In *Autonomous Robots*, 8(3), 2000. To appear.

A Probabilistic Approach to Collaborative Multi-Robot Localization

Dieter Fox[†], Wolfram Burgard[‡], Hannes Kruppa[††], Sebastian Thrun[†]

[†]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[‡]Department of Computer Science
University of Freiburg
D-79110 Freiburg, Germany

[††]Department of Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland

Abstract

This paper presents a statistical algorithm for collaborative mobile robot localization. Our approach uses a sample-based version of Markov localization, capable of localizing mobile robots in an 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 technique has been implemented and tested using two mobile robots equipped with cameras and laser range-finders for detecting other robots. The results, obtained with the real robots and in series of simulation runs, illustrate drastic improvements in localization speed and accuracy when compared to conventional single-robot localization. A further experiment demonstrates that under certain conditions, successful localization is only possible if teams of heterogeneous robots collaborate during localization.

1 Introduction

Sensor-based robot localization has been recognized as one of the fundamental problems in mobile robotics. The localization problem is frequently divided into two subproblems: *Position tracking*, which seeks to compensate small dead reckoning errors under the assumption that the initial position is known, and *global self-localization*, which addresses the problem of localization with no a priori information. The latter problem is generally regarded as the more difficult one, and recently several approaches have provided sound solutions to this problem. In recent years, a flurry of publications on localization—which includes a book solely dedicated to this problem [5]—document the importance of the problem. According to Cox [15], "Using sensory information to locate the robot in its environment is the most fundamental problem to providing a mobile robot with autonomous capabilities."

However, virtually all existing work addresses localization of a *single* robot only. The problem of cooperative multi-robot localization remains virtually unexplored. At first glance, one could solve the problem of localizing N robots by localizing each robot *independently*, which is a valid approach that might yield reasonable results in many environments. 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. The ability to exchange information during localization is particularly attractive in the context of

global localization, where each sight of another robot can reduce the uncertainty in the estimated location dramatically.

The importance of exchanging information during localization is particularly striking for heterogeneous robot teams. Consider, for example, a robot team where some robots are equipped with expensive, high-accuracy sensors (such as laser range-finders), whereas others are only equipped with low-cost sensors such as sonar sensors. By transferring information across multiple robots, sensor information can be leveraged. Thus, collaborative multi-robot localization facilitates the amortization of high-end high-accuracy sensors across teams of robots. Consequently, phrasing the problem of localization as a collaborative one offers the opportunity of improved performance from less data.

This paper proposes an efficient probabilistic approach for collaborative multi-robot localization. Our approach is based on *Markov localization* [51, 62, 34, 9], a family of probabilistic approaches that have recently been applied with great practical success to single-robot localization [7, 39, 23, 67]. In contrast to previous research, which relied on grid-based or coarse-grained topological representations of a robot's state space, our approach adopts a sampling-based representation [17, 21], which is capable of approximating a wide range of belief functions in real-time. 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 detection models are used to synchronize the individual robots' beliefs, thereby reducing the uncertainty of both robots during localization. To accommodate the noise and ambiguity arising in real-world domains, detection models are probabilistic, capturing the reliability and accuracy of robot detection. The constraint propagation is implemented using sampling, and density trees [38, 49, 52, 53] are employed to integrate information from other robots into a robot's belief.

While our approach is applicable to any sensor capable of (occasionally) detecting other robots, we present an implementation that uses color cameras and laser range-finders for robot detection. The parameters of the corresponding probabilistic detection model are learned using a maximum likelihood estimator. Extensive experimental results, carried out with two robots in an indoor environment, illustrate the appropriateness of the approach.

In what follows, we will first describe the necessary statistical mechanisms for multi-robot localization, followed by a description of our sampling-based and Monte Carlo localization technique in Section 3. In Section 4 we present our vision-based method to detect other robots. Experimental results are reported in Section 5. Finally, related work is discussed in Section 6, followed by a discussion of the advantages and limitations of the current approach.

2 Multi-Robot Localization

Let us begin with a mathematical derivation of our approach to multi-robot localization. In the remainder we assume that robots are given a model of the environment (e.g., a map [66]), and that they are given sensors that enable them to relate their own position to this model (e.g., range finders, cameras). We also assume that robots can detect each other, and that they can perform dead-reckoning. All of these senses are typically confounded by noise. Further below, we will make the assumption that the environment is Markov (i.e., the robots' positions are the only measurable state), and we will also make some additional assumptions necessary for factorial representations of joint probability distributions—as explained further below.

Throughout this paper, we adopt a probabilistic approach to localization. Probabilistic methods have been applied with remarkable success to single-robot localization [51, 62, 34, 9, 23, 8, 29], where they

have been demonstrated to solve problems like global localization and localization in dense crowds.

2.1 Data

Let N be the number of robots, and let d_n denote the data gathered by the n-th robot, with $1 \leq n \leq N$. Obviously, each d_n is a sequence of three different types of information:

1. **Odometry measurements.** Each robot continuously monitors its wheel encoders (dead-reckoning) and generates, in regular intervals, odometric measurements. These measurements, which will be denoted a, specify the relative change of position according to the wheel encoders.

2. **Environment measurements.** The robots also query their sensors (e.g., range finders, cameras) in regular time intervals, which generates measurements denoted by o. The measurements o establish the necessary reference between the robot's local coordinate frame and the environment's frame of reference. In our experiments below, o will be laser range scans or ultrasound measurements

3. **Detections.** Additionally, each robot queries its sensors for the presence or absence of other robots. The resulting measurements will be denoted r. Robot detection might be accomplished through different sensors than environment measurements. Below, in our experiments, we will use a combination of visual sensors (color camera) and range finders for robot detection.

The data of all robots is denoted d with

$$d \;=\; d_1 \cup d_2 \cup \ldots \cup d_N. \tag{1}$$

2.2 Markov Localization

Before turning to the topic of this paper—collaborative multi-robot localization—let us first review a common approach to single-robot localization, which our approach is built upon: Markov localization. Markov localization uses only dead reckoning measurements a and environment measurements o; it ignores detections r. In the absence of detections (or similar information that ties the position of one robot to another), information gathered at different platforms cannot be integrated. Hence, the best one can do is to localize each robot individually, independently of all others.

The key idea of Markov localization is that each robot maintains a belief over its position. The belief of the n-th robot at time t will be denoted $Bel_n^{(t)}(L)$. Here L is a three-dimensional random variable composed of a robot's x-y position and its heading direction θ (we will use the terms *position*, *pose* and *location* interchangeably). Accordingly, $Bel_n^{(t)}(L = l)$ denotes the belief of the n-th robot of being at a specific location l. Initially, at time $t = 0$, $Bel_n^{(0)}(L)$ reflects the initial knowledge of the robot. In the most general case, which is being considered in the experiments below, the initial position of all robots is unknown, hence $Bel_n^{(0)}(L)$ is initialized by a uniform distribution.

At time t, the belief $Bel_n^{(t)}(L)$ is the posterior with respect to all data collected up to time t:

$$Bel_n^{(t)}(L) \;=\; P(L_n^{(t)} \mid d_n^{(t)}) \tag{2}$$

where $d_n^{(t)}$ denotes the data collected by the n-th robot *up to* time t. By assumption, the most recent sensor measurement in $d_n^{(t)}$ is either an environment or an odometry measurement. Both cases are treated differently, so let's consider the former first:

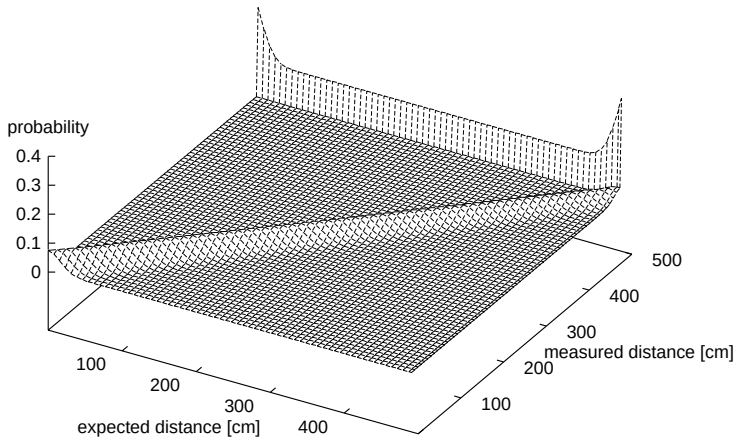

Fig. 1: Perception model for laser range finders. The x-axis depicts the expected measurement, the y-axis the measured distance, and the vertical axis depicts the likelihood. The peak marks the most likely measurement. The robots are also given a map of the environment, to which this model is applied.

---

1. **Sensing the environment:** Suppose the last item in $d_n^{(t)}$ is an environment measurement, denoted $o_n^{(t)}$. Using the Markov assumption (and exploiting that the robot position does not change when the environment is sensed), we obtain for any location l

$$\begin{aligned} Bel_n^{(t)}(L=l) &= P(L_n^{(t)}=l \mid d_n^{(t)}) \\ &= \frac{P(o_n^{(t)} \mid L_n^{(t)}=l, d_n^{(t-1)})\ P(L_n^{(t)}=l \mid d_n^{(t-1)})}{P(o_n^{(t)} \mid d_n^{(t-1)})} \\ &= \frac{P(o_n^{(t)} \mid L_n^{(t)}=l)\ P(L_n^{(t)}=l \mid d_n^{(t-1)})}{P(o_n^{(t)} \mid d_n^{(t-1)})} \\ &= \alpha\ P(o_n^{(t)} \mid L_n^{(t)}=l)\ P(L_n^{(t)}=l \mid d_n^{(t-1)}) \\ &= \alpha\ P(o_n^{(t)} \mid L_n^{(t)}=l)\ P(L_n^{(t-1)}=l \mid d_n^{(t-1)}) \\ &= \alpha\ P(o_n^{(t)} \mid L_n^{(t)}=l)\ Bel_n^{(t-1)}(L=l) \end{aligned} \tag{3}$$

where α is a normalizer that does not depend on the robot position l. Notice that the posterior belief $Bel_n^{(t)}(L=l)$ of being at location l after incorporating $o_n^{(t)}$ is obtained by multiplying the perceptual model $P(o_n^{(t)} \mid L_n^{(t)}=l)$ with the prior belief $Bel_n^{(t-1)}(L=l)$.

This observation suggests the following *incremental* update equation (we omit the time index t and the state variable L for brevity):

$$Bel_n(l) \quad \longleftarrow \quad \alpha P(o_n \mid l)\ Bel_n(l) \tag{4}$$

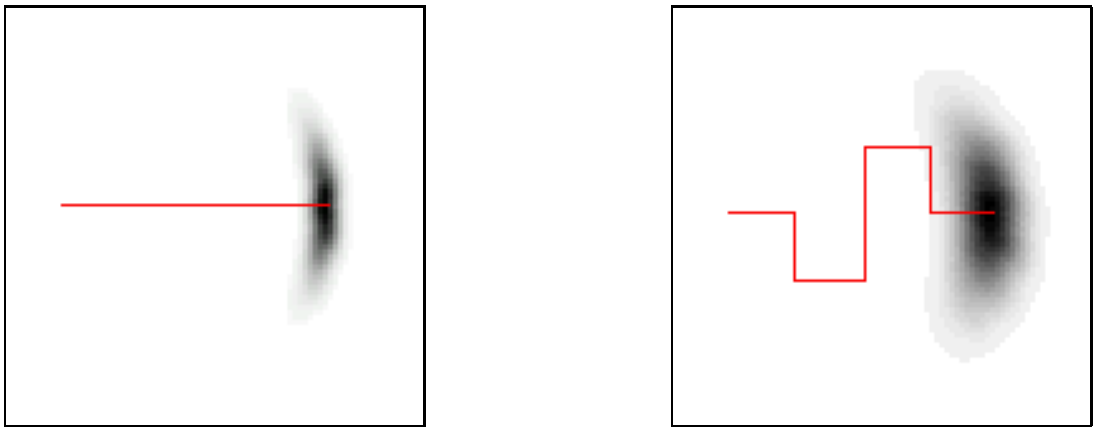

Fig. 2: Motion model representing the uncertainty in robot motion: The robot's belief starts with a Dirac distribution and the lines represent the trajectories of the robot. Both distributions are three-dimensional (in $\langle x, y, \theta \rangle$-space) and shown are their 2D projections into $\langle x, y \rangle$-space.

The conditional probability $P(o_n \mid l)$ is called the *environment perception model* of robot n and describes the likelihood of perceiving o_n given that the robot is at position l. In Markov localization, it is assumed to be given and constant over time. For proximity sensors such as ultrasound sensors or laser range-finders, the probability $P(o_n \mid l)$ can be approximated by $P(o_n \mid o_l)$, which is the probability of observing o_n conditioned on the expected measurement o_l at location l. The expected measurement, a distance in this case, is easily computed from the map using ray tracing. Figure 1 shows this perception model for laser range-finders. Here the x-axis is the distance o_l expected given the world model, and the y-axis is the distance o_n 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 [23]).

2. **Odometry:** Now suppose the last item in $d_n^{(t)}$ is an odometry measurement, denoted $a_n^{(t)}$. Using the Theorem of Total Probability and exploiting the Markov property, we obtain

$$\begin{aligned} Bel_n^{(t)}(L = l) &= P(L_n^{(t)} = l \mid d_n^{(t)}) \\ &= \int P(L_n^{(t)} = l \mid d_n^{(t)}, L_n^{(t-1)} = l')\ P(L_n^{(t-1)} = l' \mid d_n^{(t)})\ dl' \\ &= \int P(L_n^{(t)} = l \mid a_n^{(t)}, L_n^{(t-1)} = l')\ P(L_n^{(t-1)} = l' \mid d_n^{(t-1)})\ dl' \\ &= \int P(L_n^{(t)} = l \mid a_n^{(t)}, L_n^{(t-1)} = l')\ Bel_n^{(t-1)}(L = l')\ dl' \end{aligned} \tag{5}$$

which suggests the *incremental* update equation:

$$Bel_n(l) \longleftarrow \int P(l \mid a_n, l')\ Bel_n(l')\ dl' \tag{6}$$

Here $P(l \mid a_n, l')$ is called the *motion model* of robot n. Figure 2 illustrates the resulting densities for two example paths. As the figure suggests, a motion model is basically a model of robot kinematics annotated with uncertainty.

These equations together form the basis of Markov localization, an incremental probabilistic algorithm for estimating robot positions. Markov localization relies on knowledge of $P(o_n \mid l)$ and $P(l \mid a_n, l')$. The former conditional typically requires a model (map) of the environment. As noticed above, Markov localization has been applied with great practical success to mobile robot localization. However, it is only applicable to single-robot localization, and cannot take advantage of robot detection measurements. Thus, in its current form it cannot exploit relative information between different robots' positions in any sensible way.

2.3 Multi-Robot Markov Localization

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.,

$$L \;=\; L_1 \times L_2 \times \ldots \times L_N \tag{7}$$

Unfortunately, the dimensionality of this vector growths with the number of robots. Distributions over L are, hence, exponential in the number of robots. Moreover, since each robot position is described by three values (its x-y position and its heading direction θ), L is of dimension $3N$. Thus, modeling the joint distribution of the positions of all robots is infeasible already for small values of N.

Our approach maintains *factorial* representations; i.e., each robot maintains its own belief function that models only its own uncertainty, and occasionally, e.g., when a robot sees another one, information from one belief function is transfered from one robot to another. The factorial representation assumes that the distribution of L is the product of its N marginal distributions:

$$P(L_1^{(t)}, \ldots, L_N^{(t)} \mid d^{(t)}) \;=\; P(L_1^{(t)} \mid d^{(t)}) \cdot \ldots \cdot P(L_N^{(t)} \mid d^{(t)}) \tag{8}$$

Strictly speaking, the factorial representation is only approximate, as one can easily construct situations where the independence assumption does not hold true. However, the factorial representation has the advantage that the estimation of the posteriors is conveniently carried out locally on each robot. In the absence of detections, this amounts to performing Markov localization independently for each robot. Detections are used to provide additional constraints between the estimated pairs of robots, which will lead to refined local estimates.

To derive how to integrate detections into the robots' beliefs, let us assume that robot n is detected by robot m and the last item in $d_m^{(t)}$ is a detection variable, denoted $r_m^{(t)}$. For the moment, let us assume this is the only such detection variable in $d^{(t)}$, and that it provides information about the location of the n-th robot relative to robot m (with $m \neq n$). Then

$$\begin{aligned} Bel_n^{(t)}(L = l) \;&=\; P(L_n^{(t)}{=}l \mid d^{(t)}) \\ &=\; P(L_n^{(t)}{=}l \mid d_n^{(t)})\; P(L_n^{(t)}{=}l \mid d_m^{(t)}) \\ &=\; P(L_n^{(t)}{=}l \mid d_n^{(t)}) \int P(L_n^{(t)}{=}l \mid L_m^{(t)}{=}l', r_m^{(t)}) P(L_m^{(t)}{=}l' \mid d_m^{(t-1)})\; dl' \end{aligned} \tag{9}$$

which suggests the *incremental* update equation:

$$Bel_n(l) \quad \longleftarrow \quad Bel_n(l) \int P(L_n = l \mid L_m = l', r_m)\ Bel_m(l')\ dl' \tag{10}$$

Here $\int P(L_n = l \mid L_m = l', r_m)\ Bel_m(l')\ dl'$ describes robot m's belief about the detected robot's position. The reader may notice that, by symmetry, the same detection can be used to constrain the m-th robot's position based on the belief of the n-the robot. The derivation is omitted since it is fully symmetrical.

Table 1 summarizes the multi-robot Markov localization algorithm. The time index t and the state variable L is omitted whenever possible. Of course, this algorithm is only an approximation, since

for each location l **do** /* *initialize* the belief */

$$Bel_n(l) \quad \longleftarrow \quad P(L_n^{(0)} = l)$$

end for

forever do

if the robot receives new sensory input o_n **do**

for each location l **do** /* apply the *perception model* */

$$Bel_n(l) \quad \longleftarrow \quad \alpha P(o_n \mid l)\ Bel_n(l)$$

end for

end if

if the robot receives a new odometry reading a_n **do**

for each location l **do** /* apply the *motion model* */

$$Bel_n(l) \quad \longleftarrow \quad \int P(l \mid a_n, l')\ Bel_n(l')\ dl'$$

end for

end if

if the robot is detected by the m-th robot **do**

for each location l **do** /* apply the *detection model* */

$$Bel_n(l) \quad \longleftarrow \quad Bel_n(l) \int P(L_n = l \mid L_m = l', r_m)\ Bel_m(l')\ dl'$$

end for

end if

end forever

Table 1: Multi-robot Markov localization algorithm for robot number n.

it makes certain independence assumptions (e.g. it excludes that a sensor reports "I saw a robot, but I cannot say which one"), and strictly speaking it is only correct if there is only a single r in the entire run. Furthermore, repeated integration of another robots' belief according to (9) results in using the same

evidence twice. Hence, robots can get overly confident in their position. To reduce the danger arising from the factorial distribution, our approach uses the following two rules.

1. Our approach ignores *negative* sights, i.e., events where a robot does *not* see another robot.

2. It includes a counter that, once a robot has been sighted, blocks the ability to see the same robot again until the detecting robot has traveled a pre-specified distance (2.5m in our experiments). In our current approach, this distance is based purely on experience and in future work we will test the applicability of formal information-theoretic measures for the errors introduced by our factorized representation (see e.g. [6]).

In our practical experiments described below we did not realize any evidence that these two rules are not sufficient. Instead, our approach to collaborative localization based on the factorial representation still yields superior performance over robot teams with individual localization and without any robot detection capabilities.

3 Sampling and Monte Carlo Localization

The previous section left open how the belief about the robot position is represented. In general, the space of all robot positions is continuous-valued and no parametric model is known that would accurately model arbitrary beliefs in such robotic domains. However, practical considerations make it impossible to model arbitrary beliefs using digital computers.

3.1 Monte Carlo Localization

The key idea here is to approximate belief functions using a Monte Carlo method. More specifically, our approach is an extension of Monte Carlo localization (MCL), which was recently proposed in [17, 21]. MCL is a version of Markov localization that relies on sample-based representations and the sampling/importance re-sampling algorithm for belief propagation [58]. MCL represents the posterior beliefs $Bel_n(L)$ 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 and samples in MCL are of the type

$$s_i \;=\; \langle l_i, p_i \rangle \tag{11}$$

where $l_i = \langle x_i, y_i, \theta_i \rangle$ denotes a robot position, and $p_i \geq 0$ is a numerical weighting factor, analogous to a discrete probability. For consistency, we assume $\sum_{i=1}^{K} p_i = 1$. In the remainder we will omit the subscript i whenever possible.

In analogy with the general Markov localization approach outlined in Section 2, MCL proceeds in two phases:

1. **Robot motion.** When a robot moves, MCL generates K new samples that approximate the robot's position after the motion command. 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 position of this sample. The new sample's l is then generated by generating a single, random sample from $P(l \mid l', a)$, using the odometry measurement a. The p-value of the new sample is K^{-1}.

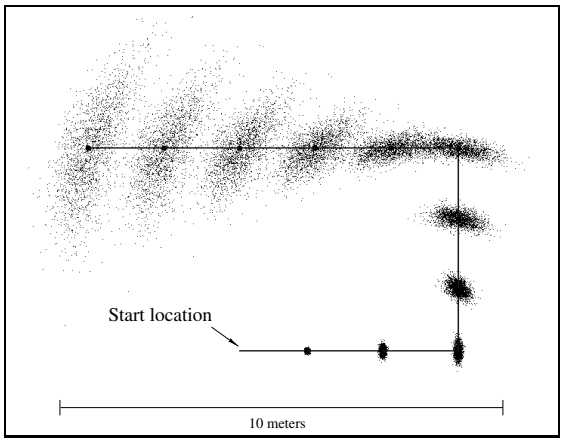

Fig. 3: Sampling-based approximation of the position belief for a non-sensing robot. The solid line displays the trajectory, and the samples represent the robot's belief at different points in time.

Figure 3 shows the effect of this sampling technique for a single robot, 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 Bayes rule in Markov localization. More specifically, let $\langle l, p \rangle$ be a sample. Then

$$p \quad \longleftarrow \quad \alpha \, P(o \mid l) \tag{12}$$

where o is a sensor measurement, and α is a normalization constant that enforces $\sum_{i=1}^{K} p_i = 1$. 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. An algorithm to perform this re-sampling process efficiently in $O(K)$ time is given in [12].

In practice, we have found it useful to add a small number of uniformly distributed, random samples after each estimation step [21]. Formally, these samples can be understood as a modified motion model that allows, with very small likelihood, arbitrary jumps in the environment. The random samples are needed to overcome local minima: Since MCL uses finite sample sets, it may happen that no sample is generated close to the correct robot position. This may be the case when the robot loses track of its position. In such cases, MCL would be unable to re-localize the robot. By adding a small number of random samples, however, MCL can effectively re-localize the robot, as documented in our experiments described in [21] (see also the discussion on 'loss of diversity' in [18]).

Another modification to the basic approach is based on the observation that the best sample set sizes can vary drastically [38]. During global localization, a robot may be completely ignorant as to where it is; hence, it's belief uniformly covers its full three-dimensional state space. During position tracking, on the other hand, the uncertainty is typically small. MCL determines the sample set size on-the-fly: It typically uses many samples during global localization or if the position of the robot is lost, and only a small number of samples is used during position tracking (see [21] for details).

Fig. 4: Global localization: (a) Initialization, (b) ambiguity due to symmetry, and (c) achieved localization.

3.1.1 Properties of MCL

MCL is based on a family of techniques generically known as *particle filters*, or sampling/importance resampling [58]. An overview and discussion of the properties of these filters can be found in [18]. Particle filters are known alternatively as the bootstrap filter [26], the Monte-Carlo filter [37], the Condensation algorithm [32, 33], or the survival of the fittest algorithm [35].

A nice property of particle filters is that they can universally approximate arbitrary probability distributions. As shown in [64], the sample-based distributions smoothly approximate the "correct" one at a rate of $1/\sqrt{K}$ as K goes to infinity and under conditions that are true for MCL. The sample set size naturally trades off accuracy and computation. The true advantage, however, lies in the way MCL places computational resources. By sampling in proportion to the likelihood, MCL focuses its computational resources on regions with high likelihood, where things really matter.

MCL also lends itself nicely to an any-time implementation [16, 72]. Any-time algorithms can generate an answer at *any* time; however, the quality of the solution increases over time. The sampling step in MCL can be terminated at any time. Thus, when a sensor reading arrives, or an action is executed, sampling is terminated and the resulting sample set is used for the next operation.

3.1.2 A Global Localization Example

Figure 4(a) – (c) illustrates MCL when applied to localization of a single mobile robot. Shown there is a series of sample sets (projected into 2D) generated during global localization of the mobile robot Rhino operating in an office building. In Figure 4(a), the robot is globally uncertain; hence the samples are spread uniformly over the free-space. Figure 4(b) shows the sample set after approximately 1.5 meters of robot motion, at which point MCL has disambiguated the robot's position mainly up to a single symmetry. Finally, after another 4 meters of robot motion, the ambiguity is resolved, the robot knows where it is. The majority of samples is now centered tightly around the correct position, as shown in Figure 4(c). All necessary computation is carried out in real-time on a low-end PC.

3.2 Multi-Robot MCL

The extension of MCL to collaborative multi-robot localization 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 are synchronized using the detection model, according to the update equation

$$Bel_n(L=l) \quad \longleftarrow \quad Bel_n(L=l) \int P(L_n = l \mid L_m = l', r_m)\, Bel_m(L=l')\, dl' \tag{13}$$

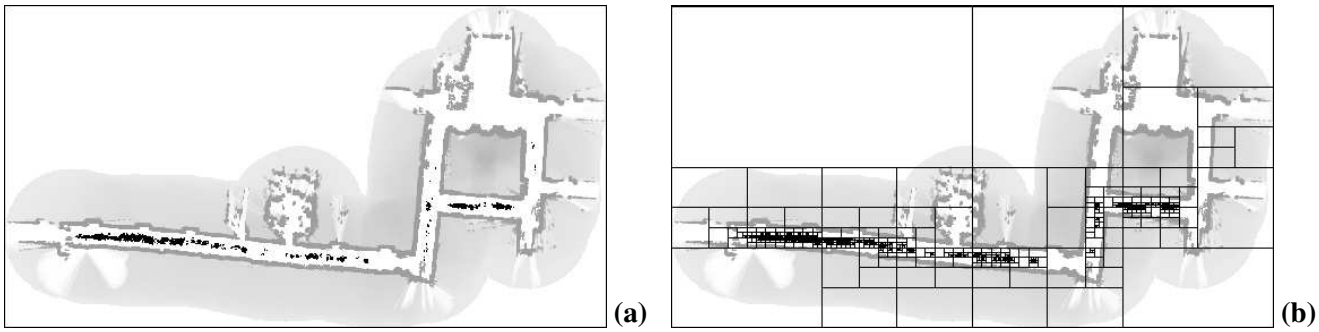

Fig. 5. (a) Map of the environment along with a sample set representing the robot's belief during global localization, and (b) its approximation using a density tree. The tree transforms the discrete sample set into a continuous distribution, which is necessary to generate new importance factors for the individual sample points representing the belief of another robot.

Notice that this equation requires the multiplication of two densities. Since samples in $Bel_n(L)$ and $Bel_m(L)$ are drawn randomly, it is *not* straightforward to establish correspondence between individual samples in $Bel_n(L)$ and $\int P(L_n = l \mid L_m = l', r_m)\ Bel_m(L = l')\ dl'$.

To remedy this problem, our approach transforms sample sets into density functions using *density trees* [38, 49, 52, 53]. 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 [69] for details). If a node is split, its interval is divided into two equally sized intervals along its longest dimension.

Figure 5 shows an example sample set along with the tree extracted from this set. The resolution of the tree is a function of the densities of the samples: the more samples exist in a region of space, the finer-grained the tree representation. After the tree is grown, each leaf's density is given by the quotient of the sum of all weights p of all samples that fall into this leaf, divided by the volume of the region covered by the leaf. The latter amounts to maximum likelihood estimation of (piecewise) constant density functions.

To implement the update equation, our approach approximates the density in Eq. 13 using samples, just as described above. The resulting sample set is then transformed into a density tree. These density values are then multiplied into each individual sample $\langle l, p \rangle$ of the detected robot n according to Eq. 14.

$$p \quad \longleftarrow \quad \alpha \int P(l \mid L_n = l', r_n)\ Bel(L_n = l')\ dl' \tag{14}$$

The resulting sample set is a refined density for the n-th robot, reflecting the detection and the belief of the m-th robot. Please note that the same update rule can be applied in the other direction, from robot n to robot m. Since the equations are completely symmetric, they are omitted here.

4 Probabilistic Detection Model

To implement the multi-robot Monte-Carlo localization technique, robots must possess the ability to sense each other. The crucial component is the detection model $P(L_n = l \mid L_m = l', r_m)$ which describes the conditional probability that robot n is at location l, given that robot m is at location l' and perceives robot n with measurement r_m. From a mathematical point of view, our approach is sufficiently general to

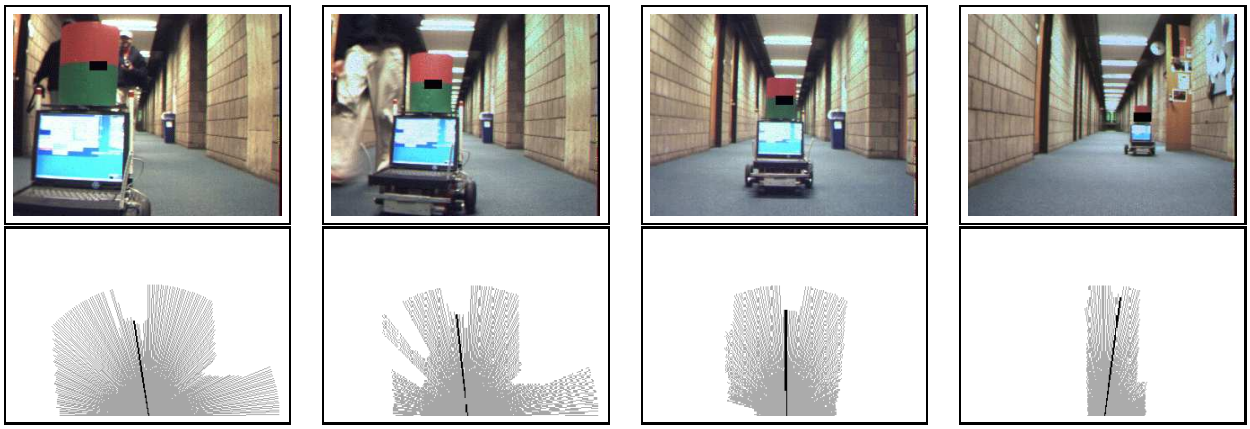

Fig. 6: Training data of successful detections for the robot perception model. Each image in the top row shows a robot, marked by a unique, colored marker to facilitate recognition. The bottom row shows the corresponding laser scans and the dark line in each diagram depicts the extracted location of the robot in polar coordinates, relative to the position of the detecting robot (the laser scans are scaled for illustration purposes).

accommodate a wide range of sensors for robot detection, assuming that the conditional density $P(L_n \mid L_m, r_m)$ is adjusted accordingly.

We will now describe a specific detection method that integrates information from multiple sensor modalities. This method, which integrates camera and range information, will be employed throughout our experiments (see [42] for more details).

4.1 Detection

To determine the relative location of other robots, our approach combines visual information obtained from an on-board camera, with proximity information coming from a laser range-finder. Camera images are used to detect other robots, and laser range-finder scans are used to determine the relative position of the detected robot and its distance. The top row in Figure 6 shows examples of camera images recorded in a corridor. Each image shows a robot, marked by a unique, colored marker to facilitate its recognition. Even though the robot is only shown with a fixed orientation in this figure, the marker can be detected regardless of the robot's orientation.

To find robots in a camera image, our approach first filters the image by employing local color histograms and decision trees tuned to the colors of the marker. Thresholding is then employed to search for the marker's characteristic color transition. If found, this implies that a robot is present in the image. The small black rectangles, superimposed on each marker in the images in Figure 6, illustrate the center of the marker as identified by this visual routine. Currently, images are analyzed at a rate of 1Hz, with the main delay being caused by the camera's parallel port interface.[1] This slow rate is sufficient for the application at hand.

Once a robot has been detected, the current laser scan is analyzed for the relative location of the robot in polar coordinates (distance and angle). This is done by searching for a convex local minimum in the distances of the scan, using the angle obtained from the camera image as a starting point. Here, tight synchronization of photometric and range data is very important, especially because the detecting robot

[1] With a state-of-the-art memory-mapped frame grabber the same analysis would be feasible at frame rate.

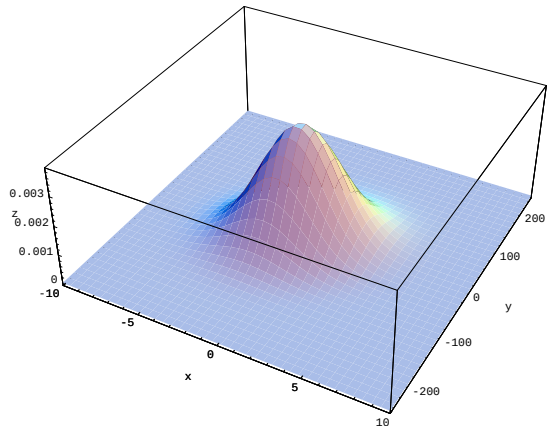

Fig. 7: Gaussian density representing the robot perception model. The x-axis represents the deviation of relative angle and the y-axis the error in the distance between the two robots.

might sense and rotate simultaneously. In our framework, sensor synchronization is fully controllable because all data is tagged with timestamps. We found that the described multi-sensor method is robust and gives accurate results even in cluttered environments. The bottom row in Figure 6 shows laser scans and the result of our analysis for the example situations depicted in the top row of the same figure. Each scan consists of 180 distance measurements with approximately 5cm accuracy, spaced at 1 degree angular distance. The dark line in each diagram depicts the extracted location of the robot in polar coordinates, relative to the position of the detecting robot. All scans are scaled for illustration purposes.

4.2 Learning the Detection Model

Next, we have to devise a probabilistic detection model of the type $P(L_n \mid L_m, r_m)$. To recap, r_m denotes a detection event by the m-th robot, which comprises the identity of the detected robot (if any), and its relative location in polar coordinates. The variable L_n describes the location of the detected robot (here n with $m \neq n$ refers to an arbitrary other robot), and L_m ranges over locations of the m-th robot. As described above, we will restrict our considerations to "positive" detections, i.e., cases where a robot m did detect a robot n. Negative detection events (a robot m does *not* see a robot n) are beyond the scope of this paper and will be ignored.

The detection model is trained using data. More specifically, during training we assume that the exact location of each robot is known. Whenever a robot takes a camera image, its location is analyzed as to whether other robots are in its visual field. This is done by a geometric analysis of the environment, exploiting the fact that the locations of all robots are known during training. Then, the image is analyzed, and for each detected robot the identity and relative location is recorded. This data is sufficient to train the detection model $P(L_n \mid L_m, r_m)$.

	robot detected	no robot detected
robot in field of view	93.3%	6.7%
no robot in field of view	3.5%	96.5%

Table 2: Rates of false-positives and false-negatives for our detection routine.

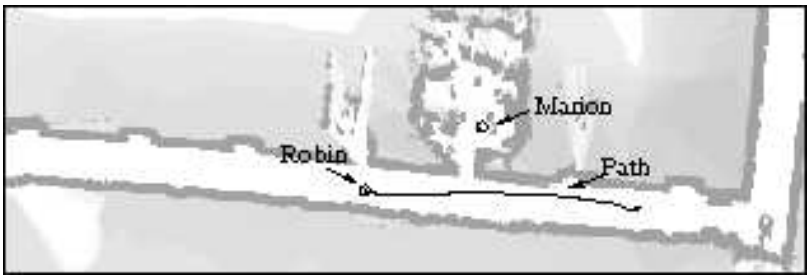

Fig. 8: Map of the environment along with a typical path taken by Robin during an experiment. Marion is operating in the lab facing towards the opening of the hallway.

In our implementation, we employ a parametric mixture model to represent $P(L_m \mid L_n, r_n)$. Our approach models false-positive and false-negative detections using a binary random variable. Table 2 shows the ratios of these errors estimated from a training set of 112 images, in half of which another robot is within the field of view. As can be seen, our current visual routines have a 6.7% chance of not detecting a robot in their visual field, and only a 3.5% chance of erroneously detecting a robot when there is none.

The Gaussian distribution shown in Figure 7 models the error in the estimation of a robot's location. Here the x-axis represents the angular error, and the y-axis the distance error. This Gaussian has been obtained through maximum likelihood estimation based on the training data. As is easy to be seen, the Gaussian is zero-centered along both dimensions, and it assigns low likelihood to large errors. The correlation between both components of the error, angle and distance, are approximately zero, suggesting that both errors might be independent. Assuming independence between the two errors, we found the mean error of the distance estimation to be 48.3cm, and the mean angular error to be 2.2 degree.

To obtain the training data, the "true" location was *not* determined manually; instead, MCL was applied for position estimation (with a known starting position and very large sample sets). Empirical results in [17] suggest that MCL is sufficiently accurate for tracking a robot with only a few centimeters error. The robots' positions, while moving at speeds like 30 cm/sec through our environment, were synchronized and then further analyzed geometrically to determine whether (and where) robots are in the visual fields of other robots. As a result, data collection is extremely easy as it does not require any manual labeling; however, the error in MCL leads to a slightly less confined detection model than one would obtain with manually labeled data (assuming that the accuracy of manual position estimation exceeds that of MCL).

5 Experimental Results

In this section we present experiments conducted with real and simulated robots. The central question driving our experiments was: *To what extent can cooperative multi-robot localization improve the localization quality, when compared to conventional single-robot localization?*

In the first set of experiments, our approach was tested using two Pioneer robots (Robin and Marian) marked optically by a colored marker, as shown in Figure 6. In order to evaluate the benefits of multi-robot localization in more complex scenarios, we additionally performed experiments in simulated environments. These experiments are described in Section 5.2.

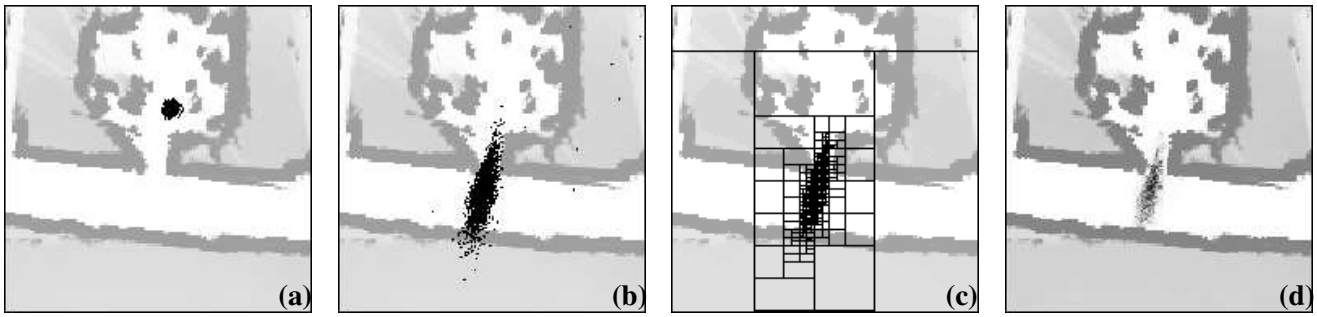

Fig. 9: Detection event: (a) Sample set of Marian as it detects Robin in the corridor. (b) Sample set reflecting Marian's belief about Robin's position. (c) Tree-representation of this sample set and (d) corresponding density.

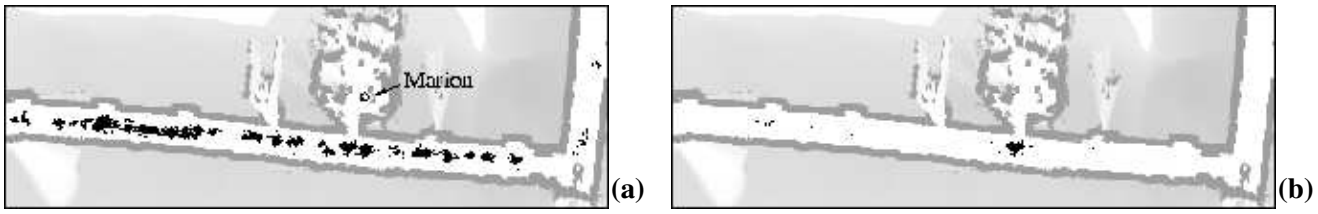

Fig. 10: Sample set representing Robin's belief (a) as it passes Marian and (b) after incorporating Marian's measurement.

5.1 Experiments Using Real Robots

Figure 8 shows the setup of our experiments along with a part of the occupancy grid map [66] 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 9 (a)). Figure 8 also shows the path taken by Robin, which was in the process of global localization. Figure 10 (a) represents the typical belief of Robin when it passes the lab in which Marian is operating. Since Robin 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 Figure 6 for the camera image and laser range scan of a characteristic measurement of this type). Using the detection model described in Section 4, Marian generates a new sample set as shown in Figure 9 (b). This sample set is converted into a density using density trees (see Figure 9 (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 run, we then computed the estimation error at the reference positions.

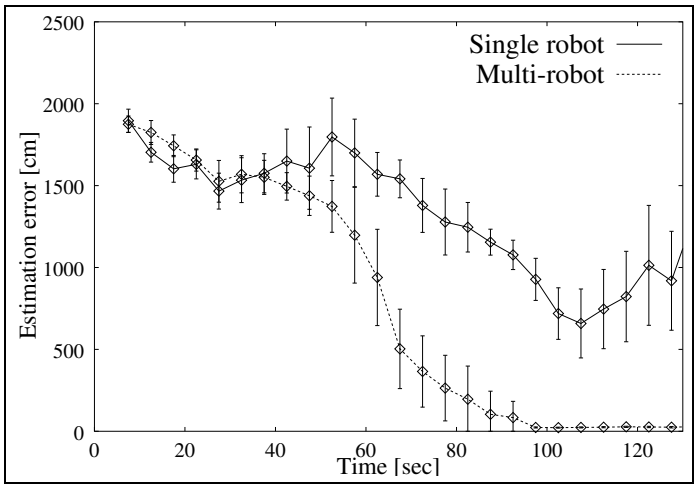

Fig. 11: Comparison between single-robot localization and localization making use of robot detections. The x-axis represents the time and the y-axis represents the estimation error obtained by averaging over ten experiments.

The estimation error is measured by the *average distance of all samples from the reference position*. The results are summarized in Figure 11. The graph plots the estimation error as a function of time, averaged over the ten experiments, along with their 95% confidence intervals (bars). As can be seen in the figure, the quality of position estimation increases much faster when using 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 in two simulation environments.

5.2 Simulation Experiments

In the following experiments we used a simulation tool which simulates robots on the sensor level, providing raw odometry and proximity measurements (see [60] for details). Since the simulation includes sensor noise, the results are directly transferable to real robots. Robot detections were simulated by using the positions of the robots and visibility constraints extracted from the map. Noise was added to these detections according to the errors extracted from the training data using our real robots. It should be noted that false-positive detections were not considered in these experiments (see Section 7.2 for a discussion of false-positive detections).

5.2.1 Homogeneous Robots

In the first simulation experiment we use eight robots, which are all equipped with ultrasound sensors. The task of the robots is to perform global localization in the hallway environment shown in Figure 12 (a). This environment is particularly challenging for single robot systems since a robot has to either pass the open space on the left corridor marked "A", or it has to move through all other hallways marked "B", "C", and "D" to uniquely determine its position. However, the localization task remains hard even if there are multiple robots which can detect each other and can exchange their beliefs. Since all robots have to

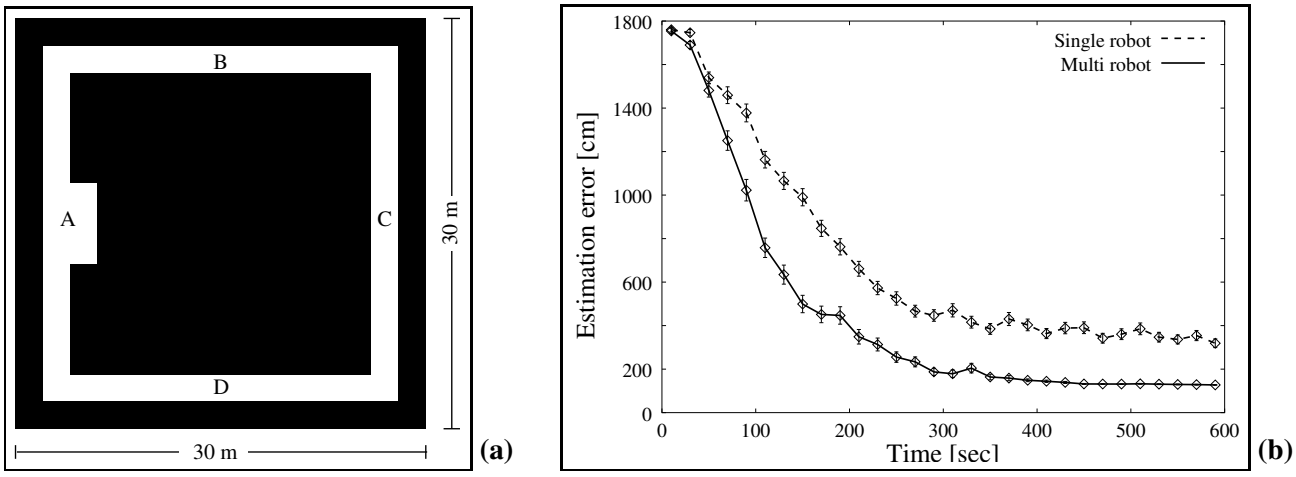

Fig. 12: (a) Symmetric hallway environment. (b) Localization error for eight robots performing global localization simultaneously. The dashed line shows the error over time when performing single-robot MCL and the solid line plots the error using our multi-robot method.

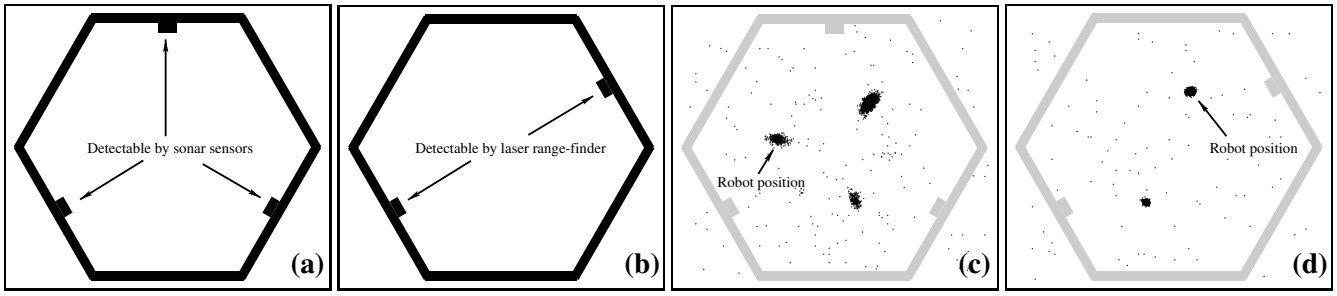

Fig. 13: Hexagonal environment with edges of length 8 meters. Distinguishing obstacles can only be detected either with (a) sonar sensors or (b) laser range-finders. Typical sample sets representing the position uncertainty of robots equipped with (a) sonar sensors or (b) laser range-finders.

perform global localization *at the same time*, several robot detections and belief transfers are necessary to significantly reduce the distance to be traveled by each robot.

As in the previous experiment, we compare the performance of our multi-robot localization approach to the performance of single-robot localization ignoring robot detections. Figure 12 (b) shows the localization errors for both methods averaged over eight runs of global localization using eight robots simultaneously in each run. The plot shows that the exploitation of detections in robot teams results in a highly superior localization performance. The surprisingly high error values for teams not performing collaborative localization are due to the fact that even after 600 seconds, some of the robots are still uncertain about their position.

Another measure of performance is the average time it takes for a robot to find out where it is. We assume that a robot has successfully localized itself, if the localization error falls below 1.5 meters. As mentioned above, this error is given by averaging over the distance of all samples from a reference position. Without making use of robot detections, a robot needs 379 ± 37 seconds to uniquely determine its position. Our approach to multi-robot localization reduces this time by 60% to 153 ± 17 seconds.

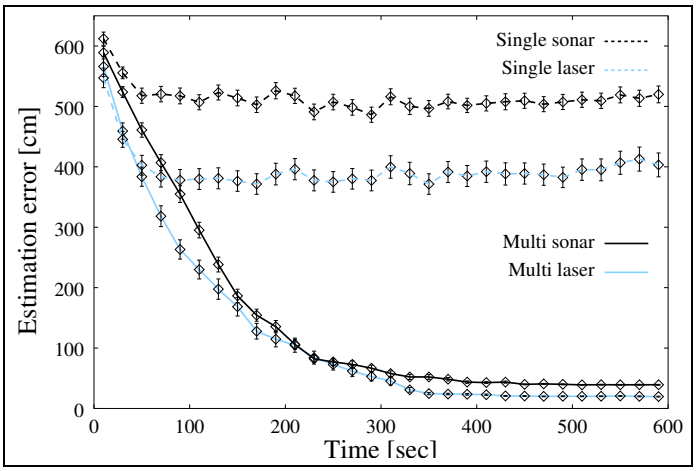

Fig. 14: Localization error for robots equipped with sonar sensors (black lines) or laser range-finders (grey lines). The solid lines summarize results obtained by multi-robot localization and the dashed lines are obtained when ignoring robot detections.

5.2.2 Heterogeneous Robots

The goal of this experiment is to demonstrate the potential benefits for heterogeneous teams of robots. Here, the heterogeneity is due to different types of sensors: One group of robots uses sonar sensors for localization and the other robots are equipped with laser range-finders. The tests are carried out in the environment shown in Figure 13. This environment is highly symmetric and only certain objects allow the robots to reduce their position uncertainty. These objects can be detected either by sonar sensors or by laser range-finders (see Figure 13 (a) and (b)). The position of these obstacles is chosen so that any robot equipped with only one of both sensor types is not able to determine uniquely where it is. Whereas robots using sonar sensors for localization cannot distinguish between three possible robot locations (see Figure 13 (c)), robots equipped with laser range-finders remain uncertain about two possible locations (see Figure 13 (d)).

As in the previous experiment, eight robots are placed in the environment and their task is to find out where they are. Four of the robots are equipped with ultraasound sensors and the other four robots use laser range-finders. The localization error for the different settings is plotted in Figure 14. Not surprisingly, the error for single-robot localization decreases in the beginning of the experiments, but remains at a significantly high level. The corresponding curves are depicted by the dashed lines (sonar black, laser grey) in Figure 14. The results obtained when the robots are able to make use of detections are presented as solid lines (sonar black, laser grey). As can be seen, both teams of robots benefit from the additional information provided by the sensors of the other robots. As a result, each robot is able to uniquely determine its position.

6 Related Work

Mobile robot localization has frequently been recognized as a key problem in robotics with significant practical importance. A recent book by Borenstein, Everett, and Feng [5] provides an excellent overview of the state-of-the-art in localization. Localization plays a key role in various successful mobile robot

architectures presented in [14, 25, 30, 44, 45, 50, 55, 57, 70] and various chapters in [40]. While some localization approaches, such as those described in [31, 41, 62, 34] localize the robot relative to landmarks in a topological map, our approach localizes the robot in a metric space, just like those methods proposed in [2, 65, 68].

Almost all existing approaches address single-robot localization only. Moreover, the vast majority of approaches is incapable of localizing a robot globally; instead, they are designed to track the robot's position by compensating small odometric errors. Thus, they differ from the approach described here in that they require knowledge of the robot's initial position; and they are not able to recover from global localizing failures. Probably the most popular method for tracking a robot's position is Kalman filtering [28, 29, 46, 48, 59, 63], which represents uncertainty by the first and second moments of the density. These approaches are unable to localize robots under global uncertainty—a problem which Engelson called the "kidnapped robot problem" [19]. Recently, several researchers proposed *Markov localization*, which enables robots to localize themselves under global uncertainty [9, 34, 51, 62, 39]. Global approaches have two important advantages over local ones: First, the initial location of the robot does not have to be specified and, second, they provide an additional level of robustness, due to their ability to recover from localization failures. Among the global approaches those using metric representations of the space such as MCL and [9, 8, 39] can deal with a wider variety of environments than those methods relying on topological maps. For example, they are not restricted to orthogonal environments containing pre-defined features such as corridors, intersections and doors.

In addition, most existing approaches are restricted in the type of features that they consider. Many approaches reviewed in [5] are limited in that they require modifications of the environment. Some require artificial landmarks such as bar-code reflectors [20], reflecting tape, ultrasonic beacons, or visual patterns that are easy to recognize, such as black rectangles with white dots [3]. Of course, modifying the environment is not an option in many application domains. Some of the more advanced approaches use more natural landmarks that do not require modifications of the environment. For example, the approaches of Kortenkamp and Weymouth [41] and Matarić [47] use gateways, doors, walls, and other vertical objects to determine the robot's position. The Helpmate robot uses ceiling lights to position itself [36]. Dark/bright regions and vertical edges are used in [13, 71], and hallways, openings and doors are used by the approaches described in [34, 61, 62]. Others have proposed methods for learning what feature to extract, through a training phase in which the robot is told its location [27, 54, 65]. These are just a few representative examples of many different features used for localization. Our approach differs from all these approaches in that it does not extract predefined features from the sensor values. Instead, it directly processes raw sensor data. Such an approach has two key advantages: First, it is more universally applicable since fewer assumptions are made on the nature of the environment; and second, it can utilize all sensor information, typically yielding more accurate results. Other approaches that process raw sensor data can be found in [39, 28, 46].

The issue of cooperation between multiple mobile robots has gained increased interest in the past (see [11, 1] 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. Kurazume and Shigemi [43], for example, divide the robots into two groups. At every point in time only one of the groups is allowed to move, while the other group remains at its position. When a motion command has been executed, all robots stop, perceive their relative position, and use this to reduce errors in odometry. While this method reduces the odometry error of the whole team of robots it is not able to perform global localization; neither can it recover from significant sensor errors. Rekleitis and colleagues [56] present a cooperative exploration method for multiple robots, which also addresses localization. To reduce the odometry error, they use an

approach closely related to the one described in [43]. Here, too, only one robot is allowed to move at any point in time, while the other robots observe the moving one. The stationary robots track the position of the moving robot, thus providing more accurate position estimates than could be obtained with pure dead-reckoning. Finally, in [4], a method is presented that relies on a compliant linkage of two mobile robots. Special encoders on the linkage estimate the relative positions of the robots while they are in motion. The author demonstrates that the dead-reckoning accuracy of the compliant linkage vehicle is substantially improved. However, all these approaches only seek to reduce the odometry error. None of them incorporates environmental feedback into the estimation, and consequently they are unable to localize robots relative to each other, or relative to their environments, from scratch. Even if the initial location of all robots are known, they ultimately will get lost—but at a slower pace than a comparable single robot. The problem addressed in this paper differs in that we are interested in collaborative localization in a global frame of reference, not just reducing odometry error. In particular, our approach addresses cooperative global localization in a known environment.

7 Conclusion

7.1 Summary

We have presented a statistical method for collaborative mobile robot localization. At its core, our approach uses probability density functions to represent the robots' estimates as to where they are. To avoid exponential complexity in the number of robots, a factorial representation is advocated where each robot maintains its own, local belief function. A fast, universal sampling-based scheme is employed to approximate beliefs. The probabilistic nature of our approach makes it possible that teams of robots perform *global localization*, i.e., they can localize themselves from scratch without initial knowledge as to where they are.

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. The "reliability" of the detection routine is modeled by learning a parametric detection model from data, using the maximum likelihood estimator. 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 amortize data collected at multiple platforms. This is particularly attractive for heterogeneous robot teams, where only a small number of robots may be equipped with high-precision sensors.

Experimental results, carried out in real and simulated environments, demonstrate that our approach can reduce the uncertainty in localization significantly, when compared to conventional single-robot localization. In one of the experiments we showed that under certain conditions, successful localization is only possible if teams of heterogeneous robots collaborate during localization. This experiment additionally demonstrates that it is not necessary to equip each robot with a sensor suit needed for global localization. In contrast, one can significantly decrease costs by spreading the different kinds of sensors among multiple platforms, thereby generating a team of heterogeneous robots. Thus, when teams of robots are placed in a known environment with unknown starting locations, our approach can yield significantly better lo-

calization results then conventional, single-robot localization—at lower sensor costs, approximate equal computation costs, and relatively small communication overhead.

7.2 Limitations and Discussion

The current approach possesses several limitations that warrant future research.

Not seeing each other: In our current system, only "positive" detections are processed. Not seeing another robot is also informative, even though not as informative as positive detections. Incorporating such negative detections is generally possible in the context of our statistical framework (using the inverse weighing scheme). However, such an extension would drastically increase the computational overhead, and it is unclear as to whether the effects on the localization accuracy justify the additional computation and communication.

Identification of robots: Another limitation of the current approach arises from the fact that it must be able to identify individual robots—hence they must be marked appropriately. Of course, simple means such as bar-codes can provide the necessary, unique labels. However, because of the inherent uncertainty of their sensors, mobile robots must be able to deal with situations in which they can detect but not identify other robots. The factorial representation, however, cannot deal with measurements such as "either robot A or robot B is straight in front of me." In the worst case, this would require to consider all possible combinations of robots and thus would scale *exponentially* in the number of robots which is equivalent to computing distributions over the joint space of all robots.

Active localization: The collaboration described here is purely passive. The robots combine information collected locally, but they do not change their course of action so as to aid localization. In [10, 22], we proposed an algorithm for *active localization* based on information-theoretic principles, where a single robot actively explores its environment so as to best localize itself. A desirable objective for future research is the application of the same principle to coordinated multi-robot localization.

False-positive detections: As discussed in Section 4, our approach to robot detection has a false-positive rate of 3.5%. This rate describes the chance of erroneously detecting a robot when there is none. While a rate of 3.5% seems to be reasonably low, it turns out to cause major problems if the robots see each other very rarely, which might happen in large environments. In this case, the ratio between true-positive and false-positive detections can fall below one, which means that more than 50% of all detections are false-positive. Our sample-based implementation of multi-robot localization is not robust to such high failure-rates and we did not model false-positive detections in our experiments. One way to handle such failures is to filter them out. First experiments based on the filter techniques introduced in [23, 24] have shown very promising results and will be pursued in future work.

Delayed integration: Finally, the robots update their position instantly whenever they perceive another robot. In situations in which both robots are highly uncertain at the time of the detection it might be more appropriate to delay the update. For example, if one of the robots afterwards becomes more certain by gathering further information about the environment or by being detected by another, certain robot, then the synchronization result can be much better if it is done retrospectively. This, however, requires that the robots keep track of their actions and measurements after detecting other robots.

Despite these open research areas, our approach does provide a sound statistical basis for information exchange during collaborative localization, and empirical results illustrate its appropriateness in practice. These results suggest that robots acting as a team are superior to robots acting individually.

Acknowledgments

Special thanks go to Dirk Schulz at the University of Bonn. Without his help, the simulation runs would not have been possible.

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 authors 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] 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.

[2] M. Betke and L. Gurvits. Mobile robot localization using landmarks. Technical Report SCR-94-TR-474, Siemens Corporate Research, Princeton, December 1993. will also appear in the IEEE Transactions on Robotics and Automation.

[3] J. Borenstein. *The Nursing Robot System*. PhD thesis, Technion, Haifa, Israel, June 1987.

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

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

[6] X. Boyen and D. Koller. Exploiting the architecture of dynamic systems. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1999.

[7] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000. To appear.

[8] 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.

[9] 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.

[10] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

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

[12] J. Carpenter, P. Clifford, and P. Fernhead. An improved particle filter for non-linear problems. Technical report, Department of Statistics, University of Oxford, 1997.

[13] T.S. Collet and B.A. Cartwright. Landmark learning in bees. *Journal of Comparative Physiology*, January 1985.

[14] 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.

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

[16] T. L. Dean and M. Boddy. An analysis of time-dependent planning. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 49–54, 1988.

[17] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.

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

[19] S. Engelson. *Passive Map Learning and Visual Place Recognition*. PhD thesis, Department of Computer Science, Yale University, 1994.

[20] H.R. Everett, D.W. Gage, G.A. Gilbreth, R.T. Laird, and R.P. Smurlo. Real-world issues in warehouse navigation. In *Proc. of the SPIE Conference on Mobile Robots IX*, Boston, MA, November 1994. Volume 2352.

[21] 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.

[22] D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.

[23] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.

[24] 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.

[25] T. Fukuda, S. Ito, N. Oota, F. Arai, Y. Abe, K. Tanake, and Y. Tanaka. Navigation system based on ceiling landmark recognition for autonomous mobile robot. In *Proc. of the International Conference on Industrial Electronics Control and Instrumentation (IECON)*, volume 1, pages 1466 – 1471, 1993.

[26] 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.

[27] R. Greiner and R. Isukapalli. Learning to select useful landmarks. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 1251–1256, Menlo Park, CA, 1994. AAAI Press / The MIT Press.

[28] 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.

[29] J.-S. Gutmann, T. Weigel, and B. Nebel. Fast, accurate, and robust self-localization in polygonal environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999.

[30] R. Hinkel and T. Knieriemen. Environment perception with a laser radar in a fast moving robot. In *Proc. of Symposium on Robot Control*, pages 68.1–68.7, Karlsruhe, Germany, October 1988.

[31] I. Horswill. Specialization of perceptual processes. Technical Report AI TR-1511, MIT, AI Lab, Cambridge, MA, September 1994.

[32] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 343–356, 1996.

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

[34] 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.

[35] 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.

[36] S. King and C. Weiman. Helpmate autonomous mobile robot navigation system. In *Proc. of the SPIE Conference on Mobile Robots*, pages 190–198, Boston, MA, November 1990. Volume 2352.

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

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

[39] K. Konolige. Markov localization using correlation. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[40] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. *AI-based Mobile Robots: Case studies of successful robot systems*, Cambridge, MA, 1998. MIT Press.

[41] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1994.

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

[43] 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.

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

[45] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):89–96, 1992.

[46] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[47] M.J. Matarić. A distributed model for mobile robot environment-learning and navigation. Master's thesis, MIT, Cambridge, MA, January 1990. also available as MIT AI Lab Tech Report AITR-1228.

[48] P.S. Maybeck. The Kalman filter: An introduction to concepts. In Cox and Wilfong [15].

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

[50] H. Neven and G. Schöner. Dynamics parametrically controlled by image correlations organize robot navigation. *Biological Cybernetics*, 75:293–307, 1996.

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

[52] S. M. Omohundro. Efficient algorithms with neural network behavior. *Journal of Complex Systems*, 1(2), 1987.

[53] 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.

[54] S. Oore, G.E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9, 1997.

[55] L. Peters, H. Surmann, S. Guo, K. Beck, and J. Huser. MORIA — Fuzzy Logik gesteuertes, autonomes Fahrzeug. In German, 1994.

[56] 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.

[57] W.D. Rencken. Concurrent localisation and map building for mobile robots using ultrasonic sensors. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1993.

[58] 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.

[59] 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.

[60] D. Schulz, W. Burgard, and A.B. Cremers. Robust visualization of navigation experiments with mobile robots over the internet. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999.

[61] H. Shatkey and L.P. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

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

[63] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial realtionships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.

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

[65] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1), 1998.

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

[67] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, J. Schulte, and D. Schulz. MINERVA: A second-generation museum tour-guide robot. In *Proc. of the International Conference on Robotics and Automation (ICRA '99)*, 1999.

[68] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. Also appeard in *Autonomous Robots* 5, pp. 253–271, joint issue.

[69] 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.

[70] G. Weiß, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1994.

[71] E. Wolfart, R.B. Fisher, and A. Walker. Position refinement for a navigating robot using motion information based on honey bee strategies. In *Proc. of the International Symposium on Robotic Systems (SIR 95)*, Pisa, Italy, 1995.

[72] S. Zilberstein and S. Russell. Approximate reasoning using anytime algorithms. In S. Natarajan, editor, *Imprecise and Approximate Computation*. Kluwer Academic Publishers, Dordrecht, 1995.