

The Ability of Robots to Locate and Interact with Objects

Play Fetch with Your Robot

By Philip Dames, Dinesh Thakur, Mac Schwager, and Vijay Kumar



IMAGE LICENSED BY INGRAM PUBLISHING

The task addressed in this article is the localization of an unknown number of targets using a mobile robot equipped with a visual sensor. The estimation of the number of targets and their locations is done using a recursive Bayesian filter over random finite sets (RFSs), and the position of the robot is assumed to be known. We present a computationally tractable control law whereby the robot follows the gradient of mutual information between target locations and detections. The method is verified through real-world experimental trials, reliably detecting multiple targets and ignoring clutter obstacles.

A Game of Fetch

The ability for robots to locate and interact with objects of interest within an unstructured environment is very impor-

tant as robots move out of controlled settings. In this article, we examine a prototypical example of playing fetch with a robot. First the robot is shown a new object and then it must go into the field and locate a small yet potentially unknown number of these objects that are scattered throughout the environment. After locating all of the objects, the robot collects them and returns them to the user. Such behavior has obvious extensions to household robots, inspection tasks, and search and rescue. Using real-world experiments with the robot shown in Figure 1, we present results showing the localization of a variety of objects, focusing on the control and estimation rather than the collection and return tasks.

The use of Bayesian filtering to estimate unknown and uncertain environments is well established, with many current methods summarized by Thrun et al. [19]. In particular, the problem of multiobject tracking has been addressed in several contexts, including simultaneous localization and mapping (SLAM), computer vision, and radar-based tracking, using a

Digital Object Identifier 10.1109/MRA.2013.2295947
Date of publication: 8 May 2014

variety of methods. To handle an unknown number of objects 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 [19]. This approach is further complicated by unknown data associations, i.e., the correspondence between a sensor measurement and a specific object, with many approaches only keeping the maximum likelihood correspondence or maintaining multiple filters for different correspondence hypotheses.

Our approach to the task of estimation is based on finite set statistics (FISST), a rigorous probabilistic framework that naturally suits problems where the dimension of the state space, i.e., the number of objects and detections, is unknown and possibly time varying. This was developed by Mahler [11] and includes several advantages over traditional methods, most notably removing the need to explicitly consider data associations. The primary tool in this field is the probability hypothesis density (PHD) filter [11], which tracks the first moment of the distribution over RFSs of object locations. The PHD filter has been used to effectively track an unknown number of moving objects using stationary sensors by Vo et al. [21] and Vo and Ma [20], among others. Recently, the use of RFSs was adopted in mobile robotics for feature-based mapping by Mullane et al. [13], [14]. Lundquist et al. [10] used this to create an obstacle map for a vehicle. Our approach differs from these works in that we do not use the PHD filter. Instead, we run a Bayesian filter over the distribution of RFSs under the assumption that the number of objects is small.

Using this estimate of object locations, the robot then moves to maximize the immediate information gain, a strategy sometimes called *information surfing* [3]. Grocholsky [7] and Bourgault et al. [2] use mutual information for object tracking and exploration tasks, but they do not use an analytic computation of the gradient. Hoffmann and Tomlin [8] use mutual information to localize objects, using particle filters to represent object locations and an iterative method to locally maximize mutual information around the sensor position. Julian et al. [9] use the gradient of mutual information to drive multiple robots for state estimation tasks. All of these previous works only consider a known number of objects. Ristic and Vo [15] and Ristic et al. [16] consider the problem of localizing an unknown number of objects using a single robot by maximizing the expected Rényi divergence, a generalization of mutual information, to select between a discrete set of actions. Our approach uses the same gradient of mutual information between the sensor readings and object positions as [9], moving the sensor footprint to follow this gradient direction.

Problem Formulation

The robot moves about in a bounded, planar environment. The robot's pose within that environment consists of its position and orientation. Although the robot moves about in continuous space, the environment is discretized into a finite collection of cells for the robot to perform the sensing and

estimation tasks. In this discrete representation, an RFS is a set of labels of occupied cells. The set of all such sets contains every combination from zero to N cells, with N being the maximum number of possible objects that are to be tracked in the environment.

As is evident from this construction, the number of such RFSs will become intractably large when either the total number of cells or the maximum number of objects becomes large. The former is mitigated by using an adaptive discretization of the environment based on a quadtree data structure while the latter is not an issue as we are considering small numbers of objects. The positions of these objects are correlated: discovering targets in one region of the environment drives down the likelihood of objects being in other regions due to the assumption that the number of targets is small compared with the number of cells. On the other hand, the PHD filter allows for tracking a larger number of objects by making the potentially restrictive assumption that their positions are independent. In addition, it is much more difficult to capture the correlations in object locations with the feature-based representations used for the PHD filter, e.g., weighted particle sets and mixtures of Gaussians.

In our quadtree representation, cells are initialized to be large, and only when the probability of occupancy exceeds

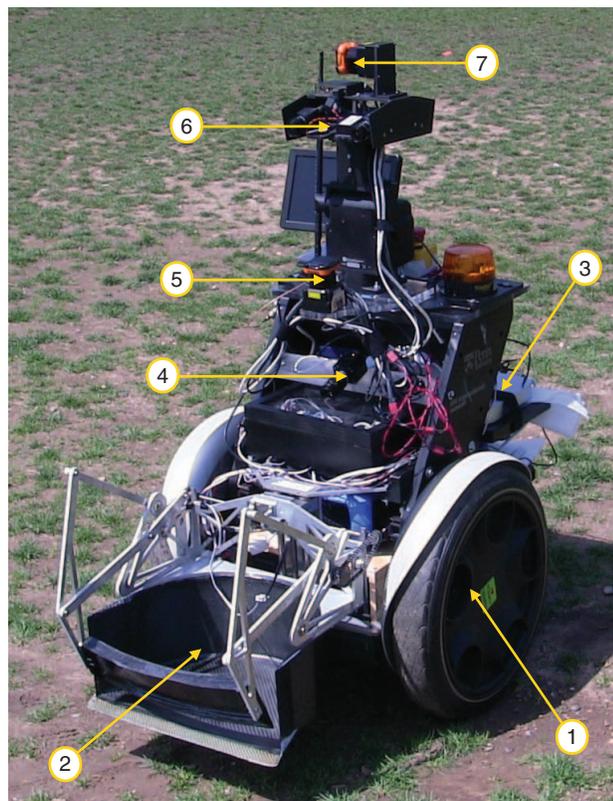


Figure 1. The robot platform used in this article, consisting of: 1) a Segway base, 2) object scoop, 3) two Mac Minis, 4) front-facing camera for object detection, 5) horizontal-scanning light detection and ranging (LIDAR) for obstacle avoidance, 6) stereo camera for visual odometry, and 7) vertical-scanning LIDAR for pitch estimation on uneven terrain.

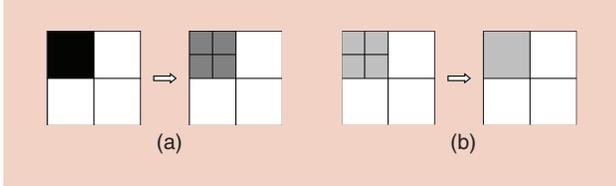


Figure 2. A simple 2×2 grid example where the shading indicates the probability that a cell is occupied, with white being zero and black being one. (a) A cell-refinement procedure where a large occupied cell is divided into four smaller cells with equal occupancy probability in each. (b) A grid-merging procedure where four empty subcells with the same parent cell are merged to form the parent cell.

some threshold (near one) do we divide the cell into four sub-cells, as shown in Figure 2. This refinement procedure continues until cells reach a minimum size, chosen to be near the standard deviation of the measurement noise, as targets cannot be located with significantly higher precision even in a continuous representation of the environment. If a large cell is divided due to a series of false positive detections, we also allow the fusion of four empty cells back into the larger parent cell, as shown in Figure 2. These cell refinement and merging procedures are done in such a way as to keep the probability of an object being within the parent cell constant, where the refinement initializes all subcells to have a uniform probability of occupancy. See [15] for pseudocode descriptions of these cell operations.

Sensing

The sensor model used for multiobject problems must consider the possibility of missing an object within the footprint (i.e., a false negative), detecting an object that is not there (i.e., a false positive or clutter detection), or returning noisy estimates of true objects. To this end, we use the general form of the FISST measurement model with Poisson clutter detections [11] as follows:

$$p(Z|X;q) = e^{-\mu} \left(\prod_{z \in Z} \kappa(z) \right) \left(\prod_{x \in X} 1 - p_d(x;q) \right) \times \left(\sum_{\theta} \prod_{j|\theta(j) \neq 0} \frac{p_d(x_j;q) g(z_{\theta(j)}|x_j;q)}{\kappa(z_{\theta(j)}) (1 - p_d(x_j;q))} \right), \quad (1)$$

where $\kappa(z)$ is the clutter PHD, $p_d(x;q)$ is the detection likelihood, $g(z|x;q)$ is the single-object measurement model, $\theta: \{1, \dots, n\} \rightarrow \{0, 1, \dots, m\}$ is a data association, and q is the pose of the robot. Note that $\theta(j) = 0$ means that the object is not detected, i.e., a false negative, and any element of $\{1, \dots, m\}$ not in the range of $\theta(\{1, \dots, n\})$ is a false positive. Also the pose, q , is shown to emphasize the dependence of the measurements on the robot's pose. Intuitively, this function averages over all possible data associations. This is not prohibitively large for small numbers of objects n and measurements m where in a single data association, θ , each measurement $z \in Z$ is either said to originate from a target $x \in X$ or be due to a false positive. The first two terms in (1)

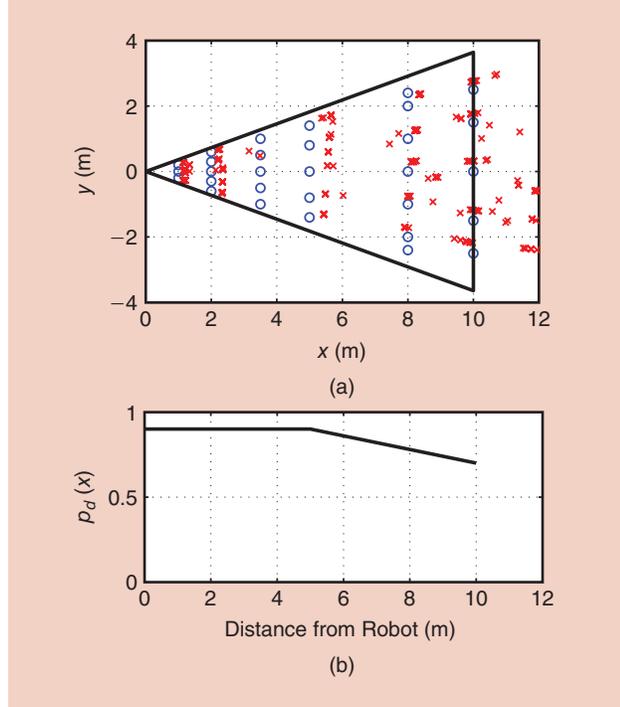


Figure 3. (a) The experimental results showing the true (blue circles) and estimated (red xs) object positions as measured in the body frame of the robot. This is superimposed on the sensor footprint and represents approximately 600 data points. (b) The detection likelihood as a function of the distance to the robot.

represent false positives, the next term represents false negatives, and the final term represents true detections.

In our case, the robot is equipped with a front-facing camera (label 4 from Figure 1) for object detection, which sees a finite subset of the environment that we call the *footprint*. Mathematically, the footprint is the set of cell labels that are at least partially visible by the robot. The system runs a template-matching algorithm to detect objects of interest within the image using shape and color and combines this with the pitch estimate to calculate the position of the objects relative to the robot. It then returns a list of cells occupied by objects with a sufficiently good match.

To determine models for the detection and measurement likelihoods, we conduct a set of experiments placing objects at known locations in front of the robot and collecting measurements. The results of this are shown in Figure 3 overlaid on the sensor footprint. Below this is the detection likelihood function, $p_d(x_c;q)$, where x_c represents a continuous domain position. The single-target measurement model is the position of the target corrupted by Gaussian noise, $g(z_c|x_c;q) = x + \mathcal{N}(0, \Sigma)$, and the noise covariance Σ is found from this training data.

Due to our discrete representation of the environment, these continuous domain detection and measurement models must be converted into a discrete form. Since the target location is uniform within a cell, we may simply average the detection model over the cell domain to obtain the detection likelihood for a cell

$$p_d(x; q) = \frac{\int_x p_d(x; q) dx_c}{\int_x dx_c},$$

which may be approximated by a finite sum over representative points within the cell. This same process may be used to find the probability of a measurement originating from a target in a given cell (of the smallest resolution), where any cell greater than three standard deviations from the detection is set to zero likelihood. This detection grid is then convolved with a candidate target cell, x , a process similar to Gaussian blurring in image processing. This process is shown in Figure 4. Note that this approach requires the small cell measurement model to have finite support, which could be taken as the bounds of the environment if the sensor is able to see the entire environment.

Finally, the clutter is modeled as a Poisson RFS, where the number of false positive detections follows a Poisson distribution with mean $\mu = \int \kappa(z) dz$, and the clutter detections are independent and identically distributed from the PHD $\kappa(z)$. Without any prior knowledge, the most natural choice is to set $\kappa(z)$ to be uniform within the sensor footprint and zero outside, as no detections can occur outside the footprint.

Multiobject Estimation

In the scenario under consideration, the robot only needs to estimate the locations of targets as we assume that the robot is well localized, i.e., the uncertainty in the pose estimate from the wheel and visual odometry is small compared with the uncertainty in sensing. As the robot moves about the environment and collects measurements, a Bayesian filter keeps track of its current belief about the state of the environment. The formulation of such a filter is standard

$$p(X | Z) = \frac{p(Z | X; q)p(X)}{\sum_{X \in \mathcal{X}} p(Z | X; q)p(X)}, \quad (2)$$

requiring only the measurement model (1) and a prior belief, which in general is initialized to the uniform distribution.

However, there are several quantities of interest that we may extract from this distribution over RFSs. In particular, we can find the probability that a given cell is occupied by adding the probability of all RFS realizations that contain that cell label; we can find out whether the cell is empty by taking the complement. This idea can be easily extended to find the probability over larger subsets of the environment by simply adding all RFS realizations that differ only outside the region of interest. Another quantity that may be of interest is the number of objects within the environment. The distribution over object number may be calculated by adding the probability of all RFS realizations containing a particular number of cells. Then, using the distribution, we can extract statistics such as the mean and variance of the estimate.

Information-Based Control

Mutual information is a concept defined by Shannon [18] that attempts to quantify the amount that can be learned about one random variable by observing another and is defined as

$$I(x, z; q) = \sum_{X \in \mathcal{X}} \sum_{Z \in \mathcal{Z}} p(X, Z; q) \log \frac{p(X, Z; q)}{p(X)p(Z; q)} \quad (3)$$

$$= \sum_{X \in \mathcal{X}} \sum_{Z \in \mathcal{Z}} p(Z | X; q)p(X) \log \frac{p(Z | X; q)}{p(Z; q)}, \quad (4)$$

where x is the space of all object locations X , z is the space of measurements Z , and q is again the pose of the robot. The measurement term within the log is computed through marginalization

$$p(Z; q) = \sum_{X \in \mathcal{X}} p(Z | X; q)p(X). \quad (5)$$

In our case, we seek to learn the object positions, which are only indirectly observable through sensor measurements. Another interpretation of mutual information is the expected difference, as measured by the Kullback–Leibler divergence, between the current and next estimates of the object positions. In this light, we wish for the robot to move to the location where the measurement is expected to change the estimate most.

Computational Considerations

One drawback of using mutual information is that it is computationally expensive: if one were to sum up all possible measurement and object location RFSs, the computation would be prohibitively slow to use for real-time localization of objects. To this end, we define a coarse sensor model based on the detection likelihood that returns a single binary reading, and so the integration over Z is reduced to a sum of two terms: either the robot sees at least one object within the footprint or it sees nothing. This coarse model can be thought of as the probability of returning a good measurement. Therefore, maximizing this should lead to faster localization of the objects, a concept we have used in other work [4], [5]. This differs from the approach by Ristic and Vo [15], who use the full sensor model but sample from it to achieve computational tractability.

Deriving this binary model is straightforward as the only way to get no detections in the footprint is to not have any false positive readings and to not see any objects within the footprint. In other words,

$$p(|Z| = \emptyset | X; q) = e^{-\mu} \cdot \prod_{x \in X} (1 - p_d(x; q)). \quad (6)$$

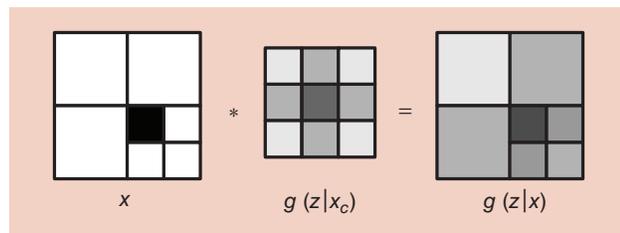


Figure 4. A simple large cell, with uniform probability of the object being in each subcell, is convolved with the small cell measurement model. The resulting array can then be converted to the large cell model by simply taking the sum of the likelihood in each of the subcells. The relative sizes of each cell are to scale, and the shading corresponds to the likelihood, with values outside the displayed cells being implicitly zero.

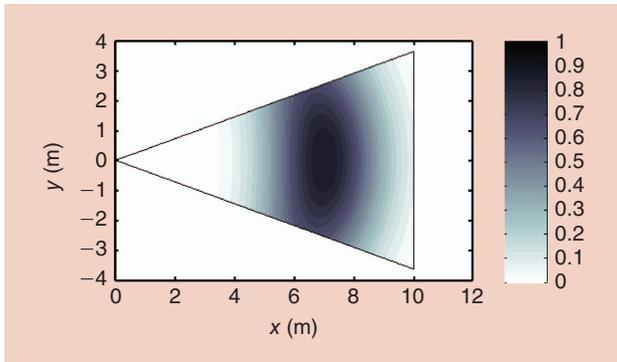


Figure 5. The experimental results showing the true and estimated object positions as measured in the body frame of the robot. The angular bias appears to be independent of the true position while the distance error is smallest for the objects placed at $x = 8$ m. Performance significantly degrades at the $x = 12$ m line. This is overlaid on the binary detection model, where darker shading indicates a higher probability of a measurement.

Then, the probability of at least one detection is simply the additive complement of this.

For the controller, we use a different detection model than that used in the estimation as the piecewise-linear model in Figure 3 has a piecewise-constant gradient that always points directly backwards. The controller detection model, shown in Figure 5, is a truncated Gaussian in polar coordinates. This functional form was chosen for two primary reasons: 1) it is differentiable everywhere except on the edge of the footprint and 2) it pushes the robot to center the camera (i.e., the peak in the model) on regions of high uncertainty.

We can also reduce the effective number of possible object RFSs by exploiting the fact that sensors have a finite footprint, as this means that robots cannot distinguish between RFSs that differ only outside the footprint. Thus, we can precompute the distribution over possible RFSs within the footprint, as this will be, in general, much smaller than the total number of RFSs within the environment. This reduced set is then used to compute mutual information.

Gradient Controller

In (4), we have used the definition of conditional probability to write the expression using the binary sensor model $p(Z | X; q)$ and filter estimate $p(X)$. Recall that the sensor model implicitly depends on the position of the robot due to the finite footprint, directionality of the sensor, and nonuniform detection likelihood. Thus, taking the gradient with respect to a vector v ,

$$\frac{\partial I(X, Z; q)}{\partial v} = \sum_{Z \in z} \sum_{X \in x} \frac{\partial p(Z | X; q)}{\partial v} p(X) \log \frac{p(Z | X; q)}{p(Z; q)} \quad (7)$$

yields a meaningful result and moves the robot in the direction of maximum local information gain, following that for a fixed distance. See [17] for a derivation of (7). The vector, v , could be taken to be the robot's pose, q , but a more natural choice for our problem is the location of the camera center projected onto the ground plane, i.e., the peak of the binary sensor model in Figure 5. The robot then moves in such a way that the camera

center follows this gradient direction, effectively directing the sensor field of view toward regions of high uncertainty.

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 is 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 closest to 0.5. The intuition here is that, because maximizing mutual information is equivalent to maximizing the expected reduction in entropy due to a sensor reading, driving toward the cell with the highest uncertainty will still lead to the desired behavior. Note that this choice ignores uncertainty in sensing and only considers marginal distributions of $P(X)$ 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.

Test Results

The platform considered in this article, shown in Figure 1, is a differential drive robot built on a Segway platform with a maximum speed of 4.4 m/s but limited to 0.55 m/s for our trials. It is equipped with a single front-facing camera, which detects objects using shape and color matching. There is also a pair of stereo cameras for visual odometry, a vertical-scanning light detection and ranging (LIDAR) for pitch estimation to correct distance measurements, wheel encoders for odometry, and a horizontal scanning LIDAR for obstacle detection. Onboard processing is done using two Mac Mini computers running Ubuntu and the Robot Operating System [1], each with 2.0-GHz Intel core i7 processors and 4 GB of RAM, mounted to the robot chassis. While the platform is outfitted with an extensive sensor suite, the front-facing color camera yields the best performance for object detection since the black-and-white stereo cameras are not as reliable and the LIDAR cannot detect small objects on the ground. For more information on the platform, see [6].

To test the performance of our proposed algorithm, we conduct a series of field tests on the robot. In general, visual sensors can be very noisy, returning false positives due to other objects in the environment (e.g., if using shape detection to locate a ball, the wheel of a car is a potential false positive) and false negatives due to variable lighting conditions and occlusions. To take this into account, we set the expected clutter rate to $\mu = 0.05$ and the probability of a false negative to $P_{fn} = 0.05$ based on empirically observed performance. In the binary sensor model (6), the peak is located 7 m directly in front of the robot and standard deviations 2 m in the radial direction and 0.5 rad in the angular direction. The sensor has a 10-m range and a 40° field of view.

Experimental Results

The environment used for field tests with the robot, shown in Figure 6(a), is the simplest example of a nontrivial topology in

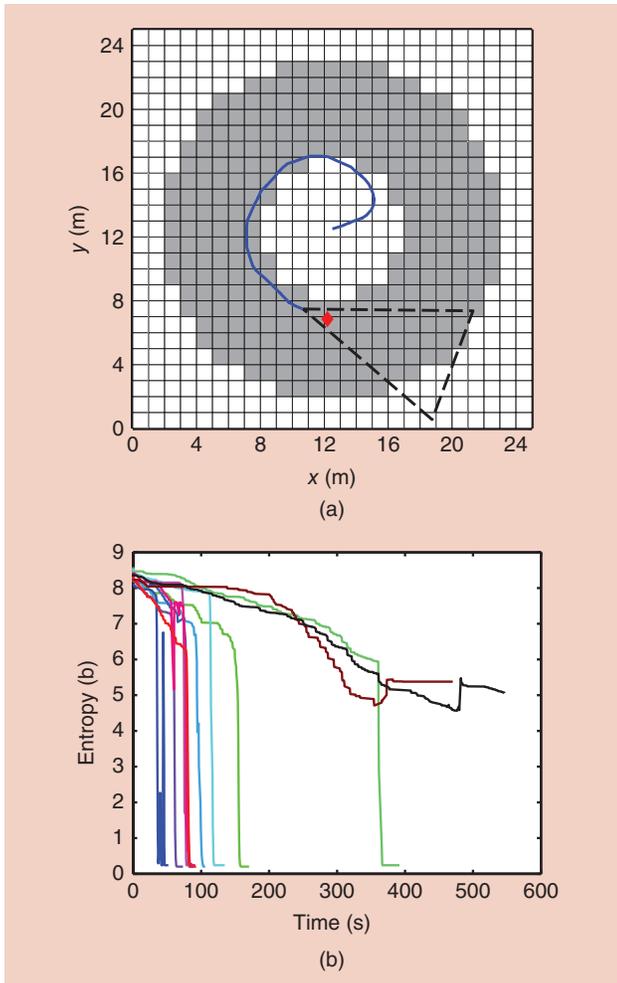


Figure 6. The sample results from the experimental data. (a) A typical path taken by the robot, starting from the center of the annulus, is indicated by the solid line, and the final position of the robot and its sensor footprint are indicated by the dashed line. The true object location is indicated by the red diamond, and clutter objects are indicated by green circles. The shaded cells correspond to the nonzero prior probability of the cell containing an object. (b) The time history of the entropy of the distribution of object locations over 12 representative runs.

the prior belief. In this scenario, the robot begins at the center of the environment and searches for targets in an annular region surrounding it, where the shaded cells have nonzero probability of occupancy in the prior. We tested two separate cases: 1) where there is a single object in the environment and the robot believes that there are either zero or one target within the environment and 2) where there are two targets (red dog bone toys) and the robot believes there are up to four. In the second case, there were also multiple false targets (dog toys of varying color and shape) placed within the environment.

Single Object

We performed 12 trials with random initial positioning of the object. Since the total number of RFSs is small (the number of cells plus the empty set), we elected to use a fixed grid with 1-m resolution. To see the progression of information gain, the resulting time history of the entropies is shown in Figure 6(b).

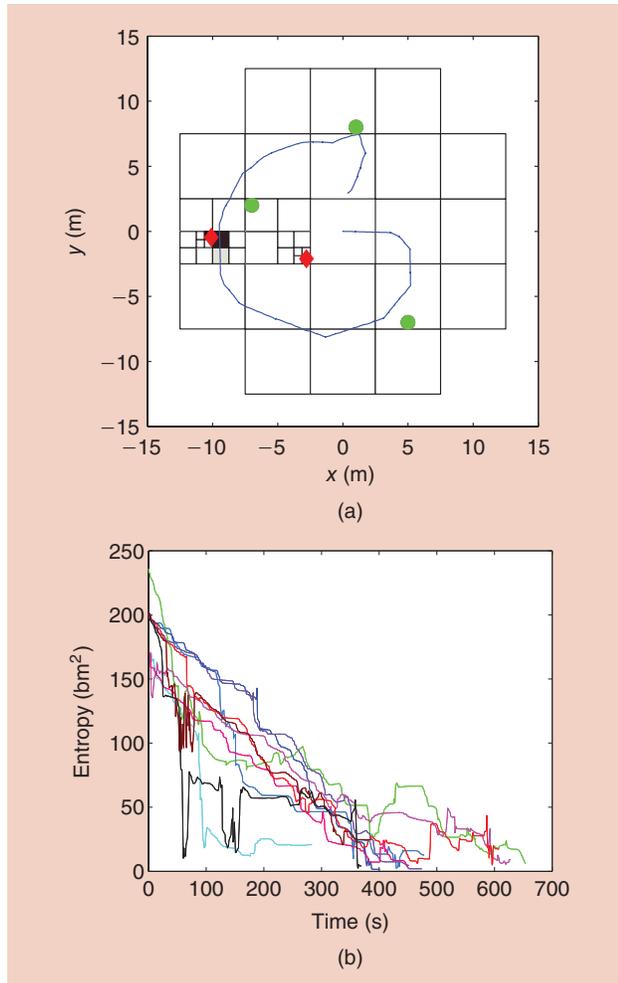


Figure 7. The sample results from the experimental data. (a) A typical path taken by the robot, starting from the center of the annulus, is indicated by the solid line. The true object locations are indicated by the red diamonds. Shaded cells correspond to the nonzero prior probability of the cell containing an object. (b) The time history of the entropy of the distribution of object locations over ten representative runs.

In ten runs, the robot correctly located the object within the precision of the grid. Initially, the entropy decreases slowly as the robot sweeps out some of the area. The sudden drop is due to the fact that the number of objects is limited to one; this causes the distribution to rapidly converge when multiple detections are made in the same cell as that means that the object cannot be in any of the other cells. 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, which are likely due to adverse lighting conditions. The system was able to recover in one such instance, shown by the dark blue line in Figure 6(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.

Two Objects

In this case, the robot began with a coarse grid of 5-m cells with a minimum resolution of 0.625 m. The trials took noticeably longer to complete as the robot must sweep out the entire area to determine that there are no targets present in the cells that are unexplored. This is a result of the maximum number of possible targets (four) being greater than the true number of targets (two). It also means that in this scenario, we do not expect to see the sudden drop in entropy at the end of a trial as we did with a single object. The time evolution of the entropy is shown in Figure 7(b), where the units are given in bits times square meters to take the variable cell size into account as a big cell contains a larger amount of the total uncertainty in an environment than a small cell with the same probability of occupancy. The initial true and clutter object locations were random, in some cases with true objects within one meter of each other. In the ten trials, the robot only failed to detect one of the 20 total targets, and this failure was due to a sudden change in the lighting conditions outdoors. The results of a typical run are shown in Figure 7(a), where the left-most target was not perfectly localized (the cell to the right has a nonzero probability of occupancy) due to the object being located on the cell boundary. However, over the course of these experimental runs (1–10 min), the robot experienced insignificant drift in the position estimate. This may become an issue for much larger environments where the robot is in use for longer periods of time. The robot also never localized a clutter object despite several isolated false positive detections.

Conclusions

In this article, 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 object locations and the probability of detection. The number and locations of objects are modeled using RFSs over an adaptive discretization of the environment, allowing for arbitrarily fine resolution while keeping the computations tractable. 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, the experimental results illustrate the performance of our proposed algorithm by reliably finding the true object locations. It should also be possible to include multiple robots by combining the techniques from this article with those from [4] and [5].

Acknowledgments

We would like to thank SRI International Sarnoff for the object detection and visual odometry work.

References

[1] (2013, Mar.). Robot Operating System. [Online]. Available: <http://www.ros.org/wiki/>

- [2] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proc. IEEE Int. Conf. Intelligent Robots Systems*, 2002, pp. 540–545.
- [3] R. A. Cortez, H. G. Tanner, R. Lumia, and C. T. Abdallah, "Information surfing for radiation map building," *Int. J. Robot. Autom.*, vol. 26, no. 1, pp. 4–12, 2011.
- [4] P. Dames and V. Kumar, "Cooperative multi-target localization with noisy sensors," in *Proc. Int. Conf. Robotics Automation*, May 2013, pp. 1877–1883.
- [5] P. Dames, M. Schwager, V. Kumar, and D. Rus, "A decentralized control policy for adaptive information gathering in hazardous environments," in *Proc. IEEE Conf. Decision Control*, Dec. 2012, pp. 2807–2813.
- [6] A. Das, D. Thakur, J. Keller, S. Kuthirummal, Z. Kira, and M. Pivtoraiko, "R-MASTIF: Robotic mobile autonomous system for threat interrogation and object fetch," in *Proc. SPIE Electronic Imaging*, 2013, pp. 866200–866200-9.
- [7] B. Grocholsky, "Information-theoretic control of multiple sensor platforms," Ph.D. dissertation, Dept. Aerospace, Mechatron. Mech. Eng., Univ. Sydney, Australia, 2002.
- [8] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Trans. Autom. Contr.*, vol. 55, no. 1, pp. 32–47, Jan. 2010.
- [9] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "A scalable information theoretic approach to distributed robot coordination," in *Proc. IEEE/RSJ Conf. Intelligent Robots Systems*, 2011, pp. 5187–5194.
- [10] C. Lundquist, L. Hammarstrand, and F. Gustafsson, "Road intensity based mapping using radar measurements with a probability hypothesis density filter," *IEEE Trans. Signal Processing*, vol. 59, no. 4, pp. 1397–1408, 2010.
- [11] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2007.
- [12] R. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, 2003.
- [13] J. Mullane, B. Vo, M. D. Adams, and V. Ba-Tuong, "A random-finite-set approach to Bayesian SLAM," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 268–282, 2011.
- [14] J. Mullane, B. N. Vo, M. D. Adams, and B. T. Vo, "Random finite sets for robot mapping and slam," in *Springer Tracts in Advanced Robotics*. Berlin, Germany: Springer-Verlag, 2011.
- [15] B. Ristic and B. N. Vo, "Sensor control for multi-object state-space estimation using random finite sets," *Automatica*, vol. 46, no. 11, pp. 1812–1818, 2010.
- [16] B. Ristic, B. N. Vo, and D. Clark, "A note on the reward function for PHD-filters with sensor control," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1521–1529, 2011.
- [17] M. Schwager, P. Dames, D. Rus, and V. Kumar, "A multi-robot control policy for information gathering in the presence of unknown hazards," in *Proc. Int. Symp. Robotics Research*, Aug. 2011.
- [18] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.
- [19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [20] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.
- [21] B. N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, 2005.

Philip Dames, University of Pennsylvania, Philadelphia, United States. E-mail: pdames@seas.upenn.edu.

Dinesh Thakur, University of Pennsylvania, Philadelphia, United States. E-mail: tdinesh@seas.upenn.edu.

Mac Schwager, Boston University, Massachusetts, United States. E-mail: schwager@bu.edu.

Vijay Kumar, University of Pennsylvania, Philadelphia, United States. E-mail: kumar@seas.upenn.edu. 