Fine Localization

Brad Saund, Shiyuan Chen, Reid Simmons
Robotics Institute
Carnegie Mellon University
{bsaund, shiyuanc, reids}@cmu.edu

I. Introduction

In automated manufacturing it is critical robots localize to the parts they are creating 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 the robot is able to perform a task without sensing the pose of the part. For the most part 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.

Localization is usually separated into two stages: Course and Fine. Course localization limits the range of the unknown state parameters to allow the slower but more accurate fine measurement steps to converge more quickly. Course localization may be accomplished through vision, jigs, laser scanners, or other means of knowing information about the pose of the part. For some tasks course localization may be sufficient.

For many machining operations precision is required and often accomplished with a touch probe or a precise laser distance sensor. A typical touch probe has a precision sphere on the end of a hardened shaft and returns a binary measurement of whether the shaft is deflected, and thus the sphere is in contact with an object. A laser distance sensor shoots a thin beam and returns the distance at which the beam was reflected from an object, either by time-of-flight or triangulation.

When sensing is required for localization a person is usually involved in designing the actions the robot must take. For high-rate production a person may design a sequence of actions that will localize all part configurations and thus the robot can repeat a task without intervention. For low-rate production it is costly to have an engineer design the localization procedure.

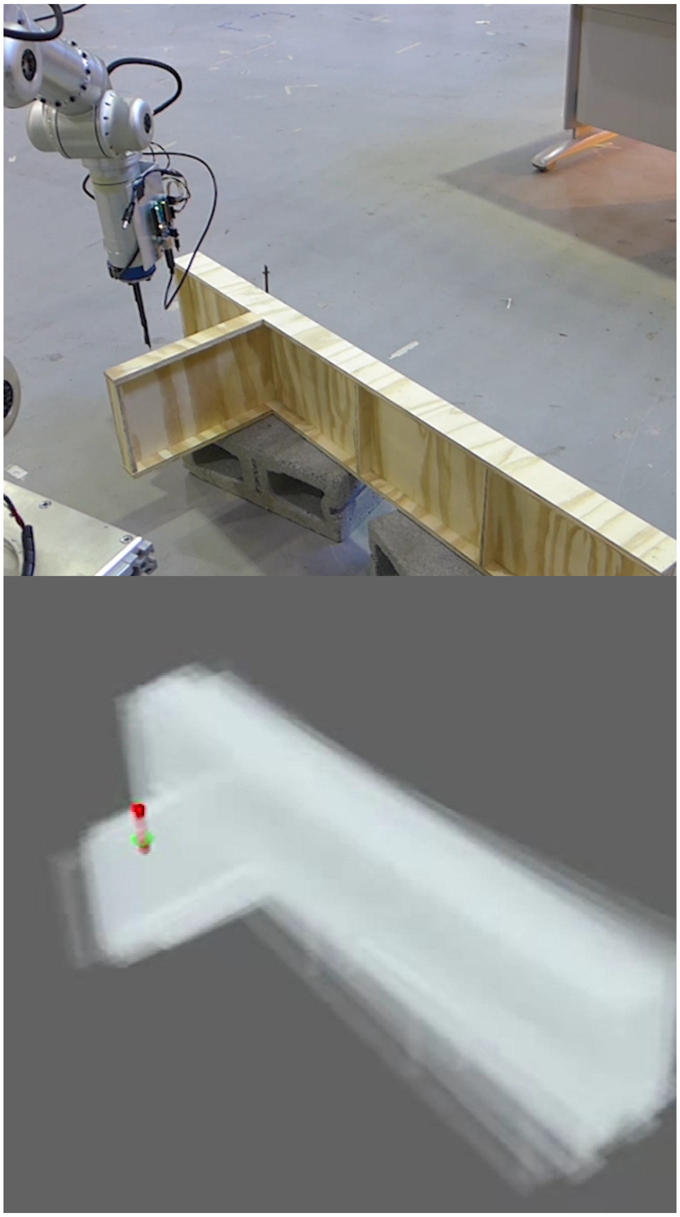


Fig. 1: *Top*: Our robot arm performing a touch measurement on the wooden object
Bottom: The simulated measurement on a set of particles representing the belief of the object pose.

Active localization of objects is accomplished by modeling the uncertainty of the object pose and choosing actions that gather information about this unknown state. In these types of localization problems we would like to use a Particle Filter to model the 6 dimensional uncertainty in pose and choose actions which maximize information gain. However a particle filter behaves poorly in this space with accurate measurements. Additionally predicting the information gain is computationally expensive. This means robots will “think” for a long time, reasoning over a few actions, perform mediocre measurement, and have an artificially poor belief of the object location.

This paper first present a solution to the to the particle filter problem. We design a particle filter with an alternate update procedure that is able to combine accurate measurement with the prior. We then tailor this update procedure to the specific measurements in our localization problem to improve the performance.

We then present a fast approximation for information gain that takes advantage of our discretized belief. This allows us to evaluate hundreds of potential measurement action in a few seconds. We demonstrate the effectiveness of these techniques by performing autonomous localization on a variety of objects

II. Review

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. However for accurate measurements 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 [**?**], [**?**], [**?**], [**?**]. Koval introduced the Manifold Particle Filter, using different sampling methods depending on the volume of the space consistent with a measurement.

Petrovskaya focused on global localization of objects via touch [**?**] and introduces 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. Inspired by these alternate sampling methods we implemented multiple sampling methods to address particle starvation in our problem.

Much of the recent work on touch measurements uses a robotic hand with contact sensors, probably due to the recent DARPA ARM-S competition, and in these works evaluation of both acutal and simulated measurements required collision checking between two meshes which is expensive.

Herbert 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.

Javdani shows that the 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 [**?**]. Our work makes the first assumption of a static object but our method allows for dynamically selecting potential measurement actions, allowing us to choose from a larger set of reasonable measurements.

III. Problem Formulation and Models

We start with an intuitive description of our problem. We seek to know the pose of an object from measurements obtained from probing. We wish to choose our probing actions to locate our part efficiently, and thus wish to avoid probing actions that provide little value. We are finished when we are reasonably confident the error of our pose estimate is below the desired tolerance.

Rather than calculating a single best estimate of pose, we represent the uncertainty in knowledge by a probability distribution, which stores the likelihood of every state given our sensor measurements. 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. Object Description and Belief

The robot estimates the pose X of an object based on a set of measurement $Z_t = \{z_1, ..., z_t\}$ 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 the object is both rigid and fixed.

At any time 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 is the main benefit of a Bayes filter, 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 $bel(x)$. We end the measurement process and consider the part localized when a specified portion of the belief is within a specified distance from the mean of the belief:

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

$$\mu_t = mean(bel(x_t)) \tag{2}$$

$$\mathcal{T} \subset SE(3) \tag{3}$$

$$\xi < \int_{\mathcal{T}+\mu_t} bel(x_t)dx \tag{4}$$

In practice the allowable error $\mathcal{T}$ for engineered parts is usually axis-aligned values, *ie* ± 90.5mm in the z direction, ± 1 °in pitch.

B. Sensors

We model two commonly use measurement techniques used on industrial robots: touch probing and linear distance sensors. A touch probe is a spherical tip mounted to a rigid shaft connected to the end effector. Various designs may implement a touch probe differently, perhaps thresholding a value from a force sensor, displacement sensor, a pressure sensitive skin, or joint torque measurements, however the end result is a sensor which returns a binary measurement of whether the probe is in contact with an object. We make the assumption every probe measurement action will involve the probe tip moving along a linear trajectory through space until contacting is made, at which point the robot will stop as to not damage the probe. Although more complicated motions are possible, a linear trajectory is the simplest and most common.

A linear distance sensor measures the euclidean distance along a vector from the sensor to the first object. A common implementation releases a narrow beam of pulsed light, the light is reflected by an object, and a sensor next to the emitter measures the round trip time of flight of the light to determine the distance. Another implementation uses triangulation: a narrow beam of light emitted, reflected, and measured by a light sensor array set a known distance from the emitter. The distance of the object determines which light sensors in the array will trigger.

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 measurements actions by a start point $\mathcal{A}_p$ and a vector $\mathcal{A}_v$ both in $\mathbb{R}^3$. These define the line segment over which the measurement is performed: either the motion of the center of the tip of the touch probe, or the line on which the range sensor looks. The measurement value z is the distance from the start point travelled along the vector until contact is made, corrupted by noise. 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.

IV. Modeling Part Uncertainty

Localization of an object requires a model of the belief of the object's state, a process modeling the evolution of the state over time, 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 to accommodate both multimodal distributions and the non-linear relationship between the object state and measurement value. Both are present during localization of an object in $SE(3)$.

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 the part is fixed in space relative to a world frame and actions the robot takes do not perturb the state. Thus our process model does not change the state, simplifying the updating process.

An overview of the measurement model was described in the previous section and will be discussed in detail below.

A. Problems with the standard particle filter

Particle filters have the unfortunate property of requiring a number of particles exponential in the dimensionally of the state. Insufficient resolution leads to particle starvation during the Importance Sampling step. This leads to the counter-intuitive result that particle filters tend to perform worse as measurement accuracy increases [**?**].

A common method of updating particles based on a measurement is Importance Sampling [**?**], yet this has the drawback that particle starvation occurs for low densities of particles and accurate measurements. In importance sampling each particles 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 be rejected but a sufficient number of particles will be resampled so to model the true belief of the state.

When a measurement is consistent with the true prior belief yet no particles are consistent with the measurement, 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 requires becomes prohibitively computationally expensive.

A common technique alleviating this problem is to initially artificially increase the measurement uncertainty, and decrease to the true measurement uncertainty when the particles begin to converge. However this ignores information from the earlier measurements, resulting in additional measuring to achieve a desired uncertainty. Another technique uses an adaptive number of particles, as to only incur a high computational cost when it is needed. This does not overcome the exponential complexity needed to compute the updated

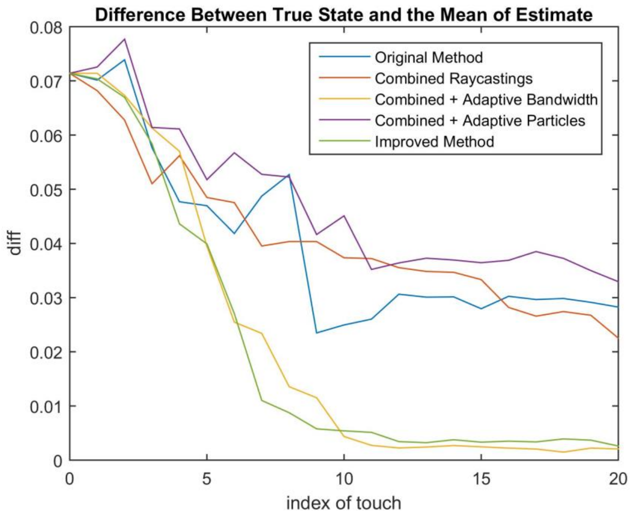


Fig. 2: [SHIYUAN, CHECK THIS AND DESCRIBE METHODS] Simulation of updating different particle filter based on performing random measurement actions. The improved method converges faster

particles for our relatively wide belief and highly accurate sensor.

B. Rejection Sampling Method

Instead we use a variation on Rejection Sampling and solve the problem with importance sampling of collapsing to an artificially accurate posterior, and while we do not bypass the computational complexity associated with accurate measurements in a high dimensional space we increase the limits of state dimensionality and measurement accuracy we are able to model.

Rather than sampling from the particles directly, as in importance sampling, we sample from a belief distribution created by applying a *Gaussian Kernel* to the particles, with the kernel covariance proportional to the covariance of the particle states. Admittedly this may produce a belief distribution different that the true belief distribution from which the particles were sampled, as for example a sharply peaked bimodal distribution will be blurred, yet as with a standard particle filter increasing the number of particles more accurately models the true distribution. We find this to be an acceptable tradeoff to solve the issues with importance sampling.

$$\Sigma = COV(\chi) \tag{5}$$

$$x_{sampled} \sim \sum_i \eta\ (\chi_i + \mathcal{N}(0, k\Sigma)) \tag{6}$$

States sampled from this approximation of the prior are then rejected if they are inconsistent with the measurement. As an accurate measurement will accept only a thin manifold in state space the probability of sampling a particle consistent with the measurement may be low, thus a lot of sampling may be required, thus we desire the rejection process to be fast.

C. Fast Evaluation of Sampled States

To reduce the computational cost for per sampled state we use discretized space, known as a *voxelized distance field*, to precompute and cache the distance from voxelized space to the object surface. 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 a cube in a 3D space. A fast 3D Triangle-Box Overlap[1] method is used to label the voxels that overlap the mesh triangles of the object surface. The voxel map can then be mapped to a binary-valued 3D array, where each value is either 1 or 0 depending on whether the corresponding voxel overlaps or not.

Voxelized Distance Field: The computation of distance field takes the input of the computed binary array, and a linear-time algorithm for 3D distance field construction [2] is then used. By using a binary array as the input, minimum convolution operation at linear-time cost can be introduced to the 3D distance field computation. The computed distance field caches the minimal distance from the voxel center to the voxelized object surface (unsigned distance), which is stored the same way as the voxel map with identical dimensions. During the evaluation process, signed distance is obtained by applying ray-casting at the corresponding voxel to check whether it is inside or outside of the object.

Fast Evaluation: Each sampled particle represents a potential configuration of the frame attached to the object. Different configuration results in different pose of the object in the workspace, which makes it difficult to compute voxel map directly. Instead, the computation of voxel map and distance field is relatively in the object frame, where the object is assumed fixed during the entire localization. Rather than transforming the object into its actual pose in the workspace, each measurement in the workspace is transformed back to the object frame using the same sampled configuration. Therefore, by transforming back to the object frame, all measurements on this same object can share the same distance transform, where the minimal distance between each measurement point and the object can be obtained directly by looking up the 3D array. The distance is then used to determine if a certain particle should be rejected or not, as the manifold in configuration space represents all possible configurations that leads to a less-than-threshold distance between the object and the measurement in the workspace.

The measurement from the range finder is the nearest intersection between the beam and the object, which is checked by applying ray-casting from the start of the beam to the measurement points. When the workspace is large enough, it will be not 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 pose is represented as a distribution, the range of the distance field DF for each measurement M_t is selected so that:

$$\xi < \int_{T(x_{t-1})^{-1}M_t \in DF} bel(x_{t-1})\ dx \tag{7}$$

Where $T(x_{t-1})^{-1}M_t$ is the measurement transformed back to the object frame. The voxel size is adjusted accordingly while keeping the number of voxels fixed. Therefore, the voxel will become larger to improve the sample speed when there is high uncertainty in bel(xt-1), and smaller as bel(xt-1) becomes more peaked when more measurements are applied, which leads to more precise results.

Adaptive Bandwidth: The selection of Gaussian kernel bandwidth h is important during the sampling process. A larger bandwidth is needed when variance of $bel(x_t)$ is large for a fixed number of particles; otherwise smaller bandwidth is preferred to avoid over-smoothing. Silvermans rule of thumb estimator [3] is used to dynamically adjust the bandwidth: $h = (\frac{4}{3n})^{1/5}\hat{\sigma}$

Adaptive Sample Size: The number of particles is determined by Kullback-Leibler distance (KL-distance) which measures the distance between the sample-based maximum likelihood estimate and the true posterior[4]. During the sampling process, assume that particles are drawn from a discrete distribution with k different bins. If the sample size n is given by:

$$\sqrt{\frac{1}{2}} \tag{8}$$

where k-12 is chi-square distribution with k-1 degrees of freedom, z1- is the upper 1- quantile of the normal distribution, then with probability 1-, the KL-distance is guaranteed less than .

V. Predicting Effective Measurement Action

Performing measurements is time-consuming, so choosing to perform measurements that likely distinguish between prior beliefs on part location is crucial to quick localization. We choose the popular metric of seeking to maximize the Kullback-Leibler divergence of the posterior belief and the prior belief, known as information gain:

$$IG = \int_x bel(x_{t+1}) \log \frac{bel(x_{t+1})}{bel(x_t)} dx \tag{9}$$

Since we determine the posterior x_{t+1} uses the true measurement value which we cannot know until performing the measurement action, we select the measurement action which maximizes the expected information gain. We make use of our discretized belief and specific formulation of measurement action to quickly approximate expected information gain taking into account arbitrary part and measurement uncertainty.

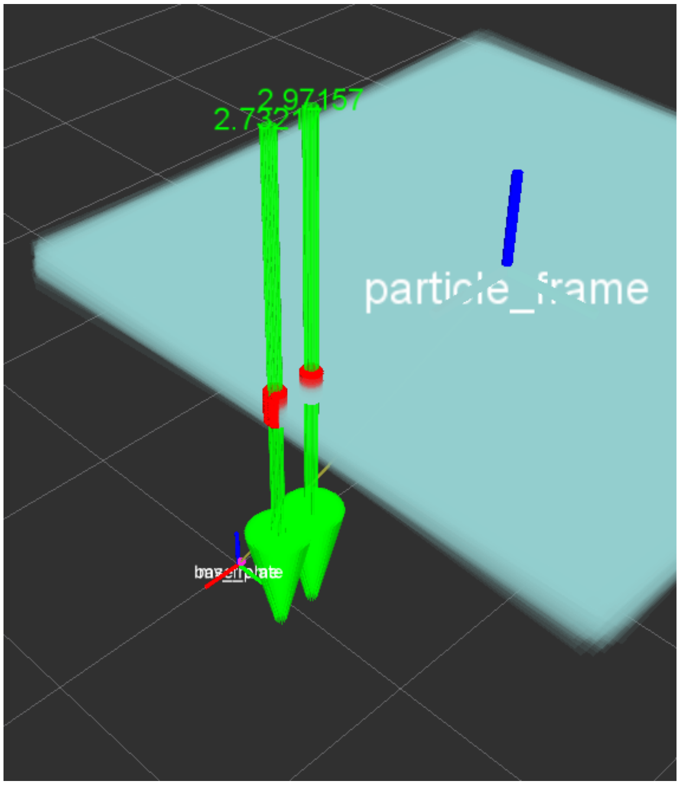


Fig. 3: Two simulated measurements on a group of particles representing square plates. Each measurement action has error, simulated by casting many rays onto the particle's meshes. The right measurement always intersects every particle while the left measurement may miss. The expected *Information Gain* of each measurement is shown

A. Information Gain

The distribution of possible configurations of the object is modelled by discrete weighted particles $\mathcal{P} = \{(w_i, x_i)\}$, such that $\sum w_i = 1$. The informational entropy of this distribution is:

$$H(\mathcal{P}) = -\sum_i w_i \log w_i \tag{10}$$

If a measurement value distinguishes between two possible configurations the entropy of the system will be reduced. Ideally we desire the measurement action that most greatly reduces entropy, however before performing the measurement the exact reduction in entropy cannot be known. The expected entropy conditioned on a measurement action $\mathcal{M}$ that has not yet been made is:

$$H(\mathcal{P}|\mathcal{M}) = \sum_j p(\mathcal{M} = m_j)\ H(\mathcal{P}|m_j) \tag{11}$$

The Information Gain of a measurement is the expected reduction in entropy of the system conditioned on the measurement action:

$$IG(\mathcal{P}|\mathcal{M}) = H(\mathcal{P}) - H(\mathcal{P}|\mathcal{M}) \tag{12}$$

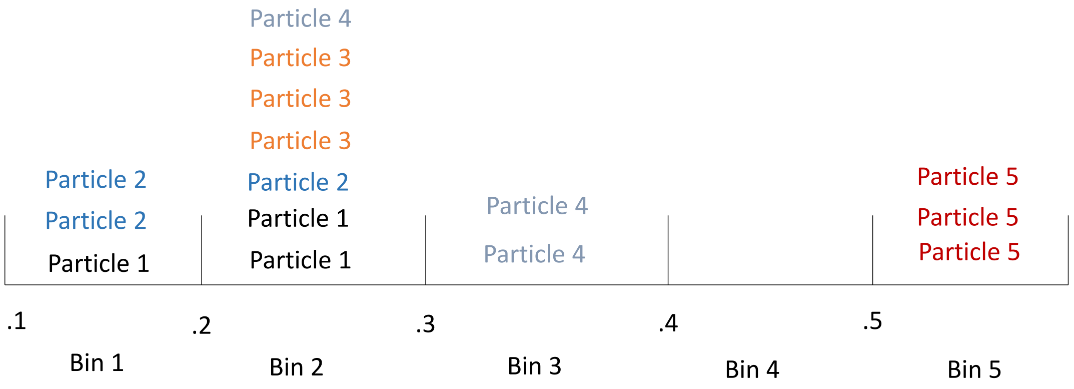


Fig. 4: Illustration of how a measurement action sorts particles into bins

Calculating the IG directly is prohibitively expensive, as evaluating $H(\mathcal{P}|\mathcal{M})$ requires updating the particles for each possible measurement. Instead we approximate the Information Gain.

B. *Measurement Simulation*

We adapt our measurement model to create a simulation of a measurement action. Rather than returning a single measurement value, we calculating a discrete distribution of possible measurements values for each particle.

Our sensor measurement will indicate a distance travelled z along the measurement action $\mathcal{A}$ until reaching the part. We start by examining the distance in $\mathbb{R}^1$ from the start point along the vector until the first intersection with the part. The benefit of this model is a given a measurement action and part pose the measurement value can be calculated as the intersection of a ray and triangular mesh. Due to heavy use in computer graphics, ray-mesh intersection algorithms have been heavily optimized and and can be computed in parallel. Modeling the measurement as a single ray is a crude approximation which we further improve while maintaining the benefits of ray casting.

Measurement Width: While a ray is infinitely thin, both of our measurement devices have thickness. The touch probe spherical tip has a non-zero diameter and the laser beam has a non-zero width, and thus will cast a cylinder rather than a ray, returning the lowest distance between the start point any cylinder-mesh intersection. We approximate the measurement cylinder by discrete uniformly spaced rays on the cylinder exterior, returning the lowest ray-mesh intersection distance. Admittedly this will fail to report intersections on the interior of the cylinder, however given the small measurement width such small features on a part are both uncommon and problematic for other reasons.

Measurement Normality: Ray-mesh intersections do not accurately model situations where the direction of measurement is far from the normal to the face the measurement intersections, and thus we choose to reject such measurement actions. For a touch probe a large motion along a direction nearly parallel to a face will produce very little deflection of the probe tip, thus the sensor will be triggered significantly after contact has been made. Further more, the sensor will be near the threshold for triggering for a long time, thus slight variations in robot motion and part defects will make the specific measurement value unpredictable. A ranged distance sensor also performs poorly during off-normal measurements as too little light is returned to the sensor because most of it is reflected in a different direction.

Measurement Error: Error is caused both by inaccurate start positions and orientations due to robot positioning error, as well as inaccuracies in the sensors. In the most extreme cases error may cause a measurement to move from barely hitting the 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 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, this is computationally feasible.

C. *Bins*

We calculate the approximate expected information gain from the simulated measurement.

A simulated measurement action yields a discrete distribution of possible measurement values for each particle, thus 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 index, measurement value> pairs. This list is histogrammed by measurement value with a bin size ϵ.

Given a particular bin b contains the true state, the probability a particular particle in that bin represents the true state $p_{i,b}$ is the particle weight w_i further weighted by the number of times particle i appears in bin b, normalized such that the sum of the particle probabilities in bin b is

1. Prior to performing a measurement the probability of the true measurement falling within bin b is sum of the weights of the particles within bin b, again normalized such that the sum of the bin probabilities is 1.

$$p_{i,b} = \eta \; w_i \text{Count[particle i in bin } b\text{]} \tag{13}$$

$$p_b = \eta \sum_i w_i \text{Count[particle i in bin } b\text{]} \tag{14}$$

The conditional entropy is the sum of the entropies of each bin weighted by the bin probability.

$$H(\text{bin } b) \approx H(\mathcal{P}|z \in b) \tag{15}$$

$$= \sum_i p_{i,b} \log p_{i,b} \tag{16}$$

$$H(\mathcal{P}|\mathcal{X}) \approx \sum_b p_b H(\mathcal{P}|z \in b) \tag{17}$$

VI. Experiments

We tested our algorithm both in simulation and on a 7-dof robot arm for a variety of objects with a variety of initial distributions. We implemented this in ROS using C++ for the computationally intensive portions.

In simulation we defined a true pose x_{true} and obtained measurement values through simulated measurements on the object at x_{true}.

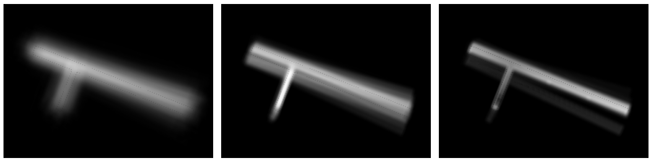
We tested on a custom 7-dof robotic arm with approximately a 1 meter reach, equipped with either a touch probe or a ranged distance sensor. During experiments both the arm and parts were rigidly fixed to the ground. Due to kinematic inaccuracies and deflections the global robot accuracy of end effector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

We modeled and constructed 3 objects: a flat plate, a cube, and a complicated support structure. The meshes agreed with the physical parts to within 3mm. For simulation the same mesh was used to model belief and perform simulated measurements. In all experiments the world frame is chosen as the robot base frame, so the estimated state x is the transformation from the robot frame to the part frame.

A. Initial Belief

To mimic practical situations we performed localization with a variety of starting beliefs. Our initial particles were drawn from an analytic belief distribution of the following forms:

1) *Unconstrained Gaussian*: A mean $\mu \neq x_{true}$ and covariance Σ
2) *Unconstrained Multi-Modal*: Multiple means μ_i and covariances$_i\Sigma$
3) *Uniform*: A range for each parameter of the pose x was specified and particles were drawn uniformly from this range.
4) *Partially Constrained Gaussian*: As a special case of the *unconstrained gaussian*, a mean and covariance were chosen such that the height, roll, and pitch of the initial belief matched as closely as possible the true height, roll, and pitch and had little variance along these dimensions. In practice this situation arises when an object is placed on a known surface.



(a) Initial Belief (b) 1 Measurement (c) 2 Measurements

Fig. 5: Heatmaps of information gain of vertical measurements during the localization process. The intensity of every pixel is the normalized expected information gain of a measurement passing through the pixel orthogonal to the image.

B. Action Selection

Candidate actions were chosen to intersect faces normal to the average particle, and the measurement action performed at each step was the measurement action with the largest expected information gain. A bounding box is constructed around the extremes of the belief meshes and all actions are sampled within this bounding box. Actions are initially sampled with a long measurement distance which is then appropriately reduced to just cover the section where intersections occur.

Candidate actions are not constrained to intersect the mesh of any particle, and as such many candidate actions will not intersect the true part and provide no information. This formulation allows us to ignore the details of the initial belief or object geometry when choosing the method of sampling candidate actions. This behavior is acceptable because actions that do not intersect any particles can be rejected very quickly (10^{-8} s). Candidate actions that are kinematically infeasible are also rejected. In our experiments for each measurement performed we evaluated candidate actions until we had modeled 500 actions with non-zero information gain.

C. Simulation Results

D. Robot Touch Probe Measurement

Our touch probe consisted of a small spherical tip mounted to a 100mm rod attached to a 6-D JR3 force/torque sensor. Contact was determined when the force exceeded a threshold of [XYZ] N, for [XYZ] time. To improve measurement accuracy a double-touch was implemented by where the robot first moved the probe tip at a higher velocity (10cm/s) until contact, backed off slightly, then probed at a much slower velocity (1cm/s). 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 in this direction. Under these conditions our sensor exhibited a 1mm repeatability.

After selecting from the set of candidate actions the robot executes the measurement action by first moving to the start

	Number of Measurements	Final Position Error (mm)	Final Angular Error (deg)	Average Computation (s)
Large Unconstrained	7	1.2345	1.2345	3.456
Small Unconstrained	7	1.2345	1.2345	3.456
Partially Constrained	3	1.2345	1.2345	3.456

TABLE I: Results of Experiments

point $\mathcal{A}_p$ by following a series of trajectory designed to avoid collision with the part. The end effector is then controlled to move in a straight line in the direction of the measurement vector $\mathcal{A}_v$ until a measurement was obtained. The robot then retreated to a safe point while the particle filter updated and the next action was selected.

E. Robot Ranged Distance Sensor Measurement

For other experiments we replaced the touch probe by a ranged distance sensor. We used a VL6180X time-of-flight sensor with a 15cm range. This sensor provides 1mm resolution and consistent measurements under similar lighting and object reflectivity. We attached a thin tube to the light sensor to narrow the measurement cone to approximately a 5mm radius at 10cm.

The robot performed measurements by first moving to $\mathcal{A}_p$ with the sensor pointing in the direction of $\mathcal{A}_v$. 1000 distance measurements were averaged, taking approximately 1 second.

F. Results

We performed localization and compared the resulting mean belief with the true state. We determined the true state by manually driving the end effector of the robot to known points of the object and best fitting the part frame. After localization we commanded the end effector 10 mm from the object in the localized frame and verified the distance with calipers.

A minimum of 6 measurements are required to constrain our object with initial uncertain in all dimensions. In practice 6 or 7 measurements sufficiently collapses our belief. We present summarized results in Table I

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

G. Conclusions and Future Work

We summarize the problems with using a particle filter in localization problems with accurate measurements and present an alternative particle filter for our problem. In addition we present a method for fast calculation of an approximation for Information Gain for our measurement techniques. These contributions allow our robot to autonomously choose good measurement actions and quickly reduce uncertainty in object pose.

The method presented here greedily selects the measurement with highest information gain. In some tasks we may wish to choose from longer sequence of actions. Slight modifications to our approximation for information gain may allow for fast analysis of longer sequences of measurements.

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 the average localization. Future work will address the complicated relationship between different features.

References

[1] Yuxin Chen, J Andrew Bagnell, and Andreas Krause. Submodular Surrogates for Value of Information. 2012.

[2] S.R. Chhatpar, M.S. Branicky, Heui Jae Pahk, Woo Jung Ahn, Anna Petrovskaya, Oussama Khatib, Zhenhua Xiong, Michael Yu Wang, Senior Member, Zexiang Li, Hong-Tzong Yau, Chia-Hsiang Menq, Shervin Javdani, Matthew Klingensmith, J. Andrew Bagnell, Nancy S. Pollard, and Siddhartha S. Srinivasa. Particle Filtering for Localization in Robotic Assemblies with Position Uncertainty. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 27(3):569–585, 2005.

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] Heui Jae Pahk and Woo Jung Ahn. Bdvanced manufacturing Technologg. 3:442–449, 1996.

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

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

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

[13] 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.

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