Touch Based Localization for High-Precision Manufacturing

Shiyuan Chen

May 2017

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Reid Simmons
Michael Kaess
Stefanos Nikolaidis

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

CMU-RI-TR-17-29

Keywords: Localization, Force and Contact Sensing, Particle Filter

Abstract

In traditional automated manufacturing, the motions of an industry robot are programmed step by step by humans for a specific task, which is costly and time-consuming. This greatly limits the application of industry robot when the production is relatively small. This thesis proposes two touch-based localization approaches extended from the importance-sampling particle filter for a potentially large and complex object in 3D workspace, both of which can achieve high precision and real-time localization speed and can be potentially used for automated manufacturing. The rigid-body particle filter assumes that an accurate geometry model of the object is provided, while the datum particle filter takes the engineering tolerance for manufactured part into consideration and provides a method to localize a task location defined by datums on an object with internal degrees of freedom. Both approaches have been evaluated in simulation. Our results show that the proposed approaches can quickly reduce the estimation accuracy in both translational and rotational dimensions within 10 measurements.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my academic advisor, Professor Reid Simmons, who has provided me with support and guidance throughout the whole course of my study. His vision, insight, and knowledge have guided me in all the time of research and writing of this work, and his patience and encouragement have always motivated me. Without his invaluable inspirations, this thesis would not have been possible.

I want to thank Professor Michael Kaess and Stefanos Nikolaidis for being my committee members and providing me with insightful comments and suggestions on this thesis.

I would also like to thank Brad Saund, with whom I have collaborated with during this work. The measurement simulation and prediction part of this thesis is heavily based on his work.

Finally, I would like to thank my parents and my family for their endless love and encouragement throughout my life. Their constant support is the greatest gift anyone has ever given me.

Contents

List of Figures

List of Tables

1 Introduction

Automated manufacturing requires high-precision localization of parts, which is currently achieved largely by human aid and the pre-design of fixed set of localization actions for robot arms. In these tasks, robots are programmed for a specific part, and are needed to be reprogrammed for a new task. Reprogramming the industry robots is costly, since the actions for each robot should be precisely designed and multiple robots should coordinate for different manufacturing and assembling processes of a single part. All of the actions of a robot arm should be planned step by step with high precision. Because the robots are fixed, sometimes a new assembly line is required for the new task. This greatly limits the application of manufacturing robot when the production of a single object is relatively small.

Great efforts have been made in order to increase the flexibility of the industry robots. One of the most heavily explored fields is how to allow the robots to perceive the environment themselves rather than programmed by human. Localization is such a task for a robot to construct the space information of its surroundings, which includes localizing itself and localizing the objects that we are interested in. This thesis will focus on object localization techniques.

There are two major categories of localization in terms of sensors that are used for perceptions. Vision-based approaches make use of visual features of objects in order to estimate their locations and poses by adopting visual sensors such as RGB-D cameras and LIDAR systems, while point-based approaches localize by either directly contacting the object surface using probes or measuring the distance using point sensors.

In this thesis, the problem is defined to localize the full $SE(3)$ pose with 6 degrees of freedom (DOFs) of an manufactured part using touch probe and achieve high precision that is comparable to the requirement in manufacturing. In more detail, the part to be localized is assumed to be fixed in the workspace, and the robot will make direct measurements on the part in order to localize it. The object can potentially be large and have complex shape. Prior information of the shape of the part is required. The geometry (CAD) model that is used in this work is triangle mesh stored in a STL file, which is a common practice in industry. This thesis starts with the assumption that the part is a rigid body and will match the given CAD model exactly. This mildly simplifies the problem, since the model will tell us exact relative position and one can localize

the whole object by localizing its sub-area.

Rather than calculating a single best estimate of the pose, the uncertainty in knowledge is represented by a probability distribution. Estimating the probability distribution is important both for allowing multiple good estimates and knowing when the object is localized sufficiently. For feasibility, this probabilistic belief is represented numerically by a list of points drawn from the true distribution called *particles*. A measurement updates the belief using a *particle filter* [28].

Unlike some other Bayesian estimation approaches such as Kalman filters [16], extended Kalman filters [15] and unscented Kalman filters [14], particle filters can easily model non-Gaussian and multi-modal probability distribution. For touch localization, contact sensors yield a highly non-linear measurement model, and the belief can frequently become multi-modal when multiple configurations are all consistent with the measurement. These properties make particle filter a popular approach for touch localization tasks.

Standard particle filters behave poorly both when the dimensionallity of the system grows large and when measurements become precise [17]. In this problem a large initial uncertainty is considered in the six-dimensional system, and measurements with low uncertainty are required to achieve high tolerance performance. Updating the belief after performing such a measurement will eliminate most particles, leaving a poor representation of the posterior. This problem is called particle starvation.

This work first presents a solution to particle starvation during touch localization (Section 3). A particle filter is designed for rigid-body objects with an alternate update procedure that is able to both combine accurate measurements with the prior and achieve real-time localization performance.

However, during the manufacturing and assembly processes there are tolerances between different sections of the assembly. A *datum* is defined as a geometric constraint within the object that is used as the reference to define the location of one section of the part with respect to another section. The tolerance is the allowed deviation of the actual manufactured dimensions from the nominal designed dimensions. A part is assumed to be able to be divided into precisely manufactured sections.

The introduction of tolerance increases the degrees of freedom of the system. These internal DOFs can be modeled as transformations with uncertainty between sections of the object. For objects with internal tolerances, perfectly localizing a single datum will not necessarily reduce the uncertainty of the full system sufficiently to perform the desired task. On the other hand, it is usually only necessary to localize a subset of the sections of an object.

The rigid-body assumption will be relaxed later in this thesis, where only the probability distributions of such geometric relationships are provided and tolerance between different rigid sections of the part is allowed. The task is defined as estimating the pose of a goal feature given multiple measurements obtained through probing. The proposed rigid-body particle filter is generalized to maintain and update the joint probabilistic distribution of different rigid sections of a part based on measurements on only some of them. This datum particle filter is able to achieve very high precision given both of pose uncertainty and internal uncertainties.

The contributions of this thesis include:

1. Proposed a probabilistic approach to update the belief of the pose of an object both in high precision and in real-time speed based on touch measurements, which can model nonlinear and non-Gaussian distributions.
2. Extended the proposed approach for object with datums given only its partial geometric information.

This thesis first summarizes the background and related work of localization using Bayesian estimation techniques, especially commonly-used importance-sampling particle filter (Section 2). The framework of rigid-body particle filter for high-precision real-time localization is then introduced (Section 3), and is later generalized to datum particle filter for objects with coupled rigid sections (Section 4). The proposed approaches were both evaluated on a simulated part. The rigid-body particle filter was also tested on our robot.

2 Background and Related Work

The localization task of a 3D object requires updating the estimate of its $SE(3)$ pose given a set of measurements and a initial $belief$ of its pose. A belief $bel(\cdot)$ is an estimated probabilistic

distribution that is either directly given or obtained during the localization process. For particle filters, a *state* represents a certain pose of the object. The state belief is estimated by a group of particles, each of which maintains a candidate discrete state of the object.

Unlike localization with cameras, the assumption that an *good* initial belief is provided is important for touch localization due to the low efficiency in perceiving a large area when point sensors are used. If the initial uncertainty of the pose is large, vision-based approaches can be applied first to refine the initial belief.

Let state $x_t \in SE(3)$ represent the pose of the object in 6-dimensional *configuration space* [20] at time t. The prior $bel(x_0)$ is the belief of the state before any measurement. Given a series of measurements $\{z_t\}$ and controls $\{u_t\}$ of the object, our goal for the localization is to estimate the posterior:

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{1}$$

For touch localization, at any time t, a good measurement z_t will result in an actual contact between the surface of the probe and the object, thus only a subset of all potential poses from $bel(x_t)$ will be consistent with this measurement z_t. When z_t is accurate, the object with a candidate pose x_t should exactly contact with the surface of the touch probe without any intersection. All poses that are consistent with this measurement will be lie on a lower dimensional manifold in the configuration space. This manifold is called a contact manifold [17]. For each accurate measurement, the updated state belief should then become the intersection of current belief and its corresponding contact manifold.

When there is measurement error from the sensors, the object and the probe may overlap with each other or have some clearance between the two surfaces due to inaccurate measurement. In this case, the probability that a pose x_t is consistent with the current measurement z_t is represented by $p(z_t|x_t, z_{1:t-1}, u_{1:t})$ and determined by the measurement error.

2.1 Particle Filter

An important assumption of the localization process is *Markov property*. Markov property assumes that the state x_t at any time t is *complete* and thus it is the best representation of the states,

observations and controls in all previous steps [28]. One such example is localizing a target object, where the observation in time t only depends on its current pose x_t, regardless its previous observations and poses, i.e. $p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t)$.

Given Bayes rules and Markov assumption, the target posterior given above can be simplified to the following equality:

$$
\begin{aligned}
bel(x_t) &= p(x_t|z_{1:t}, u_{1:t}) && (2)\\
&= \eta p(z_t|x_t, z_{1:t-1}, u_{1:t}) p(x_t|z_{1:t-1}, u_{1:t}) && (3)\\
&= \eta p(z_t|x_t, z_{1:t-1}, u_{1:t}) \int p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t}) p(x_{t-1}|z_{1:t-1}, u_{1:t}) dx_{t-1} && (4)\\
&= \eta p(z_t|x_t) \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1} && (5)
\end{aligned}
$$

Equation 3 and 4 are obtained directly from Bayes rules, where $\eta = p(z_t|z_{1:t-1}, u_{1:t})^{-1}$ is the normalization factor. Equation 5 is obtained by applying Markov assumption. Note that $p(x_{t-1}|z_{1:t-1}, u_{1:t}) = p(x_{t-1}|z_{1:t-1}, u_{1:t-1})$, since u_t happens after x_{t-1} and should have no influence on the belief $bel(x_{t-1})$. Equation 5 provides a recursive way to estimate state x_t from its previous estimation based on the new measurement and control at time t. This process is called Bayes filtering, where forward measurement probability $p(z_t|x_t)$ and state transition probability $p(x_t|x_{t-1}, u_t)$ are called *measurement model* and *motion model*, respectively.

There are different approaches that are variants to the basic Bayes filter. For nonparametric target distribution, one of the most commonly used techniques is particle filter [28]. Instead of precisely computing and maintaining the belief in parametric forms, particle filters approximate the posterior by keeping track of a finite set of discrete state samples with the corresponding weight for each sample. Each sample together with its weight here is called a particle, which is drawn from the target distribution. If each sample has unit weight, the density of the particles will correspond to the target distribution.

Let $X_t = \{{}^j x_t\}_{j=1}^N$ represents a set of N particles at time t. Directly sample from the target distribution $bel(x_t)$ is difficult. Alternatively, the particle filter algorithm proposes to sample

Algorithm 1 Importance-Sampling Particle Filter

Input: particles $X_{t-1} = \{{}^j x_{t-1}\}_{j=1}^N$
Input: observation z_t and control u_t
Output: particles $X_t = \{{}^j x_t\}_{j=1}^N$

$X_t \leftarrow \emptyset$, $\hat{X}_t \leftarrow \emptyset$
for $j = 1$ to N **do**
    ${}^j x_t \sim p(x_t | {}^j x_{t-1}, u_t)$
    ${}^j w_t \leftarrow p(z_t | {}^j x_t)$
    $\hat{X}_t \leftarrow \hat{X}_t \cup \langle {}^j x_t, {}^j w_t \rangle$
end for
for $j = 1$ to N **do**
    draw with replacement ${}^i x_t \propto {}^i w_t$
    $X_t \leftarrow X_t \cup {}^i x_t$
end for

from a *proposal distribution* $g(x_t)$:

$$g(x_t) = \int p(x_t | x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1} \tag{6}$$

This turns out to be a much easier task, since the samples from $t-1$ are already an estimation of $bel(x_{t-1})$, thus the sample from $p(x_t | {}^j x_{t-1}, u_t)$ for all particles ${}^j x_{t-1} \in X_{t-1}$ is exact the sample from the proposal distribution $g(x_t)$.

The final samples X_t can be obtained by weighting each new sample from the proposal distribution using *importance factor* ${}^j w_t$. The importance factor is defined by the quotient between the target distribution $bel(x_t)$ and the proposal distribution $g(x_t)$:

$${}^j w_t = \frac{bel({}^j x_t)}{g({}^j x_t)} = \eta p(z_t | {}^j x_t) \tag{7}$$

This process is called *importance sampling*. After the weighting, the samples $\{\langle {}^j x_t, {}^j w_t \rangle\}_j$ should be the unbias estimation of $bel(x_t)$. The original particle filter uses an extra *resampling* process to resample the particles to unit weight, as shown in Algorithm 1.

With importance-sampling particle filter, one can iteratively update the state belief $bel(x_t)$ from its prior based on the measurement model and motion model of the system. The density of final samples will represent the probabilistic distribution of the target posterior.

In this work, the importance sampling process is not used for the proposed modified particle filters due to the particle starvation problem (see Section2.2). An alternative rejection sampling process is used instead.

2.2 Related Work

Vision-based approaches have experienced great improvements over recent years. The problem is known as pose estimation, which is the process to find the transformation from the object frame in the camera coordinates system to the 3D model frame given a 2D normal image or 3D depth image. Traditional methods use handcrafted image features to extract points of interest (e.g. edges and corners) in order to match the 3D models (represented by triangle mesh or point cloud) before computing the transformation [4][13][11][9]. Learning-based approaches for pose estimation have become popular recently following the success of convolutional neural network (CNN) [19]. Many of them extended the traditional approaches by using learned features and descriptors from CNN for pose classification or regression algorithms [2] [24][29].

There have also been a variety of approaches that allow robots to localize objects solely with contact sensors. Different contact sensors have been explored and developed, which includes basic binary sensors, 6-axis force and torque sensors [5], soft tactile sensor arrays [10], and bio-inspired fingertips [7]. Localization with laser sensors, which can be regarded as an extended form of point contact sensors, has also been used in the high-precision CNC localization, where a 3D point cloud is acquired in order to estimate the transformation between the actual and planned pose [22]. Similar to visual approaches, triangle mesh and point cloud are both commonly used.

Both kinds of approaches have their pros and cons. Vision-based pose estimation approaches are efficient during localization, but they suffer from very low precision compared to the industrial requirements. Particularly, large dataset for the object being localized is also required for learning-based pipelines to work, which is impractical in industry since manufactured parts usually have very unique and irregular shapes. In the contrary, touch-based approaches are not as efficient as their counterparts, since conducting contacts requires the robot arms to perform actual movements. However, with the help of accurate contact sensors, high precision can be achieved. For automated manufacturing and assembly, we are more interested in achieving high localiza-

tion precision, which is usually less than 1mm for displacement error. Touch-based approaches proved to be more suitable for these tasks.

Particle filters have been widely used and developed since their introduction. However, they will experience *particle starvation* for measurements with very low uncertainty and objects with a high dimensional configuration space [27]. In importance sampling each particle is weighted by the probability of the measurement conditioned on the state that particle represents. This is usually followed by resampling, where particles are redrawn from the set of weighted particles with probability proportional to their weights. The effectiveness of importance sampling relies on the existence of multiple particles consistent with the measurement, such that inconsistent particles will have low weights and be unlikely to be resampled, but a sufficient number of particles will be resampled to model the true belief of the state.

Importance sampling tends to break down in situations with accurate measurements and low densities of particles. This is because when a measurement is consistent with the true prior belief yet no particles are consistent with the measurement, a situation called *particle starvation*, resampling will yield a set of particles that does not model the true posterior belief. A more accurate sensor measurement is consistent with a smaller volume of state space, thus a higher density of particles is required. For higher dimensional state spaces and more accurate sensors the number of particles required becomes prohibitively computationally expensive. This leads to the counter-intuitive result that particle filters tend to perform worse as measurement accuracy increases [17].

Koval introduced the Manifold Particle Filter to address this issue by implementing different sampling and weighting strategies compared to the traditional particle filters [18][17]. Instead of sampling particles from the motion model and weighting them based on the observation, samples are directly drawn from the contact manifold, given the observation. This provides a fast and robust solution for objects with simple shapes. For complex object geometries, as in our case, direct, efficient sampling from the contact manifold becomes difficult.

Petrovskaya tackles particle starvation during touch localization by combined Monte Carlo approaches with annealing as a smoothing technique [21]. She introduced Scaling Series algorithms for 6-DOF global tactile localization in both full-constrained and under-constrained sce-

narios to overcome particle starvation by adjusting particle density depending on the complexity of the posterior. Multiple iterations through the measurement data are used and the precision of the belief is scaled from low to high in order to avoid unnecessarily precise estimates in unlikely regions of belief space. This approach is complementary to the proposed methods in this thesis, and implementing multiple passes on these methods could lead to a faster update process.

All of the work mentioned so far assumes the prior information of the object geometry matches the real piece perfectly. Hebert et al. [12] solve a touch localization problem using a Bayes filter where an object may have additional unknown parameters describing the shape, such as a screw driver with an unknown length handle. However, for localization, they only used one additional parameter with a total of 4 degrees of freedom for the joint belief and did not solve the particle starvation problem.

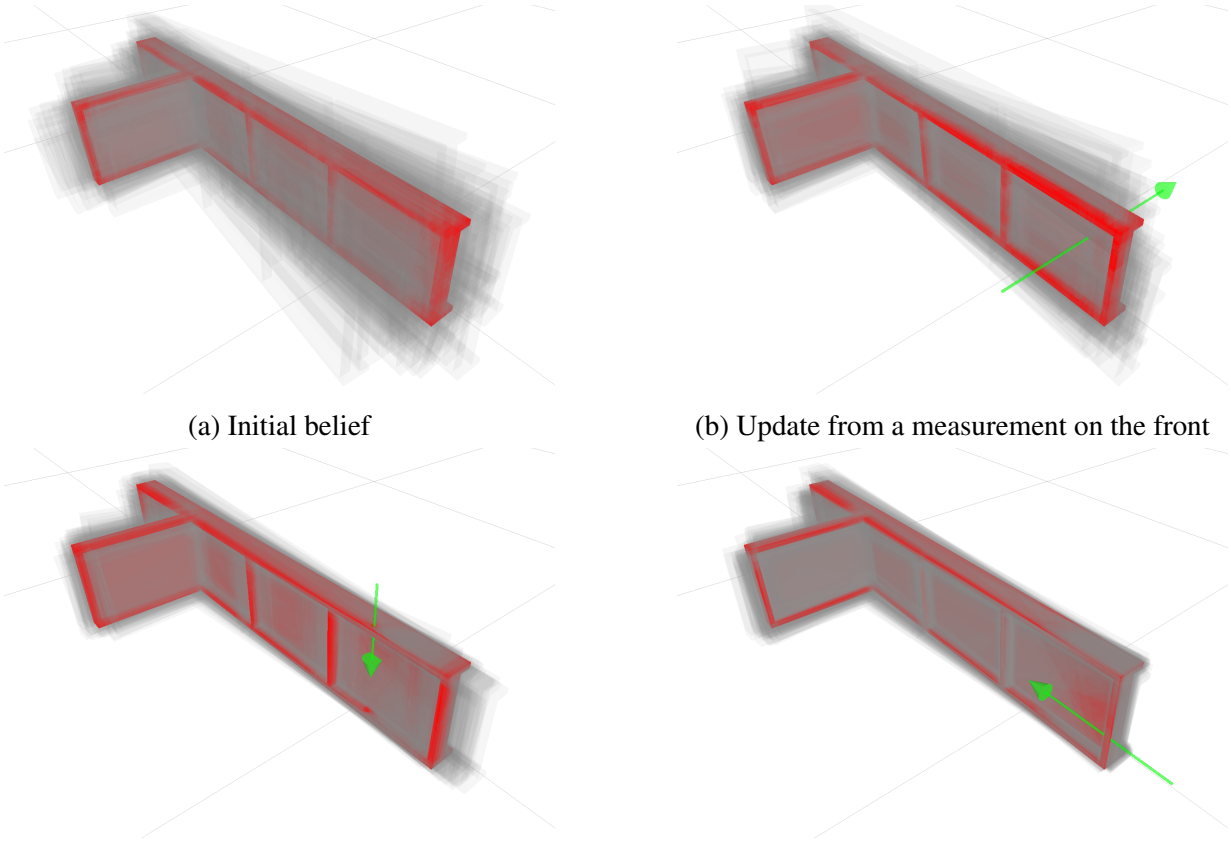
3 Rigid-Body Particle Filter

In this section, the object is assumed to be a rigid body and the given CAD model matches the object exactly. The task is to determine the pose of an object by choosing and performing touch measurements using touch probe, given the geometry of the object and prior belief over the distribution of poses.

The geometry of the object to be localized is stored in a STL file using a triangular mesh. The *part frame* is defined as the frame attached to this mesh. The pose of the object can then be defined as the transformation between this part frame and the *world frame* in the workspace. The object is assumed to be fixed in the workspace during measurement and localization, thus the configuration will not change during the localization process.

3.1 Problem Formulation

The particle filter is updated based on a set of measurements $Z_t = \{z_1, ..., z_t\}$ made directly on the object by the robot. Each particle in the particle filter represents a single potential pose $x \in SE(3)$ of the object. For a rigid body, the pose includes both translational dimensions (t_x, t_y, t_z) and rotational Euler angles (α, β, γ). The state is represented as a 6D vector $(t_x, t_y, t_z, \alpha, \beta, \gamma)$



(a) Initial belief

(b) Update from a measurement on the front

(c) Update from a measurement on the top

(d) Update from a measurement on the side

Figure 1: Visualization of rigid-body particle filter
(a): The initial belief of the object (gray) and its true state (red). For clarity in the image, only 50 particles are shown.
(b): The belief of the object after the first measurement (arrow) on the front surface. The object is partially localized.
(c): A following measurement (arrow) on the top side further localizes the object.
(d): Measuring on the side surface now has much lower probability to miss and provides more extra information about the pose of the object.

in the configuration space.

The particle filter is visualized in Figure 1. In each figure, the red part represents the ground true pose of the object during the simulation, and the parts in gray shows 50 potential poses sampled from a maximum of 500 particles in the particle filter. Green arrows represent the first three consecutive simulated measurement actions. Figure 1b to Figure 1d also show the updated belief after the corresponding measurement.

Belief

At any time step t, the target state belief is $bel(x_t) = p(x_t|Z_t)$ based on Equation 1. The omission of control series $\{u_t\}$ is due to the fact that the object is assumed fixed in the workspace during localization, which also implies that the motion model for the particle filter is static, i.e. $p(x_t|x_{t-1}) = \mathbb{1}(x_t = x_{t-1})$ where $\mathbb{1}(\cdot) = 1$ when the condition is true and zero otherwise. $\{u_t\}$ will be omitted in the rest sections for simplicity.

Measurements are a non-linear probabilistic function of the true state: $z_t \sim p(z_t|x_t)$. The belief $bel(x_t)$ is then calculated recursively as in Equation 5. Each new measurement value triggers an update to the belief $bel(x)$.

Sensor and Measurement Model

A touch probe is a spherical tip mounted to a rigid shaft connected to the end effector capable of detecting contact. The sensor returns a binary feedback of whether the probe is in contact with the object or not. From the robot arm configuration the position of the probe can be easily obtained.

A measurement action, $\mathcal{M}$ is defined by a start point $\mathcal{A}_p$ for the probe and a linear trajectory vector $\mathcal{A}_v$ both in $\mathbb{R}^3$. The measurement value m is the distance the probe travels in the direction of $\mathcal{A}_v$ until contact is made. The point of contact can then be recovered by $A_p + m\ \frac{\mathcal{A}_v}{||\mathcal{A}_v||}$. The entire information obtained from the measurement at time t is $z_t = \{\mathcal{M}_t, m_t\}$. Measurement error exists due to sensor error and robot uncertainty including positional error. For the robot arms and sensors used in manufacturing the uncertainty added typically ranges from 0.1 to 1mm

All the methods presented in this thesis also apply to a 1D laser range finder. This offers faster

measurements without requiring contact with the part directly. Similarly, for the laser sensor, the measurement $z_t = \{\mathcal{M}_t, m_t\}$, where $\mathcal{M}$ is defined by a center position $\mathcal{A}_p$ for the sensor and a linear vector $\mathcal{A}_v$ for the laser direction, and m_t is the distance measured by the sensor.

3.2 Rejection Sampling Method

The importance-sampling particle filter suffers from particle starvation problem (see Section 2). This is especially the case for real-time task, since the time complexity of particle update is linear to the number of particles and exponential to the number of dimensions of the configuration space: one can choose to have more precision result with the sacrifice of the speed or vice versa, but cannot achieve both high precision and real-time speed.

An alternate approach is to use rejection sampling [23][3]. Rejection sampling does not require a high density of particles to avoid particle starvation. It explicitly guarantees the same number of unique states by drawing more candidate samples from the motion model during the sampling process and then only accepting those that are consistent to the measurement. Rejection sampling is more robust than importance sampling when the current estimate is poor, e.g. no particle is close to the true state, since the importance-sampling particle filter resamples particles that are only better than other samples from this poor estimate, yet neither of them might be consistent to the measurement. In contrary, rejection sampling can be tuned so that it only accepts consistent particles, with the compromise of more samples drawn from the motion model. On the other hand, one can simply check the number of samples evaluated before achieving sufficient number of accepted particles to determine the quality of the estimate. Therefore, the failure of rejection sampling is far easier to notice and resolve. Most importantly, one can increase the limits of state dimensionality and measurement accuracy that can be handled efficiently.

Rejection sampling generates independent samples from a density f by sampling from a different proposal distribution g, where $g(x) > 0$ whenever $f(x) > 0$. A constant $1 < M < \infty$ is determined such that $f(x) \leq Mg(x) \; \forall \; x$. A sample ${}^j x$ drawn from $g(x)$ is accepted with probability $f({}^j x)/Mg({}^j x)$ and rejected otherwise. The process is repeated until the desired number of samples has been accepted.

For touch localization in this work, one wish to sample particles from the posterior $bel(x_t)$ but

cannot do so directly. The target distribution $f(x_t)$ is defined as the state belief $bel(x_t)$ at current step t. According to Equation 5, $f(x_t) = \eta p(z_t|x_t) \int p(x_t|x_{t-1}) bel(x_{t-1}) dx_{t-1}$. The proposal distribution $g(x_t)$ is selected to be the same with the prior of the sampling for particle filter, i.e. $g(x_t) = \int p(x_t|x_{t-1}) bel(x_{t-1}) dx_{t-1}$. The constant $M = \eta$, since $\eta = p(z_t|z_{1:t-1})^{-1} \geq 1$ is constant with x_t and x_{t-1}, and when $\eta = 1$, the target distribution is identical with the proposal distribution. For a sample ${}^j x_t$, the probability of acceptance is then given by the measurement model $f({}^j x_t)/Mg({}^j x_t) = p(z_t|{}^j x_t)$.

Broad Particles: Similar to importance-sampling particle filter, the sampling from the proposal distribution $g(x_t)$ is done by sampling ${}^j x_t \sim p(x_t|{}^j x_{t-1})$ from the motion model $p(x_t|{}^j x_{t-1})$ for each particle ${}^j x_{t-1}$. Since the motion model is static and each state ${}^j x_{t-1}$ is a discrete vector in 6-D configuration space, the unique states represented by particles X_t are always a subset of previous set of states. Because only a finite number of samples are maintained, this would lead to the same starvation problem.

Therefore, prior to sampling from the motion model, the continuous prior belief $bel(x_{t-1})$ is reconstructed by broadening each of the particles using a multivariate *Gaussian kernel* as in Equation 16. The Gaussian kernel is applied to each particle, with the kernel covariance proportional to the covariance of the particle states in this time step. This reduces particle starvation, as even if no prior particle is consistent with the measurement, the continuous belief generated fills in the gaps between particles.

Binary Measurement Model: States sampled from this continuous prior are then rejected if they are inconsistent with the measurement according to the acceptance ratio defined by the measurement model. The measurement model $p(z_t|x_t)$ defines the probability of the measurement z_t given a new sampled state x_t from the continuous prior, x_t determines the pose of the object in the workspace, and the measurement z_t determines the position of the sensor. For touch probes, the minimal distance $dist$ between the probe at the measured position and the surface of the object with the pose of x_t is a good measure of this probability: an accurate measurement will result in a zero distance, while the probability $p(z_t|x_t)$ decreases as the minimal distance increases. Therefore, it is intuitive to define the measurement model as $p(z_t|x_t) = p(dist|x_t, \sigma_m)$ where σ_m is the standard deviation of the sensor measurement noise.

While the formulation of rejection sampling allows us to model complicated measurement model $p(dist|x_t, \sigma_m)$, a binary measurement model is implemented instead. All sampled particles where the measured point is sufficiently far from the surface of the object are rejected. "sufficiently far" is defined as more than 3 standard deviations σ_m of the sensor measurement noise from both inside and outside direction of the object.

Compared to a non-binary measurement model where the acceptance ratio is defined by a continuous $p(dist|x_t, \sigma_m)$, the binary model can avoid the acceptance of particles that are poorly consistent to the measurement, which is a problem for importance sampling as discussed above. In addition, it also leads to a faster evaluation process since no computation of complicated probability is required. As a low uncertainty measurement will accept only a thin manifold in state space, the probability of sampling a particle consistent with the measurement may be low, and a lot of sampling may be required, therefore the rejection process should be very fast.

3.3 Fast Evaluation of Samples

To reduce the computational cost for evaluation per sampled state a discretized space is used, known as a *distance transform* [6], to precompute and cache the minimum unsigned distance $D_f(p)$ from point p in voxelized space to the object surface $\partial S \subseteq \mathbb{R}^3$. To improve the computation speed, the squared Euclidean distance transform $D_f^2(p)$ is first computed and defined as:

$$D_f^2(p) = \min_{q \in \mathbb{R}^3} (\|p - q\|_2^2 + f(q)) \tag{8}$$

$$f(q) = \begin{cases} 0, & \text{if } q \in \partial S \\ \infty & \text{otherwise.} \end{cases} \tag{9}$$

where $\|p - q\|_2^2$ is the square of ℓ^2-norm, and $f(q)$ equals 0 when the point q is on the surface of the object and infinity otherwise. For any given point p in 3D space, Equation 8 computes the minimum squared Euclidean distance among p and any point on the surface of the object. The Euclidean distance $D_f(p) = \sqrt{D_f^2(p)}$ is then obtained for each voxel. As the object is fixed during the localization process, voxelization can be done for the entire piece based on the given

CAD mesh model in the precomputation step.

Voxelization: Voxelization is the key part to transform the mesh model to axis-aligned discretized space, which can be stored and accessed easily as a standard array. The array form of the CAD model can greatly facilitate the computation of the distance transform, as described below. Each voxel is assumed to be a cube in 3D space. A fast 3D Triangle-Box Overlap method [1] is used to label the voxels that overlap the mesh triangles of the object surface. The voxel map is then mapped to a binary-valued 3D array $f(q)$, where each value is either 0 or ∞ depending on whether the corresponding voxel overlaps the object surface.

Voxelized Distance Transform: The computation of distance transform $D_f(p)$ takes the input of the computed binary array $f(q)$ (Equation 8), and a linear-time algorithm for 3D distance transform construction [6] is then used. For an 1D squared Euclidean distance transform, the algorithm basically computes the lowest envelope of the parabolas $(p-q)^2+f(q)$ for all voxelized q. The algorithm constructs the 3D distance transform by recursively performing the construction of 1D distance transform for each of the three dimensions in turn, with $f(q)$ replaced by the resulting distance along the previous computed axis. Let $p = (p_x, p_y, p_z)$ and $q = (q_x, q_y, q_z)$, Equation 8 can then be rewritten in terms of 1D distance transform as:

$$
\begin{aligned}
D_f^2(p_x, p_y, p_z) &= \min_{q_x,q_y,q_z} ((p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2 + f(q_x, q_y, q_z)) \\
&= \min_{q_x}((p_x - q_x)^2 + \min_{q_y}((p_y - q_y)^2 + \min_{q_z}((p_z - q_z)^2 + f(q_x, q_y, q_z)))) \\
&= \min_{q_x}((p_x - q_x)^2 + \min_{q_y}((p_y - q_y)^2 + D^2_{f|_{q_x,q_y}}(p_z))) \\
&= \min_{q_x}((p_x - q_x)^2 + D^2_{f|_{q_x}}(p_y)) \qquad (10)
\end{aligned}
$$

$$
f(q_x, q_y, q_z) = \begin{cases} 0, & \text{if } q \in \partial S \\ \infty & \text{otherwise.} \end{cases}
$$

$$
D^2_{f|_{q_x,q_y}}(p_z) = \min_{q_z}((p_z - q_z)^2 + f(q_x, q_y, q_z)) \qquad (11)
$$

$$
D^2_{f|_{q_x}}(p_y) = \min_{q_y}((p_y - q_y)^2 + D^2_{f|_{q_x,q_y}}(p_z)) \qquad (12)
$$

where $D^2_{f|_{q_x,q_y}}(p_z)$ is the 1D distance transform of f restricted to z axis indexed by (q_x, q_y) pairs, and $D^2_{f|_{q_x}}(p_y)$ is the 1D distance transform of $f' = D^2_{f|_{q_x,q_y}}(p_z)$ restricted to y axis indexed by

q_x. The 3D distance transform can then be computed by first performing 1D transforms along z axis of the voxel map for each pair (q_x, q_y), and then performing 1D transforms along y axis of the result for each q_x, and finally performing 1D transforms along x axis of the previous result. The resulting distance transform is also stored in an array for constant time access during the evaluation of sampled states.

Fast Evaluation: Different configurations result in different poses of the object in the workspace, which makes it difficult to compute the distance transform directly in the world frame. Instead, the computation of the voxel map and distance transform is relative to the part frame, where the object is assumed fixed during the entire localization. Each measurement z_t in the workspace is then transformed into the part frame, where the transformation $T(x_t)$ comes from the pose of the sampled state x_t. Therefore, by transforming back to the object frame, all measurements on this same object can share the same distance transform, where the minimal unsigned distance $dist_u(z_t, S)$ between each measurement z_t and the object $S(x_t)$ with sampled pose x_t can be obtained directly:

$$dist_u(z_t, S(x_t)) = D_f(T(x_t)^{-1} z_t) \tag{13}$$

Because $dist_u(z_t, S)$ is the absolute value of distance between the center of the spherical tip of the probe and the object, when the radius r_p of tip is non-zero, whether the center of the probe is inside the object $S(x_t)$ with sampled pose becomes important. The signed distance $dist(z_t, S(x_t))$ between the probe and the object can be obtained from the unsigned distance, as shown in Equation 14, by checking whether the voxel is inside or outside of the object. For the manifold shape object, ray-casting is applied from the corresponding voxel in a certain direction: the voxel is inside of the object only if the number of intersections between the ray and mesh is odd:

$$dist(z_t, S(x_t) = \begin{cases} dist_u(z_t, S(x_t) - r_p, & \text{if } z_t \notin S \\ -dist_u(z_t, S(x_t) - r_p, & \text{otherwise.} \end{cases} \tag{14}$$

The unsigned distance $dist(z_t, S)$ is always 0 in the ideal case, however, when evaluating a

sampled state, only those that satisfy $|dist(z_t, S)| > t_d$ will be rejected, where t_d is the tolerance selected according to the measurement uncertainty of the touch probe and the robot. If the distance is within the tolerance, ray-casting is then applied to check intersections from the start point $\mathcal{A}_p$ along the path vector $\mathcal{A}_v$ in order to determine whether the path is free of collision with other parts of the object. This will eliminate false positives which satisfy the distance condition but have collision between the robot and the object.

When the measurement is very accurate, in order to sample enough particles from the prior belief, a large number of states will get rejected, which makes the ray-casting for all sampled states computationally expensive. Instead, early rejection is applied using the upper bound of the signed distance $dist_u(z_t, S) + r_p$ before ray-casting, since $dist_u(z_t, S) + r_p \geq |dist(z_t, S)| > t_d$. Therefore, the particles that should be rejected when using the upper bound of the signed distance $dist_u(z_t, S) + r_p$ will always be rejected by using the absolute value of the actual signed distance $|dist(z_t, S)|$. This optimization will eliminate the need for the computation of signed distance for a large portion of rejected particles.

3.4 Other Improvements

In order to improve the efficiency of the rejection-sampling particle filter described above, several improvements are explored and discussed in this section.

Adaptive Voxelization

In order to achieve high precision, a good voxel size should be at least smaller than the measurement error. When the workspace is large enough while the measurement error is very small, it will not be feasible to compute a distance transform for the entire workspace while maintaining a sufficiently small voxel size due to the huge memory requirements. Instead, the distance transform for each measurement is generated prior to each update. As the belief of the object's pose is represented as a distribution, the range of the distance transform DF for each measurement z_t is selected so that the probability that the measurement position is within the distance transform

of the object with the pose from $bel(x_t)$ is larger than a predefined constant:

$$1 - \xi < \int_{T(x_t)^{-1} z_t \in DF} bel(x_t) dx \tag{15}$$

where $T(x_{t-1})^{-1} z_t$ is the measurement transformed back to the object frame, and ξ is a selected small constant. The voxel size is adjusted accordingly while keeping the number of voxels fixed. Therefore, the total memory consumption will be constant regardless of the accuracy. When there is a large uncertainty the voxel will become larger to improve the sample speed. When the uncertainty is low the smaller voxel leads to an increased precision. This is desired, since a very high accuracy is not required when the prior uncertainty is large, but it becomes more important when the prior uncertainty is getting smaller.

Adaptive Bandwidth

During sampling process, new particles X_t are sampled directly from the estimated continuous prior belief $bel(x_{t-1})$ since the motion model is static. The continuous prior is estimated by broadening each particle ${}^j x_{t-1}$ from X_{t-1} using a multivariate Gaussian kernel with kernel bandwidth $\mathbf{h}$:

$$bel(x_{t-1}) \sim \sum_{j=1}^{N} \boldsymbol{\mathcal{N}}({}^j x_{t-1},\, \mathbf{h}^2) \tag{16}$$

where $\boldsymbol{\mathcal{N}}$ represents a multivariate Gaussian distribution with the same number of dimensions as the state.

The selection of the Gaussian kernel bandwidth $\mathbf{h}$ is important in order to get reasonable estimate with relatively small number of particles. A larger bandwidth is needed when the variance of $bel(x_t)$ is large for a fixed number of particles; otherwise a smaller bandwidth is preferred to avoid over-smoothing. Silverman's rule of thumb estimator [26] is used to dynamically adjust the bandwidth: $\mathbf{h}(t) = (\frac{4}{(d+2)n})^{1/(d+4)} \hat{\boldsymbol{\sigma}}_t$ due to its simplicity for computation and the high dimensionality of the configuration space, where $\mathbf{h}(t)$ is the $d \times d$ bandwidth matrix, $d = 6$ is the number of dimensions, and $\hat{\boldsymbol{\sigma}}_t = \{\hat{\sigma}_{ij}(t)\}$ is the standard deviation of the sampled states in

configuration space:

$$\hat{\sigma}_{ij}(t) = \sqrt{cov(dim_i(x_t), dim_j(x_t))} \tag{17}$$

$$= \sqrt{E[(dim_i(x_t) - \overline{dim_i}(x_t))(dim_j(x_t) - \overline{dim_j}(x_t))])} \tag{18}$$

where $dim_i(x)$ represents the value of i-th dimension of the state, and $\overline{dim_i}$ is the mean of the i-th dimension of all states at the given time step, which is computed by averaging the state value of the i-th dimension from all particles at time t. Other techniques for bandwidth estimator are discussed in more detail in [25].

Adaptive Sample Size

The naive approach for sample size selection is to use fixed number of particles throughout the localization. However, this suffers from several major disadvantages including:

1. Precision dilemma: it is difficult to achieve both high precision and small number of particles at the same time, since small number of particles can easily lead to a poor estimate during early stages of localization.
2. Poor efficiency: if the prior is already good enough, a small number of particles will suffice.

Rather than using fixed number of particles for each update, sample size is adjusted prior to each sampling step by *Kullback-Leibler divergence* (KL-divergence) which measures the difference between the sample-based maximum likelihood estimate $\hat{p}$ and the true distribution p [8]:

$$D_{KL}(\hat{p}, p) = \sum_x \hat{p}(x) log \frac{\hat{p}(x)}{p(x)}. \tag{19}$$

Suppose that the true distribution is given by a discrete multinomial distribution with k different bins, it can be shown that with probability $1 - \delta$, the KL-divergence is less than or equal to ϵ when the sample size N is given by:

Algorithm 2 Rigid-Body Particle Filter

Input: particles $X_{t-1} = \{{}^j x_{t-1}\}_{j=1}^N$
Input: observation z_t and part mesh S
Output: particles $X_t = \{{}^j x_t\}_{j=1}^N$

$X_t \leftarrow \emptyset$, $k \leftarrow 0$
Initialize $\mathbf{h}^2$ from $Covmat(X_{t-1})$
compute center of distance transform relative in the part frame $c = mean_j(T({}^j x_{t-1})^{-1} z_t)$
build distance transform $D_f(p)$ on mesh S with center c and range DF
$j \leftarrow 1$
while $j \leq N$ **do**
    ${}^j x_t \sim p(x_t|X_{t-1})$ ▷ $p(x_t|X_{t-1}) = \sum_{j=1}^N \boldsymbol{\mathcal{N}}({}^j x_{t-1}, \mathbf{h}^2)$
    $dist \leftarrow D_f(T({}^j x_t)^{-1} z_t)$
    if $dist + r_p > t_d$ **then**
        continue
    end if
    $dist \leftarrow signed(dist)$ ▷ $signed(\cdot)$ converts the unsigned dist to signed one
    if $|dist| \leq t_d$ **then**
        $X_t \leftarrow X_t \cup {}^j x_t$
        if ${}^j x_t$ falls into a unoccupied bin b **then**
            $k \leftarrow k + 1$
            label b occupied
            if $j \geq N_{min}$ **then**
                $N \leftarrow \frac{k-1}{2\epsilon}\left(1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta}\right)^3$
            end if
        end if
        $j \leftarrow j + 1$
    end if
end while

$$N = \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta} \tag{20}$$

$$\approx \frac{k-1}{2\epsilon}\left(1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta}\right)^3 . \tag{21}$$

where $\chi^2_{k-1,1-\delta}$ is the upper $1-\delta$ quantile of χ^2-distribution with $k-1$ degrees of freedom, and $z_{1-\delta}$ is the upper $1-\delta$ quantile of the normal distribution.

Therefore, the number of particles N can be adjusted according to the number of bins with

support k, as shown in Equation 21. Each bin is implemented as a multidimensional grid with fixed size in the 6-dimensional configuration space. During sampling, the number of occupied bins k is counted whenever the newly sampled state falls into an empty bin. The current sample size is counted and each increment of k will result in an update of desired sample size N when the current sample size is larger than a predefined lower bound N_{min} to allow more robustness. When the actual number of particles reaches the desired value or a predefined maximal limit, whichever is smaller, the sampling process finishes. The rigid-body particle filter is described in Algorithm 2.

3.5 Measurement Simulation and Prediction

The rigid-body particle filter takes the input of the measurement from the robot and updates the state belief accordingly, where the measurement is defined from a measurement action and the distance along the action trajectory (Section 3.1). Since performing actual measurement is time-consuming, and a poor measurement will give little new information about the pose of the object, the performance of the particle filter relies heavily on the quality of the measurement. In this section, a pipeline to simulate and predict the effective measurement is summarized. Most of the work in this section is done by Brad Saund in our collaboration on touch localization for rigid body [23].

Measurement Simulation

The first challenge is how to simulate the measurement action with a touch probe. Since a linear trajectory is assumed for the robot from the start position of the measurement action until the contact with the object, a modified ray-casting algorithm is used: the trajectory is modeled by a ray from the start position, and the contact position is modeled by the intersection between the ray and the mesh.

While a ray is infinitely thin, the touch probe's spherical tip has a non-zero diameter, and thus will cast a cylinder rather than a ray. The true value returned by our sensor is the smallest distance until any contact with the part. The measurement cylinder is approximated by discrete uniformly spaced rays on the cylinder exterior, and return the lowest ray-mesh intersection distance.

Measurement error is caused both by inaccurate start positions and orientations due to robot positioning error, as well as inaccuracies in the sensor. In the most extreme cases error may cause a measurement to move from barely hitting an edge to completely missing the part. Thus it is clear neither adding a constant error term, nor a dependent Gaussian error will accurately model the error.

Instead error is modeled in a discrete general method. For each measurement action many simulated measurements are sampled where the initial conditions are perturbed according to an error model for the robot and the measured value is perturbed according to a model of the sensor. The above pipeline for measurement simulation is developed by Brad Saund.

A naïve approach to compute the contact position for a measurement action is to iteratively find the intersection between the ray and every triangle in the mesh and return the intersection with minimum distance. However, the time complexity of this approach is linear to the number of triangles, thus for object whose mesh contains large number of triangles, ray-casting might become a bottleneck of the system due to its repeatedly used for thousands of times during the sampling of measurement actions.

Therefore, this work uses bounding volume hierarchy (BVH) technique to organize triangles in a tree structure such that the triangles at each node are partitioned into disjoint sets. This allows logarithmic-complexity query of the intersection point for each ray-casting.

Measurement Prediction

As part of the localization task, the prediction of an effective measurement action is also expected to be autonomous without the decision from human. For the proposed rigid-body particle filter, the measurement action is determined by the Information Gain approach developed by Brad Saund. This section briefly summarizes the information gain for measurement prediction.

Prior to each measurement, many candidate measurement actions are sampled and evaluated

using the metric of expected Information Gain:

$$IG(X|\mathcal{M}) = H(X) - H(X|\mathcal{M}) \tag{22}$$

$$= \log N - \sum_j p(m_j)\, H(\mathcal{P}|m_j) \tag{23}$$

where $H(\mathcal{P})$ is the entropy of the particles and $H(\mathcal{P}|\mathcal{M})$ is the entropy of the particles conditioned on the measurement action.

The intersection points between the rays and the mesh with the pose represented by each particle are grouped into different bins according to their distance from the start position. Each bin m_j has the same fixed size, thus the probability $p(m_j)$ equals to the number of particles that fall into bin m_j.

A mean pose is computed using all of the particles during action sampling. Candidate measurement actions are chosen such that the direction of each action is normal to the faces of this mean orientation. The measurement action is selected so that it has the highest information gain IG among all of action samples.

An example for measurement prediction is shown in Figure 1. Initially measuring from the front side (or back side, where the robot cannot reach to) has higher probability to actual hit the object and is potentially able to reduce much uncertainty, thus it has higher information gain. On contrary, measurements on the top surface and side surface are more likely to miss. After the first measurement on the front of the object, the robot is now more confident to measure on the top side and thus the second measurement is selected on the top.

3.6 Evaluation

The rigid-body particle filter was evaluated both in simulation and on a physical robot. The software was implemented in ROS using C++ for both computation and visualization.

The touch probe used in the experiment consists of a 0.6mm spherical tip mounted to a 100mm rod attached to a 6-D JR3 force/torque sensor. Candidate measurement actions that are kinematically infeasible are rejected. The measurement action is selected based on its expected information gain.

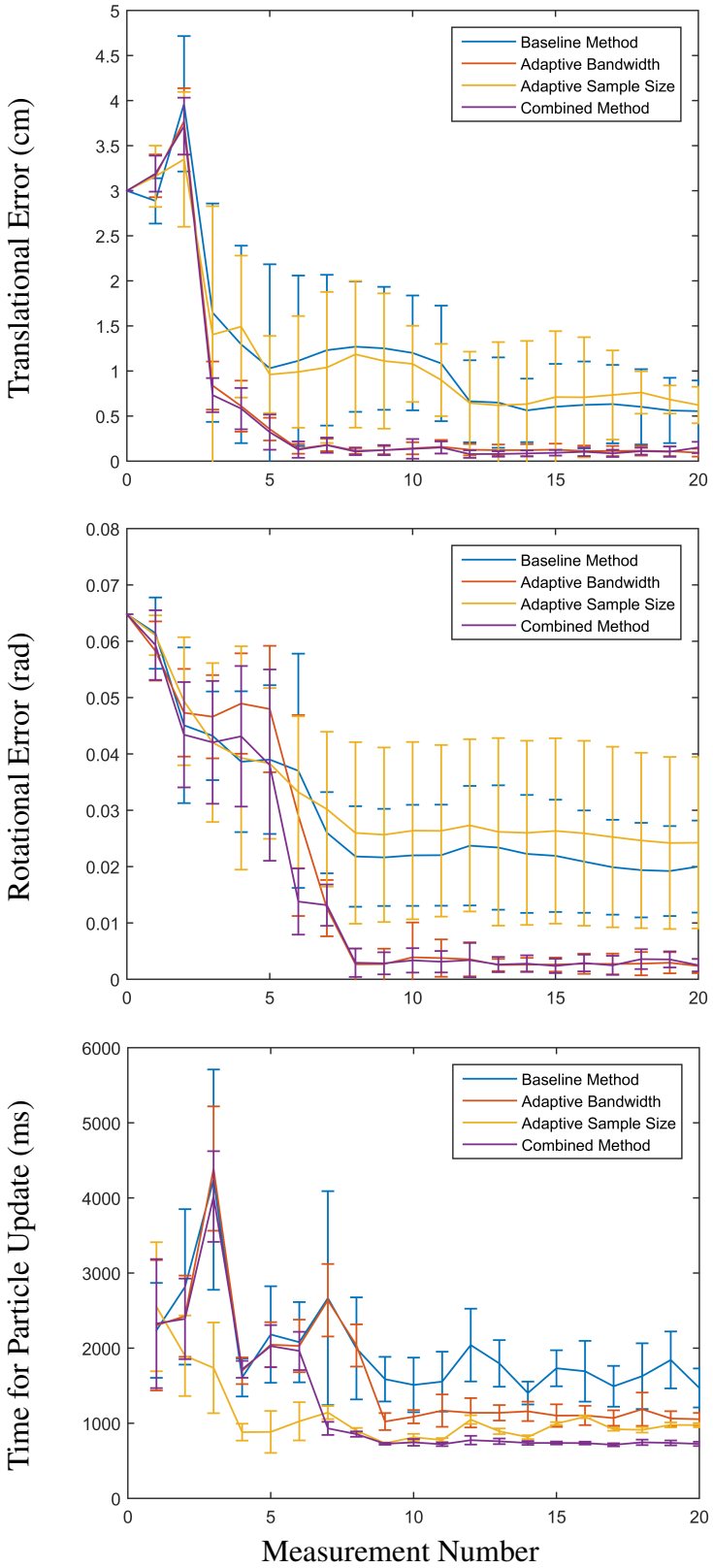


Figure 2: Comparison of the time and accuracy of the particle filter update step when using Adaptive Bandwidth and Adaptive Sample Size

Simulation Results

Figure 2 shows the comparison between the rigid-body particle filters with different improvements that are mentioned in Section 3.4. Translational error and rotational error are defined between the estimated pose of the object and its true state. The running time was recorded for particle update process and excluded the time taken for measurement prediction. The experiment for each method includes 20 trials. The standard deviation of each metric among all trials for each experiment is shown as the error bars on the corresponding curve. The object used for the experiment was a structural component used in aircraft, which was constructed typical of large manufacturing. The simulation used a mesh model for the object with a total of 39444 triangles. In addition, the initial uncertainty for each dimension is defined as a Gaussian distribution. The standard deviation for the Gaussian distribution is 3 centimeter for translational dimension and 0.05 rad for rotational dimension.

The baseline method represents the rigid-body particle filter that only implements adaptive voxelization while using a fixed kernel bandwidth $h = 0.0035$ for each dimension and a fixed number of particles ($N = 500$). Adaptive Bandwidth improves the convergence accuracy and decreases the variance among all trials significantly compared to the baseline method, while Adaptive Sample Size with $N_{min} = 50$ generally leads to poorer convergence with faster speed. The combined method applies all of the improvements above. It can achieve both faster and more accurate convergence with low variance compared to the other methods.

Robot Results

The particle filter was evaluated by Brad Saund using a custom 7-DOF robotic arm designed by TRACLabs with approximately a 1 meter reach, equipped with a touch probe. During experiments, both the base of the arm and the part were rigidly fixed to the ground. The measurement error of the end effector placement was within a few millimeters due to kinematic inaccuracy and deflections.

The object used for robot experiment was the same with the one used in simulation. The same mesh was used to model belief and predict measurement actions, which agreed with the physical part to within 3 millimeters.

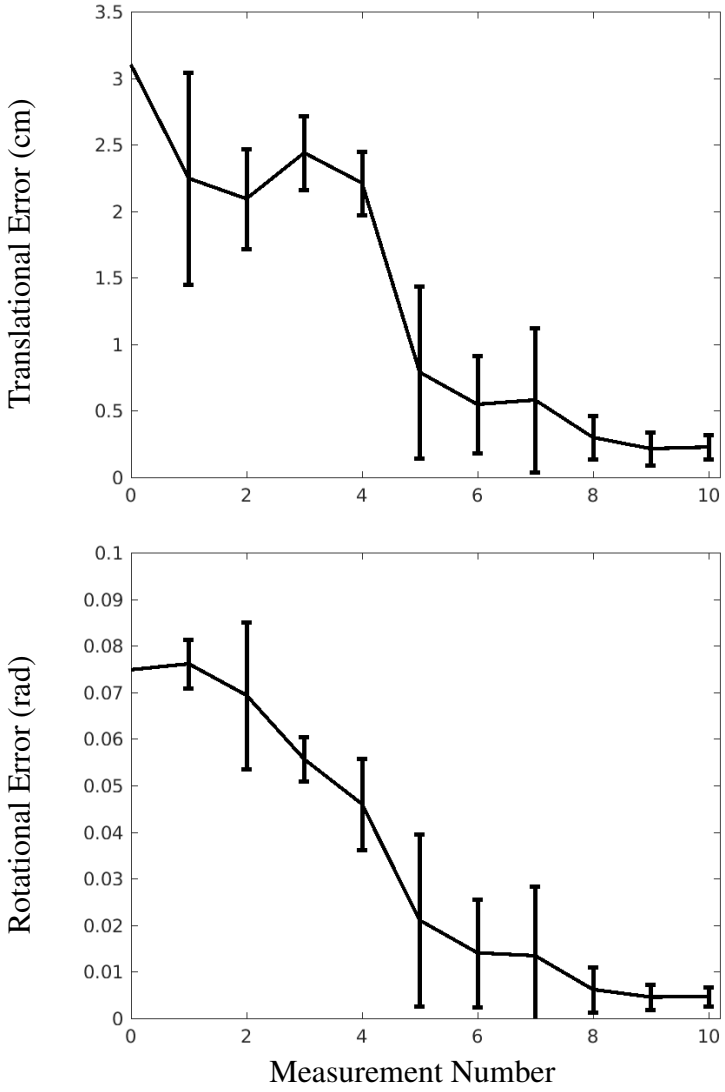


Figure 3: Translational and rotational error during localization on the physical robot

	t_x	t_y	t_z	α	β	γ
Physical Robot	0.03	0.03	0.03	0.05	0.05	0.05
Simulation baseline	0.03	0.03	0.03	0.05	0.05	0.05
Sim w/ large uncertainty	0.06	0.06	0.06	0.1	0.1	0.1
Sim w/ small uncertainty	0.01	0.01	0.01	0.02	0.02	0.02
Sim w/ partial constraint	0.0001	0.0001	0.03	0.0001	0.05	0.05
Sim w/ more accuracy	0.03	0.03	0.03	0.05	0.05	0.05

Table 1: Initial Uncertainty of Experiments

	Measurement Error (cm)	Final Position Error (cm)	Final Angular Error (rad)	Average Computation (s)
Physical Robot	0.05	0.22 $\pm$0.09	0.0047 $\pm$0.002	n/a
Simulation baseline	0.05	0.17 $\pm$0.09	0.0053 $\pm$0.0035	1.8 $\pm$0.2
Sim w/ large uncertainty	0.05	0.44 $\pm$0.14	0.0129 $\pm$0.0085	1.9 $\pm$0.2
Sim w/ small uncertainty	0.05	0.10 $\pm$0.06	0.0024 $\pm$0.0013	1.8 $\pm$0.2
Sim w/ partial constraint	0.05	0.06 $\pm$0.03	0.0008 $\pm$0.0003	1.8 $\pm$0.2
Sim w/ more accuracy	0.01	0.03 $\pm$0.02	0.0007 $\pm$0.0003	2.2 $\pm$0.2

Table 2: Results of Experiments

15 localization trials were performed with identical initial beliefs and true part location. Figure 3 shows the translational and rotational error of the mean of the belief distribution compared with our best measurement of the true state, averaged over all trials [23]. The error bars show one standard deviation.

The results from our physical system agreed with the simulation when simulating the same part orientation, initial uncertainty, and measurement uncertainty. More accurate measurements, larger initial uncertainty, smaller initial uncertainty, and a scenario where initial uncertainty in some directions was very small are then simulated. The initial uncertainty for each dimension is defined as a Gaussian distribution. The standard deviation of the Gaussian distribution for each scenario is presented in Table 1. The summarized results are presented in Table 2 after performing 10 localization measurements, where $\pm$ represents the standard deviation. The computation time was recorded for the entire step and included the process for information gain. We sampled from 500 potential measurement actions and a maximum of 500 particles to model the belief.

Compared to the simulation with baseline initial uncertainty and measurement error, using a more accurate sensor can achieve significantly smaller estimate error for both translational dimensions and rotational dimensions, with a mild increase of average computation time. This result is expected, since for lower measurement error, a smaller voxel size is required, and the tolerance for accepting particles is also smaller, thus only those particles that are more consistent with the measurement are accepted, which leads to a smaller estimate error. In contrary, more samples are required for each update since a larger portion of them will be rejected, thus the

computation time is higher. In addition, the change of the initial uncertainty will also affect the estimate error after 10 measurements accordingly, with approximately the same computation time. However, when constraining three out of a total of six DOFs, the estimation can converge much faster than the baseline scenario.

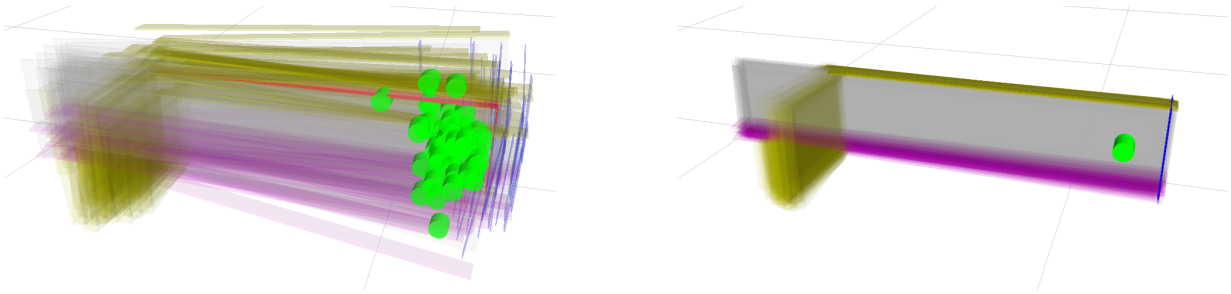
4 Datum Particle Filter

The rigid-body particle filter for object localization presented in Section 3 assumes that the object matches its CAD model exactly. However, this is usually not the case, due to tolerances in the manufacturing and assembly processes. To handle manufacturing deviations, features on parts are not located with respect to the part frame, but with respect to datums, (edges, surfaces, and holes) on the actual "as built" part. Incorporating the notion of datums, and their relationships, adds complexity because the relationship between the datum and the CAD model contains uncertainty. Thus, measuring one section of the assembly provides only uncertain updates to other sections, dependent on the specified tolerances.

The following formulation treats these as semi-rigid parts, where each complete part is composed of rigid *sections*, coupled through a probabilistic distribution of $SE(3)$ transformations connecting the section poses. The datum particle filter is introduced to allow updates on the belief of all sections of a part using the prior distribution of coupling transformations, and a measurement on a single section.

Figure 4 shows a visualization of the initial (4a) and final (4b) beliefs of the poses of the sections of the object. The task is to drill a hole, shown as a green cylinder, at a specific location defined by offsets from other sections. Internal tolerances prevent simply treating the entire system as a rigid body.

In this section, datum particle filter is proposed for localization of semi-rigid parts. The update process of the datum particle filter is extended from the rigid-body particle filter, both of which include rejection sampling and the computation of distance transform.



(a) Initial belief

(b) Belief after localization of goal feature

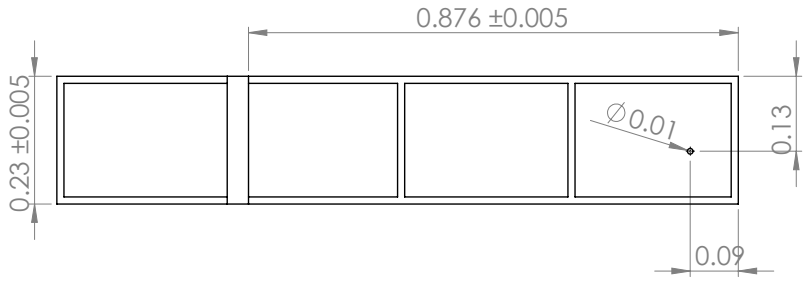
Figure 4: Visualization of the belief of the pose of each part section
(a): The prior belief of the poses. The uncertainty of the goal feature (a hole shown as a green cylinder) is too high to perform the task.
(b): The belief of the poses after performing measurements to localize the goal feature. The pose of the goal feature is now known well enough to perform the task. Although the bottom section (purple) and perpendicular section still have noticeable error, precise localization of these features is not needed.

4.1 Problem Formulation

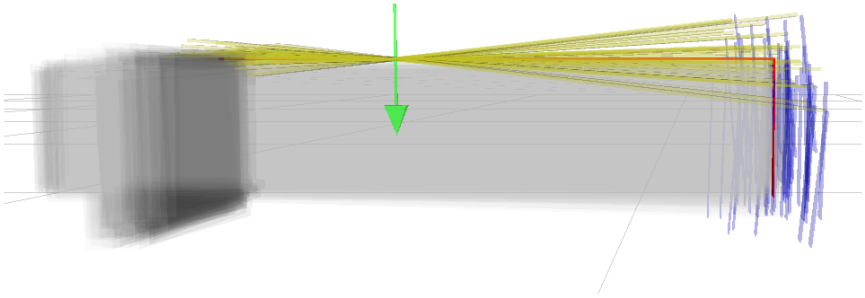
In contrast to the rigid-body particle filter in Section 3, the datum particle filter uses particles X_t to estimate the continuous beliefs of all sections. These particles are updated based on a set of measurements Z_t. Each measurement z_t^s is on a particular section s of the part. The measurement is also assumed to be made by point-based sensors, e.g. a touch probe.

In the following algorithm, the section that a measurement contacts is known. This assumption is reasonable if the uncertainty of the prior belief is small compared to the physical size of each section, and measurement are not chosen on the boundary between sections. If this assumption does not hold, localization can be performed for the whole object using the methods of Section 3 to get better estimate, before considering the object as a combination of coupled sections.

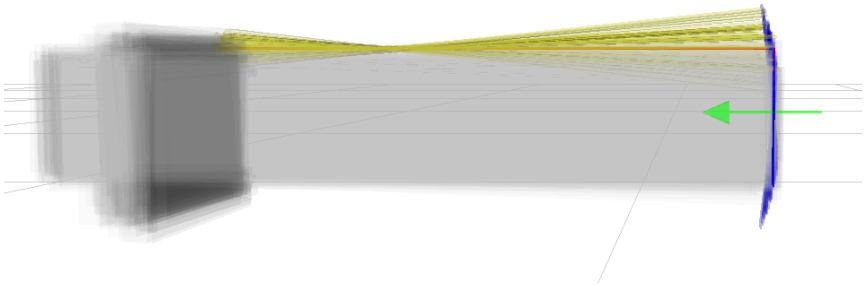
Instead of assuming a single CAD model for the entire part is given, individual CAD model S_s for each rigid section s is provided. Each CAD model is assumed to match its corresponding section.



(a) CAD schematic drawing



(b) Update from a measurement on the top datum



(c) Update from a following measurement on the right datum

Figure 5: Visualization of datum particle filter
(a): Side view of the CAD drawing with dimensions (simplified for clarity, plotted by Brad Saund in our joint paper [3]). This drawing indicates the nominal distance between the top and bottom edge is 0.23m, with a symmetric tolerance of 5mm. This drawing also defines a hole with a 1cm diameter, and the top edge as its vertical datum and side edge as its horizontal datum.
(b): The beliefs of the top (yellow) and right (blue) sections of the part are shown. The ground true location for each section is shown in red. The measurement (green arrow) on the top section partially localized the top section. For clarity in the image, the belief of the other sections are not shown, and only 50 particles are shown.
(c): A following measurement (green arrow) on the right section further localizes the part. This measurement provides information on the right section directly, and the top section indirectly.

Geometric Relationships

Geometric relationships between two part sections are defined as $SE(3)$ transformation in the configuration space relatively in a reference frame, e.g. T_k^s is the $SE(3)$ transformation from the pose of section k to the pose of section s. For a rigid object, the transformation between every pair of sections in this object is fixed and known. Let x_t^k be a candidate pose of section k from X_t. Given a pose x_t^k, one can obtain the corresponding pose x_t^s of any section s simply by $x_t^s = T_k^s \times x_t^k$.

For a semi-rigid object, the existence of the tolerance introduces uncertainty to these relationships. The belief of the transformation from section k to section s is modeled as a conditional distribution $p(T_k^s|X_t^k)$ given the belief of the pose of section k. For instance, if the pose of section s is precisely localized, while the pose of section k is poorly known, the transformation T_k^s is then determined solely by the belief $bel(x_t^k)$ of section k: for any different ${}^j x_t^k$, there is a single different transformation T_k^s that is consistent to the pose of section s.

Datum Representation

The problem is to precisely localize some feature which cannot be measured *directly* (e.g. a location to drill a hole) with respect to given datums (other sections). To localize the goal feature, certain datums must be localized in certain dimensions. For instance, Figure 5a shows a hole feature referenced to the top and right edges datums of our part. In this example, it is necessary to localize the top edge's vertical position and orientation, but not its in-page or horizontal position. Similarly, the right edge only must be localized horizontally.

Two related but different datum representations have been developed for this task: independent-state representation and full-state representation.

Independent-state representation: The independent-state representation models the system using separate coupled particle filters, each of which updates the belief of an individual section, under the assumption that the belief of the transformation between each section is fixed and independent to its pose:

$$p(T_k^s) \approx p(T_k^s|X_t^k) \tag{24}$$

where $X_t^k = \{{}^j x_t^k\}_j$ represents candidate poses from particle filter for section k. $p(T_k^s)$ is defined explicitly using either parametric probability (e.g. Gaussian or uniform distribution) or a set of discrete candidate values.

For independent-state representation, a measurement on a single section updates all individual particle filters related through the defined transformation distribution. The update to the belief $bel(x^s)$ given a measurement performed on section s itself is identical to the particle filter update for the rigid object. For a section k that references section s ($s \neq k$), each new sample x_t^k is drawn from the prior of its corresponding particle filter k. The state x_t^k is then transformed from section k to section s, $\tilde{x}_t^s$:

$$\tilde{x}_t^s = {}^i T_k^s \times x_t^k \tag{25}$$

where ${}^i T_k^s \sim p(T_k^s)$ is a sampled transformation from the distribution $p(T_k^s)$. $\tilde{x}_t^s$ is then used instead of x_t^k for the rejection sampling: if $\tilde{x}_t^s$ is accepted, its corresponding x_t^k is added to X_t^k.

This work only explored the independent-state representation with the assumption that each datum is composed by simple primitives including planes, edges and holes, following the inspiration from engineering drawing without assuming that each section is precisely manufactured. This assumption can improve the update speed since the binary measurement model in rejection sampling can be evaluated analytically using basic geometry, but also greatly limits its application to more complex object.

An alternative is to use individual CAD model for each section. The independent-state particle filter using CAD model for individual section is proposed separately by Brad Saund and is discussed in [3].

Full-state representation: The full-state representation relaxes the independence assumption by maintaining a single particle filter that stores the full joint distribution between the coupled sections. The particle filter for the full-state datum representation is called full-state particle filter and is described in more detail in Section 4.2. The constraint on the datum type is also removed by using CAD model for each individual section.

The independent-state particle filter using primitive datum type was explored initially. However, in some situations the independence assumption between the transformations and the poses

may lead to inefficient update. For instance, if the belief of the transformation $p(T_k^s|{}^j x^k)$ between two sections s and k given a specific pose ${}^j x^k$ for section k is much more converged (or at least along some dimensions) than $p(T_k^s)$, it will be more efficient to sample a consistent $\tilde{x}^s$ from ${}^j x^k$ using $p(T_k^s|{}^j x^k)$ rather than $p(T_k^s)$. In a simple 1D example, suppose that there are only two candidate pose combinations: $({}^1 x^k = 0, {}^1 x^s = 2)$ and $({}^2 x^k = 1, {}^2 x^s = 3)$. With the independence assumption, there are three possible values (${}^1 x^s - {}^1 x^k = {}^2 x^s - {}^2 x^k = 2$, ${}^1 x^s - {}^2 x^k = 1$, and ${}^2 x^s - {}^1 x^k = 3$) for the transformation T_k^s with a probability of $1/2$, $1/4$ and $1/4$ respectively. If there is a measurement on s at location 2, one need to sample from these three different transformations to compute either $\tilde{x}^s = T_k^s \times {}^1 x^k$ or $\tilde{x}^s = T_k^s \times {}^2 x^k$. When sampling the transformation for 4 times, averagely only 3 out of 8 samples will result in a consistent sample $\tilde{x}^s = 2$, but only two of them are actual correct pose combination $({}^1 x^k = 0, {}^1 x^s = 2)$. In contrary, without the independence assumption, the transformation from section k to s given the pose of section k is a single value 2, whenever the particle ${}^1 x^k = 0$ is selected, a consistent sample $\tilde{x}^s$ will be obtained, thus $1/2$ samples will result in the correct combination.

Due to this issue, full-state particle filter was explored later to model the conditional belief of the transformation in order to relax the independence assumption. This thesis will focus on the full-state particle filter. Compared to its counterpart, it requires only a single high-dimensional particle filter to be updated based on each measurement.

Throughout the rest of this thesis the following notation will be used. Each particle is a joint configuration ${}^j \mathbf{x} = \{{}^j x^k\}_{k=1}^K$ of all K sections in the part, where ${}^j x^k$ represents the state corresponding to the k-th section described by the j-th particle. $X^k = \{{}^j x^k\}_{j=1}^N$ represents the set of candidate states of section k extracted from all full-state particles. The omission of k indicates all necessary particles to represent the belief of the part: $X = \{{}^j \mathbf{x}\}_{j=1}^N$.

4.2 Full-State Particle Filter

The full-state particle filter maintains the full joint probability distribution between sections using a single set of full-state particles $X_t = \{{}^j \mathbf{x}_t\}$, each of which concatenates the state of each section. Thus, each particle represents a candidate combination of the poses of all sections (see Figure 6). The initial particles can be drawn from the prior joint distribution $bel(\mathbf{x}_0)$ using

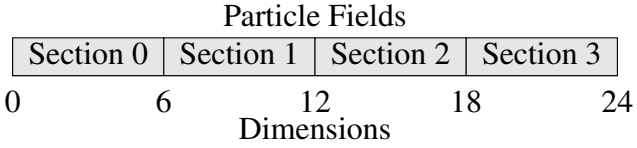


Figure 6: A 24-dimensional particle for part with 4 sections

rejection sampling if it is defined directly in parametric form or sampled from the previous set of particles obtained from prior "coarse" localization. Otherwise, one can iteratively generate a full-state sample by first sampling a state for a certain section from the prior belief, and then sampling other states based on the corresponding beliefs of transformation.

One can model the conditional beliefs of transformation using a graph. Figure 7 shows the graph of one possible definition for a part with 7 sections. Each node represents the state of a section, and each edge models the corresponding transformation belief. The edge is undirected since the transformation can be defined along either direction. For instance, the edge from A to B represents $p(T_A^B|X_t^A)$, while the edge from B to A represents $p(T_B^A|X_t^B)$. For geometric relationships, one can assume that the graph contains no cycle, since a cycle will over-define the pose of any section in that cycle (note that a transformation defines all 6DOFs between two poses). Assume that the graph is connected, then one can iteratively generate a full-state sample from this $bel(\mathbf{x}_0)$ following the direction of the path between every two sections. As for Figure 7, one possible order to sample an initial full-state particle ${}^j\mathbf{x}$ is: ${}^jx^A \sim p(x^A) \rightarrow {}^jx^B \sim p(T_A^B|{}^jx^A) \rightarrow {}^jx^E \sim p(T_B^E|{}^jx^B) \rightarrow {}^jx^C \sim p(T_A^C|{}^jx^A) \rightarrow {}^jx^F \sim p(T_C^F|{}^jx^C) \rightarrow {}^jx^G \sim p(T_C^G|{}^jx^C) \rightarrow {}^jx^D \sim p(T_A^D|{}^jx^A)$, where $j = 1$. This process is repeated until a total of N initial full-state samples are collected. However, when there is over-definition, this simplification requires the beliefs of different sections consistent with each other. For instance, when the transformation between section A and B and their individual state belief are all known, the computed state belief $\hat{bel}(x_t^B)$ of section B from belief $bel(x_t^A)$ of A and the transformation belief $p(T_A^B|X_t^A)$ should be consistent with the prior belief $bel(x_t^B)$ at the same step.

Similar to rigid-body particle filter, at each update step t, the continuous prior $bel(\mathbf{x}_{t-1}) = p(\mathbf{x}_{t-1}|Z_{t-1})$ is estimated by broadening particles X_{t-1} using multivariate Gaussian kernels in

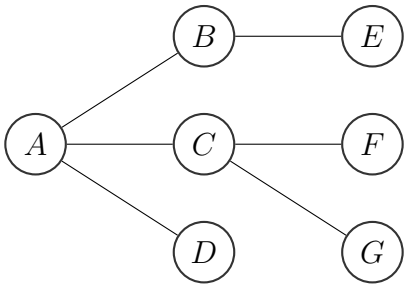


Figure 7: A graph to model the beliefs of transformation

Algorithm 3 Full-State Particle Filter

Input: number of particles N and number of sections K
Input: particles $X_{t-1} = \{{}^j\mathbf{x}_{t-1}\}_{j=1}^N$
Input: observation z_t^s on section s and meshes $S = \{S_k\}_{k=1}^K$
Output: particles $X_t = \{{}^j\mathbf{x}_t\}_{j=1}^N$

$X_t \leftarrow \emptyset$
build distance transform $D_{f^s}(p)$ on mesh S_s
$j \leftarrow 1$
while $j \leq N$ **do**
    ${}^j\mathbf{x}_t \sim p(\mathbf{x}_t|X_{t-1})$ ▷ ${}^j\mathbf{x}_t = \{{}^jx_t^k\}_{k=1}^K$
    $dist \leftarrow signed(D_f(T({}^jx_t^s)^{-1}z_t^s))$
    if $|dist| \leq t_d$ **then**
        $X_t \leftarrow X_t \cup {}^j\mathbf{x}_t$
        $j \leftarrow j + 1$
    end if
end while

the full $6n$-dimensional configuration space as in Equation 16, with the kernel covariance proportional to the covariance of the sampled states in the full-state configuration space. This is achieved by applying kernel density estimation techniques in the same way as the rigid-body particle filter (Section 3.4).

Let the current measurement z_t^k be applied on section k, and x_{t-1}^k is the portion of $\mathbf{x}_{t-1}$ representing the pose of section k. The distance transform $D_{f^k}(p)$ is constructed for section k using the mesh S_k for this particular section. During rejection sampling, new full-state samples are drawn from the estimated $bel(\mathbf{x}_{t-1})$ directly. The probability of accepting a sample ${}^j\mathbf{x}_t$ is based on the measurement z_t^k, which is computed using the same binary measurement model $p(z_t^k|{}^jx_t^k)$ introduced in Section 3.2 by extracting the 6-dimensional pose ${}^jx_t^k$ for section k from the full-state sample ${}^j\mathbf{x}_t$. Equation 13 and 14 are then used for the computation of unsigned

distance and signed distance to evaluate the particle, except that the transformation $T({}^j x_t^k)$ from the extracted ${}^j x_t^k$ is used to transform the measurement. If the absolute value of signed distance $dist(z_t^k, S_k({}^j x_t^k))$ is sufficiently large, this full-state sample ${}^j\mathbf{x}_t$ is rejected, otherwise accepted. The algorithm of full-state particle filter is presented in Algorithm 3). The full-state particle filter also adopts all of the improvements introduced for the rigid-body particle filter. For simplicity, the pseudo-code does not include these improvements.

Although it seems at first that the high dimensional state space will require a prohibitively huge number of particles to approximate the true distribution, in practice, we have not found this to be the case. Firstly, this is due to the small internal tolerances compared to the uncertainty of the pose of the object as a whole, so particles tend to cluster in a small subset of the full state space. Note that if there were no uncertainty in the transformation between sections, then the system reduces to a single 6 dimensional state space. In addition, the Gaussian kernel applied to the particles is capable of creating broad beliefs in specific dimensions [23][3], thus this particle filter is able to model the mixture of precisely localized, and poorly localized dimensions. Finally, the sample acceptance probability, $p(z_t|\mathbf{x}_t)$, depends only on the 6 dimensional subspace corresponding to the pose of the section being touched, and although only a thin manifold is accepted with high probability, the particle starvation challenge is no worse than previously addressed for rigid-body particle filter in Section 3.

During sampling, the beliefs of transformation between sections may change as each full-state particle is broadened with a Gaussian kernel. Since the goal feature cannot be measured directly, depending on the desired accuracy, one may need to maintain a separate set of particles for the goal feature and manually compute each state of the goal feature using its corresponding full-state particle to avoid loss of precision rather than concatenate it with other sections during rejection sampling. This can be achieved by first extracting the combination of poses of its referenced sections from each full-state particle and then determining its state according to the extracted poses. Another alternative is to apply a separate sample and update process for the particles of the goal feature in the same manner as in the independent-state particle filter after the update of other sections using full-state particle filter. Since the uncertainties between the state of the goal feature and the state of each referenced section along the dimensions that we are

interested in are very small, the independence assumption is no longer an issue here.

4.3 Measurement Simulation and Prediction

The measurement is simulated in a similar way as the rigid-body particle filter (Section 3.5) except that the mesh for individual section is used instead for the ray-casting. This thesis explores the measurement prediction approach for full-state particle filter, which is established upon the measurement prediction framework for rigid-body localization developed by Brad Saund [23].

Let $X^s = \{{}^j x^s\}_{j=1}^N$ represents all candidate poses of the section s extracted from full-state particles X. The information gain from a measurement action $\mathcal{M}$ is defined as:

$$IG(X^s|\mathcal{M}) = H(X^s) - H(X^s|\mathcal{M}) \tag{26}$$

where $H(X^s)$ is the entropy of X^s and $H(X^s|\mathcal{M})$ is the entropy of the particles conditioned on the measurement action.

Similarly, the entropy of a discrete distribution of states depends only on the probabilities of each state occurring: $H(X^s) = -\sum_i w_i \log w_i$, where w_i is the weight of particle i. For full-state particle filter, uniform weight is used. In this case, $H(X^s) = \log N$.

Given a measurement action $\mathcal{M}$ on section s, a set of discrete measurement values $\{m_{i,j}\}_j$ can be computed for each candidate pose ${}^i x^s$ from the measurement simulation (Section 3.5). $H(X^s|\mathcal{M})$ is calculated by dividing the measurement values $\{m_{i,j}\}_{i,j}$ of all candidate poses X^s into discrete bins with fixed size, b_k. The conditional entropy $H(X^s|\mathcal{M})$ of this measurement action then becomes:

$$H(X^s|\mathcal{M}) = \sum_k p(b_k)\ H(X^s|b_k) \tag{27}$$

where $p(b_k)$ is the prior probability that this measurement will fall into bin b_k and $H(X|b_k)$ is the entropy of the particles within bin b_k. Let W_k of bin b_k is the cumulative weight of particles

that fall within it:

$$W_k = \sum_{i,j} \mathbb{1}(m_{i,j} \in b_k) \cdot w_i \tag{28}$$

where $\mathbb{1}(\cdot)$ equals to one when the condition is true and zero otherwise, then the likelihood of a bin becomes :

$$p(b_k) = \frac{W_k}{\sum_k W_k} = \frac{W_k}{\sum_{i,j} w_i} \tag{29}$$

The probability of a specific particle given a bin b_k is:

$$p({}^i x^s | b_k) = \frac{\sum_j \mathbb{1}(m_{i,j} \in b_k) \cdot w_i}{W_k} \tag{30}$$

Then the entropy of the bin can be calculated by:

$$H(X^s | b_k) = -\sum_i p({}^i x^s | b_k) \log(p({}^i x^s | b_k)) \tag{31}$$

Equation 26 provides a way to extend the Equation 22 for rigid body to semi-rigid body with multiple coupled sections: the expected information gain from measuring on section s at given location when updating belief of section s can be computed from its corresponding candidate poses X^s.

If the sections that the goal features are on can be measured directly, to update their beliefs one can sample many measurement actions randomly on those sections and compute the expected information gain for the individual goal section that is selected by each sampled measurement action. The measurement action with the highest information gain is chosen, and its corresponding section will be updated based on the selected action. Although different from the assumption of the datum particle filter where the goal feature cannot be measured directly, datum particle filter can still be useful here and is more efficient than separately measuring each goal feature in turn regardless of the tolerance information, since the datum particle filter allows the "joint" update of the other sections based on the measurement on individual section.

A more common and challenging task is that when a goal feature cannot be measured directly. In this case, one needs to maximize the expected information gain of the goal feature based on measurements on other sections. The strategy mentioned above will only try to select the measurement that can achieve the highest information gain for the section being measured. However, localizing some of these sections may not be able to provide much information about the goal feature and thus should be avoided, even if it has higher information gain. An example can be found in Figure 4. Localizing the top section and side section is sufficient to localize the goal feature.

Alternatively, let X^G represents the candidate poses of the goal section and X^S be the candidate poses of a section of the same part that can be measured directly. The distribution of the transformation from section G to section S due to tolerance is $p(T_G^S|X^G)$. A temporary set of poses $\tilde{X^S} = \{\tilde{x^S}\}$ can be created by applying the transformations sampled from T_G^S to each state ${}^j x^G$ from the original X^G:

$$T \sim p(T_G^S|{}^j x^G) \tag{32}$$

$$\tilde{x^S} \leftarrow T \times {}^j x^G \tag{33}$$

For each particle from X^G, multiple transformations can be sampled, and thus $\tilde{X^S}$ may contain larger number of particles than X^G. $\tilde{X^S}$ can then be used to evaluate the information gain based on Equation 26.

Since the full-state particle filter does not explicitly maintain the transformation between two sections of a part, but rather each full-state particle, ${}^j\mathbf{x}$, represents the full $6 \times n$ state. Ideally $p(T_G^S|X^G)$ can be estimated by directly computing the transformation between the state of section S and the corresponding state of section G in the same full-state particle. This will give us the transformation conditioned on that specific state of section G. However, this is not practical since X^G is only a set of small number of discrete particles that are sampled from the continuous belief. When there only a small number of particles, there may only be a single value T_G^S for each identical X^G.

Instead, the prior $p(T_G^S)$ is used to estimate the conditional posterior $p(T_G^S|X^G)$. $p(T_G^S)$ is

created as a list of potential transformations by calculating T_G^S for the state ${}^j x^S$ of section S and the corresponding state ${}^j x^G$ for section G in every full-state particle, where $1 \leq j \leq N$. The list $p(T_G^S)$ is sampled uniformly when used to transform ${}^j x^G$ to $\tilde{x^S}$.

Note that this estimation seems to contradict to what full-state particle is trying to solve, where the dependence between the transformation $p(T_G^S|X^G)$ and X^G is maintained to achieve high precision. However, as the expected information gain is only used to evaluate the quality of a candidate measurement action, it will not affect the precision of the full-state particle filter directly.

For larger number of particles, there are two alternative approaches which relax this assumption in different ways, both of which involve grouping particles using discrete 6-dimensional bins b_j^G by their corresponding state values of the goal feature, where b_j^G is the bin that ${}^j x^G$ falls within. This idea is similar to that for adaptive sample size by using bins to discretize the continuous distribution, and the size of the bin b^G can also be adjusted according to the covariance of the states for goal feature similar to technique for adaptive bandwidth.

The first approach estimates the conditional probability $p(T_G^S|{}^j x^G)$ by $p(T_G^S|b_j^G)$. For each ${}^j x^G$, the bin b_j^G is determined, thus the transformation sampled from $p(T_G^S|b_j^G)$ is used to transform ${}^j x^G$ to $\tilde{x^S}$ as in Equation 33. Rather than explicitly estimates $p(T_G^S|{}^j x^G)$, the second approach combines particles that produce similar configuration for the goal feature, i.e. ${}^j x^G$ falls within the same bin b_j^G. To calculate $H(X^G|\mathcal{M})$ using particles X^S, measurement actions are used to sort the particles into bins b for measurement values. Then, particles that produce sufficiently similar goal feature configurations (i.e. fall within the same bin b_j^G) are treated as identical particles when computing entropy, by combining these into groups L. The updated equations for grouped particles are:

$$p({}^L x|b_k) = \frac{\sum_{i,j} \mathbb{1}(m_{i,j} \in b_k)\mathbb{1}(i \in L) \cdot w_i}{W_k} \tag{34}$$

$$H(X|b_k) = -\sum_L p({}^L x|b_k) \log(p({}^L x|b_k)) \tag{35}$$

In practice, these two approaches did not work as expected, partially because that the small number of particles used in the experiment cannot maintain sufficient density for this descretiza-

tion of goal feature configuration to produce meaningful group sizes. Instead, the original approach that samples the transformations directly from $p(T_G^S)$ is used for the measurement prediction.

4.4 Evaluation

The full-state particle filter was evaluated in simulation. The software was implemented in ROS using C++. This experiment simulated a specific task which is common in manufacturing: localizing a target location, defined by datums, to drill a hole on an object.

The datum particle filter was simulated on the same object as used in the rigid-body particle filter, with adaptations made to allow internal degrees of freedom. The object is composed of 6 precisely manufactured sections, and tolerance between sections was determined by engineering drawings (for precisely defined features), and our assumptions (for loosely defined relationships).

The target hole is localized by measuring its referenced datums. Specifically, the pose of the hole feature (green cylinder) in Figure 4 is defined by an offset distance from the top and right sections shown in Figure 5a, and the axis of the hole is orthogonal to the front plane. The radius of the hole is 1cm with a depth of 3.5cm. The hole does not exist yet, and thus cannot be measured directly. In order to localize the hole location precisely without direct measurement on the hole, the transformations between the target location and its defining datum sections is assumed to have very small uncertainty along some dimensions, e.g. the vertical distance between the center of the hole and the top plane. The particles of the hole are computed and maintained separately.

At each step, the measurement is simulated using the same ray-casting algorithm for rigid-body particle filter. Each measurement is assumed to have a small measurement error (0.02cm). Candidate measurement actions are sampled in the workspace. The information gain for each measurement is calculated based on Equation 26. In this experiment, for each measurement performed 500 candidate actions are sampled with non-zero information gain. Only the measurement action with the largest expected information gain is "performed" in simulation and used to update the beliefs.

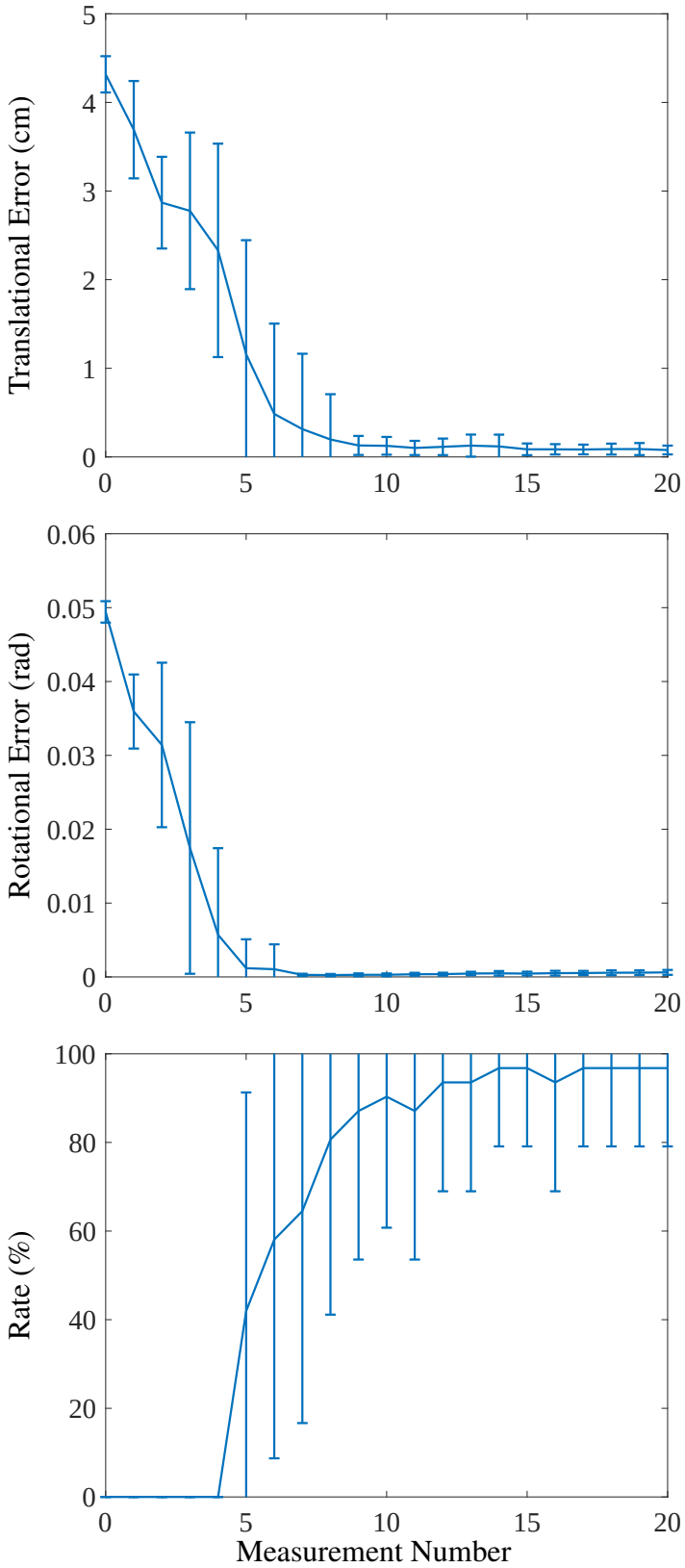


Figure 8: Simulation result for full-state particle filter

Results

A total of 20 measurements are simulated during each trial. The number of full-state particles are constrained between a minimum of 50 and a maximum of 800. The simulation uses a total of 6 mesh models, one for each section. After each update, the average estimated pose of the hole is computed by averaging the hole poses produced from all particles.

Figure 8 shows the performance of the full-state particle filter. Translational error and rotational error are defined between the estimated pose of the hole and its true state. The rate in the third figure is defined as the probability among all 30 trials that a cylinder with radius equal to 90% of the original radius of the hole positioned in the ground true state will not collide with the hole positioned in the average estimated state after each measurement. The probability is defined as $rate = 1 - \#\, collisions/\#\, trials$, where $\#\, collisions \in [0, \#\, trials]$ is the times of trials where a collision is happen. The results show that the errors of the estimated pose decreased rapidly after each new measurement is applied on the system. The datum particle filter can converge to less than 1 mm for translational error and 0.0004 rad for rotational error after around 10 simulated measurements. The probability that there is no collision between the estimated hole and the imagined fitting cylinder also increases rapidly after the first 5 measurements. At the end of the 20 measurements, there is still one trial among a total of 30 trials that results in a collision. However, considering the difference of the radius between the hole and the fitting cylinder is only 0.1cm and the depth is 3.5cm, this is a reasonably good result.

5 Conclusion and Future Work

This thesis introduced the *rigid-body particle filter* and *datum particle filter* for real-time localization with touch probe. The rigid-body particle filter uses rejection sampling and a precomputed distance transform, which overcomes the particle starvation problem and can achieve both high precision and real time localization for rigid-body object. The full-state datum particle filter is built upon the rigid-body particle filter, but takes the engineering tolerance for manufactured part into consideration. It provides a method to localize a task location defined by datums on an object with internal degrees of freedom.

For future work, we would like to implement the full-state particle filter on a physical system for further evaluation. In addition, we would like to combine the full-state particle filter presented in this thesis and the independent-state particle filter [3] together: use individual full-state particle filter for each group of sections with strictly-defined internal relationships to minimize the loss of information, while using a independent-state particle filter to maintain different full-state filters for different groups of sections and other loosely-defined sections, each of which is represented by a single full-state particle filter, for better scalability.

Bibliography

[1] Tomas Akenine-Möllser. Fast 3D Triangle-Box overlap testing. *Journal of Graphics Tools*, 6(1):29–33, 2001. ISSN 1086-7651. doi: 10.1080/10867651.2001.10487535. 3.3

[2] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for RGB-D based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013. 2.2

[3] Shiyuan Chen, Brad Saund, and Reid Simmons. The Datum Particle Filter: Localization for objects with coupled geometric datums [PREPRINT]. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, submitted. (document), 3.2, 5, 4.1, 4.2, 5

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.177. URL `http://dx.doi.org/10.1109/CVPR.2005.177`. 2.2

[5] Andrea Del Prete, Francesco Nori, Giorgio Metta, and Lorenzo Natale. Control of contact forces: The role of tactile feedback for contact localization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4048–4053. IEEE, 2012. 2.2

[6] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Cornell Computing and Information Science Technical Report TR20041963*, 4:1–15, 2004. ISSN 1557-2862. doi: 10.4086/toc.2012.v008a019. URL `http://ecommons.library.cornell.edu/handle/1813/5663`. 3.3, 3.3

[7] Jeremy A Fishel and Gerald E Loeb. Sensing tactile microvibrations with the BioTac-Comparison with human sensitivity. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pages 1122–1127. IEEE, 2012. 2.2

[8] Dieter Fox. Adapting the sample size in particle filters through KLD Sampling. *Intl Jour of Robotics Research*, 22(12):985–1004, 2003. 3.4

[9] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998. 2.2

[10] Frank L Hammond, Rebecca K Kramer, Qian Wan, Robert D Howe, and Robert J Wood. Soft tactile sensor arrays for micromanipulation. In *Intelligent Robots and Systems (IROS),*

2012 IEEE/RSJ International Conference on, pages 25–32. IEEE, 2012. 2.2

[11] Robert M Haralick, Hyonam Joo, Chung-Nan Lee, Xinhua Zhuang, Vinay G Vaidya, and Man Bae Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426–1446, 1989. 2.2

[12] Paul Hebert, Thomas Howard, Nicolas Hudson, Jeremy Ma, and Joel W. Burdick. The next best touch for model-based localization. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 99–106, 2013. ISSN 10504729. doi: 10.1109/ICRA.2013.6630562. 2.2

[13] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012. URL `http://dx.doi.org/10.1007/978-3-642-37331-2_42`. 2.2

[14] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997. 1

[15] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(3):95–108, 1961. 1

[16] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 1

[17] Michael C. Koval, Mehmet R. Dogar, Nancy S. Pollard, and Siddhartha S. Srinivasa. Pose estimation for contact manipulation with manifold particle filters. *IEEE International Conference on Intelligent Robots and Systems*, (Section 3):4541–4548, 2013. ISSN 21530858. doi: 10.1109/IROS.2013.6697009. 1, 2, 2.2

[18] Michael C Koval, Nancy S Pollard, and Siddhartha S Srinivasa. Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty. *The International Journal of Robotics Research*, 35(1-3):244–264, 2016. 2.2

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 2.2

[20] Tomas Lozano-Perez. Spatial planning: A configuration space approach. In *Autonomous robot vehicles*, pages 259–271. Springer, 1990. 2

[21] Anna Petrovskaya and Oussama Khatib. Global localization of objects via touch. *IEEE Transactions on Robotics*, 27(3):569–585, 2011. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5784199`. 2.2

[22] Mabaran Rajaraman, Michael Dawson-Haggerty, Kenji Shimada, and David Bourne. Automated workpiece localization for robotic welding. In *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, pages 681–686. IEEE, 2013. 2.2

[23] Brad Saund, Shiyuan Chen, and Reid Simmons. Touch based localization of parts for high

precision manufacturing [PREPRINT]. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, in press. 3.2, 3.5, 3.6, 4.2, 4.3

[24] Max Schwarz, Hannes Schulz, and Sven Behnke. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1329–1335. IEEE, 2015. 2.2

[25] Simon J Sheather. Density estimation. *Statistical Science*, 19(4):588–597, 2004. 3.4

[26] Bernard W Silverman. Density estimation for statistics and data analysis. *Biometrical Journal*, 30(7), 1986. ISSN 03233847. doi: 10.1002/bimj.4710300745. 3.4

[27] Sebastian Thrun, D Fox, and W Burgard. Monte Carlo localization with mixture proposal distribution. *Proceedings of the National Conference on*, pages 859–865, 2000. URL `http://www.aaai.org/Papers/AAAI/2000/AAAI00-132.pdf`. 2.2

[28] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005. 1, 2.1, 2.1

[29] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3D pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3109–3118, 2015. 2.2