A Monte Carlo Algorithm for Multi-Robot Localization

Dieter Fox, Wolfram Burgard,
Hannes Kruppa, Sebastian Thrun

March 1999
CMU-CS-99-120

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

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 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.

This research is sponsored in part by NSF, 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.

Keywords: mobile robots, localization, Markov localization, robotic vision, multi-robot co-operation, Monte Carlo methods

1 Introduction

Sensor-based robot localization has been recognized as of the fundamental problems in mobile robotics. The localization problem is frequently divided into two subproblems: *Position tracking*, which seeks to identify and 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 [6]—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 believes 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 location estimated 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 ultrasonic range finders. By transferring information across multiple robots, high-accuracy sensor information can be leveraged. Thus, collaborative multi-robot localization facilitates the amortization of high-end high-accuracy sensors across teams of robots. Thus, 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* [53, 64, 37, 9], a family of probabilistic approaches that have recently been applied with great practical success to single-robot localization [7, 70, 19, 29]. In contrast to previous research, which relied on grid-based or coarse-grained topological representations, our approach adopts a sampling-based representation [17, 23], 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, detection models are used to synchronize the individual robots' believes, 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 [42, 51, 54, 55] 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 for robot detection. Color images are continuously filtered, segmented, and analyzed, to detect other robots. To obtain accurate probabilistic models of the detection process, a statistical learning technique is employed to learn the parameters of this model using the maximum likelihood estimator. Extensive experimental results, carried out using data collected in two indoor environments, illustrate the appropriateness of the approach.

In what follows, we will first describe the Monte Carlo Localization algorithm for single robots. Section 2 introduces 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. Throughout the derivation, it is assumed that robots are given a model of the environment (e.g., a map [69]), 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 [53, 64, 37, 9, 25, 8], 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 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 queries 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.

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

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

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 $Bel_n^{(t)}(\xi)$. Here ξ denotes a *robot position* (we will use the terms *position*, *pose* and *location* interchangeably), which is typically a three-dimensional value composed of a robot's x-y position and its heading direction θ. Initially, at time $t = 0$, $Bel_n^{(0)}(\xi)$ 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 $Bel_n^{(0)}(\xi)$ is initialized by a uniform distribution.

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

$$Bel_n^{(t)}(\xi) \quad = \quad P(\xi_n^{(t)} \mid d_n^{(t)}) \tag{2}$$

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

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

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

where α is a normalizer that does not depend on $\xi_n^{(t)}$. Notice that the posterior belief $Bel_n^{(t)}(\xi)$ after incorporating $o_n^{(t)}$ is obtained by multiplying the perceptual model $P(o_n^{(t)} \mid \xi_n^{(t)})$ with the prior belief. This observation suggest the *incremental* update equation:

$$Bel_n(\xi_n) \quad \longleftarrow \quad P(o_n^{(t)} \mid \xi_n^{(t)})\ Bel_n(\xi_n) \tag{4}$$

The conditional probability $P(o_n \mid \xi_n)$ is called the *environment perception model* of robot n. In Markov localization, it is assumed to be given. The probability $P(o_n \mid \xi_n)$ can be approximated by $P(o_n \mid o_\xi)$, which is the probability of observing o_n conditioned on the expected measurement o_ξ at location ξ. The expected measurement is easily computed using ray tracing. Figure 1 shows this perception model for laser range finders. Here the x-axis is the distance o_ξ 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 density and a geometric distribution. 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 [22]).

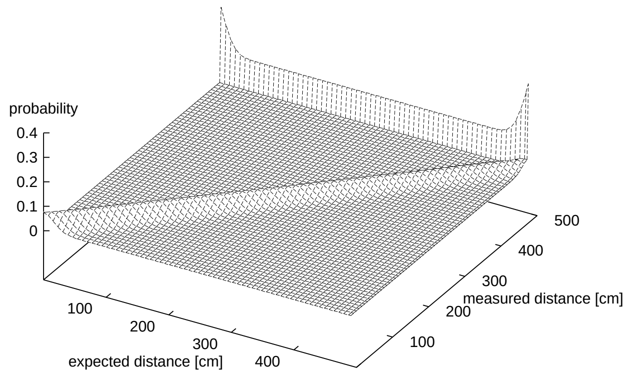

Fig. 1: Perception model for laser range finders. The x-axis depicts the expected measurement, the y-axis the measured distance, and the vertical axis depicts the likelihood. The peak marks the most likely measurement. The robots are also given a map of the environment, to which this model is applied.

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

$$\begin{aligned} Bel_n^{(t)}(\xi) &= P(\xi_n^{(t)} \mid d_n^{(t)}) \\ &= \int P(\xi_n^{(t)} \mid d_n^{(t)}, \xi_n^{(t-1)})\ P(\xi_n^{(t-1)} \mid d_n^{(t)})\ d\xi_n^{(t-1)} \\ &= \int P(\xi_n^{(t)} \mid a_n^{(t)}, \xi_n^{(t-1)})\ P(\xi_n^{(t-1)} \mid d_n^{(t-1)})\ d\xi_n^{(t-1)} \end{aligned} \tag{5}$$

which suggests the *incremental* update equation:

$$Bel(\xi_n) \quad \longleftarrow \quad \int P(\xi_n \mid a_n^{(t)}, \xi_n')\ Bel(\xi_n')\ d\xi_n' \tag{6}$$

Here $P(\xi \mid a, \xi')$ is called the *motion model* of robot n. Figure 2 shows an example for the mobile robots used in our experiments. The straight line represents the trajectory of the robot, which moved straight from left to right. In the beginning $Bel_n^{(0)}(\xi)$ was initialized by a Dirac-Distribution. After 30 meters the robot is highly uncertain about its location which is represented by the "‘banana-shaped"’ distribution $Bel_n^{(t)}(\xi)$. 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. The Markov localization algorithm consists of the following steps:

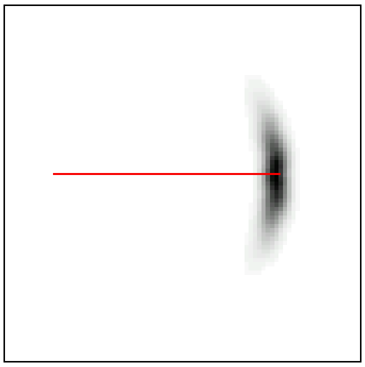

Fig. 2: Motion model representing the uncertainty in robot motion.

Step 1. Initialize $Bel_n(\xi)$ by a uniform distribution.

Step 2.1. For each environment measurement o_n do

$$Bel_n(\xi_n) \quad \longleftarrow \quad P(o_n \mid \xi_n)\ Bel_n(\xi_n). \tag{7}$$

Step 2.2. For each odometry measurement a_n do

$$Bel(\xi_n) \quad \longleftarrow \quad \int P(\xi_n \mid a_n, \xi_n')\ Bel(\xi_n')\ d\xi_n' \tag{8}$$

Thus, Markov localization relies on knowledge of $P(o \mid \xi)$ and $P(\xi \mid a, \xi')$, 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.,

$$\xi \quad = \quad \{\xi_1, \ldots, \xi_N\} \tag{9}$$

Unfortunately, the dimensionality of this vector growths with the number of robots: If each robot position is described by three values (its x-y position and its heading direction θ), ξ is of dimension

$3N$. Distributions over ξ are, hence, exponential in the number of robots. Thus, modeling the joint distribution of the positions of all robots is infeasible for larger 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 ξ is the product of its N marginal distributions:

$$P(\xi_1^{(t)}, \ldots, \xi_N^{(t)} \mid d^{(t)}) \quad = \quad P(\xi_1^{(t)} \mid d^{(t)}) \cdot \ldots \cdot P(\xi_N^{(t)} \mid d^{(t)}) \tag{10}$$

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 of pairs of robots, which will lead to refined local estimates.

To derive how to integrate detections into the robots' beliefs, let us assume the last item in $d_n^{(t)}$ is a detection variable, denoted $r_n^{(t)}$. 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 m-th robot relative to robot n (with $m \neq n$). Then

$$\begin{aligned} Bel_m^{(t)} &= P(\xi_m^{(t)} \mid d^{(t)}) \\ &= P(\xi_m^{(t)} \mid d_m^{(t)}) \; P(\xi_m^{(t)} \mid d_n^{(t)}) \\ &= P(\xi_m^{(t)} \mid d_m^{(t)}) \int P(\xi_m^{(t)} \mid \xi_n^{(t)}, r_n^{(t)}) P(\xi_n^{(t)} \mid d_n^{(t-1)}) \; d\xi_n^{(t-1)} \end{aligned} \tag{11}$$

which suggests incremental update equation:

$$Bel(\xi_m) \quad \longleftarrow \quad Bel(\xi_m) \int P(\xi_m^{(t)} \mid \xi_n^{(t)}, r_n^{(t)}) \; Bel(\xi_n) \; d\xi_n \tag{12}$$

Of course, this is only an approximation, since it makes certain independence assumptions (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. However, this gets us around modeling the joint distribution $P(\xi_1, \ldots, \xi_N \mid d)$, which is computationally infeasible as argued above. Instead, each robot basically performs Markov localization with these additional probabilistic constrains, hence estimates the marginal distributions $P(\xi_n | d)$ separately.

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. The derivation is omitted since it is fully symmetrical.

2.4 Additional Considerations

If the data set contains more than one constraint r between two robots m and n, the situation becomes more complicated. Basically, repeated integration of different robots' belief according to (11) can lead to using the same evidence twice; hence, robots can get overly confident in their position.

In our approach, this effect is diminished by a set of rules that basically reduce the danger arising from the factorial distribution.

1. To diminish these effects, our approach ignored *negative* sights, i.e., events where a robot does *not* see another robot.
2. It also includes timer that, once a robot has been sighted, blocks the ability to see the same robot again for a pre-specified duration.

In practice, these two restrictions are sufficient to yield superior performance, as demonstrated below. However, the reader should notice that they imply that detection information may not be used. At the current point, we are not aware of an approach that would utilize more information yet maintain the highly convenient factorial representations.

3 Sampling and Monte Carlo Localization

The previous section left open how the belief 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.

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, 23]. MCL is a version of Markov localization that relies on sample-based representation and the sampling/importance re-sampling algorithm for belief propagation [60]. MCL represents the posterior beliefs $Bel_n(\xi)$ 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 approximates the "correct" one at a rate of $1/\sqrt{K}$ as K goes to infinity [66].

A particularly elegant algorithm to accomplish this has recently been suggested independently by various authors. It is known alternatively as the bootstrap filter [27], the Monte-Carlo filter [40], the Condensation algorithm [35, 36], or the survival of the fittest algorithm [38]. These methods are generically known as *particle filters*, or sampling/importance re-sampling [60], and an overview and discussion of their properties can be found in [18].

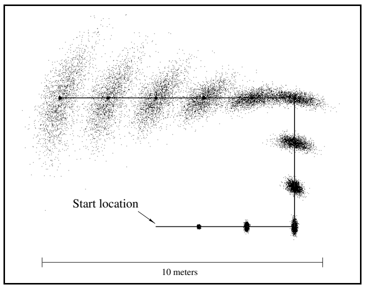

Fig. 3: Sampling-based approximation of the position belief for a non-sensing robot. The solid line displays the actions, and the samples represent the robot's belief at different points in time.

Samples in MCL are of the type

$$\langle\langle x, y, \theta\rangle, p\rangle \tag{13}$$

where $\langle x, y, \theta\rangle$ denote a robot position, 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 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 ξ' denote the position of this sample. The new sample's ξ is then generated by generating a single, random sample from $P(\xi \mid \xi', a)$, using the action a as observed. The p-value of the new sample is K^{-1}.

   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 easily, 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

$$\langle \xi, p\rangle \tag{14}$$

   be a sample. Then

$$p \quad \longleftarrow \quad \alpha\, P(o \mid \xi) \tag{15}$$

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 $P(o \mid \xi)$, 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 [23]. 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 [23] (see also the discussion on 'loss of diversity' in [18]).

3.1 Properties of MCL

A nice property of the MCL algorithm is that it can universally approximate arbitrary probability distributions. As shown in [66], the variance of the importance sampler converges to zero at a rate of $1/\sqrt{N}$ (under conditions that are true for MCL). Thus, at least theoretically MCL is superior to all previous localization approaches that the authors are aware of, in that it can approximate a much larger class of distributions. 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 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, 75]. 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.

3.2 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 each own, local sample set. When one robot detects another, both sample sets are synchronized using the detection model, according

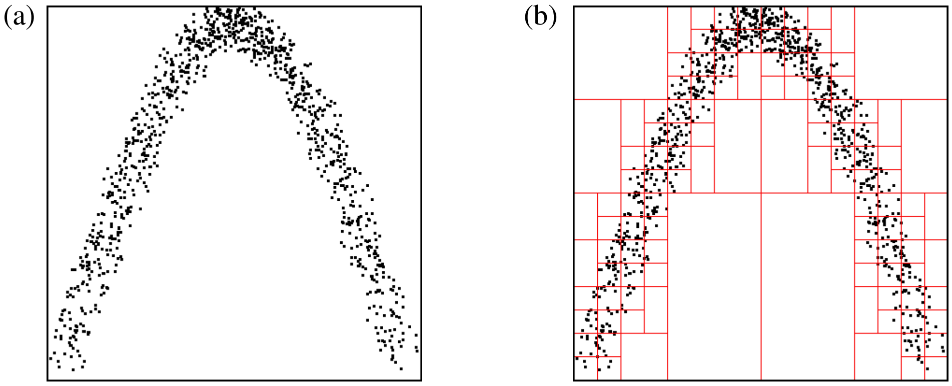

Figure 4: (a) Sample set that corresponds to a detection r, and (b) its approximation using a density tree. The tree transforms the discrete sample set into a continuous distribution, which is necessary to generate new importance factors for the individual sample points representing each robot's belief.

to the update equation

$$Bel(\xi_m) \quad \longleftarrow \quad Bel(\xi_m) \int P(\xi_m^{(t)} \mid \xi_n^{(t)}, r_n^{(t)}) \; Bel(\xi_n) \; d\xi_n \tag{16}$$

Notice that this equation requires the multiplication of two densities. Since samples in $Bel(\xi_m)$ and $Bel(\xi_n)$ are drawn randomly, it is *not* straightforward to establish correspondence between individual samples in $Bel(\xi_m)$ and $\int P(\xi_m^{(t)} \mid \xi_n^{(t)}, r_n^{(t)}) \; Bel(\xi_n) \; d\xi_n$.

To remedy this problem, our approach transforms sample sets into density functions using *density trees* [42, 51, 54, 55]. These methods approximate sample sets using piecewise constant density functions represented by a tree. 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.

Figure 4 shows an example sample set along with the tree that represents this set. Our specific algorithm grows trees by recursively splitting in the center of each coordinate axis, terminating the recursion when the number of samples is smaller than a pre-defined constant. 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 above, our approach approximates the density

$$\int P(\xi_m^{(t)} \mid \xi_n^{(t)}, r_n^{(t)}) \; Bel(\xi_n) \; d\xi_n \tag{17}$$

using samples, just as described above. The resulting sample set is then transformed into a density tree. These density values are then multiplied into the weights (importance factors) of the samples in $Bel(\xi_m)$, effectively multiplying both density functions. The result is a refined density for the m-th robot, reflecting the detection and the belief of the n-th robot.

The same update rule is applied in the other direction, from robot m to robot n. Since the equations are completely symmetric, they are omitted here.

3.3 Adaptive Sampling

In practice, the best sample set sizes can vary drastically [42]. 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 and often focused on lower-dimensional manifolds. For example, when a robot knows its relative position to an adjacent wall but does not know what hallway it is in, the belief is focused on a one-dimensional sub-manifold similar to a road-map. Thus, many more samples are needed during global localization to approximate the true density with high accuracy, than are needed for position tracking.

MCL determines the sample set size on-the-fly. The idea is to use the divergence of $P(\xi_n)$ and $P(\xi_n \mid o_n)$, the belief *before* and *after* sensing, to determine the sample sets. More specifically, both motion data and sensor data is incorporated in a single step, and sampling is stopped whenever the non-normalized sum of weights p (before normalization!) exceeds a threshold η. If the position predicted by odometry is well in tune with the sensor reading, each individual p is large and the sample set remains small. If, however, the sensor reading carries a lot of surprise, as is typically the case when the robot is globally uncertain or when it lost track of its position, the individual p-values are small and the sample set is large.

MCL directly relates to the well-known property that the variance of the importance sampler is a function of the mismatch of the sampling distribution (in our case $P(\xi_n)$) and the distribution that is being approximated with the weighted sample (in our case $P(\xi_n \mid o_n)$). The less these distributions agree, the larger the variance (approximation error). The idea is here to compensate such error by larger sample set sizes, to obtain approximately uniform error.

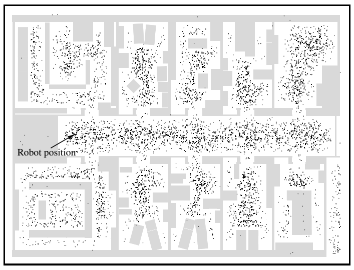

Fig. 5: Global localization: Initialization.

Fig. 6: Ambiguity due to symmetry.

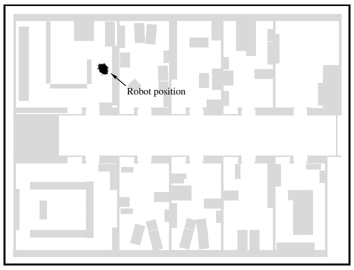

Fig. 7: Achieved localization.

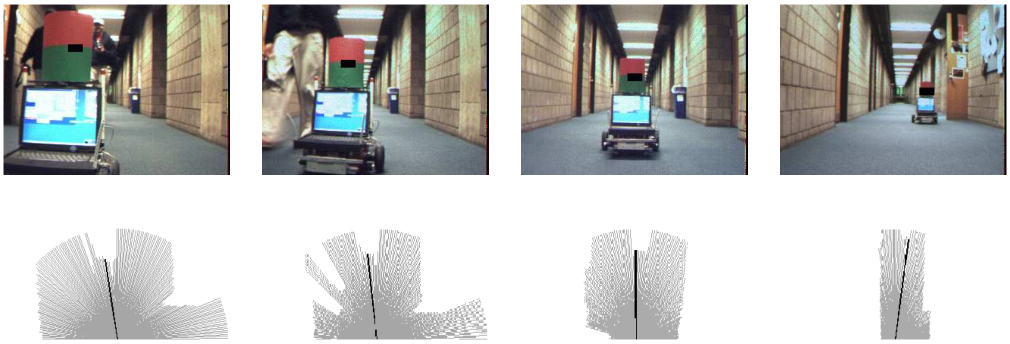

Fig. 8: Training data of successful detections for the robot perception model.

3.4 A Global Localization Example

Figure 5 to 7 illustrate 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 5, the robot is globally uncertain; hence the samples are spread uniformly over the free-space. Figure 6 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 7. All necessary computation is carried out in real-time on a low-end PC.

4 Learning Visual Detection Models

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 $P(\xi_m \mid \xi_n, r_n)$ which describes the conditional probability that robot m is at location ξ_m, given that robot n perceives robot m with measurement r_n. From a mathematical point of view, our approach is sufficiently general to accommodate a wide range of sensors for robot detection, assuming that the conditional density $P(\xi_m \mid \xi_n, r_n)$ 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.

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. Below, in our experiments, only one of the robots is equipped with a camera; however, despite the asymmetry, the information conveyed by a detection enables both robots to refine their internal belief as to where they are, utilizing the other robot's belief.

Camera images are used to detect other robots, and laser ranger finder scans are used to determine the relative position of the detected robot and its distance. The top row in Figure 8 shows examples of camera images recorded in the corridor. Each image shows a robot, marked by unique, colored markers to facilitate their recognition. Even though the robot is only shown with a fixed orientation in this figure, the markers can be detected regardless of a robot's orientation.

To find robots in a camera image, our approach first filters the image using Gaussian color filters tuned to the colors of the markers (see e.g., [34]). The center of the colors are then obtained by local smoothing, and thresholding is applied to determine whether or not a robot can be seen in the image. The small black rectangles, superimposed at the center of each marker in the images in Figure 8, 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 parallel port over which images are transferred from the camera to the computer.[1] This slow rate is sufficient for the application at hand.

Once a robot has been detected, a 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. We found that this method is robust and gives accurate results even in cluttered environments.

The bottom row in Figure 8 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 approx. 5 cm 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. 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 88.7cm, and the mean angular error to be 2.36 degree.

4.2 Learning the Detection Model

Next, we have to devise a probabilistic detection model of the type $P(\xi_m \mid \xi_n, r_n)$. To recap, r_n denotes a detection event by the n-th robot, which comprises the identity of the detected robot (if

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

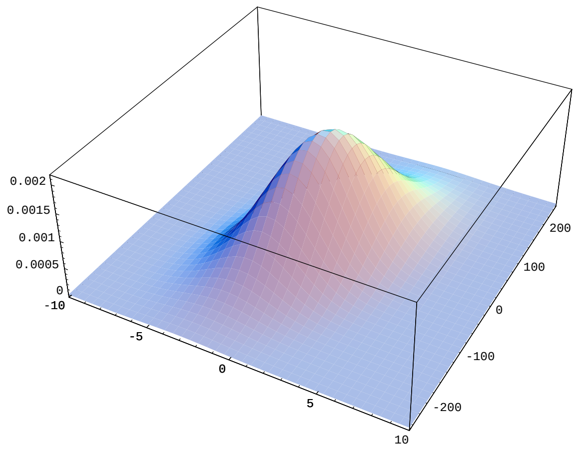

Fig. 9: Gaussian density representing the robot perception model. The x-axis represents the deviation of relative angle and the y-axis the error in the distance between the two robots.

any), and its relative location in polar coordinates. The variable ξ_m is the location of the detected robot (here m with $m \neq n$ refers to an arbitrary other robot), and ξ_n is the location of the n-th robot. As described above, we will restrict our considerations to "positive" detections, i.e., cases where a robot n did detect a robot m. Negative detection events (a robot n does *not* see a robot m) 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 $P(\xi_m \mid \xi_n, r_n)$.

	robot detected	no robot detected
robot in field of view	64.3%	35.7%
no robot in field of view	6.90%	93.1%

Table 1: Rates of false-positives and false-negatives for our detection routine.

In our implementation, we employ a parametric mixture model to represent $P(\xi_m \mid \xi_n, r_n)$. Our approach models false-positive and false-negative detections using a binary random variable. Table 1 shows the ratios of these errors in the training set. As can be see there, our current visual routines have a 35.7% chance of not detecting a robot in their visual field, but only a 6.9% chance to erroneously detecting a robot when there is none.

The Gaussian distribution shown in Figure 9 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. 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.

In our experiments, 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 then 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 that one would obtain with manually labeled data (assuming that the accuracy of manual position estimation exceeds that of MCL).

5 Experimental Results

Our approach was evaluated systematically using the two mobile robots (Robin and Marian) shown in Figure 10. Both robots were marked optically by a colored marker, as shown in Figure 8. The central question driving our experiments war: *Can cooperative multi-robot localization improve the localization accuracy, when compared to conventional single-robot localization?* Put differently, *can the task of global localization sped up significantly when multiple robots cooperate during localization?*

To shed light onto these questions, we operated the robots over extended periods of time in our university building. Figure 11 shows a map of the environment which was learned using a probabilistic mapping algorithm [69, 72]. Notice the long corridor. Due to the lack of features, global localization is quite difficult when the robots operate in this corridor. Previous publications (e,g,. [17, 23]) have analyzed in detail the performance of Markov localization and MCL. Thus, in this paper we will focus on the utility of collaboration and detections in multi-robot localization.

Throughout our experiments, we consistently found that the collaboration reduced the time required for global localization, and it also improved the overall accuracy. Figures 11 to 15 show an example in detail, obtained in one of our experiments.

In particular, Figure 11 shows the belief state of one of the robots, Robin, at a specific point in time while performing global localization. In this specific experiment, the robot previously

Fig. 10: Two of the robots used for testing: Marian (left) and Robin (right).

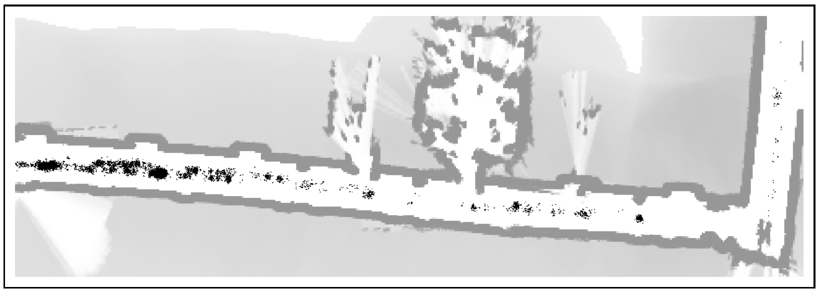

Fig. 11: Belief state of Robin during global localization in a long corridor.

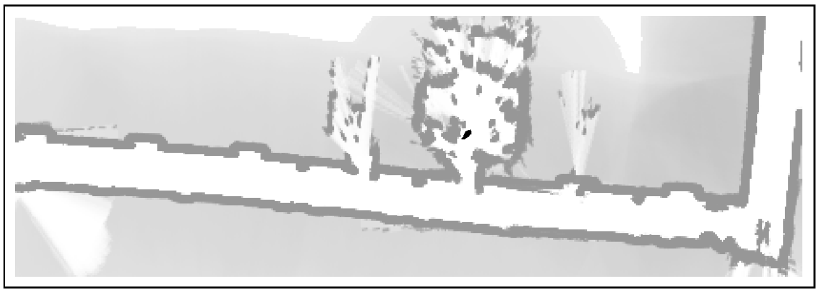

Fig. 12: Belief state of Marian operating in the room.

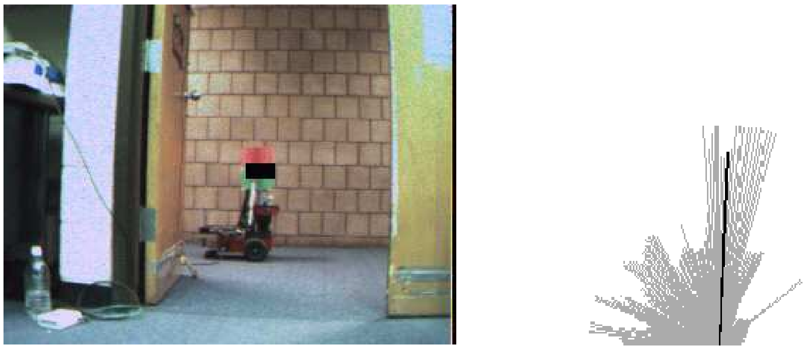

Fig. 13: Image and laser scan Marian uses to determine the relative angle and distance of Robin.

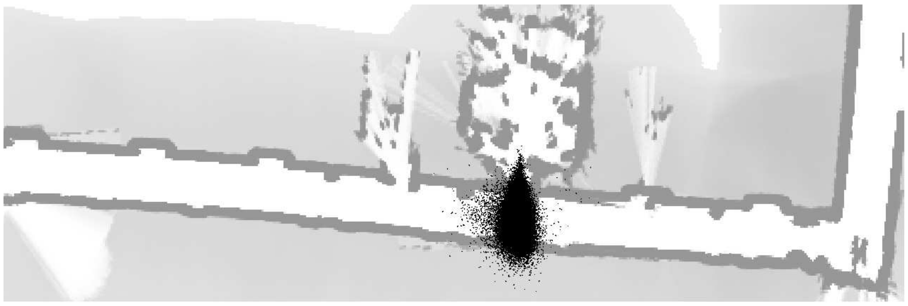

Fig. 14: Sampling-based representation of the density generated by Marian according to the detection of Robin in the current image.

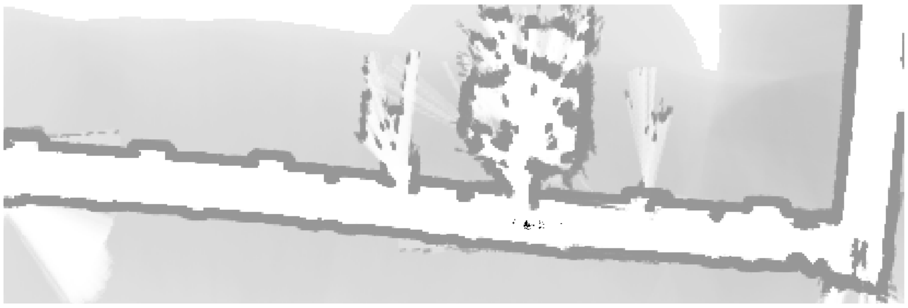

Fig. 15: Belief state of robin after incorporating the measurement of Marian.

traversed the corridor from the right to the left, developing a belief that is centered along the main axis of the corridor. However, the robot is unaware of its exact location within the corridor; neither does it know its global heading direction.

The second robot, Marian, operates in our lab, which is the cluttered room adjacent to the corridor. Its belief is shown in Figure 12. Because of the non-symmetric nature of the lab, the robot knows fairly well where it is.

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. Figure 13 shows the image and the laser scan, along with the estimated distance and orientation. Using the detection model described in Section 4 Marian generates the density shown in Figure:14. It then transmits this density to Robin which integrates it into its current belief. Robin's resulting density is shown in Figure 15. As this figure illustrates, this single incident resolves entirely the uncertainty in Robin's belief—which would have taken minutes if the robots were unable to detect each other.

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. Nevertheless, we consistently observed similarly good performance even when operating the robots in other parts of the environment, e.g., when they both operated in the corridor.

6 Related Work

Mobile robot localization has frequently been recognized as a key problem in robotics with significant practical importance. Cox [15] 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, Everett, and Feng [6] provides an excellent overview of the state-of-the-art in localization. Localization plays a key role in various successful mobile robot architectures [14, 26, 32, 46, 47, 52, 57, 59, 73] and various chapters in [43]. While some localization approaches, such as those described in [33, 44, 64] localize the robot relative to some landmarks in a topological map, our approach localizes the robot in a metric space, just like those methods proposed in [3, 67, 71].

Almost all existing approach 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 [30, 31, 48, 50, 61, 65], which represent uncertainty by single-modal distributions. These approaches are unable to localize robots under global uncertainty—a problem which Engelson called the "kidnapped robot problem" [20]. Recently, several researchers proposed *Markov localization*, which enables robots to localize themselves under global uncertainty [9, 37, 53, 64]. 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 land [9, 8] 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 features that they consider. Many approaches reviewed in [6], a recent book on this topic, are limited in that they require modifications of the environment. Some require artificial landmarks such as bar-code reflectors [21], reflecting tape, ultrasonic beacons, or visual patterns that are easy to recognize, such as black rectangles with white dots [4]. 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 [44] and Matarić [49] use gateways, doors, walls, and other vertical objects to determine the robot's position. The Helpmate robot uses ceiling lights to position itself [39]. Dark/bright regions and vertical edges are used in [13, 74], and hallways, openings and doors are used by the approach described in [41, 62, 63]. Others have proposed methods for learning what feature to extract, through a training phase in which the robot it told its location [28, 56, 67, 68]. 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 [30, 31, 48].

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 how to reduce the odometry error using a cooperative team of robots. Kurazume and Shigemi [45], 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 [58] 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 [45]. 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 [5], 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 roots 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 other 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, represented by tree-like structure, that tie one robot's belief to another robot's belief function. To combine different sample sets collected generated at different robots (each robot's belief is represented by a separate sample set), our approach transforms detections into density trees, which transform discrete sample sets into 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.

Experimental results, carried out in an indoor environment, demonstrate that our approach can reduce the uncertainty in localization significantly, when compared to conventional single-robot localization. Thus, when teams of robots are placed in a known environment with unknown starting locations, our approach can yield much faster localization then conventional, single-robot location—at approximate equal computation costs and relatively small communication overhead.

7.2 Implications for Heterogeneous Robot Teams

Even though the experiment reported here were carried out using homogeneous robots, the work reported here offers some interesting perspectives for teams of heterogeneous robots. Traditionally, heterogeneity has often been suggested as a means to achieve a wide range of tasks, requiring a collection of different actuators, manipulators, or locomotion modalities (wheels, legs). In the context of behavior-based robotics, heterogeneity has often studied the effect of different software architectures on the overall task performance [2].

Our approach can exploit heterogeneity in the robots' sensors. Consider, for example, a team of robots where only a small number of robots are equipped with sensors that support high-accuracy localization. For example, laser range finders typically provide highly accurate range measurements, but they are bulky, expensive, and they consume significantly more energy than comparable, low-accuracy sensors such as sonar sensors. It might therefore be desirable to equip only a small number of robots with laser range finders.

As noted above, our approach makes it possible to amortize sensor data across multiple robotic platforms during localization. Thus, it potentially enables a heterogeneous team of robots to maintain highly accurate location estimates, even if only a small number of robots are equipped with the necessary high-accuracy sensors. In the extreme, one might think of heterogeneous robot teams where only a small number of robots is capable of performing localization. Our approach would enable these robots to localize other robots in the team, not capable of localizing themselves autonomously, thereby provide a unique service to the entire heterogeneous team.

7.3 Limitations and Discussion

The current approach possesses several limitations that warrant future research.

- 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 weighting 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.

- Another limitation of the current approach arises from the fact that our detection approach 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, from an academic stand point of view it might be interesting to devise methods that can detect, but not identify robots. The general problem with such a setting lies in our factorial representation, which cannot model statements such as "either robot A or robot B is straight in front of me." To model such situations, one would have to compute distributions over the joint space of all robots' coordinates, which would make it impossible that each robot carries its own, local position estimate. In addition, the complexity of the estimation routine would now depend super-linearly on the number of robots (as pointed out above, in the worst case it would scale *exponentially* instead of *linearly*). In fact, the latter observation is the key reason as to why factorial representations are chosen here.

- The collaboration described here is purely passive, in that robots combine information collected locally, but they do not change their course of action so as to aid localization. In [10, 24], we proposed an algorithm based on information-theoretic principles, for *active localization*, 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, information-theoretic principle, to coordinated multi-robot localization.

- Finally, the robots update their 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. While we were forced to carry out this research on two platforms only, we conjecture that the benefits of collaborative multi-robot localization increase with the number of available robots.

Acknowledgement

The authors like to thank Frank Dellaert and the members of CMU's Robot Learning Laboratory for many inspiring discussions.

References

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

[2] T.R. Balch. The impact of diversity on performance in multi-robot foraging. In *Agents '99*, Seattle, WA, 1999.

[3] 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.

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

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

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

[7] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc.of the Fifteenth National Conference on Artificial Intelligence*, 1998.

[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 *Pro. 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 Thirteenth National Conference on Artificial Intelligence*, 1996.

[10] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proc. of the Fifteenth International Joint Conference on Artificial Intelligence*, 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 *Proceeding of Seventh National Conference on Artificial Intelligence AAAI-92*, pages 49–54, Menlo Park, CA, 1988. AAAI, AAAI Press/The MIT Press.

[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 and Automation*, 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] H. Endres, W. Feiten, and G. Lawitzky. Field test of a navigation system: Autonomous cleaning in supermarkets. In *Proc. of the 1998 IEEE International Conference on Robotics & Automation (ICRA 98)*, 1998.

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

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

[22] D. Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Naviagation*. PhD thesis, Department of Computer Science, University of Bonn, Germany, December 1998.

[23] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization—efficient position estimation for mobile robots. In *Proc.of the Sixteenth National Conference on Artificial Intelligence*, 1999. To appear.

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

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

[26] 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. Int'l Conference on Industrial Electronics Control and Instrumentation (IECON'93)*, volume 1, pages 1466 – 1471, 1993.

[27] 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.

[28] R. Greiner and R. Isukapalli. Learning to select useful landmarks. In *Proceedings of 1994 AAAI Conference*, pages 1251–1256, Menlo Park, CA, 1994. AAAI Press / The MIT Press.

[29] J.-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch. The CS Freiburg Robotic Soccer Team: Reliable self-localization, multirobot sensor integration, and basic soccer skills. In M. Asada, editor, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag. To appear.

[30] J.-S. Gutmann and B. Nebel. Navigation mobiler Roboter mit Laserscans. In *Autonome Mobile Systeme*. Springer Verlag, Berlin, 1997. In German.

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

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

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

[34] M. Hunke and A. Waibel. Face locating and tracking for human-computer interaction. In *Proceedings of the Twenty-Eight Asilomar Conference on Signals, Systems and Computers*, Monterey, California, November 1994.

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

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

[37] 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*, 1996.

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

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

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

[41] S. Koenig and R. Simmons. Passive distance learning for robot navigation. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996.

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

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

[44] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 979–984, Menlo Park, July 1994. AAAI, AAAI Press/MIT Press.

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

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

[47] 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.

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

[49] 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.

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

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

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

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

[54] S. M. Omohundro. Efficient algorithms with neural network behavior. *Journal of Complex Systems*, 1(2):273–347, 1987.

[55] 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.

[56] S. Oore, G.E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9:683–699, 1997.

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

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

[59] W.D. Rencken. Concurrent localisation and map building for mobile robots using ultrasonic sensors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2129–2197, Yokohama, Japan, July 1993.

[60] 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.

[61] B. Schiele and J.L. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1628–1634, 1994.

[62] H. Shatkay and L.P. Kaelbling. Learning hidden markov models with geometric information. Technical Report CS-97-04, Computer Science Department, Brown University, Providence, RI, April 1997.

[63] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan. A layered architecture for office delivery robots. In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, February 1997.

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

[65] 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.

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

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

[68] S. Thrun. Finding landmarks for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1998.

[69] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[70] 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 *Proceedings of the International Conference on Robotics and Automation (ICRA '99)*, 1999. To appear.

[71] 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.

[72] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proc.of the Fifteenth National Conference on Artificial Intelligence*, 1998.

[73] 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 *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 595–601, 1994.

[74] E. Wolfart, R.B. Fisher, and A. Walker. Position refinement for a navigating robot using motion information based on honey bee strategies. In *Proceesings of the International Symposium on Robotic Systems (SIR 95)*, Pisa, Italy, 1995. also appeared as DAI Research Paper No 751, University of Edinburgh, Department of Artificial Intelligence, U.K.

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