Bayesian Estimation for Autonomous Object Manipulation Based on Tactile Sensors

Anna Petrovskaya, Oussama Khatib, Sebastian Thrun, Andrew Y. Ng
Computer Science Department
Stanford University
Stanford, California 94305
{ anya, ok, thrun, ang }@cs.stanford.edu

***Abstract*— We consider the problem of autonomously estimating position and orientation of an object from tactile data. When initial uncertainty is high, estimation of all six parameters precisely is computationally expensive. We propose an efficient Bayesian approach that is able to estimate all six parameters in both unimodal and multimodal scenarios. The approach is termed Scaling Series sampling as it estimates the solution region by samples. It performs the search using a series of successive refinements, gradually scaling the precision from low to high. Our approach can be applied to a wide range of manipulation tasks. We demonstrate its portability on two applications: (1) manipulating a box and (2) grasping a door handle.**

I. Introduction

Today most robots operate in structured environments built specifically with the robot in mind. Precise position control and pre-defined trajectories can be used in these environments as all parameters are known in advance. The challenge of bringing robots into environments built for humans is in making robots act autonomously when environment parameters are uncertain. Thus robots need to estimate these parameters from sensory information. A variety of sensors can be used to obtain sensory information, including laser, vision, sonar, and tactile. In this paper, we focus on the use of tactile sensors to estimate parameters for an object to be manipulated in a process called localization. While these sensors have received relatively little attention in the literature as means of localizing objects, they have certain advantages over other types of sensors. Tactile sensors tend to provide better precision (up to sub millimeter). They behave well in all lighting conditions and regardless of object material. They are able to provide additional information such as surface normals, object stiffness, texture, and friction. In contrast, vision and lasers are influenced by lighting and shadows, and do not perform well with glass objects for example. Sonar sensors tend to be considerably less precise than tactile.

Bayesian techniques are the state of the art for acting under uncertainty. These techniques are widely used in mobile robotics [1-3], visual feature tracking [4, 5], and speech recognition [6]. Recently some work has been done in application of Bayesian techniques to tactile perception. In [7], authors have developed a variant of Kalman filter to estimate environment parameters during compliant motions. In [8], histogram filters were used for haptic mapping and localization. They sited that computational costs forced them to restrict the problem to a small 20 x 20 grid. The closest prior art to the problem we are approaching in this paper is [9], where authors have advocated the use of particle filters for object localization using tactile sensors. The localization was restricted to 3 degrees of freedom (DOF), due to computational costs. Using 2 million particles they were able to perform the search in about 1 second. The authors pointed out that the approach would need to be significantly modified to be feasible for 6 DOF localization (i.e. 3 for position and 3 for orientation of the object). This is because the size of search space goes up exponentially with dimensionality. Thus simple-minded extrapolation of the same approach to 6 DOF, would require roughly (2 million)-squared, i.e. 4 trillion particles. Even assuming memory was available to perform operations with that many particles, running time of particle filter is linear in the number of samples. Hence one could expect to accomplish localization in 2 million seconds, i.e. approximately 23 days.

In this paper we consider object localization in 6 DOF. We divide the problem into two cases: over-constrained and under-constrained. In the over-constrained case, the measure-

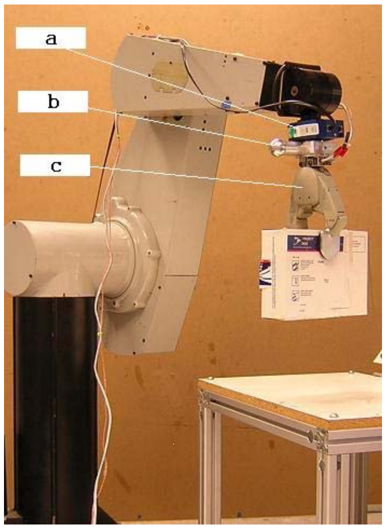


Fig. 1. We used this 6 DOF PUMA robotic manipulator in our box grasping experiments. The manipulator is equipped with (a) 6D JR3 force/torque sensor, (b) robotic finger with a spherical end, and (c) gripper. We used the robotic finger for sensing tasks and the gripper for grasping tasks. The spherical end on the robotic finger simplified contact point estimation during sensing.

ments are sufficient to uniquely identify the solution. In this case, optimization search techniques are applicable and we demonstrate the use of gradient descent for object localization. In under-constrained cases, the measurements do not yield a unique solution. Solutions may form entire regions of non-zero dimensionality. In these cases we explore the use of sampling techniques (such as particle filters and importance sampling) for search. Given high computational costs of search using standard sampling techniques in 6 DOF, we propose a modified sampling technique, we call Scaling Series approach. The search approximates the solution regions by samples and performs a series of successive refinements, scaling granularity from low to high. The technique allows efficient estimation of solution regions in 6 DOF. In addition it estimates the posterior within a small neighborhood of the solution region.

The approaches presented here are easily applicable to any object represented as a poly-mesh. We demonstrate the portability of our techniques by applying them to two different applications. In the first, we localize and manipulate a box. In the second, done as part of the STAIR (STanford AI Robot) project, we localize a door handle, so as to turn the handle and open the door.

In the following section we provide the necessary background. In section III, we address the use of optimization search techniques for over-constrained cases. In section IV, we consider under-constrained cases and present our modified sampling approach. Experimental results are discussed in section V.

II. Background

Consider a simple example of having measurements from 5 different sides of a rectangular box (see Fig. 2). Let us assume that each measurement contains contact position and surface normal. How to best estimate position and orientation of the box from these measurements? A simple approach would be to take averages of normals on opposing sides, then fit orthogonal basis to the resulting normals, then perform best fit of corresponding box faces. This approach will work for a box with 6 sides, but will not generalize to arbitrary polygonal meshes of complex objects or even datasets of larger size taken from the same object.

Bayesian approach provides means of parameter estimation for arbitrary objects and datasets. The measurements are considered as being caused by the world with certain probability, called measurement model $p(Y|X,m)$. Here Y is a measurement consisting of contact position $Y_p = (x_p, y_p, z_p)$ and surface normal $Y_n = (n_x, n_y, n_z)$, X is position and orientation $(x, y, z, \alpha, \beta, \gamma)$ of the object and m is the model of the object (i.e. poly-mesh). Given a set of measurements, D, the goal is to find the most likely state given the measurements and the model. In other words find the maximum likelihood of $p(X|D,m)$ (also known as posterior distribution). In the rest of this paper, we will drop the model m from equations for the sake of brevity, although conditioning on the model will be always assumed.

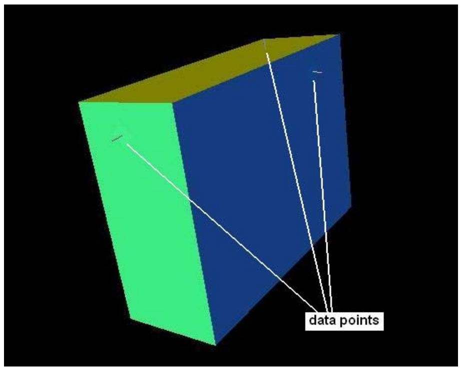


Fig. 2. To estimate position and orientation of a box we take 5 measurements from different sides.

There is a simple relationship between the measurement model and posterior. Applying Bayes rule to the posterior, we obtain:

$$p(X|D) = p(D|X)\frac{p(X)}{p(D)}$$

Here $p(X)$ and $p(D)$ are prior probabilities of state and measurements, that we assume to be uniform. Thus the posterior is proportionate to $p(D|X)$. Furthermore, we assume the measurements to be independent of each other. Hence, if $D = \{Y^{(j)}\}$, the posterior can be written as a product of measurement likelihoods:

$$p(X|D) = \eta \prod_j p(Y^{(j)}|X) \tag{1}$$

This equation is often utilized in localization algorithms.

III. Over-Constrained Localization

First let us consider the case when our dataset, D, is large enough so that only one solution to the localization problem is possible. In this case the posterior is unimodal as it has only one well-defined maximum. For example, if we take 5 measurements from 5 different sides of a rectangular box, then the posterior will be unimodal. In this case, localization is a search problem that can be solved using optimization search techniques. We solve this problem using gradient descent.

A. Measurement Model

Since our object model is represented by a polygonal mesh consisting of faces $\{f_i\}$, we model the measurement likelihood as follows. For each face, f_i, we compute the likelihood of the measurement being caused by that face (and a given state X). Then we take maximum likelihood over all faces as the likelihood of the measurement. For convenience, let us introduce correspondence variables $\{c_i\}$. We will assume that $c_i = 1$ when face f_i has caused the measurement, and $c_i = 0$ otherwise. When conditioning, we will write c_i as a shorthand for $c_i = 1$. Thus our measurement model is defined by

$$p(Y|X) = \lambda \max_i\{p(Y|X,c_i)\}, \tag{2}$$

where λ is the normalizing factor given by

$$\lambda = \frac{1}{\int \max_i\{p(Y|X, c_i)\} dY}.$$

Since we do not impose any limitations on the measurement space, λ is independent of the state X. In practice we never need to compute numeric value of this factor as it is taken care of during normalization step.

Recall that each measurement, Y, consists of two parts: contact position, Y_p, and surface normal, Y_n. When computing how likely a measurement to be caused by a face f_i, we consider the two parts of measurement to be independent. We use state parameters X to transform the measurement into the coordinate system of the object and denote transformed measurement components Y_p^X and Y_n^X respectively. Thus equation 2 becomes:

$$p(Y|X) = \lambda \max_i\{p(Y_p^X|c_i)\ p(Y_n^X|c_i)\}$$

We assume Gaussian distribution for the two likelihoods, with variance err_p^2 and err_n^2 respectively. Thus, the likelihoods can be computed as follows:

$$\begin{aligned} p(Y_p^X|c_i) &= \frac{1}{\sqrt{2\pi}\ err_p} \exp\{-\frac{1}{2}\frac{d(Y_p^X, f_i)^2}{err_p^2}\} \\ p(Y_n^X|c_i) &= \frac{1}{\sqrt{2\pi}\ err_n} \exp\{-\frac{1}{2}\frac{||Y_n^X - normal(f_i)||^2}{err_n^2}\} \end{aligned}$$

Here $d(Y_p^X, f)$ is the shortest Euclidean distance from Y_p^X to face f_i.

B. Cost Function

Due to the negative exponentiation involved in the measurement model, it is convenient to optimize negative log of the posterior. Thus we search for the global minimum of the following cost function:

$$F = -\log(p(X|D))$$

Substituting from equation 1 we obtain:

$$F = -\log \eta \prod_j p(Y^{(j)}|X) = C - \sum_j \log p(Y^{(j)}|X)$$

For a given state X, let τ_j be the index of the face that is most likely to cause measurement $Y^{(j)}$. From our measurement model equations we obtain

$$F = C - \sum_j \log \lambda p(Y_p^{(j)X}|c_{\tau_j})\ p(Y_n^{(j)X}|c_{\tau_j})$$

Substituting in definitions for H_p and H_n and subsuming constants into C, we obtain:

$$F = C + \frac{1}{2}\sum_j \frac{d(Y_p^{(j)X}, f_{\tau_j})^2}{err_p^2} + \frac{||Y_n^{(j)X} - normal(f_{\tau_j})||^2}{err_n^2}$$

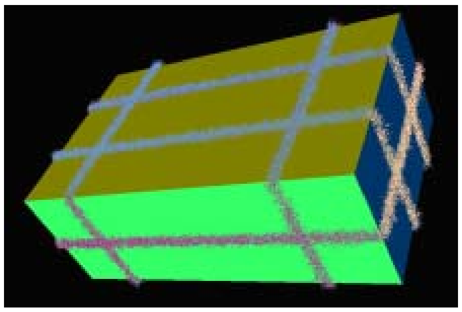


Fig. 3. Possible solutions form several intersecting rims for a dataset consisting of two measurements taken from two non-parallel sides of the box.

Finally, we remove uninteresting constants and obtain the following cost function:

$$F = \sum_j \frac{d(Y_p^{(j)X}, f_{\tau_j})^2}{err_p^2} + \frac{||Y_n^{(j)X} - normal(f_{\tau_j})||^2}{err_n^2}$$

It is difficult to differentiate F in closed form because of $argmax$ involved for computation of τ_j and dependence on the shape of object faces. Therefore we performed numerical differentiation for the purposes of gradient descent.

Optimization search is prone to local minima. In order to find global minimum, we run gradient descent many times from random starting points until we obtain a sufficiently low value. We discuss performance and estimation precision in the experimental results section.

IV. UNDER-CONSTRAINED LOCALIZATION

So far we have considered the case where there is a unique solution to the localization problem. However, often there are many placements of the object that fit the dataset. In some cases there could be several solutions, in others there could be infinitely many solutions. For example, figure 3 shows the set of possible solutions for the case when two measurements were collected from two non-parallel sides of a rectangular box. In this case the set of solutions is a one dimensional region (a collection of intersecting rims).

In these cases it is useful to estimate the solution region, because this information can be used to make decisions on where to sense next. Sampling and gridding techniques have been widely used for estimation in multi-modal scenarios. For example in [9], the authors used a particle filtering technique for a similar box localization problem. Since the problem is static (i.e. the object does not move), importance sampling is another viable alternative. Splitting up the search space into grids deterministically or semi-deterministically is also possible, although gridding methods tend to have more implementation overhead. In this paper we present our methods based on particle filters, however we have also implemented Scaling Series with importance sampling and found the performance to be comparable. Gridding methods could also be substituted in even though we have not tried to implement that variation.

As we pointed out earlier, the main difficulty with sampling techniques in 6 DOF search space is that the number of samples required for precise estimation explodes exponentially with the space dimensionality. Large numbers of particles lead to computation times that are unacceptable. On the other hand the problem with using fewer particles is that uniform sampling is extremely unlikely to produce any samples near the actual solution. For example, suppose we are performing localization in 40 cm x 40 cm x 40 cm x 360 degrees x 360 degrees x 360 degrees space, with desired deviation of 1mm and 1 degree respectively. If we consider the 6-D sphere around the solution with radius of 1 desired deviation, the volume of this sphere is 3 quadrillion times smaller than the volume of the search space. If we utilize 1,000 particles, we are very unlikely to sample one within desired deviation of the solution.

A. Representing Regions of Space with Particles

Usually each particle is seen as a point in search space, but let us consider what happens if each particle represents a region of search space. For a parameter δ, we will call the 6-D sphere with radius δ around a particle a δ-sphere. We will think of each particle as representing the entire region within its δ-sphere. If δ is large, it is clearly easy to cover the search space with even a small number of particles. For example, if δ is larger than the diameter of the search space, one particle would suffice.

Now that our particles are regions of space, we need to understand how to apply the measurement model to compute particle likelihood weights. To parameterize the measurement model relative to δ, we simply update the measurement error based on δ. We set:

$$(err_p, err_n) \leftarrow (\delta, r\delta),$$

where r is the ratio between actual position and normal measurement errors.

The above equations amount to "pretending" that measurement noise is inflated to be δ. Artificially inflating measurement noise is not an uncommon practice in application of particle filters, see for example [1]. This technique allows for particles to survive better by making the likelihood weights less discriminative.

B. Scaling Series Approach

We explore the performance of particle filters depending on the value of δ on simulated datasets. For a spectrum of values for δ, we ran 100 particle filters with 1,000 samples each. The results are presented in Fig. 4. As we can see, when δ is small (less than 1 cm), the precision of estimates falls drastically and variance rises. Thus in effect we are getting random guesses of the correct state. We did not include results for δ settings below 4 mm, because at these settings many experiments fail as all particles end up with zero likelihood weights. When δ is 1 mm, the variance used in the measurement model corresponds to the actual noise variance. At this setting all 100 experiments failed. On the other side of the spectrum, for large values of

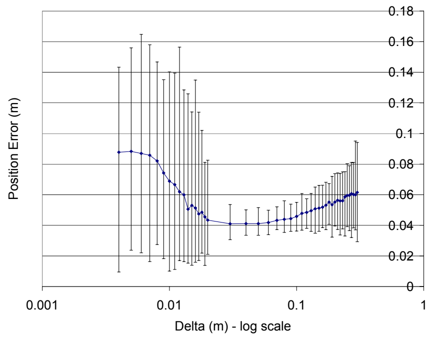


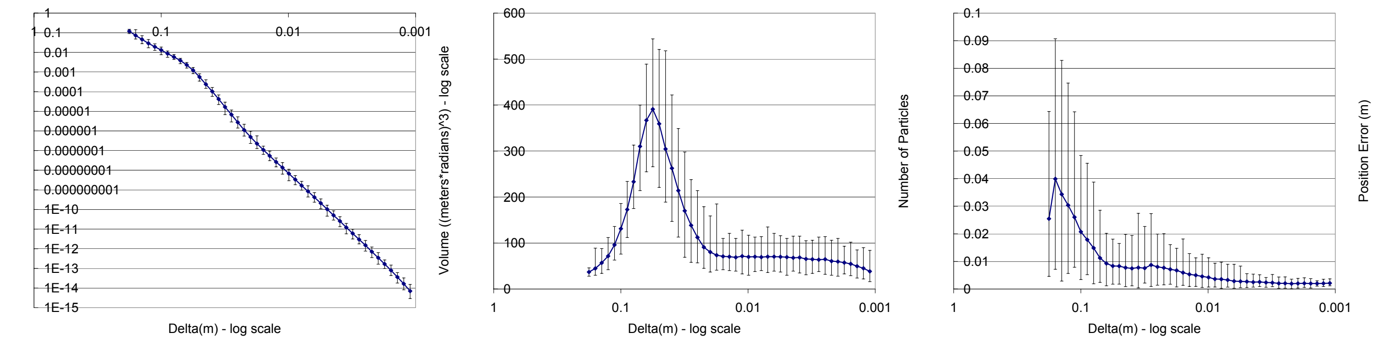
Fig. 4. Performance of standard particle filter for varying settings of δ. Standard particle filter was ran 100 times for each setting of δ on simulated dataset. The resulting position estimation error is plotted in the graph. The vertical bars represent absolute min/max values during all 100 experiments.

δ precision gradually falls as the measurement model is not discriminative enough to distinguish between good and bad guesses effectively. We note that even for the best settings (of 1 to 5 cm), the estimates were 4 cm off on average, which is not an acceptable precision of estimation for manipulation tasks.

Since no one setting of δ gives good results, we propose to run a series of successive particle filters, reducing the value of δ from one filter to the next. The intuition behind this approach is that the first filter in the series finds regions of high likelihood at a very coarse resolution. The next filter focuses the search in the smaller subspace found by the previous filter, but at a finer resolution. In this manner, we can keep reducing δ until it corresponds to the actual noise variance. Thus, the last filter will approximate true posterior although over a small subspace of the initial search space. This intuition is formalized in the next subsection into an algorithm we call Scaling Series Particle Filter.

An illustration of the approach is given in Fig. 5. It represents results of 100 runs of the Scaling Series algorithm on realistic simulated data for box localization. The progression of the series is from left to right, with corresponding δ values noted on the horizontal axis (in log scale). It clearly shows how the volume of the search space shrinks drastically as the series progresses. At the same time estimation errors fall with the series progression. The number of samples remained small throughout all of the experiments, with the absolute maximum being below 600. The number of particles is highest for δ values between 5 and 10 cm. At these settings the distribution is multi-modal, corresponding to 6 possible sides of the box. As these possibilities are ruled out, the number of particles goes down.

We note that gradually reducing the value of δ during the series progression, changes the measurement model from less discriminative initially to more discriminative towards the end of the series. This technique is a variant of annealing, which has been used in other settings for particle filters. See for example [10], where the authors applied an annealing filter to articulated motion capture from vision data.



(a) Reduction of search space volume.

(b) Number of particles utilized.

(c) Error of position estimate.

Fig. 5. Performance of Scaling Series Particle Filter on simulated dataset during 100 experiments. Each graph shows progression of the series from left to right. Corresponding value of δ is noted on the horizontal axis. Vertical bars represent absolute min/max values during all 100 runs.

C. Algorithm Details

The algorithm consists of a series of particle filters. We start by running a filter with a large value of δ (i.e. radius of initial search space V). After a complete pass over the dataset D, the filter produces a set of particles concentrated in the region of high likelihood. This region, denoted V_1, is the union of δ-spheres around the particle set. Since V_1 is smaller than the original search space, we can cover it with smaller particles. Thus we reduce the value of δ and run a second particle filter, but this time restrict our search to V_1. The second particle filter produces a new search space, V_2, that represents region of highest likelihood for this setting of δ. We repeat the process until we reach the desired value for δ, corresponding to desired precision. Refer to Alg. 1 for a complete listing of the algorithm.

We select the scaling factor, $zoom$, so that the volume of δ-sphere is halved during scaling (line 2). We also take care to maintain a healthy density of particles in each search space. This is controlled by desired number of particles per δ-sphere, M.

```
   Scaling_Series_PF($V_0, M, D, \delta_{desired}$)
1: $\delta \leftarrow radius(V_0)$
2: $zoom \leftarrow 1/\sqrt[n]{2}$
3: $T \leftarrow log_2(Vol(V_0)/Vol(S_{\delta_{desired}}))$
4: for $t = 1$ to $T$ do
5:     $\{X_i\} \leftarrow$ Uniform_Sample_From_Subspace($V_{t-1}, M$)
6:     $\{X_i\} \leftarrow$ Standard_PF($\{X_i\}, D$)
7:     $V_t \leftarrow$ Union_Delta_Spheres($\{X_i\}, \delta$)
8:     $\delta \leftarrow zoom \cdot \delta$
9: end for
```

Alg. 1: Scaling Series Particle Filter

During each run of the standard particle filter, the likelihood weights are parameterized by δ as described above. Since the process is stationary, we add random white noise during mutation step in standard particle filter. We set the mutation parameter to be $\delta/\sqrt{|D|}$. This is because each standard PF invocation performs a total of $|D|$ mutations. This way the cumulative mutation distribution has a deviation of δ.

At each iteration t we consider our search space V_t to be the union of δ-spheres centered around the current particle set $\{X_i\}$. Thus, we need an algorithm for sampling uniformly from V_t at the start of each standard filter run. It is important for each particle filter in the series to be initialized with a uniform spread of particles over its search space. This prevents from compounding the error from one filter to the next and allows for better tracking of multi-modal solutions. Many implementations of uniform sampling are possible. For example one could approximate V_t with small grids and sample directly from the grids. Alternatively one could sample from existing particles in inverse proportion to local particle density (this was the method we used in our implementation). One version of uniform sampling algorithm was suggested by an anonymous reviewer. Since it is the simplest algorithm we have come across, we provide its listing here (Alg. 2). It is based on rejection sampling.

```
   Uniform_Sample_From_Subspace($V, M$)
1: // space $V$ is represented as union of spheres $\{S_i\}$
2: $X \leftarrow \{\}$
3: for $i = 1$ to $|\{S_i\}|$ do
4:     for $j = 1$ to $M$ do
5:         sample point $x$ from $S_i$
6:         reject $x$ if it is in union of $S_1 \ldots S_{i-1}$
7:         otherwise add $x$ to $X$
8:     end for
9: end for
```

Alg. 2: Uniform Sampling from Subspace

V. Experimental Results

We utilized polygonal models of objects. These models were constructed by hand from measurements taken with a ruler, but they could be obtained by other means. For example one could use tri-meshes from stereo vision or other range sensors. Each model also included optimal grasping points determined by a human. Once localization is performed, grasping configuration is derived from the estimated parameters.

We implemented our localization techniques in Java running on 1.2GHz laptop computer. We then applied our approach to two different problems: locating and picking up a box and manipulating a door handle.

A. Application 1: Locating and picking up a box

We applied our approach to the task of localizing, grasping and picking up a rectangular box (see Fig. 6). The manipulator used was a 6 DOF PUMA robot, equipped with a 6D JR3 force/torque sensor (see Fig. 1). Its end-effector included a gripper and robotic finger configuration. To simplify contact point estimation, tactile sensing was performed with the robotic finger that had a spherical end.

For the over-constrained scenario, a simple active sensing procedure (specific to the box) probed 5 different sides of the box recording contact position and surface normal for each data point. Care was taken to make sure the box did not move during sensing as it would introduce considerable noise into measurements.

The model of the box was constructed by hand from measurements taken with a ruler. Two grasp points were manually defined on the model. Each grasp point consisted of 3 points: one for each side of the gripper and one for wrist position. Thus each grasp point fully defined position and orientation of the gripper. After localization, the grasp point with the highest Z-coordinate was selected (Z-coordinates increase vertically upwards). The gripper orientation, position and approach vector were derived from the selected grasp point and estimated parameters. Note the precise fit required for grasping in Fig. 6(b).

The localization was performed in a 40cm x 40cm x 40cm area with unrestricted orientation (i.e. 360 x 360 x 360 degrees). Desired precision was set to 1 mm for position and 2 degrees for orientation. Sensing procedure took 30 seconds. Localization was performed in less than 1 second on our laptop computer with a 1.2GHz processor. We performed 30 trials on the real robot. In our experiments, localization and grasping had 100% success ratio on completed datasets. The active sensing strategy had a 70% success ratio. Failures during sensing were due to hardware issues and motion of the object. We performed localization using SSPF during real robot trials.

We also performed a number of simulated trials, where ground truth was easily available for evaluation of localization success and precision. We performed 1,000 simulated trials using gradient descent and 1,000 trials using SSPF. As described before, we run gradient descent multiple times from random starting locations until we obtain a solution with cost function below a threshold. Our threshold corresponded to cost of a solution that placed all measurements within 3 standard deviations of measurement noise. Only 16% of gradient descent runs found the global minimum, while the rest settled in local minima. On average 5 seconds of gradient descent runs resulted in a solution with cost below the threshold. 99.8% of simulated SSPF trials found the solution successfully and had an average running time of about 1 second. Since the object to be localized was symmetric, we added symmetry compensation to rule out symmetric solutions. This allowed for easy automatic identification of correct localization results. Gradient descent gave slightly more precise results. Average precision of gradient descent was 1.9mm, while average precision of SSPF was 2.1mm over the 1000 simulated trials.

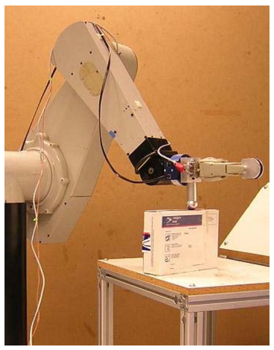
We note that our experiments were performed on a relatively simple object, consisting of only 6 faces. For more complex meshes, measurement likelihood evaluation will be linear in the number of faces. However, it is possible to implement efficiency improvements that only consider a subset of faces during measurement likelihood evaluation. Additionally, we expect that gradient descent will have lower percentage of success on more complex objects, as there will be more local minima in the cost function. Experimentation with more complex objects is a possible direction for future work.

We also performed experiments for under-constrained scenarios. In this case the datasets consisted of 2 - 3 measurements from different sides of the box. For real robot experiments, we took subsets of measurements from our completed real robot trials. We verified that the estimated region included the true state of the object, as it was estimated from complete datasets. We also examined the estimated region visually to make sure it corresponded to the correct solution region in each under-constrained scenario (Fig. 7). In addition, we performed 100 simulated trials where ground truth was available. The true state was included in resulting solution set in all 100 trials.

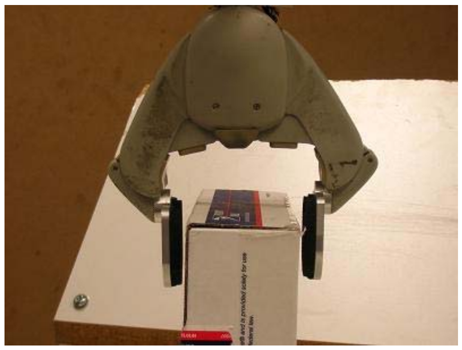
Since the solution region is broad, high precision settings result in large numbers of particles. For example for a dataset consisting of two measurements, SSPF generated 1,000 particles for δ setting of 5cm, 7,000 particles for δ setting of 6mm and 21,000 particles for δ setting of 2mm. The running time increases with the number of particles generated. Operations with a few thousand particles take a few seconds, but 20,000 particles take 40-50 seconds to process. Thus it is possible to trade off precision of estimation for running time. As more measurements arrive, the solution region shrinks and higher precision can be achieved with fewer particles.

B. Application 2: Manipulating door handles for building navigation

In a second application, we carried out experiments with door handle manipulation as part of the STanford AI Robot (STAIR) project. The goal of the STAIR project is to build a robot capable of performing a broad range of tasks in home and office environments. Over the long term, the envisioned tasks include fetching a book from an office, showing guests around a research lab, tidying up after a party, and using tools to assemble a bookshelf. In order to carry out these tasks, STAIR will need to navigate home and office environments, which means being able to open doors. We will do this by accurately localizing, and then manipulating, the door handle.

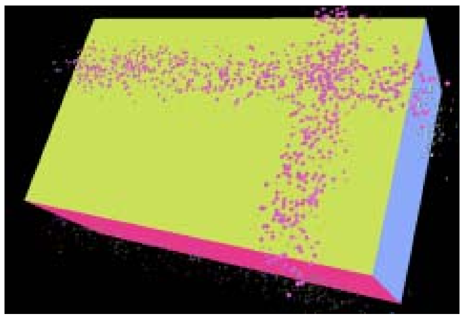


(a) sensing

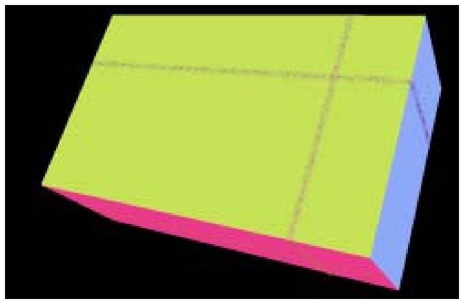


(b) grasping

Fig. 6. The stages of our box manipulation experiment. (a) Sensing the box with a robotic finger. (b) Grasping the box using grasping configuration defined as part of the box model. Note the precise fit required to perform the grasp.

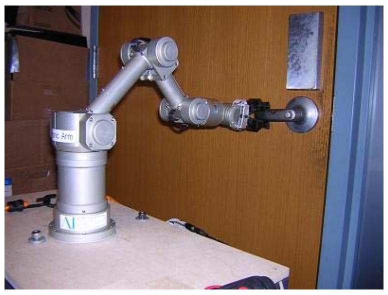


(a) 11mm precision

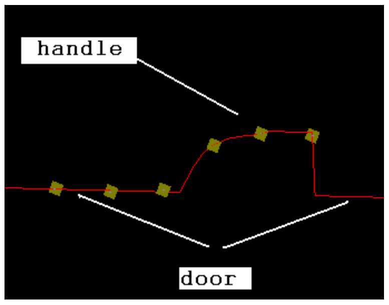


(b) 1mm precision

Fig. 7. Examples of under-constrained solution estimation for datasets consisting of 2 measurements (includes symmetry compensation). (a) With δ setting of 11mm, 4,000 particles were generated by SSPF (b) With δ setting of 1mm 29,000 particles were generated.



(a)



(b)

Fig. 8. (a) A 5 DOF Harmonic Arm 6M manipulator performing one of our door handle grasping experiments. (b) The 2D model of a door handle was constructed from depth measurements made with a ruler every 1cm along the length of the handle. The squares represent data points from one of our experiments.

Once the robot navigates to the area in front of a door (using its laser sensors for approximate localization), we use tactile feedback to accurately estimate the position and orientation of the door handle. We performed experiments on a 5 DOF Harmonic Arm 6M manipulator, which has about 1mm end-effector positioning precision. (See Fig. 8(a).) The height of the handle as well as 2 orientation angles were fixed, which reduces the localization task to a 3 DOF problem. Our algorithm used a 2D model of the door that was constructed by hand using ruler measurements. Specifically, we took door handle depth measurements every 1cm along its length in a horizontal plane through the center of the handle. This gave a 2D model consisting of line segments (see Fig. 8(b)). For the door used in these experiments, we decided to grasp the handle at the endpoint of one of the line segments in the model. The sensing used in this experiment gave only position measurements, and did not include surface normals.

For each experimental trial, the STAIR robot takes 6 measurements in a 30 degree span (at $0^\circ, 6^\circ, \ldots, 30^\circ$). Each data point thus consisted of range to the contact point and an orientation angle. The sensing procedure took between 1 and 2 minutes; using these six measurements, our algorithm is then able to localize the door handle in a fraction of a second. In these experiments, we restricted the dimensions of the search space (to 6cm x 6cm x 30 degrees) because of the limited operational range of the manipulator. Out of 100 independent trials, our algorithm successfully completed the sensing in 98 trials. In all of these 98 trials, our algorithm then successfully localized, grasped, and turned the door handle, and opened the door. The two failures during sensing were caused by a hardware glitch in communication with the robot.

VI. Conclusions

Sensory perception is vital for robots performing autonomous object manipulation in uncertain environments. Bayesian estimation of all six parameters for position and orientation of objects from tactile data has been known to be computationally expensive ([9]). We split the problem into two cases: over-constrained and under-constrained. In the over-constrained case, measurements identify the solution uniquely. In the under-constrained case, multiple solutions are possible up to entire regions of non-zero dimensionality. We solved over-constrained cases using optimization search. We also proposed an efficient sampling approach, termed Scaling Series Particle Filter, which works in both over-constrained and under-constrained cases. SSPF approximates the solution region by samples. It performs search by successively refining the search region and scaling granularity of search from low to high.

Our approach does not utilize any special properties of the manipulated objects and can be easily applied to any object represented as a polygonal mesh. We have demonstrated its portability by applying it to two different tasks: manipulating a box and grasping a door handle.

For over-constrained cases, real time performance has been achieved in our experiments. For under-constrained cases, running time depends on the precision desired and size of the solution region. However, it is possible to trade off precision of estimation for running time. Coarse estimates can be obtained quickly when few measurements are available. As more measurements arrive, the solution region shrinks and so more precise estimates can be obtained in a timely fashion.

We note that running time depends linearly on the complexity of objects (i.e. number of faces in the mesh model). It is possible however, to implement efficiency improvements that only consider a small subset of faces during each measurement evaluation. In addition, we expect that average running time of optimization search will go up for more complex objects, due to the fact that cost function will have more local minima.

In future we would like to work with more complex objects and develop a generic active sensing procedure that is capable of collecting datasets with high efficiency and reliability.

Acknowledgments

We are grateful to Jaeheung Park, Irena Paschenko, Dongjun Shin, and Peter Thaulad for their help with the PUMA robot. We give warm thanks to Gary Bradski, Jimmy Zhang, David Li, Ashutosh Saxena, Jamie Schulte, and Francois Conti for help with the STAIR robot. Our sincere thanks to David Stavens, Daphne Koller, Luis Sentis, and Vincent De Sapio for their feedback on earlier versions of this paper. We also thank anonymous reviewers for their insightful comments and one anonymous reviewer in particular for proposing a simpler version of uniform sampling algorithm presented in this version of the paper.

References

[1] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient position estimation for mobile robots. In Proc. of the National Conference on Artificial Intelligence, 1999.

[2] S. Lenser and M. Veloso. Sensor resetting localization for poorly modeled mobile robots. In Proc. of the IEEE International Conference on Robotics & Automation, 2000.

[3] P. Jensfelt, O. Wijk, D. Austin, and M. Andersson. Feature based condensation for mobile robot localization. In Proc. of the IEEE International Conference on Robotics & Automation, 2000.

[4] J. Vermaak, A. Doucet, and P. Perez . Maintaining Multi-Modality Through Mixture Tracking. Int. Conf. on Computer Vision, pages 1110-1116, 2003.

[5] A. Lookingbill, D. Lieb, D. Stavens, and S. Thrun. Learning activity-based ground models from a moving helicopter platform. To be published in Proceedings of the International Conference on Robotics and Automation (ICRA 2005), 2005.

[6] J. Vermaak, C. Andrieu, A. Doucet, and S.J. Godsill. Particle methods for Bayesian modeling and enhancement of speech signals. IEEE Transactions on Speech and Audio Processing, 2002.

[7] P. Slaets, J. Rutgeerts, K. Gadeyne, T. Lefebvre, H. Bruyninckx, and J. De Schutter. Construction of a Geometric 3-D Model from Sensor Measurements Collected during Compliant Motion. In Proceedings of the International Symposium on Experimental Robotics, Singapore, Singapore, june 2004. Springer Tracts in Advanced Robotics (In press).

[8] M. Schaeffer and A. M. Okamura, "Methods for Intelligent Localization and Mapping during Haptic Exploration," 2003 IEEE International Conference on Systems, Man and Cybernetics, pp. 3438-3445.

[9] K. Gadeyne and H. Bruyninckx. Markov techniques for object localization with force-controlled robots. In Proceeding of the 10th Int. Conf. on Advanced Rob., Budapest, Hungary, Aug 2001.

[10] Jon Deutscher, Andrew Blake and Ian Reid. Articulated Body Motion Capture by Annealed Particle Filtering. Proc. Conf. Computer Vision and Pattern Recognition (CVPR), 2000.