MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Entropy-Based Motion Selection for Touch-Based Registration Using Rao-Blackwellized Particle Filtering

Taguchi, Y.; Marks, T.K.; Hershey, J.R.

TR2011-067 September 25, 2011

Abstract

Registering an object with respect to a robot's coordinate system is essential to industrial assembly tasks such as grasping and insertion. Touch-based registration algorithms use a probe attached to a robot to measure the positions of contact, then use these measurements to register the robot to a model of the object. In existing work on touch-based registration, the selection of contact positions is not typically addressed. We present an algorithm for selecting the next robot motion to maximize the expected information obtained by the resulting contact with the object. Our method performs 6-DOF registration in a Rao-Blackwellized particle filtering (RBPF) framework. Using the 3D model of the object and the current RBPF distribution, we compute the expected information gain from a proposed robot motion by estimating the expected entropy that the RBPF distribution would have as a result of being updated by the proposed motion. The motion that provides the maximum information gain is selected and used for the next measurement, and the process is repeated. We compare various methods for estimating entropy, including approximations based on kernel density estimation. We demonstrate entropy-based motion selection in fully automatic and human-guided registration, both in simulations and on a real robotic platform.

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Entropy-Based Motion Selection for Touch-Based Registration Using Rao-Blackwellized Particle Filtering

Yuichi Taguchi, Tim K. Marks, and John R. Hershey
Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA
{taguchi, tmarks, hershey}@merl.com

***Abstract*— Registering an object with respect to a robot's coordinate system is essential to industrial assembly tasks such as grasping and insertion. Touch-based registration algorithms use a probe attached to a robot to measure the positions of contact, then use these measurements to register the robot to a model of the object. In existing work on touch-based registration, the selection of contact positions is not typically addressed. We present an algorithm for selecting the next robot motion to maximize the expected information obtained by the resulting contact with the object. Our method performs 6-DOF registration in a Rao-Blackwellized particle filtering (RBPF) framework. Using the 3D model of the object and the current RBPF distribution, we compute the expected information gain from a proposed robot motion by estimating the expected entropy that the RBPF distribution would have as a result of being updated by the proposed motion. The motion that provides the maximum information gain is selected and used for the next measurement, and the process is repeated. We compare various methods for estimating entropy, including approximations based on kernel density estimation. We demonstrate entropy-based motion selection in fully automatic and human-guided registration, both in simulations and on a real robotic platform.**

I. Introduction

Industrial robots require accurate registration to a world or object coordinate system in order to perform assembly tasks such as grasping and insertion. A classical but extremely labor-intensive approach is to manually teach the robot every pose that is required for the assembly task.

Touch-based registration methods [1]–[5] enable automatic registration of a robot to an object with known 3D shape by using a probe attached to the robot to measure several positions of contact with the object. Fig. 1 shows our system setup for touch-based registration. Successful registration enables the robot to perform any task at any positions defined in the object coordinate system.

Previous approaches do not address the selection of touch positions. In this paper, we consider optimizing the touch positions to improve registration accuracy as a function of the number of touches. We present an algorithm for selecting the next robot motion in order to achieve this goal. The basic idea, inspired by active localization algorithms commonly used in mobile robotics [6], is to consider a set of candidate motions and choose the one that maximizes the expected information obtained by the resulting contact with the object. Our method for full six-degrees-of-freedom (6-DOF) registration represents the relative pose between robot and object using a Rao-Blackwellized particle filtering [7] framework: We represent the 3D position of the probe with respect to the object as a set of particles, and define a 3D Gaussian angle distribution for each particle. Using the current particle distribution and a 3D model of the object, we compute the expected information gain from a proposed robot motion by estimating the expected entropy that the RBPF distribution would have after being updated by the proposed motion. The candidate motion that yields the minimum expected entropy is used for the next motion and measurement. We consider multiple methods for estimating the entropy of the RBPF distribution, including approximations using weighted kernel density estimation, and analyze the strengths and weaknesses of each. We test our entropy-based motion selection algorithm in simulations and on a real robotic platform, for both fully automatic and human-guided registration.

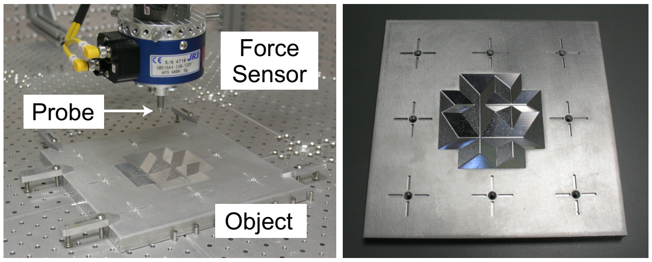

Fig. 1. *Left:* Overview of our system. We perform robot-to-object registration by using a probe attached to a robot arm to measure the positions of contact with an object that has a known 3D shape. *Right:* Close up of the metal object we used in experiments.

A. Related Work

Online Touch-Based Registration: Chhatpar and Branicky presented a registration method using probing and particle filtering for lock-key assembly [1] and peg-in-hole problems [2]. As a preprocessing step, they exhaustively probe every (x, y) location on the object with the peg to obtain a contact configuration-space map, which describes all possible transformations in which the peg has a contact with the object. Using the obtained map, they perform particle filtering by sequentially probing the object and using the contact positions as observations. Thomas et al. [3] used as an object model a force/torque map, consisting of contact force and torque at every possible contact pose. This map was generated as a preprocessing step by probing the object using a probe with a force/torque sensor, or using a physics-based estimate computed from CAD models of the probe and object. They also used particle filtering to match each force/torque observation to the map. Because these previous methods use particle filtering for all dimensions, they are not well-suited to full 6-DOF registration, since the number of particles required for standard particle filtering increases roughly exponentially with the number of dimensions in the search space. Although the formulations of these methods were described for 6-DOF uncertainty, in practice they were only used for localization in lower-dimensional search spaces (2- or 3-DOF).

To reduce the number of particles for 6-DOF registration, Petrovskaya et al. [4] presented a coarse-to-fine approach by assigning a region (instead of a single point) to each particle. They first determine regions of high likelihood at a coarse resolution by artificially inflating the measurement noise, then iteratively resample particles inside the regions while reducing the noise level. In [5], we used Rao-Blackwellized particle filtering (RBPF) for the 6-DOF registration problem. The probability distribution is factorized into a 3-DOF position distribution, represented using particles, and a 3-DOF angular distribution, approximated by a Gaussian distribution conditioned on the position of each particle. This factorization's analytical representation of angular uncertainty reduces the number of dimensions to be represented using particles, greatly reducing the number of particles needed.

None of these previous methods address how to select probing locations. Although [4] described a simple active sensing procedure, it was specific to rectangular shapes. In this paper, we develop an entropy-based motion selection algorithm based on the RBPF framework we presented in [5] that can be used for any shape of object that is represented as a piecewise planar mesh (e.g., a CAD model). In addition to using our algorithm to improve the accuracy and reduce the number of measurements required for the fully automatic registration framework used in [5], we also present a novel application of our method to human-guided registration.

Offline Registration: Given a set of point measurements and a 3D model of an object as a set of planes, several registration algorithms have been proposed. Methods such as [8], [9], which are based on a minimal number of correspondences, are suitable for use as hypothesis generators in hypothesize-and-test frameworks such as RANSAC. Olsson et al. [10] presented a method based on branch and bound that uses point-to-plane correspondences to achieve globally optimal registration. Such globally optimal registration is difficult for iterative closest point (ICP) algorithms, which require good initialization [11].

Although these offline methods provide solutions to the point-plane registration problem, they assume that a complete set of contact measurements has already been collected. They provide no method for selecting a good next motion based on an already-collected subset of contact measurements, nor do they provide a termination condition for determining when enough measurements have been collected.

Motion Selection: The general problem of motion selection (planning where to touch next) has not previously been addressed in touch-based registration. However, motion selection is well studied for exploration in mobile robotics [6], such as in active simultaneous localization and mapping (active SLAM) applications in which a mobile robot plans where to move next in order to improve its location estimate as well as its map of the environment [12]–[16]. In mobile robotics, selecting where to move is often addressed using entropy-based measures [12]–[15] such as expected information gain (or expected reduction in entropy). In this paper, we demonstrate that an entropy estimation method that is typically used in mobile robotics, computing the entropy of a particle distribution using a Gaussian approximation [12], is not well suited to touch-based registration, and we propose alternative entropy estimation methods that prove to be more effective in this application.

II. Touch-Based 6-DOF Registration Using Rao-Blackwellized Particle Filtering

This paper builds upon our method for touch-based registration using RBPF [5], which we briefly summarize here. Our problem is to find the pose of a needle-like probe with respect to an object by measuring 3D contact positions between them. The 6-DOF uncertainty between the probe and the object is represented as $(\boldsymbol{s}, \boldsymbol{\theta})$, where $\boldsymbol{s} = (x, y, z)^{\mathrm{T}}$ and $\boldsymbol{\theta} = (\alpha, \beta, \gamma)^{\mathrm{T}}$ are relative positions and angles, respectively, between the probe and the object. Given the sequence of motion commands (defined in the robot coordinate system) from time 1 to t, $\boldsymbol{u}_{1:t}$, and the sequence of observations, $z_{0:t}$, our goal is to infer the posterior distribution over the position and angle, $p(\boldsymbol{s}_{0:t}, \boldsymbol{\theta}|z_{0:t}, \boldsymbol{u}_{1:t})$.

The key to our approach is to factorize the probability distribution separately into positions and angles as follows:

$$p(\boldsymbol{s}_{0:t}, \boldsymbol{\theta}|z_{0:t}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{s}_{0:t}|z_{0:t}, \boldsymbol{u}_{1:t})p(\boldsymbol{\theta}|\boldsymbol{s}_{0:t}, z_{0:t}). \quad (1)$$

The probability distribution over position is represented by particles, which enables us to represent multimodal distributions for position. The probability distribution over 3D rotation angle is represented as a Gaussian distribution for each particle, conditioned on the current and previous positions of the particle. Equation (1) can be efficiently solved, since the distribution over angle $p(\boldsymbol{\theta}|\boldsymbol{s}_{0:t}, z_{0:t})$ can be estimated using extended Kalman filtering (EKF) once the position of each particle $\boldsymbol{s}_{0:t}$ is determined by particle filtering. More specifically, we first update the position of each particle using the motion model, then compute the particle's weight using the measurement model and the particle's previous Gaussian distribution over angle. We then update the particle's Gaussian estimate of angle using EKF.

III. Entropy Estimation for Motion Selection

Unlike in SLAM, in which both the map and the location are estimated over time, in our touch-based registration application the map (the object model) is already known, and only the location (relative pose between robot and object) is unknown. Nonetheless, our use of the Rao-Blackwellized particle filter (RBPF) is analogous to its use in SLAM applications. In RBPF approaches to SLAM, the robot location is represented using particles, and the map is represented as a continuous distribution (typically Gaussian) for each particle. Similarly, in our RBPF registration framework (described in more detail in [5]), the relative 3D position of the robot with respect to the object is represented using particles, and the relative 3D angle of the robot with respect to the object is represented using a Gaussian distribution for each particle.

In [6], [12], the expected utility of various possible motions for SLAM is determined by estimating the expected information gain, or decrease in entropy, of the filtering distribution of an RBPF. The motion selected is the one that minimizes the expected entropy of the RBPF, which can be seen as minimizing the uncertainty over both the robot location and the map of the environment. In our approach to active localization, we also choose from several candidate motions by minimizing the expected entropy of our RBPF filtering distribution (1), which can be seen as minimizing the uncertainty over both the relative position and the relative angular orientation between the object and the robot.

However, we demonstrate that the entropy estimate used successfully in the SLAM scenarios of [6], [12], based on a Gaussian approximation of the particle distribution, is not the best choice for the present application. We contrast this Gaussian approximation with other entropy estimation methods that produce better results in the present scenario: an approximation based only on particle weights, and non-parametric entropy estimates based on weighted kernel density estimators.

A. Entropy of a Rao-Blackwellized Particle Filter

The differential entropy of the posterior distribution (1) of the RBPF can be represented as follows [6], [12]:

$$\begin{aligned} &H(p(\boldsymbol{s}_{0:t}, \boldsymbol{\theta} | z_{0:t}, \boldsymbol{u}_{1:t})) = \\ &\qquad H(p(\boldsymbol{s}_{0:t} | z_{0:t}, \boldsymbol{u}_{1:t})) + \mathrm{E}_{\boldsymbol{s}_{0:t}}[H(p(\boldsymbol{\theta} | \boldsymbol{s}_{0:t}, z_{0:t}))], \end{aligned} \tag{2}$$

where $\mathrm{E}_{\boldsymbol{s}_{0:t}}[\,]$ denotes expectation with respect to the posterior distribution $p(\boldsymbol{s}_{0:t} | z_{0:t}, \boldsymbol{u}_{1:t})$. Note that in this paper, we use the term *entropy* to mean differential entropy when referring to continuous distributions. The first term in (2) represents the entropy of the position distribution, which is represented using particles, and the second term represents the entropy of the angular distribution. We describe the computation of the second term in (2), which is relatively straightforward, in Section III-B.

To compute the first term in (2), we need to estimate the entropy of a continuous distribution based on a collection of weighted samples, which is not as straightforward. We discuss a variety of alternatives for computing this entropy in Section III-C, then analyze the strengths and weaknesses of each in Section III-D.

B. Entropy of Angular Distribution

Since the distribution over position $\boldsymbol{s}_{0:t}$ is represented by particles, we can compute the second term in (2) using the following approximation:

$$\mathrm{E}_{\boldsymbol{s}_{0:t}}[H(p(\boldsymbol{\theta} | \boldsymbol{s}_{0:t}, z_{0:t}))] = \sum_{j=1}^{J} w_t^{[j]} H(p(\boldsymbol{\theta} | \boldsymbol{s}_{0:t}^{[j]}, z_{0:t})), \tag{3}$$

where $\boldsymbol{s}_{0:t}^{[j]}$ is the position history up to time t of the jth particle, $w_t^{[j]}$ is the weight of the jth particle at time t (normalized so that $\sum_j w_t^{[j]} = 1$), and J is the total number of particles. Since $p(\boldsymbol{\theta} | \boldsymbol{s}_{0:t}^{[j]}, z_{0:t}) = N(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, its entropy is

$$H\big(N(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})\big) = \frac{1}{2} \log\big((2\pi e)^d |\boldsymbol{\Sigma}|\big). \tag{4}$$

C. Entropy of Position Distribution

Computing the entropy of the position distribution, which is represented by a collection of weighted particles, amounts to computing the entropy of an unknown continuous density distribution based on a finite number of samples. This is inherently an ill-posed problem because no finite sample is sufficient to determine an arbitrary density distribution, and thus some assumption must be made about either the functional form of the distribution or its smoothness [17].

We compute the first term in (2) by approximating it as $H\big(p(\boldsymbol{s}_t | z_{0:t}, \boldsymbol{u}_{1:t})\big)$; i.e., we consider only the current positions of the particles and not their previous position history. We let $f(\boldsymbol{s}_t)$ denote the distribution whose entropy we wish to estimate:

$$f(\boldsymbol{s}_t) = p(\boldsymbol{s}_t | z_{0:t}, \boldsymbol{u}_{1:t}). \tag{5}$$

Since the remaining equations in Section III are only concerned with the distribution at time t, we will simplify notation by omitting the subscript t from the variables $\boldsymbol{s}$ and w. For instance, we denote the function in (5) as simply $f(\boldsymbol{s})$, and we use $\boldsymbol{s}^{[j]}$ and $w^{[j]}$ to respectively represent the position and weight of particle j at time t.

Our goal is to estimate the entropy of the unknown continuous density distribution $f(\boldsymbol{s})$ using the positions and weights of J particles. We compare several methods for estimating this entropy. First, we approximate the entropy of this distribution by assuming the distribution is Gaussian; this is analogous to the approach used for SLAM in [12]. If the distribution being estimated is far from Gaussian, however, this parametric approach will not be accurate. Thus, we also consider a non-parametric approach to entropy estimation, computing the entropy using a weighted kernel density estimate of the posterior distribution [17]–[19] with various kernels. Finally, we consider estimating the entropy using only the particle weights (ignoring the spatial locations of the particles), by treating the set of weights as a set of probability masses in a discrete distribution.

Gaussian Approximation of the Particle Distribution: To fit a Gaussian to the particle distribution at time t, we compute the weighted average $\bar{\boldsymbol{s}}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$\bar{\boldsymbol{s}} = \sum_{j=1}^{J} w^{[j]} \boldsymbol{s}^{[j]}, \quad \boldsymbol{\Sigma} = \frac{\sum_{j=1}^{J} w^{[j]} (\boldsymbol{s}^{[j]} - \bar{\boldsymbol{s}})(\boldsymbol{s}^{[j]} - \bar{\boldsymbol{s}})^{\mathrm{T}}}{1 - \sum_{j=1}^{J} \big(w^{[j]}\big)^2}, \tag{6}$$

where $w^{[j]}$ is the weight of particle j at time t (normalized so that $\sum_j w^{[j]} = 1$), and $\boldsymbol{s}^{[j]}$ is the position of particle j in the object coordinate system at time t. We have thus approximated $f(\boldsymbol{s})$ as a multivariate Gaussian, $\hat{f}(\boldsymbol{s}) = N(\bar{\boldsymbol{s}}, \boldsymbol{\Sigma})$, whose entropy $H\big(\hat{f}(\boldsymbol{s})\big)$ we compute using (4).

Kernel Density Estimate of the Particle Distribution: Here we use a non-parametric method to estimate the posterior [19] and compute its entropy [17], [18]. We form a weighted kernel density estimate from the collection of position samples $\boldsymbol{s}^{[j]}$ and corresponding nonnegative weights $w^{[j]}$ (normalized so that $\sum_j w^{[j]} = 1$), to approximate the unknown density $f(\boldsymbol{s})$:

$$\hat{f}(\boldsymbol{s}) = \sum_{j=1}^{J} w^{[j]} \, k(\boldsymbol{s} - \boldsymbol{s}^{[j]}). \tag{7}$$

The kernel function $k()$ (also known as a *Parzen window*) is a proper density function: that is, $k(x) \geq 0$ and $\int k(x)\,\mathrm{d}x = 1$. We can estimate the entropy of $f(\boldsymbol{s})$ by the entropy of $\hat{f}(\boldsymbol{s})$:

$$\begin{aligned} H\big(\hat{f}(\boldsymbol{s})\big) &= -\int \hat{f}(\boldsymbol{s}) \log \hat{f}(\boldsymbol{s})\, d\boldsymbol{s} \\ &= -\mathrm{E}_{\hat{f}(\boldsymbol{s})}\big[\log \hat{f}(\boldsymbol{s})\big], \end{aligned} \tag{8}$$

where $\mathrm{E}_{p(\boldsymbol{x})}[g(\boldsymbol{x})]$ represents the expected value of a function $g(\boldsymbol{x})$ when the random variable $\boldsymbol{x}$ is distributed according to the pdf $p(\boldsymbol{x})$.

Unfortunately, for a wide variety of choices of kernel function $k()$, including a simple Gaussian kernel, the integral in (8) is intractable. However, we can approximate the expectation over the probability distribution $\hat{f}(\boldsymbol{s})$ in (8) as an expectation over the weighted samples $\boldsymbol{s}^{[j]}$, which is similar to the approximations in [17], [18]. Our entropy estimate then simplifies to an easily computed sum:

$$\begin{aligned} \hat{H}\big(\hat{f}(\boldsymbol{s})\big) &= -\mathrm{E}_{\{\boldsymbol{s}^{[j]},w^{[j]}\}}\big[\log \hat{f}(\boldsymbol{s})\big] \\ &= -\sum_{j=1}^{J} w^{[j]} \log\left(\sum_{i=1}^{J} w^{[i]}\, k\big(\boldsymbol{s}^{[j]} - \boldsymbol{s}^{[i]}\big)\right). \end{aligned} \tag{9}$$

A wide variety of kernel functions $k(\boldsymbol{s})$ can be used. We consider a family of zero-mean, radially symmetric kernel functions of the form

$$k(\boldsymbol{s}) \propto e^{-\frac{1}{2}\|\boldsymbol{s}\|_2^n/\sigma^n}, \tag{10}$$

in which the parameter σ controls the width of the kernel function, and the exponent n controls how heavy-tailed the kernel function is. Note that when the parameter $n = 2$, the kernel function (10) is a spherically symmetric Gaussian kernel whose marginal variance in each dimension is σ^2.

Note also that we can safely ignore the kernel function's constant of proportionality in (10). This is because the constant will be fixed for any particular choice of values of n and σ, and merely adds a constant to each entropy estimate. These constants do not affect comparisons between entropy estimates that use the same kernel function.

Estimating Entropy using only the Particle Weights: As a contrast to the two methods above, we consider a discrete entropy estimate that uses only the particle weights, completely ignoring their spatial locations. This estimate treats the entire set of weights $\{w^{[j]}\}$ as a set of probability masses in a discrete probability distribution and simply computes the entropy of this discrete distribution:

$$H\big(\{w^{[j]}\}\big) = \sum_{j=1}^{J} -w^{[j]} \log w^{[j]}. \tag{11}$$

This simple entropy estimate was considered in [14] in the context of RBPF SLAM. However, it was dismissed there because every time the set of particles is resampled, the new particles will have equal weights, and thus the weight-based entropy is reset to its maximum value.

D. Comparing Estimates of Entropy of Position Distribution

We now compare the three different types of entropy estimates described in Section III-C.

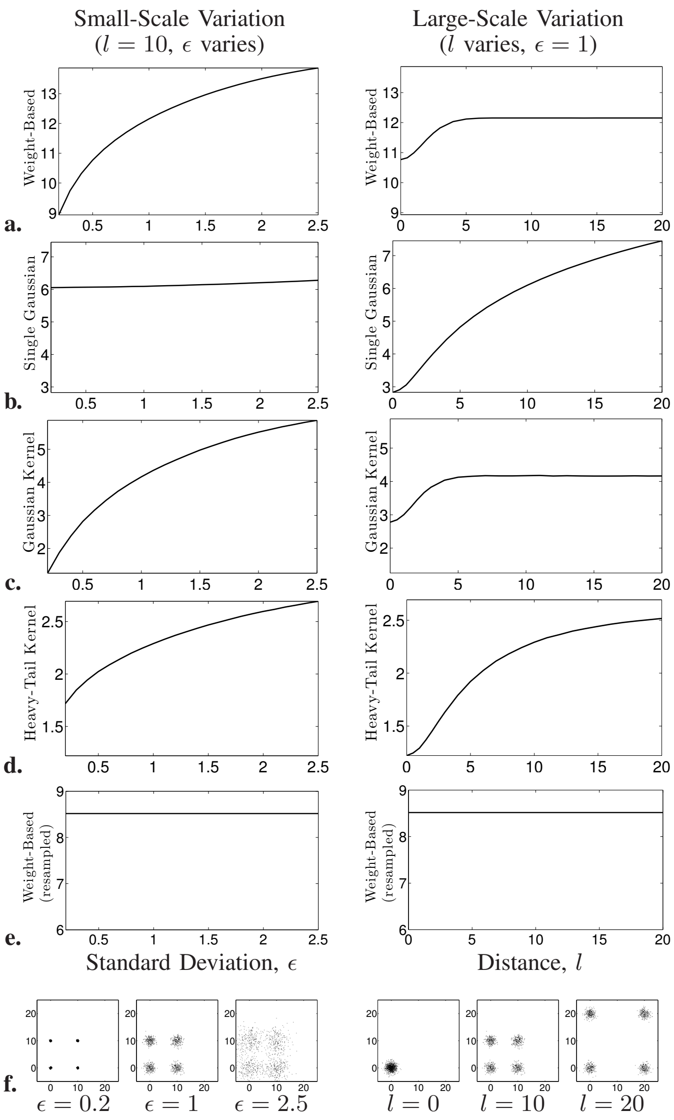

Fig. 2. *Rows a–e:* Estimated entropy of a distribution (mixture of four Gaussians) as the distribution parameters change. *Left column:* Small-scale variation experiment. Distance between Gaussians is fixed at $l = 10$, while the standard deviation ϵ of each Gaussian varies (see row f). *Right column:* Large-scale variation experiment. Distance between Gaussians l varies, while the standard deviation of each Gaussian is fixed at $\epsilon = 1$ (see row f). Note that although the vertical axes differ in different rows, we ensure they are the same for both graphs within each row, since only the relative slopes of the graphs for the same entropy measure are important here.

The Test Distributions: For comparison, we use all of the methods to estimate the entropy of the same set of 2D test distributions: a mixture of four radially symmetric 2D Gaussians centered at the four corners of a square of side length l, each of which has marginal standard deviation ϵ in each dimension. The parameters l and ϵ basically set the two characteristic scales of the test distribution, and the entire set of test distributions is generated by varying the values of these parameters. Fig. 2f shows some examples from this set of test distributions. We divide the set of test distributions into two experiments. The first experiment (left column of Fig. 2) explores small-scale variation: the standard deviation of each Gaussian is varied from $\epsilon = 0.2$ to $\epsilon = 2.5$, while the distance between Gaussians is fixed at $l = 10$ (e.g., the three distributions pictured in the left column of Fig. 2f). The second experiment (right column of Fig. 2) explores large-scale variation: the distance between Gaussians is varied from $l = 0$ to $l = 20$, while the standard deviation of each Gaussian is fixed at $\epsilon = 1$ (e.g., the three distributions pictured in the right column of Fig. 2f).

Two Sampling Methods: We used two different methods for sampling from each test distribution. For sampling method I, we randomly drew samples from a spatially uniform 2D distribution. Each sample was then assigned a weight proportional to the value of the test distribution at the sample's location. Sampling method I resulted in a set of particles that were distributed basically uniformly in space and used unequal weights to represent the test distribution. In contrast, sampling method II drew samples according to the test distribution and assigned them all equal weight. This resulted in a set of particles with equal weights whose spatial distribution represented the test distribution.

Our experiments showed that only one of the entropy estimates described Section III-C is affected by which sampling method is used: the weight-based estimate (11), whose results are shown in rows a and e of Fig. 2. When particles are sampled using method I, the weight-based estimate (Fig. 2a) provides an excellent Monte Carlo estimate of each test distribution's Shannon entropy. On the other hand, when particles are sampled using method II (Fig. 2e), the weight-based entropy measure is completely useless, giving a constant value for every test distribution. We label Fig. 2e *resampled* because the equal weights of particles sampled using method II resemble the filtering distribution of a particle filter after the particle set is resampled. The crippling effect that resampling has on the simple weight-based entropy estimate explains why RBPF approaches to active SLAM do not use this estimator [14]. For each of the other entropy estimators (shown in Fig. 2b–d), both sampling methods give the exact same result, which shows that when applied to a particle filter, these other estimators will be robust in the presence of particle resampling.

Fig. 2b shows the Gaussian approximation of entropy (6), in which the entire particle distribution is approximated by a single Gaussian whose entropy is computed. Fig. 2c–d shows the kernel density estimate (9) of the entropy using two different kernels (10): Fig. 2c used a Gaussian kernel ($n=2$) with standard deviation $\sigma=0.25$, while Fig. 2d used a heavy-tailed kernel function ($n=0.5$) with width $\sigma=0.25$.

Comparison of Entropy Estimation Methods: As noted above, Fig. 2a is approximately the Shannon entropy. Notice that it is sensitive to small-scale changes in the test distributions. However, it is not sensitive to large-scale changes when the Gaussians in the test distribution do not overlap.

In contrast, the single Gaussian approximation (Fig. 2b) is sensitive to large-scale changes in the test distribution but not to small-scale changes. A particle filter whose distribution resembles the 1st (leftmost) picture in Fig. 2f will typically be much closer to convergence than one whose distribution looks like the 3rd picture, but the Gaussian approximation registers almost no difference between them. On the other hand, for some applications we may be interested in distinguishing between distributions like the 5th and 6th pictured in Fig. 2f, for which the Gaussian approximation is better than the Shannon entropy (or the weight-based estimate).

The kernel density estimate using a small Gaussian kernel (Fig. 2c) provides an excellent estimate of the Shannon entropy across the entire range of test distributions (compare parts a and c of Fig. 2). But perhaps Shannon entropy is not what a particular application demands. Some applications may require a measure that provides small-scale sensitivity similar to that in Fig. 2a, but also large-scale sensitivity similar to that provided by the single Gaussian estimate (Fig. 2b). For such applications, the kernel density estimate with a heavy-tailed kernel, which has both a sharper central peak and longer tails than the Gaussian kernel, provides sensitivity to both small-scale and large-scale variation, as demonstrated in Fig. 2d.

IV. Motion Selection Algorithm

This section presents an algorithm for selecting an optimal motion using the RBPF entropy measures described in Section III. Compared to random motion selection, this algorithm enables us to reduce the number of measurements needed for a given accuracy, or to achieve higher accuracy using a given number of measurements. We first present an overview of our entropy-based motion selection algorithm, then describe how to apply it to two different applications: fully automatic registration, and human-guided registration.

A. Overview of the Algorithm

Algorithm 1 contains pseudocode for our motion selection algorithm. Given $\mathcal{X}_t$, the set of particles at time t, we use a greedy approach that selects the best next motion from M candidates, $\{\Gamma_1, \ldots, \Gamma_M\}$. We compute the expected entropy for each candidate and select as the next motion the candidate with the minimum expected entropy. For efficiency of entropy computations, we approximate the particle set $\mathcal{X}_t$ by its subset $\mathcal{Y}_t$, the K particles with highest weights.

Each motion candidate Γ_m is randomly proposed. For each motion candidate Γ_m, we simulate the measurement process C times, each time selecting a random particle $q^{[c]}$ from $\mathcal{Y}_t$ with probability proportional to the particle's weight. For each $q^{[c]}$, we compute the answer to the following question: If $q^{[c]}$ is correct about the relative pose between the object and the robot, then when we perform the motion dictated by Γ_m, what contact position measurement would result? The answer gives us the estimated motion $\hat{\boldsymbol{u}}_t^{[c]}$, which is used to update the particle set $\mathcal{Y}_t$. This results in an updated particle set, $\mathcal{Y}_{t+1}^{[c]}$, whose entropy we compute using one of the estimation methods from Section III. We repeat this simulation procedure C times and compute the expected entropy as the average over the C simulations.

Compared to the random motion selection used in [5], our motion selection algorithm requires the following additional computations: KMC operations of particle update, and MC entropy computations. Note that we assume that the cost to move the probe to the next position is independent of the position, since in our application the time required to move the robot between measurement positions is much less than the time required to precisely measure a single contact location. For applications in which different motion candidates have different costs (e.g., if moving to a more

Algorithm 1 Motion selection algorithm

```
Motion Selection($\mathcal{X}_t$)
for $m = 1$ to $M$ do
    Copy the $K$ particles with largest weight from $\mathcal{X}_t$ to $\mathcal{Y}_t$
    Propose a randomly chosen motion candidate $\Gamma_m$
    for $c = 1$ to $C$ do
        Sample $q^{[c]} \in \mathcal{Y}_t$ with probability given by its weight
        Compute the expected motion $\hat{\boldsymbol{u}}_t^{[c]}$ based on $q^{[c]}$ and $\Gamma_m$
        Update all $K$ particles in $\mathcal{Y}_t$ based on $\hat{\boldsymbol{u}}_t^{[c]}$, to obtain $\mathcal{Y}_{t+1}^{[c]}$
        $H_m^{[c]} =$ Compute Entropy($\mathcal{Y}_{t+1}^{[c]}$)
    end for
    $H_m = \frac{1}{C} \sum_{c=1}^{C} H_m^{[c]}$
end for
$m_0 = \arg\min_m H_m$
return $\Gamma_{m_0}$
```

distant position takes more time), one could incorporate the movement into the cost function in addition to the entropy.

B. Application to Automatic Registration

In the automatic registration approach we presented in [5], each observation consists of moving the probe to a random (x, y) position in the robot coordinate system, then measuring the z position by moving the probe towards the object, parallel to the robot's z-axis, until contact. To apply our entropy-based motion selection algorithm in this framework, each motion candidate Γ_m describes a random displacement in the x and y directions (in robot coordinates), and the measurement of the z position is simulated for each particle $q^{[c]}$. Particle $q^{[c]}$ maintains an estimate of the relative pose between robot and object, which consists of a 3D position $\boldsymbol{s}_t^{[c]}$ and a Gaussian distribution over angle $\boldsymbol{\theta}$ with mean $\boldsymbol{\mu}_\theta^{[c]}$. We use the particle's position $\boldsymbol{s}_t^{[c]}$ and mean angle $\boldsymbol{\mu}_\theta^{[c]}$, together with the object model, to estimate the location at which the probe would contact the object. The estimated motion in the z direction, combined with the (x, y) displacement Γ_m, constitute the expected motion $\hat{\boldsymbol{u}}_t^{[c]}$.

C. Application to Human-Guided Registration

The automatic registration process described above uses a fixed probing direction, typically parallel to the z-axis of the robot. For some objects, however, it is helpful or necessary to probe the object from multiple directions. In such cases, the automatic registration process may be impractical—for instance, there might be other objects nearby that restrict the possible motions of the robot. In these cases, letting a human operator move the robot to measure the contact positions is often safer and more practical, but the human operator may not know the best contact locations for registering the object. We now present a visualization system that uses our entropy-based motion selection method (Algorithm 1) to suggest an optimal touch position to a human operator, in order to achieve fast and accurate registration.

Our system is initiated by a human operator: The operator moves the peg to any position on the object surface and specifies the position to the system. The system initializes all of its particles near the location indicated by the operator, with an angular distribution selected from a range that includes every possible initial angle of contact of the probe with the surface (shown on the left in Fig. 5). In this human-guided case, each candidate motion Γ_m is simply a randomly selected point on the object surface, in object coordinates. For each particle $q^{[c]}$, we compute the expected motion in object coordinates as the difference between Γ_m and the particle's position $\boldsymbol{s}_t^{[c]}$, then transform the motion to robot coordinates by using the particle's position $\boldsymbol{s}_t^{[c]}$ and mean angle $\boldsymbol{\mu}_\theta^{[c]}$ to obtain $\hat{\boldsymbol{u}}_t^{[c]}$.

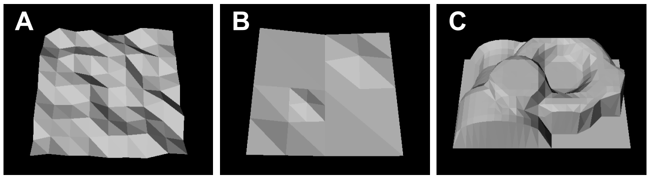

Fig. 3. Mesh models of objects used in simulations.

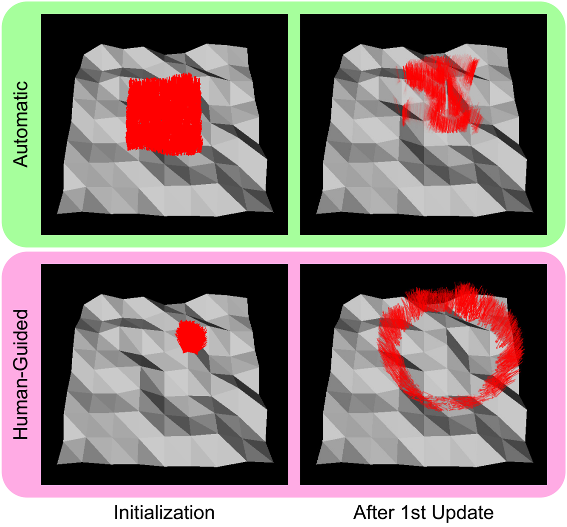

Fig. 4. Typical distributions of particles at the initial step (left) and after the first update (right), for automatic registration (top) and human-guided registration (bottom). For each particle, its position and the mean of its angle distribution are represented by the position and the orientation of a red line. The weight of each particle is represented using transparency.

The algorithm selects the motion candidate Γ_m that minimizes the expected entropy and directs the operator to probe next at that location on the object, as shown in Fig. 5 (right). Note that the operator does not have to move the peg precisely to the suggested position—the RBPF framework will work properly as long as the next contact happens anywhere on the surface of the object. However, since the suggested position achieves the minimum entropy of all of the motion candidates, using the algorithm's suggested location will enable accurate registration to be performed with fewer measurements.

V. Experiments

Existing approaches to touch-based registration [1]–[3], [5] use random motion selection to determine where to probe for each measurement. In this section, we describe several experiments comparing touch-based registration using random motion selection [5] (labeled *Random* in results tables and graphs) to that using our entropy-based motion selection algorithm. We test our algorithm using the three

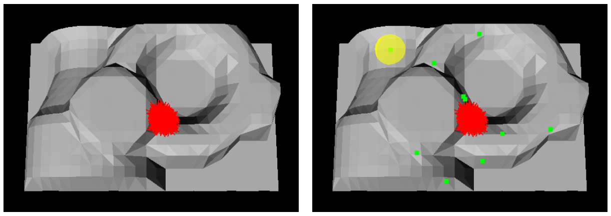

Fig. 5. Visualization for human-guided registration. *Left:* Initialization: The user specifies the first touch position on the object surface, and the system initializes particles nearby. Each particle is represented by a red line as in Fig. 4. *Right:* Candidate touch positions, randomly sampled on the object surface, are shown in green (these would not usually be shown to the user). The suggested next touch position is displayed for the user as a yellow region.

entropy estimation methods described in Section III: Gaussian approximation of the distribution (*Single Gauss.*), kernel density estimate with Gaussian kernel (*Kernel Gauss.*), and simple weight-based (*Weight*). We first present simulation results using three different objects, then present the results of experiments performed using a robot arm with a metal probe and object. The accompanying video shows our touch-based registration system in action on the robot arm.

A. Simulations

Fig. 3 shows the objects we used in simulations. Object A was generated by placing vertices every 10 units in a square (x, y) grid, choosing each z value randomly from the range $[0, 10]$, and connecting the vertices using a triangle mesh. Object B was generated similarly, but in a hierarchical manner, in order to have a surface that is less complex, with some regions more informative than others. Object C is a mesh model of a stamped part (obtained from http://www.cs.caltech.edu/~njlitke/meshes/), scaled to have a size similar to that of objects A and B.

For both automatic registration and human-guided registration, we added Gaussian noise with standard deviation 0.1 to the contact point measurements. The measurement uncertainty in the mesh model (see [5]) was Gaussian with standard deviation 0.3 for all features (faces, edges, and vertices). We initialized our process with 3200 particles and decreased the number of particles as the process proceeds, as described in [5]. We used $M = 10$, $C = 5$, and set K equal to $1/10$ of the current number of particles.

Automatic Registration: For each trial, we randomly initialized the pose of the probe by uniformly sampling its position from the range ± 15 units from the center of the object in the x and y directions, and its angle from the range $\pm 10°$ in each angular dimension. Each particle's initial position was sampled uniformly from this square region on the object's surface, with Gaussian angular distribution initialized to mean $(0°, 0°, 0°)^T$ and covariance matrix $10^2\boldsymbol{I}$, where $\boldsymbol{I}$ is the 3×3 identity matrix. Fig. 4 (top) shows an example particle distribution at the initial step and after the first update. Since we assumed a small initial angular uncertainty, all the particles are updated with similar displacements in the object coordinate system.

Table Ia shows the number of contact position measurements required for convergence using each entropy estimator. Fig. 6a compares mean translation and rotation errors when all estimators use the same number of contact measurements.

TABLE I

NUMBER OF MEASUREMENTS REQUIRED FOR CONVERGENCE (MEAN OF N TRIALS $\pm$ STANDARD ERROR)

a. Simulations of Automatic Registration ($N = 100$)

Object	Random	Weight	Single Gauss.	Kernel Gauss.
A	11.5 ± 0.3	8.7 ± 0.1	8.9 ± 0.2	8.9 ± 0.2
B	68.6 ± 5.7	33.3 ± 1.9	54.9 ± 4.4	38.0 ± 2.7
C	13.5 ± 0.6	8.5 ± 0.2	8.9 ± 0.2	8.5 ± 0.2

b. Simulations of Human-Guided Registration ($N = 100$)

Object	Random	Weight	Single Gauss.	Kernel Gauss.
A	12.3 ± 0.2	10.7 ± 0.2	11.6 ± 0.2	10.4 ± 0.2
B	26.9 ± 1.2	12.5 ± 0.3	17.2 ± 0.6	11.4 ± 0.3
C	8.9 ± 0.2	7.8 ± 0.1	8.6 ± 0.1	8.0 ± 0.1

c. Automatic Registration using a Robot Arm ($N = 30$)

Random	Weight	Single Gauss.	Kernel Gauss.
15.2 ± 0.5	12.7 ± 0.5	12.3 ± 0.6	12.1 ± 0.3

Human-Guided Registration: For each trial, we assumed the operator specified the initial contact position on the surface of the object within ± 3 units of the true position (uniformly sampled) in the x and y directions, and we sampled the initial angle uniformly from the range $(\pm 20°, \pm 20°, \pm 180°)$. Each particle's initial position was sampled uniformly from this square region on the object's surface, and its initial Gaussian angular distribution had mean sampled uniformly from the range $(\pm 20°, \pm 20°, \pm 180°)$ and covariance matrix $20^2\boldsymbol{I}$. Fig. 4 (bottom) shows an example particle distribution for this application. Since the initial angular distribution is wide, the first update spreads the particles out in the object coordinate system.

Table Ib shows the number of contact measurements required for convergence. Fig. 6b compares registration errors when all estimators use the same number of measurements.

B. Experiments Using a Robot Arm

We implemented our algorithm using a Mitsubishi MELFA RV-6SL 6-axis robot arm, using a JR3 force/torque sensor to measure contact positions (see Fig. 1). In automatic registration, our system continuously reads the force/torque values while moving the robot towards the object, detecting contact when the force values exceed a threshold. Our system first detects a contact position roughly using a large motion step size, then repeats the contact using a small motion step size to get precise measurements (see the accompanying video). In human-guided registration, our system displays the force/torque readings to the operator, who controls the robot motion and uses the readings to determine contact positions.

We used the object shown in Fig. 1. The arrangement of planes at the center of the object is used for registration. The surrounding holes, whose poses are defined in the object coordinate system, can be used to validate the estimated registration (see the accompanying video). We collected numerical results for automatic registration. Table Ic shows the number of measurements required for convergence. Fig. 6c compares mean registration errors when all entropy estimators use the same number of contact measurements.

C. Discussion of Experimental Results

It is clear from Table I that both in simulations and in experiments using a robot arm, our entropy-based motion selections are better than random motion selection. Interestingly,

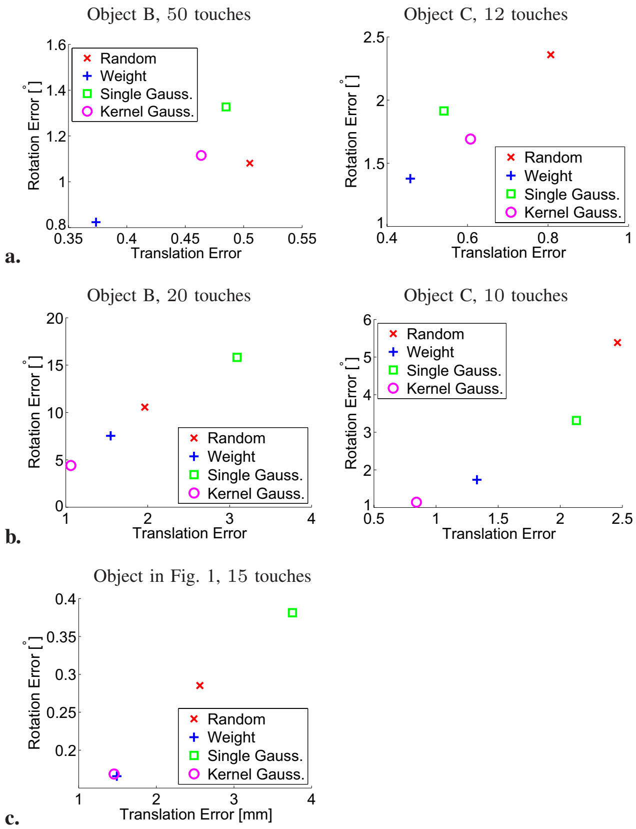

Fig. 6. Comparing mean translation and rotation errors for different entropy estimators using the same number of measurements. **a.** Simulations of automatic registration. **b.** Simulations of human-guided registration. **c.** Automatic registration using a robot arm.

the single Gaussian approximation, which has been used successfully for entropy computation in active SLAM [6], [12], generally performs the worst of the three entropy estimates. Fig. 6 also demonstrates that kernel-based and weight-based entropy estimates perform better than random motion selection, but the single Gaussian approximation is even worse than random in some cases. The reason is that in touch-based registration, the particle distribution is often far from Gaussian—for example, it can be multimodal (similar to the rightmost image in Fig. 2f) or can have most of its mass far from the mean (as in Fig. 4, bottom right). In fact, the single Gaussian approximation greatly overestimates the entropy of distributions similar to these two examples, which in some instances causes this approximation to choose the least informative candidate motions.

It is perhaps surprising that the weight-based entropy estimate performs so well, comparable to the kernel density approximation. Weight-based entropy estimates do not give a good measure of the entropy as the particle filter evolves over a long period of time, because whenever the particles are resampled, the weight-based entropy resets to its maximum value. This is why they do not work well in the context of active SLAM [14], because there the entropy must be averaged over several time steps of the predicted RBPF distribution. However, for just estimating the relative entropy one time step in the future (as in our motion selection algorithm), weight-based entropy can provide quite an effective estimate.

VI. Conclusion

We presented a method for online motion selection in touch-based registration of a robot to an object whose 3D model is known. Unlike previous touch-based registration methods, which randomly choose each motion, our RBPF-based method chooses several candidate motions, then uses estimates of expected entropy to select the best candidate. We applied our method not only to fully automatic registration (the application addressed by previous work), but also to human-guided registration. In both simulations and experiments with a robot arm, our entropy-based motion selection method significantly outperforms random motion selection. In addition, we found that entropy estimators that have been used successfully in previous work on RBPF-based active SLAM do not correspond to the entropy measures that perform best for touch-based registration.

Acknowledgments: We thank Jay Thornton and Haruhisa Okuda for their helpful comments and feedback, and John Barnwell for his support in building the experimental setup.

References

[1] S. R. Chhatpar and M. S. Branicky, "Localization for robotic assemblies using probing and particle filtering," in *AIM*, 2005.

[2] ——, "Particle filtering for localization in robotic assemblies with position uncertainty," in *IROS*, Aug. 2005, pp. 3610–3617.

[3] U. Thomas, S. Molkenstruck, R. Iser, and F. M. Wahl, "Multi sensor fusion in robot assembly using particle filters," in *ICRA*, 2007.

[4] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *ICRA*, 2006.

[5] Y. Taguchi, T. K. Marks, and H. Okuda, "Rao-Blackwellized particle filtering for probing-based 6-DOF localization in robotic assembly," in *ICRA*, 2010.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.

[7] A. Doucet, N. de Freitas, K. P. Murphy, and S. J. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, June 2000.

[8] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," MIT A. I. Memo 738, 1983.

[9] S. Ramalingam, Y. Taguchi, T. K. Marks, and O. Tuzel, "P2Π: A minimal solution for registration of 3D points to 3D planes," in *ECCV*, Sept. 2010.

[10] C. Olsson, F. Kahl, and M. Oskarsson, "The registration problem revisited: Optimal solutions from points, lines and planes," in *CVPR*, 2006, pp. 1206–1213.

[11] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *PAMI*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[12] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *RSS*, 2005.

[13] N. Fairfield and D. Wettergreen, "Active SLAM and loop prediction with the segmented map using simplified models," in *FSR*, 2009.

[14] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. Gonzalez, "An entropy-based measurement of certainty in Rao-Blackwellized particle filter mapping," in *IROS*, 2006.

[15] J. Blanco, J. Fernández-Madrigal, and J. Gonzalez, "A novel measure of uncertainty for mobile robot SLAM with Rao-Blackwellized particle filters," *IJRR*, vol. 27, no. 1, 2008.

[16] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri, "An application of kullback-leibler divergence to active SLAM and exploration with particle filters," in *IROS*, 2010.

[17] P. Viola, N. N. Schraudolph, and T. J. Sejnowski, "Empirical entropy manipulation for real-world problems," in *NIPS*, 1996.

[18] D. Erdogmus, K. E. Hild, J. C. Principe, M. Lazaro, and I. Santamaria, "Adaptive blind deconvolution of linear channels using Renyi's entropy with Parzen window estimation," *IEEE Trans. Sig. Proc.*, vol. 52, no. 6, pp. 1489–1498, 2004.

[19] M. West, "Approximating posterior distributions by mixture," *Journal of the Royal Statistical Society*, vol. 55, no. 2, pp. 409–422, 1993.