Efficient Multi-Robot Localization Based on Monte Carlo Approximation

Dieter Fox[1], Wolfram Burgard[2], Hannes Kruppa[3] and Sebastian Thrun[1]

[1]Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15213, {dfox,thrun}@cs.cmu.edu

[2]University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany, burgard@informatik.uni-freiburg.de

[3]ETH Zürich, Department of Computer Science, CH-8092 Zürich, Switzerland, kruppa@inf.ethz.ch

Abstract

This paper presents a probabilistic algorithm for collaborative mobile robot localization. Our approach uses a sample-based version of Markov localization, capable of localizing mobile robots in any-time fashion. When teams of robots localize themselves in the same environment, probabilistic methods are employed to synchronize each robot's belief whenever one robot detects another. As a result, the robots localize themselves faster, maintain higher accuracy, and high-cost sensors are amortized across multiple robot platforms. The paper also describes experimental results obtained using two mobile robots, using computer vision and laser range-finding for detecting each other and estimating each other's relative location. The results, obtained in an indoor office environment, illustrate drastic improvements in localization speed and accuracy when compared to conventional single-robot localization.

1. Introduction

This paper presents an efficient, probabilistic algorithm for collaborative multi-robot localization. Throughout the last decade, sensor-based position estimation has been recognized as a key problem in mobile robotics [1]. The localization problem comes in two flavors: *global localization* and *position tracking*. The second is by far the most-studied problem; here a robot knows its initial position and "only" has to accommodate small errors in its odometry as it moves. The global localization problem involves a robot which is not told its initial position; hence, it has to solve a much more difficult localization problem, that of estimating its position from scratch.

The majority of existing work has focused on the tracking problem, mostly using Kalman-filters [2] to integrate sensor-information over time. However, these approaches are not able to represent the ambiguities occurring during global localization. To overcome the limitations of these techniques, several researchers have recently developed a highly successful family of approaches called *Markov localization* [3, 4, 5, 6]. The central idea of Markov localization is to employ discrete, but *multi-modal* representations for the robot's belief. Implementations of Markov localization have been shown to be able to globally localize a robot even when using highly uncertain sensors such as ultrasound sensors. However, these approaches either provide only coarse information about the robot's position (topological approaches), or they are inefficient in their usage of memory and time (grid-based approaches).

Virtually all existing work addresses localization of a *single* robot only. At first glance, one could solve the problem of localizing multiple robots by localizing each robot *independently*. However, if robots can detect each other, there is the opportunity to do better. When a robot determines the location of another robot relative to its own, both robots can refine their internal beliefs based on the other robot's estimate, hence improve their localization accuracy.

In this paper we propose an efficient, probabilistic approach to collaborative multi-robot localization. By applying a fast, sample-based representation of the robot's belief state, our approach overcomes the disadvantages of earlier techniques. It can globally localize a robot from scratch and is able to efficiently and accurately keep track of its position. Furthermore, the technique makes use of the additional information available when localizing multiple robots simultaneously, thereby significantly increasing the localization performance. To transfer information across different robotic platforms, probabilistic "detection models" are employed to model the robots' abilities to recognize each other. When one robot detects another, these models are used to synchronize the individual robots' beliefs, thereby reducing the uncertainty of both robots during localization.

In what follows, we will first describe our sample-based localization algorithm for single robots and then introduce the necessary statistical mechanisms for multi-robot localization. Thereafter, we present a vision-based method that enables robots to detect each other. Experimental results are presented in Section 5. Finally, we discuss related work and present the conclusions.

2. Monte Carlo Localization

In this section we will introduce our sampling-based approach to single-robot localization. It is based on Markov localization [3, 4, 5, 6], which provides a general framework for estimating the position of a mobile robot. Markov localization maintains a belief $Bel(L)$ over the complete three-dimensional state space of the robot. Here, L denotes a random variable and $Bel(L = l)$ denotes the robot's belief of being at location l, representing it's x-y coordinates (in some Cartesian coordinate system) and its heading direction θ. The belief over the state space is updated whenever the robot moves and senses.

Monte **C**arlo **L**ocalization (in short: MCL [7]) relies on sample-based representations for the robot's belief and the sampling/importance resampling algorithm for belief propagation [8, 9]. The sampling/importance resampling algorithm has been introduced for Bayesian filtering of nonlinear, non-Gaussian dynamic models. It is known alternatively as the bootstrap filter [10], the Monte-Carlo filter [11], the Condensation algorithm [12], or the survival of the fittest algorithm [13]. All these methods are generically known as *particle filters*, and a discussion of their properties can be found in [14].

More specifically, MCL represents the posterior beliefs $Bel(L)$ over the robot's state space by a set of K weighted random samples, or *particles*, denoted $S = \{s_i \mid i = 1..K\}$. A sample set constitutes a discrete distribution. However, under appropriate assumptions (which happen to be fulfilled in MCL), such distributions smoothly approximate the "correct" one at a rate of $1/\sqrt{K}$ as K goes to infinity [15]. Samples in MCL are of the type $\langle l, p \rangle$, where l denotes a robot position in x-y-θ-space, and $p \geq 0$ is a numerical weighting factor, analogous to a discrete probability. For consistency, we assume $\sum_{i=1}^{K} p_i = 1$.

In analogy with the general Markov localization approach, MCL propagates the belief as follows:

1. **Robot motion.** When a robot moves, MCL generates K new samples that approximate the robot's position after a motion measurement a. Each sample is generated by *randomly* drawing a sample from the previously computed sample set, with likelihood determined by their p-values. Let l' denote the $\langle x, y, \theta \rangle$ position of this sample. The new sample's l is then determined by generating a single, random sample from the distribution $P(l \mid l', a)$, using the observed motion a. The p-value of the new sample is K^{-1}. Here $P(l \mid l', a)$ is called the *motion model* of the robot. It models the uncertainty in robot motion [1]. The update of the sample positions corresponds to the *prediction* phase in Kalman filtering.

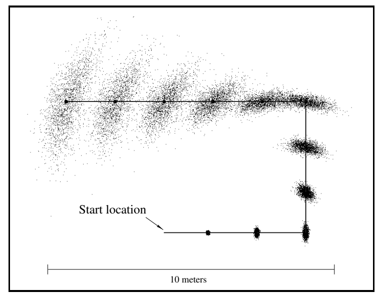

Figure 1: Sampling-based approximation of the position belief for a non-sensing robot.

Figure 1 shows the effect of this sampling technique, starting at an initial known position (bottom center) and executing actions as indicated by the solid line. As can be seen there, the sample sets approximate distributions with increasing uncertainty, representing the gradual loss of position information due to slippage and drift.

2. **Environment measurements** are incorporated by re-weighting the sample set, which is analogous to the application of Bayes rule to the belief state using importance sampling. More specifically, let $\langle l, p \rangle$ be a sample. Then

$$p \quad \longleftarrow \quad \alpha \, P(o \mid l) \tag{1}$$

where o is a sensor measurement, and α is a normalization constant that enforces $\sum_{i=1}^{K} p_i = 1$. $P(o \mid l)$, also called the *environment perception model*, describes the likelihood of perceiving o given that the robot is at position l. The incorporation of sensor readings is typically performed in two phases, one in which p is multiplied by $P(o \mid l)$, and one in which the various p-values are normalized. The reader may notice that this scheme corresponds to the *correction* phase in Kalman filtering.

For proximity sensors such as ultrasound sensors or laser range-finders, the probability $P(o \mid l)$ can be approximated by $P(o \mid o_l)$, which is the probability of observing o conditioned on the expected measurement

[1]Please note, that the motion model can additionally incorporate information about obstacles in the environment by setting the probability of occupied positions to zero.

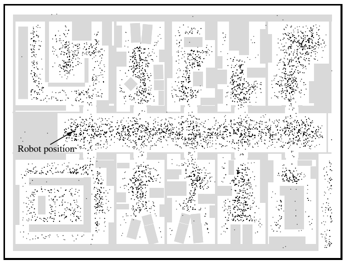

Figure 3: Global localization: Initialization.

Figure 4: Ambiguity due to symmetry.

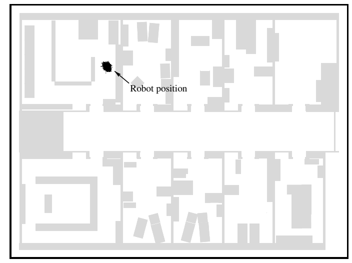

Figure 5: Achieved localization.

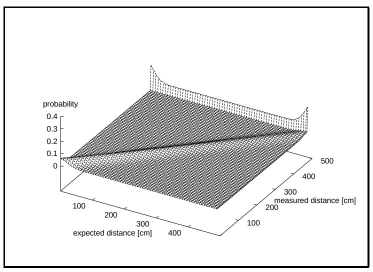

Figure 2: Perception model for sonar sensors. The x-axis depicts the expected measurement, the y-axis the measured distance, and the vertical axis depicts the likelihood. The peak marks the most likely measurement. The robots are also given a map of the environment, to which this model is applied.

o_l at location l. The expected measurement, a distance in this case, is easily computed from the map of the environment using ray tracing. Figure 2 shows a perception model for sonar sensors. Here the x-axis is the distance o_l expected given the world model, and the y-axis is the distance o measured by the sensor. The function is a mixture of a Gaussian (centered around the correct distance o_l), a Geometric distribution (modeling overly short readings) and a Dirac distribution (modeling max-range readings). It integrates the accuracy of the sensor with the likelihood of receiving a "random" measurement (e.g., due to obstacles not modeled in the map [6]).

Figures 3 to 5 illustrate single-robot MCL in practice. Shown there is a series of sample sets (projected into 2D) generated during global localization of our robot RHINO, an RWI B21 robot, as it operates in an office building. In Figure 3, the robot is globally uncertain; hence the samples are spread uniformly through the free-space. Figure 4 shows the sample set after approximately 1 meter of robot motion and several environment measurements obtained by the robot's ultrasound sensors. At this point MCL has disambiguated the robot's position up to a single symmetry. Finally, after another 2 meters of robot motion, the ambiguity is resolved and the robot knows where it is. The majority of samples is now centered tightly around the correct position, as shown in Figure 5.

In practice, we have found it useful to add a small number of uniformly distributed, random samples after each estimation step. By doing so, MCL can recover from localization failures, as documented in our experiments described in [7]. In addition, we apply an adaptive sampling scheme that enables MCL to adjust the number of samples in proportion to the amount of surprise in the sensor data. Consequently, MCL uses few samples when tracking the robot's position, but increases the sample set size when the robot loses track of its position, or otherwise is forced to globally localize itself.

Using our previously developed technique for filtering proximity sensor readings that are caused by obstacles not represented in the map, MCL is able to reliably localize a robot even in highly dynamic environments such as a populated museum [16, 6].

3. Multi-Robot MCL

In this section we will first describe the basic statistical mechanism for multi-robot localization and then it's implementation using MCL.

3.1. Basic Idea

The key idea of multi-robot localization is to integrate measurements taken at different platforms, so that each robot can benefit from data gathered by robots other than itself. At first glance, one might be tempted to maintain a single belief over all robots' locations, i.e., $L = \{L_1, \ldots, L_N\}$, where L_i describes the position of the i-th robot. Unfortunately, the complexity of such joint distributions over L is exponential in the number of robots, which becomes infeasible for larger values of N. To overcome this problem, our approach maintains *factorial* representations; i.e., each robot maintains its own belief function that models only its own uncertainty. Hence, each robot integrates its motion and environment measurements using Monte Carlo localization, as described in the previous section. Occasionally, e.g., when a robot sees another one, informa-

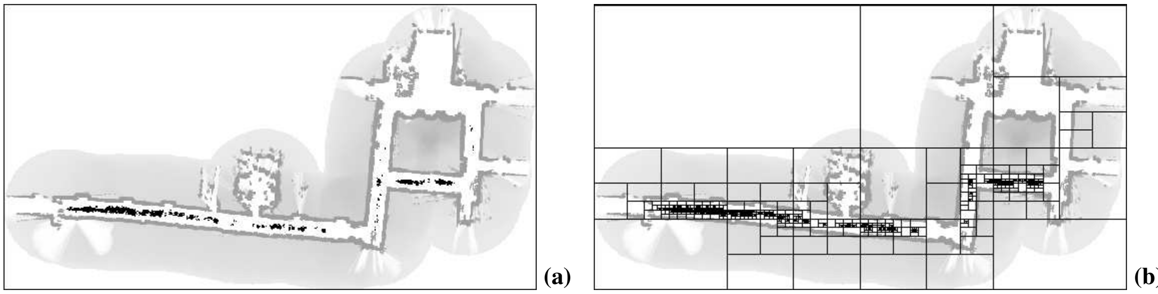

Fig. 6. (a) Map of the environment along with a sample set representing the robot's belief during global localization, and (b) its approximation using a density tree. The tree transforms the discrete sample set into a continuous distribution, which is necessary to generate new importance factors for the individual sample points representing the belief of another robot.

tion from one belief function is transferred to the belief function of another robot.

To see, let us assume that the n-th robot detects robot m and that the detection variable, denoted r_n, provides information about the relative location of the two robots (with $m \neq n$). Then $Bel(L_m = l)$, namely the belief of the detected robot m of being at location l, is updated as follows [2]:

$$Bel(L_m = l) \longleftarrow Bel(L_m = l) \cdot \int P(L_m = l \mid L_n = l', r_n)\ Bel(L_n = l')\ dl' \quad (2)$$

In this equation the term $P(L_m = l \mid L_n = l', r_n)$ is the *robot perception model*. It describes the probability of robot m being at location l, given the current position l' of the detecting robot n and the detection r_n. In our approach, r_n describes the relative location of the detected robot in polar coordinates with Gaussian uncertainty. The reader may notice that, by symmetry, the same detection can be used to constrain the n-th robot's position based on the belief of the m-the robot.

3.2. Sample-based Implementation

The implementation of this scheme using the sample-based density representation of MCL is *not* straightforward. This is because under our factorial representation, each robot maintains its own, local sample set. When one robot detects another, both sample sets have to be synchronized according to Eq. (2). Notice that this equation requires the multiplication of two *densities* which means that we have to establish a correspondence between the individual samples of robot m and the density representing robot n's belief about the position of robot m given by $\int P(L_m = l \mid L_n = l', r_n)\ Bel(L_n = l')\ dl'$.

To solve this problem, our approach transforms sample sets into density functions using *density trees* [18, 19, 20, 21]. These methods approximate sample sets using piecewise constant density functions represented by a tree. Each node in a density tree is annotated with a hyper-rectangular subspace of the three-dimensional state space of the robot. Initially, all samples are assigned to the root node, which covers the entire state space. The tree is grown by recursively splitting each node until a certain stopping condition is fulfilled (see [21] for details). If a node is split, its interval is divided into two equally sized intervals along its longest dimension.

Figure 6(a) shows a map of our testing environment along with a sample set obtained during global localization. The tree generated from this sample set is depicted in Figure 6(b). As can be seen there, the resolution of the tree is a function of the densities of the samples: the more samples exist in a region of space, the more fine-grained the tree representation.

Once a tree representation for the belief of the detecting robot about the detected robot's position is generated, we can integrate the detection into the sample set of the detected robot using importance sampling for each individual sample $\langle l, p \rangle$:

$$p \longleftarrow \alpha \int P(l \mid L_n = l', r_n)\ Bel(L_n = l')\ dl' \quad (3)$$

Please note the similarity to the integration of environment measurements in single-robot MCL in Eq 1.

4. Visual Robot Detection

To implement collaborative multi-robot localization, robots must possess the ability to sense each other. In this section, we briefly describe one possible detection method which integrates camera and range information to estimate the relative position of robots (see [22] for more details).

Our implementation uses camera images to detect robots and to extract their relative position. After detecting another robot, it uses laser ranger finder scans

[2] See [17] for a derivation and a detailed discussion of the involved independence assumptions.

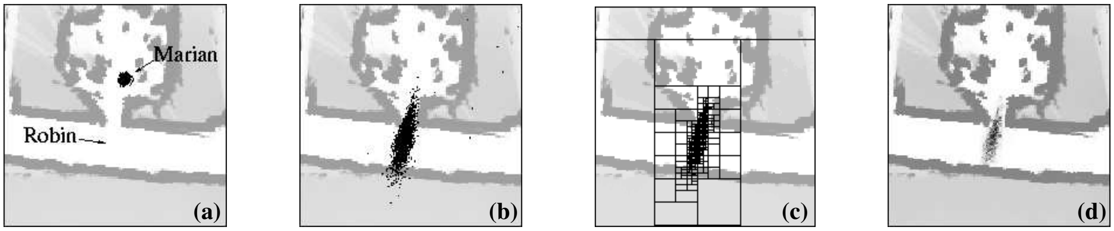

Figure 7. Detection event: (a) Sample set of Marian as it detects Robin in the corridor. (b) Sample set reflecting Marian's belief about Robin's position (see *robot detection model* in Eq. (2)). (c) Tree-representation of this sample set and (d) corresponding density.

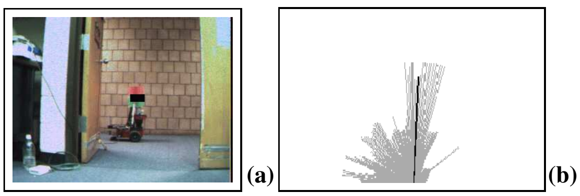

Figure 8: (a) Camera image and (b) laser scan of a detection.

to determine the distance to the robot. Figure 8(a) shows a camera image taken by one of the robots. Each robot is marked by a unique, colored marker to facilitate the recognition. This marker is used to determine the relative angle to the detected robot. Figure 8(b) shows the corresponding laser scan consisting of 180 distance measurements. The dark line in the scan depicts the extracted location of the robot in polar coordinates, relative to the position of the detecting robot. Based on a dataset of 54 successful robot detections, which were labeled by the "true" positions of both robots, we found the mean error of the distance estimation to be 48.26 cm, and the mean angular error to be 2.2 degree. Note that our detection model additionally considers a 6.9% chance of erroneously detecting a robot when there is none.

5. Experimental Results

Our approach was evaluated using two Pioneer robots (Robin and Marian) marked optically by a colored marker, as shown in Figure 8 (a).

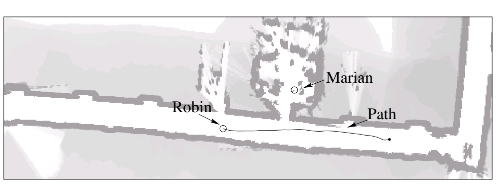

Figure 9: Map of the environment along with a typical path taken by Robin during an experiment.

Figure 9 shows the setup of our experiments along with a part of the occupancy grid map [23] used for position estimation. Marian operates in our lab, which is the cluttered room adjacent to the corridor. Because of the non-symmetric nature of the lab, the robot knows fairly well where it is (the samples representing Marian's belief are plotted in Figure 7 (a)). Figure 9 also shows the path taken by Robin during global localization. Figure 10 (a) represents the typical belief of Robin when it passes the lab in which Marian is operating. Since at this point, Robin has already moved several meters in the corridor, it developed a belief which is centered along the main axis of the corridor. However, the robot is still highly uncertain about its exact location within the corridor and even does not know its global heading direction. Please note that due to the lack of features in the corridor the robots generally have to travel a long distance until they can resolve ambiguities in the belief about their position.

The key event, illustrating the utility of cooperation in localization, is a detection event. More specifically, Marian, the robot in the lab, detects Robin, as it moves through the corridor (see camera image and laser range scan in Figure 8 for a characteristic measurement of this type). Using the detection model described in Section 3.2., Marian generates a new sample set as shown in Figure 7 (b). This sample set is converted into a density using density trees (see Figure 7 (c) and (d)). Marian then transmits this density to Robin which integrates it into its current belief. The effect of this integration on Robin's belief is shown in Figure 10 (b). It shows Robin's belief after integrating the density representing Marian's detection. As this figure illustrates, this single incident almost completely resolves the uncertainty in Robin's belief.

We conducted ten experiments of this kind and compared the performance to conventional MCL for single robots which ignores robot detections. To measure the performance of localization we determined the true locations of the robot by measuring the starting position of each run and performing position tracking off-line using MCL. For each global localization run, we then compared for each sample set the positions of the samples to the reference positions. Fig. 9(c) shows the average distance of the samples from the reference posi-

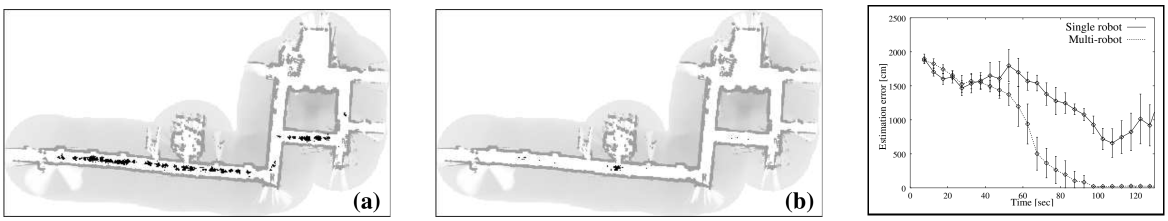

Figure 10: Effect of robot detection: Sample set representing Robin's belief during global localization (a) as it passes Marian in the lab and (b) after incorporating Marian's detection. (c) Estimation error during global localization with and without robot detection.

tions as a function of time, averaged over the ten experiments, along with their 95% confidence intervals (bars). As can be seen here, the quality of position estimation increases much faster when making use of the robot detections (multi-robot localization). Please note that the detection event typically took place 60-100 seconds after the start of an experiment.

Obviously, this experiment is specifically well-suited to demonstrate the advantage of detections in multi-robot localization, since the robots' uncertainties are somewhat orthogonal, making the detection highly effective. In order to test the performance of our approach in more complex situations, we additionally performed experiments involving eight robots in simulation environments. These experiments show that our approach to multi-robot localization significantly speeds up global localization, even when *all* robots perform global localization *at the same time*. Furthermore, one of the experiments shows that under certain conditions, successful localization is only possible if teams of heterogeneous robots collaborate during localization (see [17] for details).

6. Related Work

Mobile robot localization has frequently been recognized as a key problem in robotics with significant practical importance. Cox [24] noted that "Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities." A recent book by Borenstein et al. [25] provides an overview of the state-of-the-art in localization.

Almost all existing approaches address single-robot localization only. These approaches can be distinguished by the way they represent the state space of the robot.

Kalman filter-based techniques. Most of the earlier approaches to robot localization apply Kalman filters [26]. The vast majority of these approaches is based on the assumption that the uncertainty in the robot's position can be represented by a unimodal Gaussian distribution. Sensor readings, too, are assumed to map to Gaussian-shaped distributions over the robot's position. For these assumptions, Kalman filters provide extremely efficient update rules that can be shown to be optimal (relative to the assumptions) [27]. Kalman filter-based techniques [28, 29, 30] have proven to be robust and accurate for keeping track of the robot's position. However, these techniques do not represent multi-modal probability distributions, which frequently occur during global localization. In practice, localization approaches using Kalman filters typically require that the starting position of the robot is known. In addition, Kalman filters rely on sensor models that generate estimates with Gaussian uncertainty—which is often unrealistic (see for example Dellaert et al. [31]). Recently, Jensfelt and Kristensen [32] introduced an approach based on multiple hypothesis tracking, which allows to model typical probability distributions as they occur during global localization.

Topological Markov localization. To overcome the limitations of Kalman filter techniques, different approaches have used increasingly richer schemes to represent uncertainty, moving beyond the Gaussian density assumption inherent in the vanilla Kalman filter. These different methods can be roughly distinguished by the type of discretization used for the representation of the state space. In [33, 3, 4], Markov localization is used for landmark-based corridor navigation and the state space is organized according to the coarse, topological structure of the environment. The coarse resolution of the state representation limits the accuracy of the position estimates. Topological approaches typically give only a rough sense as to where the robot is.

Grid-based Markov localization. To deal with multi-modal and non-Gaussian densities at a fine resolution (as opposed to the coarser discretization in the above methods), grid-based approaches perform numerical integration over an evenly spaced grid of points [5, 34, 6]. This involves discretizing the interesting part of the state space, and use it as the basis for an approximation of the state space density, e.g.

by a piece-wise constant function. Grid-based methods are powerful, but suffer from excessive computational overhead and *a priori* commitment to the size and resolution of the state space. The computational requirements have an effect on accuracy as well, as not all measurements can be processed in real-time, and valuable information about the state is thereby discarded. Recent work [34] has begun to address these problems, using *oct-trees* to obtain a variable resolution representation of the state space. This has the advantage of concentrating the computation and memory usage where needed, and addresses the limitations arising from fixed resolutions.

The issue of cooperation between multiple mobile robots has gained increased interest in the past (see [35, 36] for overviews). In this context most work on localization has focused on the question of how to reduce the odometry error using a cooperative team of robots [37, 38, 39]. While these approaches are very successful in reducing the odometry error, none of them exploits environmental feedback. Even if the initial locations of all robots are known, they ultimately will get lost although at a slower pace than a comparable single robot. The problem addressed here differs in that we are interested in collaborative localization in a global frame of reference, not just reducing odometry error.

7. Conclusions

In this paper we presented an approach to multi-robot localization that uses a sample-based representation of the state space of a robot, resulting in an extremely efficient and robust technique for *global* position estimation. During localization, robots can detect each other. Here we use a combination of camera images and laser range scans to determine another robot's relative location. During localization, detections are used to introduce additional probabilistic constraints that tie one robot's belief to another robot's belief function. To combine sample sets generated at different robots (each robot's belief is represented by a separate sample set), our approach transforms detections into density trees, which approximate discrete sample sets by piecewise constant density functions. These trees are then used to refine the weighting factors (importance factors) of other robots' beliefs, thereby reducing their uncertainty in response to the detection.

As a result, our approach makes it possible to share sensor information collected at different robot platforms. Experimental results demonstrate that our approach yields significantly better localization results than conventional, single-robot localization.

Acknowledgment

This research is sponsored in part by the National Science Foundation, DARPA via TACOM (contract number DAAE07-98-C-L032) and Rome Labs (contract number F30602-98-2-0137), and also by the EC (contract number ERBFMRX-CT96-0049) under the TMR programme. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of NSF, DARPA, TACOM, Rome Labs, the United States Government, or the EC.

References

[1] I.J. Cox. Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, 1991.

[2] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer Verlag, 1990.

[3] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[4] L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1996.

[5] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1996.

[6] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)*, 11:391–427, 1999.

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

[8] D.B. Rubin. Using the SIR algorithm to simulate posterior distributions. In M.H. Bernardo, K.M. an DeGroot, D.V. Lindley, and A.F.M. Smith, editors, *Bayesian Statistics 3*. Oxford University Press, Oxford, UK, 1988.

[9] A.F.M Smith and A.E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *American Statistician*, 46(2):84–88, 1992.

[10] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Procedings F*, 140(2):107–113, 1993.

[11] G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1), 1996.

[12] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[13] K. Kanazawa, D. Koller, and S.J. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proc. of the 11th Annual Conference on Uncertainty in AI (UAI)*, Montreal, Canada, 1995.

[14] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310, Department of Engineering, University of Cambridge, 1998.

[15] M.A. Tanner. *Tools for Statistical Inference*. Springer Verlag, New York, 1993. 2nd edition.

[16] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.

[17] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3), 2000. To appear.

[18] D. Koller and R. Fratkina. Using learning for approximation in stochastic processes. In *Proc. of the International Conference on Machine Learning (ICML)*, 1998.

[19] A.W. Moore, J. Schneider, and K. Deng. Efficient locally weighted polynomial regression predictions. In *Proc. of the International Conference on Machine Learning (ICML)*, 1997.

[20] S. M. Omohundro. Bumptrees for efficient function, constraint, and classification learning. In R. P. Lippmann, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*. Morgan Kaufmann, 1991.

[21] S. Thrun, J. Langford, and D. Fox. Monte Carlo hidden Markov models: Learning non-parametric models of partially observable stochastic processes. In *Proc. of the International Conference on Machine Learning (ICML)*, 1999.

[22] H. Kruppa. Relative multi-robot localization: A probabilistic approach. Master's thesis, ETH Zürich, 1999.

[23] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):27–71, 1998.

[24] I.J. Cox and G.T. Wilfong, editors. *Autonomous Robot Vehicles*. Springer Verlag, 1990.

[25] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[26] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME, Journal of basic engineering*, 82:35–45, March 1960.

[27] P.S. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, New York, 1979.

[28] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic, Boston, MA, 1992.

[29] B. Schiele and J.L. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1994.

[30] J.-S. Gutmann and C. Schlegel. AMOS: Comparison of scan matching approaches for self-localization in indoor environments. In *Proc. of the 1st Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

[31] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[32] P. Jensfelt and S. Kristensen. Active global localisation for a mobile robot using multiple hypothesis tracking. In *Proc. of the IJCAI-99 Workshop on Reasoning with Uncertainty in Robot Navigation*, 1999.

[33] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2), Summer 1995.

[34] W. Burgard, A. Derr, D. Fox, and A.B. Cremers. Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[35] Y.U. Cao, A.S. Fukunaga, and A.B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:1–23, 1997.

[36] R.C. Arkin and T. Balch. Cooperative multiagent robotic systems. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, Cambridge, MA, 1998.

[37] R. Kurazume and N. Shigemi. Cooperative positioning with multiple robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1994.

[38] I.M. Rekleitis, G. Dudek, and E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

[39] J. Borenstein. Control and kinematic design of multi-degree-of-freedom robots with compliant linkage. *IEEE Transactions on Robotics and Automation*, 1995.