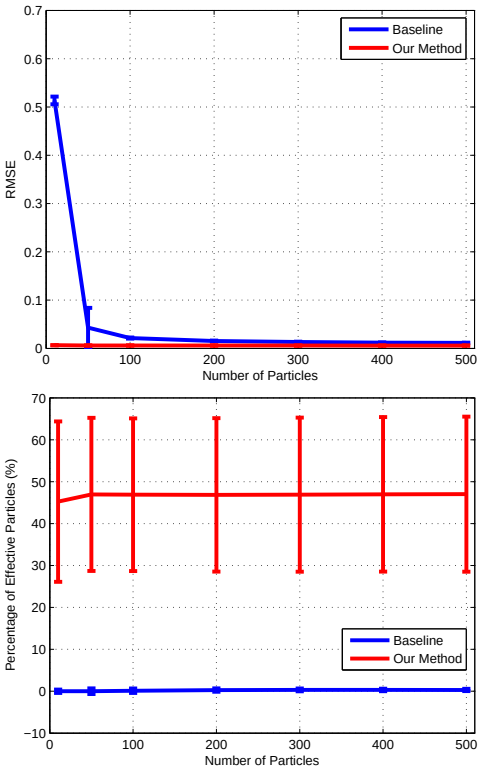


Fig. 11: RMSE and number of effective particle performance for the PR2 sequence in Fig. 10 are shown.

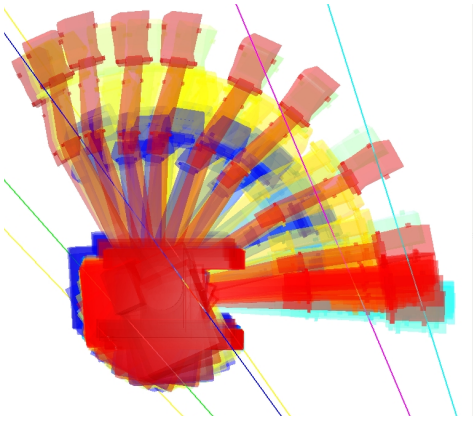


Fig. 12: The excavator loads a dump truck three times. Viewed from above, this graphic shows the entire 1645-frame sequence, color coded by time. The three loads correspond to blue, yellow, and red, in that order.

3) Excavator: We also visited the Heavy Construction Academy (HCA) in Brentwood, NH and observed an excavator operating. We collected data with a PointGrey CMLN-13S2C camera and Velodyne HDL-32E 3D LIDAR. (Velodyne data was used only for manually establishing ground truth.)

The excavator was a Caterpillar model CAT322BL, with a tracked base and rotating cab/engine; its heavy lift arm consists of three links. We constructed a URDF of the CAT322BL excavator from third-party datasheets. The datasheets were not complete, but provided enough information to make a reasonably accurate model.

The image-based observations were made using seven independent TLD trackers. The trackers were manually initialized during the first frame on the points shown in Fig. 1. The pixel observations included two points on the tracks, two points on the cab (the point on the back of the cab is not visible in this frame), a point on the boom, a point on the stick, and a point on the bucket. Each of the pixel observations defines a ray in 3D space (see Fig. 13), corresponding to two constraints, and is used to estimate the 10-DOF state of the excavator (six DOFs for the base pose and four arm joints). Fig. 12 shows the ground-truth path executed by the excavator for one scenario.

The pixel observations do not completely define the link locations. Combined with the internal DOFs, the excavator can "slide" along the rays; thus many configurations explain the observations, as shown in Fig. 15. This multi-modal distribution of configurations motivates our use of the particle filter.

From Fig. 14, we can draw several conclusions. First, our method exhibits a lower RMSE than the UKF. This is because the excavator kinematics and observation pixels/rays result in a multi-modal distribution, and the UKF periodically tracks the wrong mode. Second, the constant-velocity model did not improve performance for the UKF or baseline. This highlights the difficulty in creating a good state transition model: although we might expect a higher order model to perform better, it fails here because the excavator's motion is

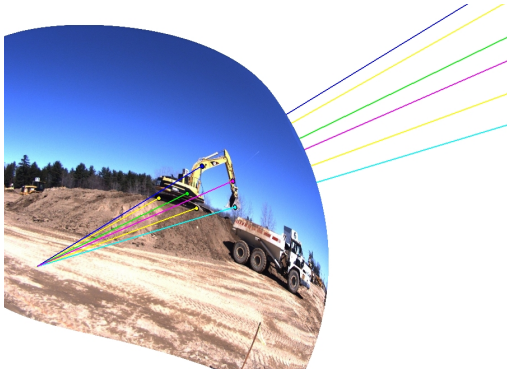


Fig. 13: Each pixel observation defines a ray observation in 3D space, emanating from the camera origin.

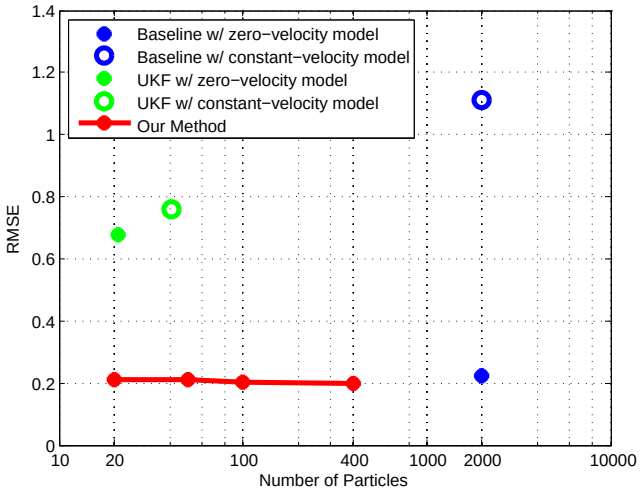


Fig. 14: Our method demonstrated lower RMSE with fewer particles than the UKF and the baseline particle filter. A video of the comparison is available at http://rvsn.csail.mit.edu/articulated.

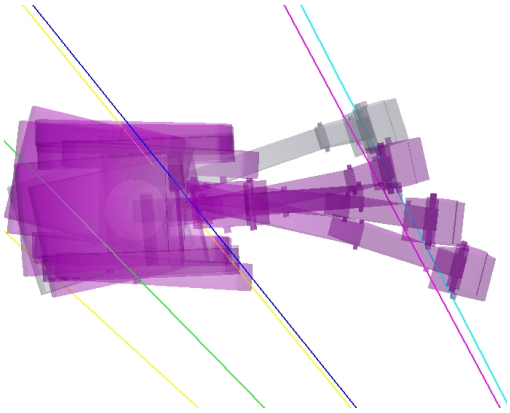


Fig. 15: Shown from above, the observations (colored rays) are generated by the actual configuration of the excavator (gray). Each colored ray intersects the excavator at the same colored point in Fig. 1. Other configurations (purple) explain the observations as well as the true configuration.

sufficiently irregular and the frame rate sufficiently low that constant velocity is inappropriate. Third, the baseline method requires 100 times as many particles as our method – and runs roughly 8 times slower than our method – to achieve the same RMSE.

A. *Frame rate*

We have demonstrated at least an order-of-magnitude reduction in the number of required particles to track at similar levels of RMS error. This enables a corresponding increase in our method's frame rate. Fig. 16 compares frame rates for the dishwasher, PR2, and excavator examples. In each case, our method was 4 to 8 times faster than the baseline method for comparable RMS error. Since the frame rate of the particle filter is linearly proportional to the number of particles, we might have expected a factor of 10 improvement. The algorithm did not reach this level, however, because of the discrete approximation required to calculate particle weights.

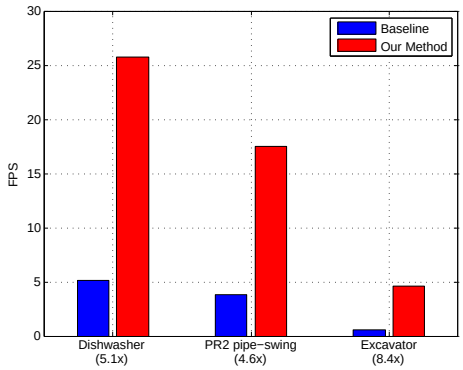


Fig. 16: The improved particle generation in our method resulted in a factor of 4-8 speed up over the baseline method. Results were generated on a 3.4 GHz processor running Matlab 2013.

VI. Conclusion

Our approach is most appropriate when the object's motion cannot be well predicted (relative to the quality of the observations). When the motion can be well predicted, a UKF or baseline particle filter approach might be a more suitable choice. It is also interesting to consider an increasing number of missing observations. When observations are unavailable, the algorithm relies on the state transition model. As less data is available, the performance converges to that of the baseline method (which always proposes with the state transition model).

We give an algorithm that augments observations with a model describing the constraints between them. It estimates the joint positions of an articulated object, and we demonstrate more than an order-of-magnitude reduction in the number of particles over a baseline implementation, and a corresponding increase in frame rate.

REFERENCES

[1] R. Cabido, D. Concha, J. Pantrigo, and A. Montemayor. High speed articulated object tracking using GPUs: A particle filter approach. In *ISPAN*, 2009.

[2] A. Comport, E. Marchand, and F. Chaumette. Kinematic sets for real-time robust articulated object tracking. *Image and Vision Computing*, 25(3):374–391, 2007.

[3] J. Deutscher, A. Blake, and I. D. Reid. Articulated body motion capture by annealed particle filtering. In *CVPR*, pages 126–133, 2000.

[4] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[5] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture. *IJCV*, 87, 2010.

[6] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *ICRA*, 2005.

[7] S. Hauberg and K. Pedersen. Data-driven importance distributions for articulated tracking. In Y. Boykov, F. Kahl, V. Lempitsky, and F. Schmidt, editors, *EMMCVPR*, volume 6819 of *Lecture Notes in Computer Science*. 2011.

[8] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *IJCV*, 29, 1998.

[9] S. Julier. The scaled unscented transformation. In *IEEE American Control Conference*, volume 6, 2002.

[10] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. *International Conference on Pattern Recognition*, 2010.

[11] D. Katz, M. Kazemi, J. Bagnell, and A. Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown articulated objects. Technical report, RI, 2012.

[12] D. Simon. *Optimal State Estimation*. Wiley & Sons, 2006.

[13] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *JAIR*, 41, Aug 2011.

[14] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[15] R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *CVPR*, 2006.

[16] R. van der Merwe, A. Doucet, N. de Freitas, and E. A. Wan. The unscented particle filter. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. 2000.

[17] Willow Garage. URDF, May 2013. URL http://www.ros.org/wiki/urdf.

[18] J. Ziegler, K. Nickel, and R. Stiefelhagen. Tracking of the articulated upper body on multi-view stereo image sequences. In *CVPR*, pages 774–781, 2006.