Touch Based Localization of Parts for High Precision Manufacturing

Brad Saund[1], Shiyuan Chen[2], Reid Simmons[3]

***Abstract*— Performing detailed work on objects requires precise localization. Currently humans aid machines in localization either by direct operation, or implicitly by designing a sequence of actions a robot follows. Our approach to automate localization is to reason over many potential actions, perform the best information gathering action, and then use the measurement obtained to update a non-Gaussian belief. We propose a method for autonomous localization of objects with initial 6DOF uncertainty capable of reasoning about and performing measurements with low uncertainty and arbitrary error models. Surprisingly, common methods capable of modeling arbitrary belief distributions perform poorly as measurement uncertainty decreases, so we modify a *particle filter* to handle these accurate measurements produced by tactile or laser sensors. We then show how the expected *information gain* of the proposed measurement can be calculated efficiently from these particles. We present experiments, both in simulation and on hardware, that show our method is both fast and accurate.**

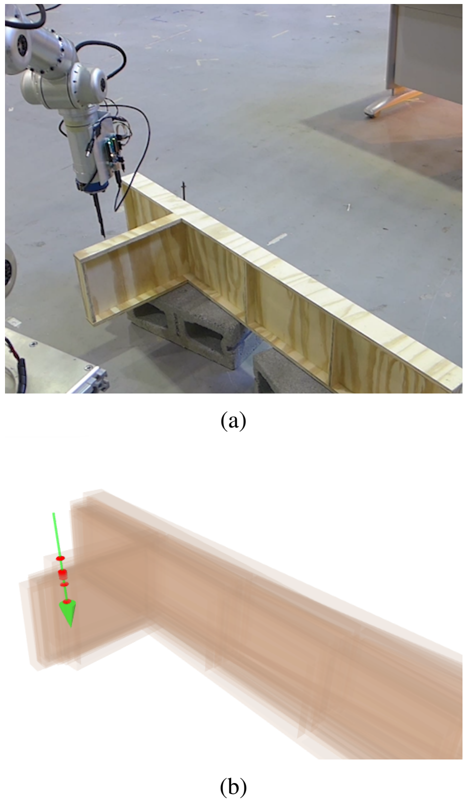
I. Introduction

In automated manufacturing it is critical that robots localize to the parts they are working on before performing their operations. With sufficiently consistent parts and a high rate of production this localization can be accomplished though external jigs forcing every part into the same configuration, so that the robot is able to perform a task without sensing the pose of the part. Many of these tasks have already been automated. However when there is large variation in parts, or when constructing dedicated jigs is cost-prohibitive for low rates of production, a robot must use sensing to localize each part.

In autonomous localization we seek to know the pose of an object from measurements obtained from probing. We wish to choose probing actions to locate parts efficiently, and thus wish to avoid probing actions that provide little value. We are finished when we are reasonably confident that the error in the pose estimate is below some desired tolerance.

Rather than calculating a single best estimate of the pose, we represent the uncertainty in knowledge by a probability distribution. Estimating the probability distribution is important both for choosing effective measurements and knowing when we have 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* [1].

However particle filters behave poorly both when the dimensionallity of the system grows large and when measurements become precise [2]. In the problems we consider there is a large initial uncertainty in the six-dimensional system, and we require measurements with low uncertainty 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.


(a)

(b)

Fig. 1: **(a)**: The robot performing a touch measurement **(b)**: The belief of the part location and measurement

This paper first presents a solution to particle starvation during touch localization (section IV). We design a particle filter with an alternate update procedure that is able to combine accurate measurements with the prior. We then tailor this update procedure to the specific measurement process in our localization problem to further improve the performance.

To minimize the time taken to localize the part, the robot should choose informative measurement actions. To achieve this we sample many actions and select the one with the highest expected *information gain*. Fully predicting this is also computationally expensive, thus previous methods using this metric introduce delays during which the robot pauses between measurements and yet still only samples a small number of actions. Our approach involves a fast approximation for information gain that takes advantage of the discretized belief from the particle filter (section V). This allows us to evaluate hundreds of potential measurement actions in a few seconds.

We demonstrate our method both in simulation and on a real robot (section VI). Figure 1 shows our robot equipped with a touch probe, the belief distribution, and the expected points of contact during the probing (red dots). Even with relatively inaccurate sensors we are able to localize the 6 DOF of a part to within 2mm with an average of only 7 measurements.

II. Related Work

Particle filters, touch measurements, and maximal information gain measurement selection have all been used in localization tasks. In particular the recent DARPA ARM-S competition led to progress with robotic arms localizing and grasping objects.

Since their introduction particle filters have been popular due to their ease of implementation and ability to model complex distributions, process models, and measurement models [1]. However, as we will describe, for a measurement with low uncertainty there exists only a thin manifold of states consistent with that measurement, yielding a low probability of any particle existing on that manifold, leading to particle starvation [3].

To address this, Koval introduced the Manifold Particle Filter, using different sampling methods depending on the volume of the space consistent with a measurement [4], [2]. This allows a quick update of the belief when the contact sensor is not in contact with the part, and only requires addressing the harder thin manifold update problem when contact is made. Koval used multiple methods when updating from measurements when on this thin manifold, and shows that rejection sampling requires the fewest restrictions on prior knowledge of the environment, but naive rejection sampling is time consuming. His efficient methods require direct sampling from the contact manifold, which is not feasibly for a complex part.

Petrovskaya focused on global localization of objects via touch [5] and introduced the Series Scaling algorithm to overcome particle starvation. The Series Scaling algorithm adaptively alters the particle density depending on the complexity of the posterior. Multiple passes through the measurement data are used and the precision of the modeled belief is scaled from low to high, avoiding unnecessarily precise estimates in exceedingly unlikely regions of belief space. This is complementary to our approach, and implementing multiple passes on our methods could lead to a faster update process.

Alternative approaches to modeling touch localization adapt a Kalman Filter to model the belief distribution. A Kalman filter requires a Gaussian belief of the estimated state, which is inaccurate in touch localization. For example, multi-modal distributions appear when it is ambiguous whether a close edge or far edge was touched.

A Kalman filter also requires a linear measurement models, which does not exist for touch localization, so this too must be approximated. The Extended Kalman Filter performs first-order linear approximations of the measurement model, but this diverges for pose estimation with large initial error [6]. The Unscented Kalman Filter approximates the measurement model through evaluation at many "sigma" points chosen by tuning parameters. As the pose likelihood given a measurement varies by many orders of magnitude in different dimensions, these parameters are sensitive and unreliable. Srivatsan constructs a linear measurement model using two measurements and assuming the correspondence between the probe tip and the touched point on the object [7]. However, this correspondence is unknown, and additional effort is needed, otherwise approximating correspondence (for example through Iterative Closest Point methods) can lead to convergence on local, not global, minima.

Much of the recent work on touch measurements uses a robotic hand with contact sensors. In these works evaluation of both actual and simulated measurements required collision checking between two meshes which is computationally expensive.

Hebert et. al. use geometric (CAD) models of objects such as screwdrivers and door handles as well as the geometric model of their robotic arm to autonomously choose touch actions that localize objects sufficiently to perform everyday tasks, such as grasping and opening doors. Their algorithm greedily selects the next best touch action from a list of candidate actions to maximize information gain [8].

Javdani shows that selecting the next touch to maximize information gain is submodular under assumptions of a static object and an action cost independent of object and robot state, explaining the effectiveness of the greedy approach [9]. This provides a sound theoretical basis for our approach. Javdani demonstrates the computation of information gain is time consuming and proposes an alternative method of hypothesis pruning. Our formulation computes the expected information gain about two orders of magnitude faster, and thus we are able to evaluate many more potential measurement actions and model the belief using more particles.

III. Problem Formulation and Models

The robot estimates the pose X of an object based on a set of measurements $Z_t = \{z_1, ..., z_t\}$ made by probing the object. We use a triangular mesh to describe the part geometry and attach a frame to this mesh. The state X is the $SE(3)$ transformation from a fixed world frame to this part frame. This is a 6-dimensional state stored as position (x, y, z) and orientation angles (α, β, γ). We assume our geometry is rigid, thus the state can be fully described as the $SE(3)$ configuration of a frame attached to the part. We further assume that the part is fixed in space relative to a world frame and that actions the robot takes do not perturb the state. Thus our process model does not change the state.

A. Belief

At any time step we have a belief of the state $bel(x_t) = p(x_t|Z_t)$. Measurements are a non-linear probabilistic function of the true state: $z_t \sim p(z_t|X)$. As in all Bayesian filters, the belief $bel(x_{t+1})$ is calculated recursively as follows [1]:

$$bel(x_{t+1}) \leftarrow \eta \; p(z_t|x_t) \; bel(x_t) \tag{1}$$

with η as a normalization factor. Each new measurement value triggers an update to the belief $bel(x)$.

B. Sensors

We model a measurement technique common to industrial robots: touch probing. A touch probe is a spherical tip mounted to a rigid shaft connected to the end effector capable of detecting contact. We make the assumption that every probe measurement action will involve the probe tip moving along a linear trajectory through space until contact is made, at which point the robot will stop.

All the methods presented in the paper also apply to a 1D laser range finder. This offers faster measurements without requiring contact with the part. We implemented the techniques presented in the paper on robots equipped with both sensors and have observed similar results.

C. Measurement Model

A measurement is defined by a measurement action and a measurement value: $\mathcal{M} = \{\mathcal{A}, z\}$. The measurement action is chosen by the robot while the measurement value is the probabilistic result of taking that action given the part state.

We define measurement actions by a start point $\mathcal{A}_p$ and a vector $\mathcal{A}_v$ both in $\mathbb{R}^3$. These define the line segment of the motion of the center of the tip of the touch probe. The measurement value z is the distance the start point travels along the vector until contact is made. The point of contact in R^3 is then easily recoverable as $A_p + z \; \frac{\mathcal{A}_v}{||\mathcal{A}_v||}$. Note that this point may not lie on the surface of the part, as the probe tip has width, the action is not perfectly followed by the robot, and the sensor will add noise to the measured value. For the robot arms and sensors used in manufacturing the uncertainty added typically ranges from 0.1 to 1mm [10].

IV. Modeling Part Uncertainty

Localization of an object requires a model of the belief of the object's state, and a method of updating the belief given a measurement. We choose to model the belief of the object state by a finite list of particles, $\mathcal{P} = \{(x_i \in SE(3))\}$, to accommodate both multimodal distributions and the nonlinear relationship between the object state and measurement value. Both are present during localization of an object in $SE(3)$.

A. Problems with the standard particle filter

A common method of updating particles based on a measurement is importance sampling [3]. 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 [2].

B. Rejection Sampling Method

An alternate approach is to use rejection sampling. Rejection sampling does not require a high density of particles to avoid particle starvation and failure of rejection sampling is far easier to notice and resolve. Most importantly, we 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 distribution g. A constant M is determined such that $f(x) \leq Mg(x) \; \forall \; x$. A sample x^* drawn from $g(x)$ is accepted with probability $f(x^*)/Mg(x^*)$ and rejected otherwise. The process is repeated until the desired number of samples has been accepted. We wish to sample particles from the posterior $bel(x_{t+1})$ but cannot do so directly. Instead we sample from our continuous prior belief $bel(x)$ and possibly reject based on the measurement.

Broad Particles: We first reconstruct the continuous prior belief by broadening each of the particles. We apply a *Gaussian kernel* to the particles, with the kernel covariance proportional to the covariance of the particle states. 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.

C. Fast Evaluation of Sampled States

States sampled from this continuous prior are then rejected if they are inconsistent with the measurement. While the formulation of rejection sampling allows us to model complicated measurement error, we implement a binary measurement model. We reject all sampled particles where the measured point is sufficiently far from all faces of the object. We define "sufficiently far" as more than 3 standard deviations of the sensor measurement noise. As a low uncertainty measurement will accept only a thin manifold in state space, the probability of sampling a particle consistent with

[1] The full update of a Bayesian filter also includes a process model. Our assumption of a fixed part yields the static process model and this simpler formulation

the measurement may be low, and a lot of sampling may be required, therefore we desire the rejection process to be fast.

To reduce the computational cost per sampled state we use discretized space, known as a *distance field* [11], to precompute and cache the minimum unsigned distance $D_f(p)$ from point p in voxelized space to the object surface $\partial S \subseteq \mathbb{R}^3$:

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

$$f(q) = \begin{cases} 0, & \text{if } q \in \partial S \\ \infty & \text{otherwise.} \end{cases} \tag{3}$$

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 field, as described below. Each voxel is assumed to be a cube in 3D space. A fast 3D Triangle-Box Overlap method [12] 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 Field: The computation of distance field $D_f(p)$ takes the input of the computed binary array $f(q)$ (Eq. 3), and a linear-time algorithm for 3D distance field construction [11] is then used. The resulting distance field 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 field directly in the world frame. Instead, the computation of the voxel map and distance field is relative to the object frame, where the object is assumed fixed during the entire localization. Each measurement M_t in the workspace is then transformed into the part frame, where the transform $T(x_{t+1})$ comes from the pose of the sampled state x_{t+1}. Therefore, by transforming back to the object frame, all measurements on this same object can share the same distance field, where the minimal unsigned distance $dist_u(M_t, S)$ between each measurement M_t and the object $S(x_{t+1})$ can be obtained directly:

$$dist_u(M_t, S(x_{t+1})) = D_f(T(x_{t+1})^{-1}M_t) \tag{4}$$

The signed distance $dist(M_t, S(x_t))$ between the probe and the object can be obtained from the unsigned distance, as shown in Eq. 5, 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

$$dist(M_t, S(x_t) = \begin{cases} dist_u(M_t, S(x_t) - r_p, & \text{if } M_t \notin S \\ -dist_u(M_t, S(x_t) - r_p, & \text{otherwise.} \end{cases} \tag{5}$$

The unsigned distance $dist(M_t, S)$ is always 0 in the ideal case, however, when evaluating a sampled state, only those that satisfy $|dist(M_t, S)| > T_d$ 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.

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 a greater distance $dist_u(M_t, S) + r_p$ before the computation of signed distance.

D. Improvements on the Particle Filter

Adaptive Voxelization: When the workspace is large enough, 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 field for each measurement is generated prior to the update. As the belief of the object's pose is represented as a distribution, the range of the distance field DF for each measurement M_t is selected so that:

$$\xi < \int_{T(x_t)^{-1}M_t \in DF} bel(x_t)dx \tag{6}$$

The voxel size is adjusted accordingly while keeping the number of voxels fixed. When there is large uncertainty the voxels will become larger to improve the sample speed. When the uncertainty is low the precision is increased.

Adaptive Bandwidth: The selection of the Gaussian kernel bandwidth h is important during the sampling process. 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. Silvermans rule of thumb estimator [13] is used to dynamically adjust the bandwidth: $h(t) = (\frac{4}{(d+2)n})^{1/(d+4)}\hat{\sigma}_t$, where $\hat{\sigma}_t$ is the standard deviation of the sampled states and number of dimensions $d = 6$.

Adaptive Sample Size: The number of particles is determined by *Kullback-Leibler divergence* (KL-divergence) which measures the difference between the sample-based maximum likelihood estimate $\hat{p}$ and the true distribution p:

$$D_{KL}(\hat{p}, p) = \sum_x \hat{p}(x) log \frac{\hat{p}(x)}{p(x)}. \tag{7}$$

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 [14]:

$$n = \frac{1}{2\epsilon}\chi^2_{k-1,1-\delta} \tag{8}$$

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

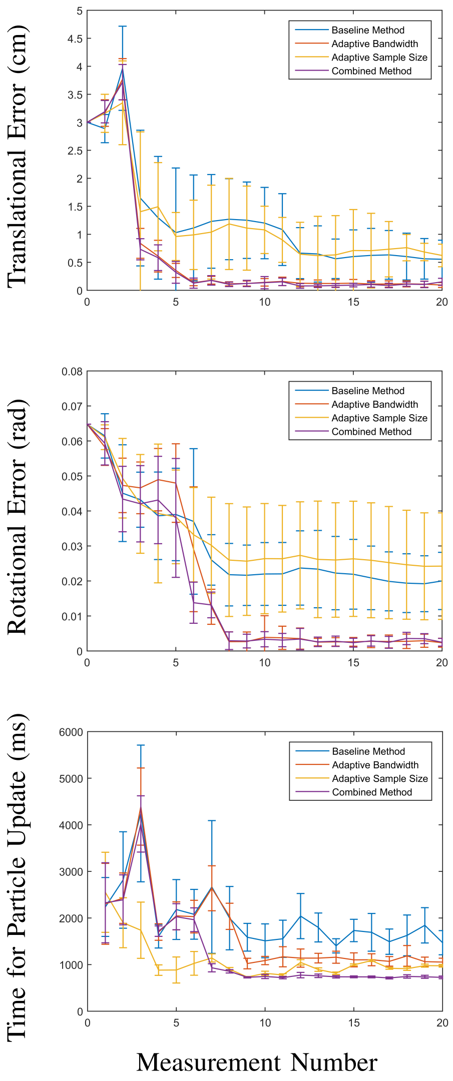


Fig. 2: Comparison of the time and accuracy of the particle filter update step when using Adaptive Bandwidth and Adaptive Sample Size

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 k of bins with support, as shown in Eq. 4. The bins are implemented as a multidimensional grid with fixed size in the configuration space. During sampling, the number k of occupied bins 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 actual number of particles reaches the desired value or a predefined maximal limit, whichever is smaller, the sampling process finishes.

Figure 2 shows the comparison between the particle filters with different improvements that are mentioned above. The simulation uses a mesh model with 39444 triangles. The set of measurements used for each method is fixed and predefined. The baseline method represents the particle filter that only implements adaptive voxelization while using fixed kernel bandwidth $h = 0.0035$ and fixed number of particles $(n = 500)$. Adaptive Bandwidth improves the convergence accuracy significantly compared to the baseline method, while Adaptive Sample Size 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 compared to the other methods.

V. Predicting Effective Measurement Action

Performing measurements is time-consuming, so it is crucial to perform only measurements that greatly reduce uncertainty. We select many candidate actions, evaluate their effectiveness using the metric of expected Information Gain, and perform only the best one. We make use of our discretized belief and specific measurement formulation to quickly calculate the expected information gain taking into account arbitrary part and measurement uncertainty.

While measurements are valuable because they influence our future belief, we avoid the expensive computation of $bel(x_{t+1})$ for every possible measurement value. Instead we treat this as a decision problem over our discrete particle states and ignore the continuous distribution the particles represent.

A. *Information Gain*

The Information Gain of a discrete distribution due to a probabilistic measurement is defined as the expected reduction of entropy.

$$IG(\mathcal{P}|\mathcal{M}) = H(\mathcal{P}) - H(\mathcal{P}|\mathcal{M}) \tag{10}$$

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.

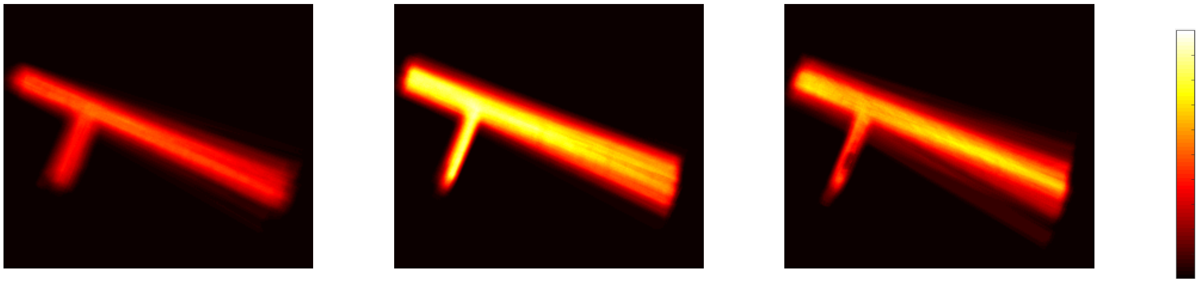
The entropy of a discrete distribution of states depends only on the probabilities of each state occurring.

$$\begin{aligned} H(\mathcal{P}) &= -\sum_i w_i \log w_i && (11) \\ &= \log n && (12) \end{aligned}$$

where n is the number of particles. Equation 12 follows as our particles are uniformly weighted: $w_i = \frac{1}{n}$. Note that the entropy does not depend on the location of the particles in the 6D pose space, and that a set of particles spread out can have the same entropy as a tightly packed set. This counter-intuitive definition is due to only considering the entropy of the discrete distribution of particles where every particle is distinct from every other particle, and ignoring continuous belief which the particles represent.

To calculate the entropy of the particles conditioned on a measurement action, we first discretize the possible measurement values, transforming a continuous measurement into a discrete measurement $m_j = (\mathcal{A}, z_j)$. The conditional entropy of the as yet unknown measurement is then:

$$H(\mathcal{P}|\mathcal{M}) = \sum_j p(m_j)\ H(\mathcal{P}|m_j) \tag{13}$$



(a) Initial Belief

(b) 1 Measurement

(c) 2 Measurements

Fig. 3: Heat maps showing information gain

The term $p(m_j)$ is the probability that after performing this measurement the discretized measurement value is z_j. The term $H(\mathcal{P}|m_j)$ is the entropy of the particles once the measurement is known to be m_j. Once the measurement value is known some particles may be eliminated entirely, some may become less likely, and some may become more likely. These now weighted particles will have a lower entropy than the original set.

We visualize the information gain of our particles in Figure 3 by using the information gain for a grid of parallel rays as the "temperature" in a heat map. Initially **(a)** probing into the page is likely to miss the part, so information gain is low for all these measurements and we instead probe sideways. After this first measurement **(b)** the uncertainty is reduced, and probing into the page is more likely to hit the part and provide information. Once this second measurement is performed **(c)**, another measurement in the same place will not provide more information.

The details of efficently calculating the terms in equation 13 are described below in section V-C. However we first describe the measurement simulation in detail.

B. Measurement Simulation

We need a method of predicting the measurement value obtained from a measurement action: $p(m_j)$. We again choose to model the measurement uncertainty by a discrete sampling of the continuous distribution. For a single measurement action, and for each particle a sampling of measurement values is drawn from the distribution of measurement values that would be obtained if that particle was the true state.

Our sensor measurement will indicate a distance z traveled along the measurement action $\mathcal{A}$ until reaching the part. We start by examining the distance from the start point along the vector until the first intersection with the part. Though a crude approximation of the true measurement value, the benefit of this model is that given a measurement action and part pose, the measurement value can be calculated as the intersection of a ray and a triangular mesh. Due to their heavy use in computer graphics, ray-mesh intersection algorithms have been heavily optimized and can be computed in parallel. We now describe refinements to this approximation while maintaining the benefits of ray casting.

Measurement Width: 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. We approximate the measurement cylinder by discrete uniformly spaced rays on the cylinder exterior, and return the lowest ray-mesh intersection distance.

Measurement Error: 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 we choose to model the error in a discrete general method. For each measurement action we make many simulated measurements where we perturb the initial conditions according to an error model for the robot and perturb the measured value according to a model of the sensor. Because our ray-mesh intersection method is cheap, the additional cost these extra simulations add is acceptable.

C. Discrete Measurement Approximation Using Bins

The values from a simulated measurement are used to sort the particles into bins as illustrated in Figure 4. The fraction of particles in each bin determines the measurement probability $p(m_j)$, while the number and diversity of particles in each bin determines the entropy $H(\mathcal{P}|m_j)$.

A simulated measurement action yields a discrete distribution of possible measurement values for each particle. For each particle there will be a list of possible measurements approximating a continuous distribution. We combine these lists into a single list of <particle, measurement value> pairs. This list is sorted into bins by measurement value, such that all $<$ particle i, $m_j >$ pairs are sorted into bin m_j.

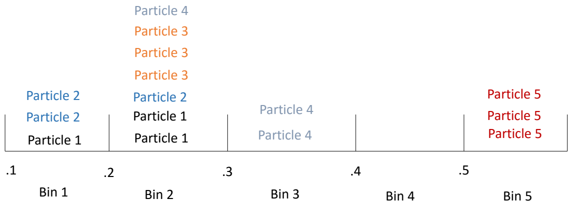


Fig. 4: Illustration of how a potential measurement sorts particles into bins

The probability of the true measurement falling within bin m_j is proportional to the number of <particle, measurement> pairs in the bin. Given that a particular bin m_j contains the true state, the probability that a particular particle i in that bin represents the true state, $p_{i,b}$, is proportional to the number of times pairs containing particle i appear in bin b.

$$p_{i,j} = \eta \text{ Count[particle i in bin } m_j] \tag{14}$$

$$p_j = \eta \sum_i \text{Count[particle i in bin } m_j] \tag{15}$$

The conditional entropy can then be directly calculated.

$$H(\mathcal{P}|m_j) = H(\text{bin } m_j) \tag{16}$$

$$= \sum_i p_{i,b} \log p_{i,b} \tag{17}$$

$$H(\mathcal{P}|\mathcal{M}) = \sum_j p_j H(\mathcal{P}|m_j) \tag{18}$$

VI. Experiments

We evaluated our algorithm both in simulation and on a physical system. The software was implemented in C++ using ROS for visualization and integration with the robot. Our results are summarized in Table I.

We tested on a custom 7-DOF robotic arm with approximately a 1 meter reach, equipped with a touch probe. During experiments, both the arm and parts were rigidly fixed to the ground. Due to kinematic inaccuracies and deflections the global robot accuracy of the end effector placement had an error of a few millimeters, though we had no method of measuring this more accurately.

We modeled and constructed an object typical of large manufacturing (see Figure 1). The meshes agreed with the physical parts to within 3mm. In simulation the same mesh was used to model belief and perform simulated measurements.

A. Action Selection

A mean pose was calculated using all of the particles and candidate actions were chosen in the directions normal to the faces of this mean orientation. The measurement action performed at each step was the candidate measurement action with the largest expected information gain.

Candidate actions were not constrained to intersect the mesh of any particle, however actions that miss most configurations provide little information and are thus not preferred. Candidate actions that are kinematically infeasible are rejected. In our experiments for each measurement performed we evaluated candidate actions until we had modeled 500 actions with non-zero information gain.

B. Robot Results

Our touch probe consists of a 0.6mm spherical tip mounted to a 100mm rod attached to a 6-D JR3 force/torque sensor. Although contacts in any direction were possible, the sensor was significantly more accurate with the contact force parallel to the shaft, and measurements actions were always chosen to place the contact force in this direction. Under these conditions our sensor exhibited a 1mm repeatability.

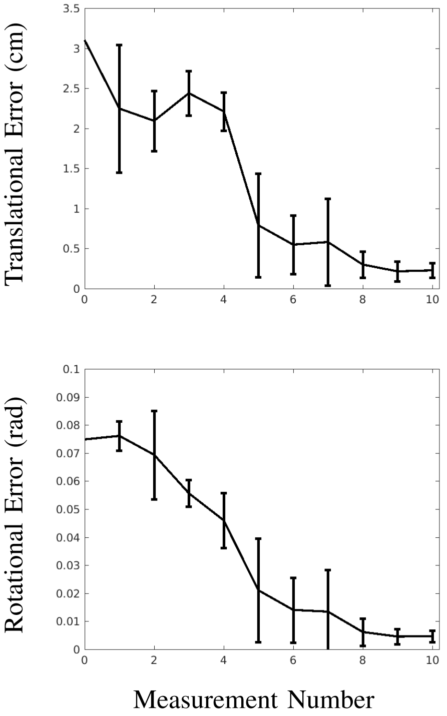


Fig. 5: Translational and rotational error during localization on the physical robot

After selecting from the set of candidate actions the robot executed the measurement action by first following a trajectory to the start point $\mathcal{A}_p$, designed to avoid collision with the part. The end effector was then controlled to move in a straight line in the direction of the measurement vector $\mathcal{A}_v$ until contact was detected. To improve measurement accuracy a double-touch was implemented: the robot first moved the probe tip at a higher velocity (1cm/s) until contact, backed off slightly, then probed at a much slower velocity (2mm/s). The robot then retreated to a safe point while the particle filter updated and the next action was selected.

We performed 15 localization trials with identical initial beliefs and true part location. Figure 5 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. The error bars show one standard deviation.

Full videos of the robot localization are available online[2].

C. Simulation Results

Simulations allowed us to tightly control experiments and quickly test many scenarios. Our robot arm and touch probe are less accurate than those found in precision manufacturing, but simulation allowed us to examine the effects

[2] https://www.youtube.com/watch?v=Gc_TSRL_Azo, https://www.youtube.com/watch?v=-EDO8XbQ3yI

	Measurement Error (cm)	Init. Position Uncertainty (cm)	Init. Angular Uncertainty (rad)	Final Position Error (cm)	Final Angular Error (rad)	Average Computation (s)
Physical Robot	0.05	3	0.08	0.22 ±0.09	0.0047 ±0.002	7.0 ±0.3
Simulation of Robot	0.05	3	0.08	0.17 ±0.09	0.005 ±0.003	7.0 ±0.3
Simulation of Accurate Robot	0.01	3	0.08	0.03 ± 0.02	0.0004 ±0.0002	7.8 ±0.3

TABLE I: Results of Experiments

of adjusting the measurement and robot accuracy. We also easily varied the ground truth and initial belief distributions.

We confirmed that our physical system results agreed with our simulation results by simulating the same part orientation, initial uncertainty, and measurement uncertainty. We then simulated less accurate measurements, more accurate measurements, larger initial uncertainty, and a scenario where initial uncertainty in some directions was very small. We present summarized results in Table I after performing 10 localization measurements. We sampled from 500 potential measurement actions and 500 particles to model the belief.

VII. Conclusions and Future Work

This paper summarized the difficulty of tactile localization problems when using accurate measurements and presented an alternative particle filter design that uses rejection sampling and a precomputed distance transform. In addition we presented a method for fast calculation of information gain using the particle filter to approximate the continuous distribution. Combined, these contributions allowed our robot to autonomously localize a part with high accuracy using few measurements. This ability is crucial for robots performing tasks on a diverse set of objects with less human direction.

For future work, we would like to extend our approach to tasks that require only that the part pose is accurately in some dimensions, but large error is other dimensions is acceptable. In addition we currently assume the mesh model matches our object, yet in reality parts are manufactured with tolerances. In many cases it is important to localize to specific features at the expense of the average localization. We are currently working to efficiently model and reason about the complicated relationships between different features.

VIII. Acknowledgments

We gratefully acknowledge our collaborators at Boeing for their support and suggestions. This material is based upon research supported by (while Dr. Simmons was serving at) the National Science Foundation.

References

[1] Sebastian Thrun, Burgard Wolfram, and Fox Dieter. Probabilistic Robotics. pages 1999–2000, 2000.

[2] 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.

[3] Sebastian Thrun, D Fox, and W Burgard. Monte carlo localization with mixture proposal distribution. *Proceedings of the National Conference on*, pages 859–865, 2000.

[4] Michael C Koval, Nancy S Pollard, and Siddhartha S Srinivasa. Pre- and Post-Contact Policy Decomposition for Planar Contact Manipulation Under Uncertainty. pages 1–27, 2011.

[5] Anna Petrovskaya and Oussama Khatib. Global Localization of Objects via Touch. *IEEE Transactions on Robotics*, 27(3):569–585, 2011.

[6] D Choukroun, Y Oshman, Senior Member, Control Conference, Los Angeles, and Y Oshman. Novel Quaternion Kalman Filter. 42(1), 2006.

[7] Rangaprasad Arun Srivatsan, Gillian T Rosen, D Feroze Naina Mohamed, and Howie Choset. Estimating SE (3) elements using a dual quaternion based linear Kalman filter. (3).

[8] 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.

[9] Shervin Javdani, Matthew Klingensmith, J. Andrew Bagnell, Nancy S. Pollard, and Siddhartha S. Srinivasa. Efficient touch based localization through submodularity. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1828–1835, 2013.

[10] Bradley Saund and Russell DeVlieg. High accuracy articulated robots with CNC control systems. *SAE International Journal of Aerospace*, 6(2):1–6, 2013.

[11] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Cornell Computing and Information Science Technical Report TR20041963*, 4:1–15, 2004.

[12] Tomas Akenine-Möllser. Fast 3D Triangle-Box Overlap Testing. *Journal of Graphics Tools*, 6(1):29–33, 2001.

[13] B. W. Silverman. Density Estimation for Statistics and Data Analysis. *Biometrical Journal*, 30(7), 1986.

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

[15] Kaijen Hsiao and Leslie Pack Kaelbling. Task-Driven Tactile Exploration. *Proceedings of Robotics Science and Systems*, 2010.

[16] Shervin Javdani, Andreas Krause, Yuxin Chen, and J Andrew Bagnell. Near Optimal Bayesian Active Learning for Decision Making. 33, 2014.

[17] Heui Jae Pahk and Woo Jung Ahn. Advanced manufacturing Technology. 3:442–449, 1996.

[18] Zhenhua Xiong, Michael Yu Wang, Senior Member, and Zexiang Li. A Near-Optimal Probing Strategy for Workpiece Localization. 20(4):668–676, 2004.

[19] Hong-Tzong Yau and Chia-Hsiang Menq. An automated dimensional inspection environment for manufactured parts using coordinate measuring machines. *International Journal of Production Research*, 30(7):1517–1536, 1992.

[20] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.

[21] Anna Petrovskaya and Andrew Y. Ng. Probabilistic mobile manipulation in dynamic environments, with application to opening doors. *IJCAI International Joint Conference on Artificial Intelligence*, pages 2178–2184, 2007.

[22] Cyrill Stachniss, G Grisetti, and W Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. *Robotics: Science and Systems*, pages 65–72, 2005.