

Adaptive Information Gathering Using Visual Sensors

Philip Dames¹, Dinesh Thakur¹, Mac Schwager², and Vijay Kumar¹

Abstract—This paper proposes an algorithm to drive a robot equipped with a noisy, visual sensor to localize an unknown number of objects in an environment. The control strategy is based upon the analytic gradient of mutual information between the sensor readings and the object locations. An adaptive cellular decomposition is used to represent the environment, increasing resolution only in regions likely to contain an object. The unknown number and locations of both objects and sensor readings are modeled using random finite sets and a recursive Bayesian filter maintains the robot’s belief over the distribution of object locations. Utilizing the fact that a visual sensor can only see a finite subset of the whole environment, the complexity of the Bayesian filter update and mutual information gradient computations are significantly reduced. Numerical simulations and experimental results are used to illustrate the performance of the filter and controller.

I. INTRODUCTION

Due to decreasing cost and increasing capabilities, visual sensors such as cameras and the Microsoft Kinect have many possible applications in robotics, including the detection and localization of objects in an environment using cameras mounted to an autonomous mobile robot platform. Scenarios where such technology would be useful include search and rescue as well as inspection of buildings or vehicles, where the number of objects of interest (eg. trapped victims, or damaged areas) is in general unknown a priori. The two questions are then how to estimate both the number of objects and their locations, and subsequently how to use the robot’s belief about object locations to drive the robot to quickly explore the environment.

Our approach to the task of estimation makes use of random finite sets, as these provide a rigorous probabilistic framework for problems where the dimension of the state space, i.e. number of objects and detections, is unknown and possibly time-varying. The underlying environment is adaptively discretized into a collection of cells, giving finer resolution to regions where there is high probability of an object and coarsening the resolution in regions that are likely to be empty. This has the advantage of decreasing computational load while still allowing for arbitrarily fine precision in localization. Our algorithm then uses a recursive Bayesian filter over this discrete space to update the robot’s belief about object locations using the sensor readings.

*This work was funded in part by ONR MURI Grants N00014-07-1-0829, N00014-09-1-1051, and N00014-09-1-1031, the AFOSR MURI Grant FA9550-10-1-0567, and the SMART Future Mobility project. We are grateful for this financial support.

¹P. Dames, D. Thakur, and V. Kumar are with The GRASP Lab, University of Pennsylvania, Philadelphia, PA 19104, USA {pdames, tdinesh, kumar}@seas.upenn.edu

²M. Schwager is with the Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA schwager@bu.edu

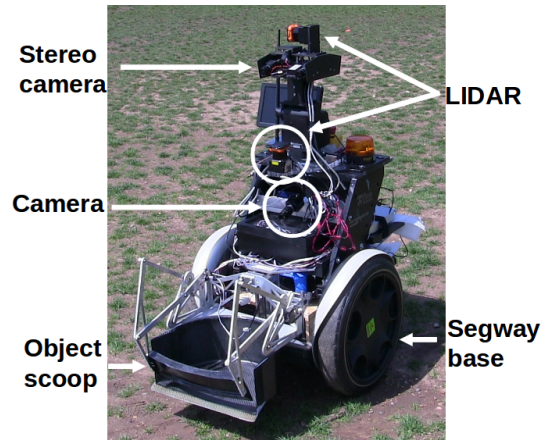


Fig. 1. Photograph of the robot platform used in this work.

Additionally the number of object detections is time varying, both because the robot’s motion causes the number of objects in the field of view of the sensor (the sensor footprint) to change and due to noisy measurements, i.e. false positive detections and false negatives. To take this uncertainty into account the sensor model is also based on random finite sets. Furthermore, leveraging the fact that the sensor footprint only covers a finite subset of the environment, we are able to reduce the computational complexity of the algorithm.

Using the resulting estimate of object locations, the control algorithm moves the robot to maximize the immediate information gain, a strategy sometimes known as “information surfing” [3]. More precisely, the controller computes the gradient of mutual information between the sensor readings and object positions using an approximation to the sensor model. The robot then moves such that the sensor footprint follows this gradient direction.

In our previous work, we introduced the analytic gradient of mutual information and used this to localize objects in a hazardous environment using a team of robots with simple binary sensors, which return a positive value if an object is detected anywhere within the sensor footprint [16]. Then in [4] we reformulated the problem using finite set statistics and developed a decentralized implementation of the previous algorithm based on an approximation to mutual information. This paper builds on our previous work in two ways. Firstly we consider more complex sensor models for estimation, providing position information of objects, allowing for directional sensing, and the possibility of multiple detections. Secondly we use an approximation to the sensor model in the controller, which decreases the computational complexity while still leading to reliable localization of objects and exploration of the environment.

A. Related Work

The use of Bayesian filtering to estimate unknown and uncertain environments is well established, with many current methods summarized in [18] by Thrun, Burgard, and Fox. In particular, the problem of multi-target tracking has been addressed in several contexts, including SLAM, computer vision, and radar-based tracking, using a variety of methods. To handle an unknown number of targets in many traditional SLAM implementations, a random vector of a specified size is initialized and the size of this vector is increased when there is sufficiently high confidence that a new object has been detected, as described in [18]. This approach is further complicated by unknown data associations, i.e. correspondence between a sensor measurement and a specific target, with many approaches only keeping the maximum likelihood correspondence or maintaining multiple filters for different correspondence hypotheses.

The development of finite set statistics in [9] by Mahler more naturally describes such problems, including removing the need to explicitly consider data associations. This has been used to effectively track an unknown number of moving targets using stationary sensors in works such as [20] by B.N. Vo, Singh, and Doucet and [19] by B.N. Vo and Ma. Recently the use of random finite sets has been adopted in mobile robotics, being used for feature-based mapping by Mullane, et al. in [13], [12]. Lundquist, et al. use this to create an obstacle map for a vehicle in [8]. All of these applications use the Probability Hypothesis Density filter introduced by Mahler in [10], which tracks the first moment of the distribution over random finite sets of target locations.

Mutual information as a control objective for active estimation has appeared in several works. Grocholsky in [5] and Bourgault, et al. in [1] use mutual information for target tracking and exploration tasks, but do not use an analytic computation of the gradient. Hoffmann and Tomlin in [6] use mutual information to localize a known number of targets, using particles filters to represent object locations and an iterative method to locally maximize mutual information around the sensor position. In [7] Julian, et al. use the identical gradient of mutual information as our work to drive multiple robots for state estimation tasks. All of these previous works only consider a known number of targets. Ristic et al. consider the problem of localizing an unknown number of targets using the expected Rényi divergence, a generalization of mutual information, to select between a discrete set of actions in [14], [15].

B. Platform Description

The platform considered in this work, shown in Fig. 1, is a differential drive robot built on a Segway platform. It is equipped with a single front-facing camera which detects objects using shape and color matching. There are a pair of stereo cameras, a vertical-scanning LIDAR, and an IMU used for odometry and a horizontal scanning LIDAR for obstacle detection. Onboard processing is done using two Mac Mini computers with 2.0 GHz Intel core i7 processors and 4 GB of RAM mounted to the robot chassis.

The organization of the remainder of the paper is as follows. In Sect. II we formulate the problem, with the Bayesian filter given in Sect. III and the mutual information gradient controller in Sect. IV. The adaptive cellular decomposition of the environment is discussed in Sect. V. Results from simulations and experiments are given in Sect. VI and Sect. VII contains our concluding remarks.

II. PROBLEM FORMULATION

Let the robot move in a bounded, planar environment $Q \subset \mathbb{R}^2$, where the vector $\mathbf{x}^t = [x, y, \psi]^T$ is the pose of the robot at time t . While the robot moves about in continuous space, the environment is discretized into a finite collection of cells $\{Q_j\}_{j=1}^m$. Then a random finite set (RFS), $S \in \mathcal{S}$, describing object locations will be a set of labels of occupied cells, where \mathcal{S} is the set of all possible RFSs for a given discretization and choice of maximum number of objects.

The robot is equipped with a camera which sees a finite subset of the environment in front of the robot, which we call the footprint of the sensor. Mathematically, the footprint is the set of cell labels that are at least partially visible by the robot and is denoted F . Thus, since the robot is only able to disambiguate environment hypotheses (i.e. RFSs) based upon what is currently visible, we define the projection $r : \mathcal{S} \rightarrow \mathcal{V}$ to be $r(S) = S \cap F$, where \mathcal{V} is the set of RFSs within the footprint of the robot. Note that in general multiple S will map to the same V , thus the map is surjective but not injective. This means that we may only define the right inverse $r^{-1}(V) = \{S \mid r(S) = V\}$, so $r(r^{-1}(V)) = V$ but $r^{-1}(r(S)) \neq S$ in general.

To take the finite footprint into account in the sensor model, the measurements are assumed to be conditionally independent of objects outside the footprint given the objects within the footprint, so that

$$\mathbb{P}(Y \mid S) = \mathbb{P}(Y \mid r(S)) = \mathbb{P}(Y \mid V), \quad (1)$$

where $Y \in \mathcal{Y}$ is the RFS of observations. Furthermore we assume that detections $\{y_1, \dots, y_n\} \in Y$ are conditionally independent of one another given the environment and that the number of detections is given by a Poisson distribution, so that Y is a Poisson RFS and the measurement model is

$$\mathbb{P}(Y \mid V) = e^{-\lambda} \prod_{y \in Y} D_V(y), \quad (2)$$

where $D_V(y)$ is the intensity function and $\lambda = \int_F D_V(y) dy$ is the expected number of detections, both of which depend on the environment V . As the location of the object inside each cell is unknown, the simplest approach is to let $D_V(y)$ be a piecewise constant function, with $D_V(y) = (1 - P_{\text{fn}})U_j(y)$ in cells that are occupied, $D_V(y) = (1 - P_{\text{fn,n}})U_j(y)$ in cells sufficiently close to occupied cells, and $D_V(y) = P_{\text{fp}}U_j(y)$ in cells that are empty. Here P_{fn} is the probability of a false negative ($P_{\text{fn,n}} > P_{\text{fn}}$), P_{fp} is the probability of a false positive, and $U_j(y)$ is the uniform distribution over cell Q_j . The choice of $P_{\text{fn,n}}$ will depend upon the consistency of the sensor, with $P_{\text{fn,n}} \approx P_{\text{fn}}$ when the noise in the measured position of an object is on the same length scale as the underlying grid.

III. BAYESIAN ESTIMATION

As the robot moves about the environment and collects measurements, a Bayesian filter keeps track of the current belief about the state of the environment. Let $\varphi^t(S) = \mathbb{P}(S | Y^{1:t})$ be the estimated distribution over RFSs at time t given all observations up to time t . Then the general form of the Bayesian update is

$$\varphi^t(S) = \frac{\mathbb{P}(Y^t | S)\varphi^{t-1}(S)}{\sum_S \mathbb{P}(Y^t | S)\varphi^{t-1}(S)}. \quad (3)$$

However this can be made more computationally efficient using the fact that sensors have a finite footprint.

Theorem 1: The Bayesian update over the full environment can be computed from the Bayesian update over the neighborhood \mathcal{V} as

$$\varphi^t(S) = \frac{\varphi^t(\mathcal{V})}{\varphi^{t-1}(\mathcal{V})}\varphi^{t-1}(S).$$

Proof: See Theorem 1 in [4]. ■

In other words, the updated belief for a RFS S is proportional to the prior belief and the updated belief over the footprint of the robot. While this operation still requires updating all RFSs over the full environment, it saves on computations as it does not require computing the measurement likelihood for each possible environment.

IV. MUTUAL INFORMATION GRADIENT CONTROLLER

The mutual information between two random variables is defined by Shannon in [17] to be

$$I(S; y) = \int_S \int_{\mathcal{Y}} \mathbb{P}(S, Y) \log \frac{\mathbb{P}(S, Y)}{\mathbb{P}(S)\mathbb{P}(Y)} dY dS. \quad (4)$$

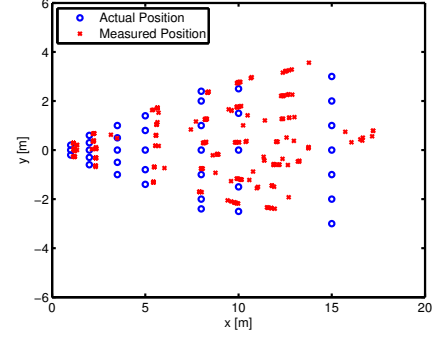
While it is possible to integrate over all possible measurement and object location RFSs, we use an approximation to reduce the complexity of these computations, similar to the motivation for the development of the PHD filter in [10]. To this end we define a new, coarse sensor model that returns a single binary reading, so the integration over Y is reduced to a sum over two terms. This coarse model can be thought of as the probability of returning a “good” measurement, so maximizing this should lead to faster localization of the targets. The intuition behind this is similar to the posterior expected number of targets (PENT) control objective proposed by Mahler and Zajic in [11], which seeks to maximize the number of targets in the footprint of the sensor.

Based on experimental results of the error in the estimated position, shown in Fig. 2(a), this binary model is

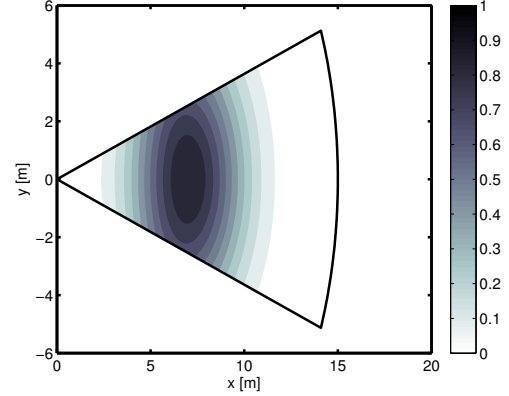
$$\mathbb{P}(y = 1 | s) = (1 - P_{\text{fn}}) \exp\left(-\frac{(r(\mathbf{x}) - R_0)^2}{\sigma_r^2} - \frac{\theta(\mathbf{x})^2}{\sigma_\theta^2}\right) \quad (5)$$

$$r(\mathbf{x}) = \sqrt{(x_s - x)^2 + (y_s - y)^2} \quad (6)$$

$$\theta(\mathbf{x}) = \tan^{-1}\left(\frac{y_s - y}{x_s - x}\right) - \psi \quad (7)$$



(a) Experimental Results



(b) Binary Measurement Model

Fig. 2. (a) Experimental results showing the true and estimated object positions as measured in the body frame of the robot. The angular bias appears to be independent of the true position while the distance error is smallest for the objects placed at $x = 8\text{m}$. Performance significantly degrades at the $x = 15\text{m}$ line. (b) An example binary measurement model, where darker shading indicates the highest probability of a “good” measurement.

which is a truncated Gaussian in polar coordinates centered at $(r, \theta) = (R_0, 0)$. Here (x_s, y_s) is the centroid of cell $s \in S$, $\sigma_r^2, \sigma_\theta^2$ are the covariances in the radial and angular directions, and \mathbf{x} is the pose of the robot. A contour plot of this function is shown in Fig. 2(b).

Given the binary detection model, we now derive a new expression for $\mathbb{P}(y | S)$. As the only way to get no detection is to not see each of the objects in the footprint and not have a false positive, we can write the conditional probability

$$\mathbb{P}(y = 0 | S) = (1 - P_{\text{fp}}) \prod_{s \in S} P(y = 0 | s), \quad (8)$$

where $P(y = 0 | s)$ is the additive complement of (5). Similarly the probability of a detection is the additive complement, $\mathbb{P}(y = 1 | S) = 1 - \mathbb{P}(y = 0 | S)$.

Our proposed controller follows the analytic gradient of mutual information with respect to the position of the peak of (5). This gradient is given in the following theorem, where the subscript \mathbf{x} indicates that the quantity depends upon the pose of the robot.

Theorem 2: Let random vector \mathcal{Y} and random finite set \mathcal{S} be jointly distributed with distribution $\mathbb{P}_{\mathbf{x}}(\mathcal{S}, \mathcal{Y})$ that is differentiable with respect to the parameter vector \mathbf{z} . Also, suppose that the support $\mathcal{S} \times \mathcal{Y}$ of $\mathbb{P}_{\mathbf{x}}(\mathcal{S}, \mathcal{Y})$ does not depend

on \mathbf{z} . Then the gradient of mutual information with respect to the parameters \mathbf{z} is

$$\frac{\partial I_{\mathbf{x}}(\mathcal{S}; \mathcal{Y})}{\partial \mathbf{z}} = \int_{\mathcal{Y}} \int_{\mathcal{S}} \frac{\partial \mathbb{P}_{\mathbf{x}}(S, Y)}{\partial \mathbf{z}} \log \frac{\mathbb{P}_{\mathbf{x}}(S, Y)}{\mathbb{P}(S) \mathbb{P}_{\mathbf{x}}(Y)} \delta S dY. \quad (9)$$

Proof: See Theorem 2 in [16]. ■

Utilizing the fact that the sensor footprint is finite we are able to significantly reduce the complexity of the gradient computations. This follows from application of the conditional independence assumption stated in (1) and the fact that the mutual information between independent random variables is identically zero, so that $I(\mathcal{S}; \mathcal{Y}) = I(\mathcal{V}; \mathcal{Y})$. The gradient computation can be simplified analogously.

Writing (9) in terms of known quantities, we have

$$\frac{\partial I_{\mathbf{x}}(\mathcal{V}; \mathcal{Y})}{\partial \mathbf{x}_p} = \sum_{y \in \{0,1\}} \sum_{V \in \mathcal{V}} \frac{\partial \mathbb{P}_{\mathbf{x}}(y | V)}{\partial \mathbf{x}_p} \varphi^t(V) \times \log \frac{\mathbb{P}_{\mathbf{x}}(y | V)}{\sum_{V \in \mathcal{V}} \mathbb{P}_{\mathbf{x}}(y | V) \varphi^t(V)}, \quad (10)$$

where $\mathbf{x}_p = [x_p, y_p]^T = [x + R_0 \cos \psi, y + R_0 \sin \psi]^T$ is the position of the peak of (5). Here $\varphi^t(V)$ comes from (3) and $\mathbb{P}_{\mathbf{x}}(y | V)$ comes from (8), the gradient of which is

$$\frac{\partial \mathbb{P}(y = 0 | V)}{\partial \mathbf{x}_p} = -\mathbb{P}(y = 0 | V) \times \sum_{v \in V} \frac{\frac{\partial \mathbb{P}(y=1|V)}{\partial r} \frac{\partial r}{\partial \mathbf{x}_p} + \frac{\partial \mathbb{P}(y=1|V)}{\partial \theta} \frac{\partial \theta}{\partial \mathbf{x}_p}}{1 - \mathbb{P}(y = 1 | V)}, \quad (11)$$

and the gradient of $\mathbb{P}(y = 1 | V)$ is the negative of this.

Using these results, our proposed controller moves the sensor field of view according to

$$\mathbf{x}_p^{t+1} = \mathbf{x}_p^t + k \frac{\frac{\partial I(\mathcal{V}; \mathcal{Y})}{\partial \mathbf{x}_p}}{\left\| \frac{\partial I(\mathcal{V}; \mathcal{Y})}{\partial \mathbf{x}_p} \right\| + \epsilon}, \quad (12)$$

where k is the maximum step size and $\epsilon \ll 1$ avoids singularities near critical points. Note that (5) is not differentiable on the boundary of the footprint, as the probability of detection instantaneously drops to zero outside of the footprint. Sub-gradient methods can be used, defining the gradient to be identically zero on the boundary of the footprint, so the optimization will perform analogously to the gradient ascent for differentiable functions.

In the event that the estimate has nearly converged within the footprint of the sensor, the mutual information and its gradient will be near zero so the local, greedy controller may get stuck. Longer time-horizon path planning would be the best way to prevent this, however even with the reductions in complexity, mutual information is prohibitively expensive for such searches. Instead, when mutual information is below some threshold, $\tau_I \ll 1$, the robot drives toward the cell with the highest entropy in the probability of occupancy, i.e. with probability nearest 0.5. The intuition here being that, because maximizing mutual information is equivalent to maximizing the expected reduction in entropy due to a sensor reading, driving toward the cell with highest uncertainty

will still lead to the desired behavior. Note that this choice ignores uncertainty in sensing and only considers marginal distributions of φ^t over individual cells, so while it is sufficient to perturb the robot away from local extrema in the greedy controller, it will not perform as well for local searches.

The only remaining question is how to move the robot itself in order to place the field of view at the desired location. The simplest method, used here, is to turn the robot to face x_p^{t+1} and then drive forward/backward as necessary. We are currently testing other methods to determine their impact on the performance of the algorithm.

V. ADAPTIVE CELLULAR DECOMPOSITION

Looking at (10), the number of computations needed to compute the mutual information gradient is $O(|\mathcal{V}|)$. While this is a linear dependence upon the number of RFSs in the footprint of the robot, this number depends upon the number of cells in the footprint, $|F|$, and the maximum number of objects, N_{\max} , according to

$$|\mathcal{V}| = \sum_{n=0}^{N_{\max}} \binom{|F|}{n}. \quad (13)$$

So $|\mathcal{V}|$ is $O(2^{|F|})$ when $N_{\max} \approx |F|$ and $O(|F|^{N_{\max}})$ when $N_{\max} \ll |F|$.

To keep the number of cells to a tractable level, we must either consider a small number of objects or employ an adaptive cellular decomposition of the environment. One such method used in grid-based mapping and localization problems is the quadtree structure, such as in the work of Chung, et al. in [2]. The idea here is that an initial representation of the environment is very coarse, and only refined in areas that are likely to contain an object. Furthermore we allow the refinement to be reversed, in case areas are incorrectly refined due to false positive detections. The two basic operations to this procedure are the addition and subtraction of a cell from the environment, given in Alg. 1 and Alg. 2, respectively. In the case of a quadtree, a refinement consists of the removal of the occupied cell followed by the addition of four new cells and takes place if the probability that a cell is occupied surpasses a specified threshold τ_o . Similarly a merging process occurs when the probability that a cell is empty drop below another threshold $\tau_e < \tau_o$, and consists of the removal of four empty cells followed by the addition of a single new cell. Examples of these processes are shown in Fig. 3.

Algorithm 1: Add Cell

- 1: $S' \leftarrow S$
- 2: **for** $S \in \mathcal{S} \mid |S| < \max$ number of objects **do**
- 3: $\varphi'(S) \leftarrow \frac{1}{2} \varphi(S)$
- 4: $\varphi'(S \cup \{m_s + 1\}) \leftarrow \frac{1}{2} \varphi(S)$
- 5: $S' \leftarrow S' \cup \{S \cup \{m_s + 1\}\}$
- 6: **end for**
- 7: $m'_s \leftarrow m_s + 1$

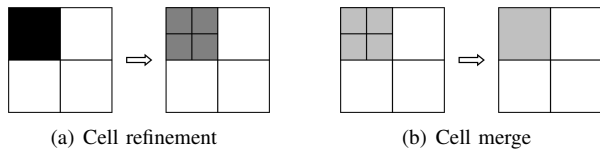


Fig. 3. A simple 2×2 grid example where the shading indicates the probability that a cell is occupied with white being 0 and black being 1. A cell refinement procedure is shown in (a), where a large occupied cell is divided into four smaller cells with lesser occupancy probability. A grid merging procedure is shown in (b), where four empty sub-cells with the same parent cell are merged to form the parent cell.

Algorithm 2: Remove Cell

- 1: **for** $V \in \mathcal{V}_{j^c}$ **do**
- 2: $\varphi(V) \leftarrow \sum_{S|r_j(S)=V} \varphi(S)$
- 3: $S' \leftarrow S' \setminus \{V \cup \{j\}\}$
- 4: **end for**
- 5: $m_s \leftarrow m_s - 1$

VI. TEST RESULTS

To test the performance of our proposed algorithm we conduct a series of simulations in Matlab and field tests on the robot. In general visual sensors can be very noisy, returning many false positives due to other objects in the environment (eg. if using shape detection to locate a ball, the wheel of a car is a potential false positive) and many false negatives (eg. variable lighting conditions and occlusions). To take this into account we set the probability of a false positive to be $P_{fp} = 0.1$ and the probabilities of a false negative to be $P_{fn} = 0.1$ and $P_{fn,n} = 0.5$ based on empirically observed performance. In the binary sensor model (5) we set $R_0 = 7\text{m}$, $\sigma_r = 3\text{m}$ and $\sigma_\theta = 0.7\text{rad}$, and for the control law $k = 0.25\text{m}$, $\epsilon = 10^{-20}$, and $\tau_I = 10^{-5}$. The sensor has a 10m range and a 40° field of view.

Simulation Results: The open-field environment, shown in Fig. 4(a) along with a typical path and final location, is used for simulations. As can be seen the robot tends to sweep through the environment, exploring new regions with every loop. Once an object appears within the field of view of the sensor and the robot receives many consecutive detections in the same area, the robot stays in that area and will continue to observe the object until the probability that it is in a particular cell is near 1. In other words, the robot will perform a depth-first search of the quadtree until it determines the leaf of minimum size (chosen to be 1m here) containing the object.

Fig. 4(b) shows the final estimate of object locations for the path in Fig. 4(a), with darker shading indicating a higher probability of occupancy in a cell. The two dark cells, which are the true cells containing the objects, have a probability of occupancy near 1 (differing by $\sim 10^{-9}$) and all other cells are empty with probability of occupancy ≤ 0.0068 . We assume a maximum of three objects in the environment.

Fig. 4(c) shows the time evolution of the entropy of the distribution over object position over 20 trials of the simulation. Overall there is a decreasing trend, with small local increases due to noisy measurements, i.e. false positive and false negative detections. The brief sections with large fluctuations occur when an object is in the field of view of the sensor: there is a steep drop in entropy as the probability

of occupancy increases followed by instantaneous increases as the grid is refined, repeating until the estimate converges. In each trial the filter correctly found the objects.

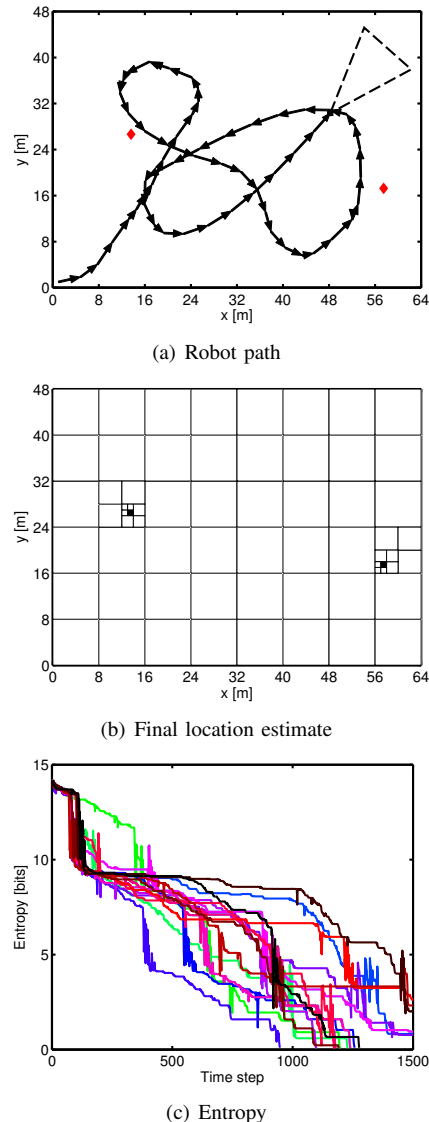
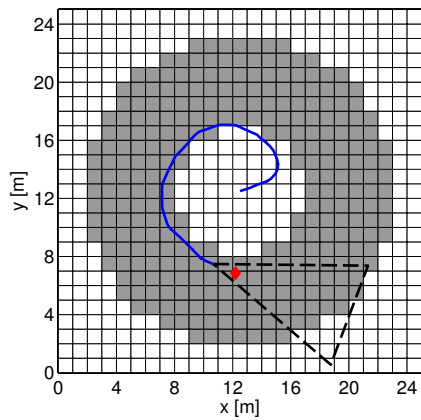


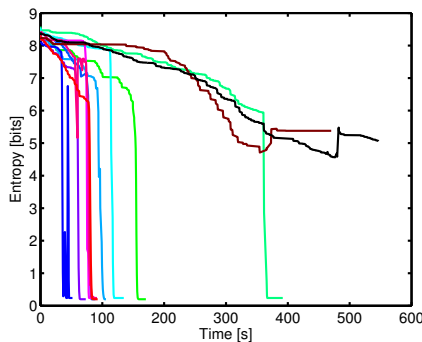
Fig. 4. Sample results from simulated data. (a) A typical path taken by the robot, starting from the origin, is indicated by the solid line and the final position of the sensor footprint is given by the dashed line. True object locations are given by the red diamonds. (b) The final object position estimates for the path shown in (a). The shading in each cell is proportional to the probability of occupancy, with white being 0 and black being 1. (c) Time history of the entropy of the distribution of object locations over 20 representative runs.

Experimental Results: The second environment, shown in Fig. 5(a), is used for field tests with the robot and is the simplest example of a non-trivial topology in the prior belief. In this scenario the robot begins at the center of the environment, with a single object located in the surrounding annular region show by the shaded cells, which have non-zero probability of occupancy in the prior.

We performed 12 trials with random initial position of the object with the resulting time history of the entropies in Fig. 5(b). In 10 runs the robot correctly located the object within the 1m precision of the grid. The sudden drop in



(a) Prior belief and path



(b) Entropy

Fig. 5. Sample results from experimental data. (a) A typical path taken by the robot, starting from the center of the annulus, is indicated by the solid line and the final position of the robot and its sensor footprint are given by the dashed line. The true object location is given by the red diamond. Shaded cells correspond to non-zero prior probability of the cell containing an object. (b) Time history of the entropy of the distribution of object locations over ten representative runs.

entropy is due to the fact that the number of objects is known, causing the distribution to rapidly converge when multiple detections are made in the same cell. The variation in time to convergence is due to the random placement of the object, with short times corresponding to the object being placed nearer the initial footprint of the robot. The robot failed to localize the object after a full sweep of the environment in two runs due to failures in the perception system. Uneven terrain can cause large jumps in the measured location of the target as we do not yet take the tilt of the robot into account when sensing, causing the filter to lose track of the object. The system was able to recover in once such instance (blue line in Fig. 5(b)), nearly converging to the incorrect cell before being switching to the correct cell, causing the large spike in entropy near the end of the trial.

VII. CONCLUSIONS

In this paper we proposed a method to drive a robot equipped with a visual sensor to localize an unknown number of objects in an environment by following the gradient of mutual information between the objection locations and the probability of detection. The number and locations of objects is modeled using random finite sets over an adaptive discretization of the environment, allowing for arbitrarily fine

resolution while reducing the computational complexity. A recursive Bayesian filter maintains the robot's belief of object locations. The complexity is further reduced by noting that real sensors have a limited field of view in the environment, thus sensor measurements will be conditionally independent of objects that are not visible. Finally, simulated and experimental results illustrate the performance of our proposed algorithm, reliably finding the true object locations. It should also be possible to include multiple robots by combining the techniques from this work and [4].

REFERENCES

- [1] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *Proceedings of the IEEE Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 540–545, 2002.
- [2] T.H. Chung and S. Carpin. Multiscale search using probabilistic quadtrees. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2546–2553, 2011.
- [3] R.A. Cortez, H.G. Tanner, R. Lumia, and C.T. Abdallah. Information surfing for radiation map building. *International Journal of Robotics and Automation*, 26(1), 2011.
- [4] P. Dames, M. Schwager, V. Kumar, and D. Rus. A decentralized control policy for adaptive information gathering in hazardous environments. In *Proceedings of the IEEE Conf. on Decision and Control (CDC)*, December 2012. Submitted.
- [5] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sydney, 2002.
- [6] G. M. Hoffmann and C. J. Tomlin. Mobile sensor network control using mutual information methods and particle filters. *IEEE Trans. on Automatic Control*, 55(1):32–47, January 2010.
- [7] B. J. Julian, M. Angermann, M. Schwager, and D. Rus. A scalable information theoretic approach to distributed robot coordination. In *Proceedings of the IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, 2011.
- [8] C. Lundquist, L. Hammarstrand, and F. Gustafsson. Road intensity based mapping using radar measurements with a probability hypothesis density filter. *IEEE Trans. on Signal Processing*, 59(4):1397–1408, 2010.
- [9] R. Mahler. *Statistical multisource-multitarget information fusion*. 2007. Artech House, Norwood, MA.
- [10] R. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Trans. on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.
- [11] R.P.S. Mahler and T.R. Zajic. Probabilistic objective functions for sensor management. In *Proceedings of SPIE*, volume 5429, page 233, 2004.
- [12] J. Mullane, B. Vo, M.D. Adams, et al. A random-finite-set approach to bayesian slam. *IEEE Trans. on Robotics*, 27(2):268–282, 2011.
- [13] J. Mullane, B.N. Vo, MD Adams, and B.T. Vo. Random finite sets for robot mapping and slam. *Springer Tracts in Advanced Robotics*, 2011.
- [14] B. Ristic and B.N. Vo. Sensor control for multi-object state-space estimation using random finite sets. *Automatica*, 46(11):1812–1818, 2010.
- [15] B. Ristic, B.N. Vo, and D. Clark. A note on the reward function for phd filters with sensor control. *IEEE Trans. on Aerospace and Electronic Systems*, 47(2):1521–1529, 2011.
- [16] M. Schwager, P. Dames, D. Rus, and V. Kumar. A multi-robot control policy for information gathering in the presence of unknown hazards. In *Proceedings of the International Symposium on Robotics Research (ISRR)*, August 2011.
- [17] C. E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 1948.
- [18] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [19] B.N. Vo and W.K. Ma. The gaussian mixture probability hypothesis density filter. *IEEE Trans. on Signal Processing*, 54(11):4091–4104, 2006.
- [20] B.N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *IEEE Trans. on Aerospace and Electronic Systems*, 41(4):1224–1245, 2005.