Markov Localization for Reliable Robot Navigation and People Detection

Dieter Fox[†‡], Wolfram Burgard[†], and Sebastian Thrun[‡]

[†]University of Bonn, Department of Computer Science III, Bonn, Germany
[‡]Carnegie Mellon University, School of Computer Science, Pittsburgh, PA

Abstract Localization is one of the fundamental problems in mobile robotics. Without knowledge about their position mobile robots cannot efficiently carry out their tasks. In this paper we present Markov localization as a technique for estimating the position of a mobile robot. The key idea of this technique is to maintain a probability density over the whole state space of the robot within its environment. This way our technique is able to globally localize the robot from scratch and even to recover from localization failures, a property which is essential for truly autonomous robots. The probabilistic framework makes this approach robust against approximate models of the environment as well as noisy sensors. Based on a fine-grained, metric discretization of the state space, Markov localization is able to incorporate raw sensor readings and does not require predefined landmarks. It also includes a filtering technique which allows to reliably estimate the position of a mobile robot even in densely populated environments. We furthermore describe, how the explicit representation of the density can be exploited in a reactive collision avoidance system to increase the robustness and reliability of the robot even in situations in which it is uncertain about its position. The method described here has been implemented and tested in several real-world applications of mobile robots including the deployments of two mobile robots as interactive museum tour-guides.

1 Introduction

The problem of estimating the position of a mobile robot within its environment belongs to the fundamental problems of mobile robotics [10,2]. The knowledge about its position enables a mobile robot to carry out its tasks efficiently and reliably. In general, the problem is to estimate the location of the robot, i.e. its current state in its three-dimensional (x, y, θ) configuration space within its environment given a map and incoming sensory information. Methods of this type are regarded as so-called map-matching techniques, since they match measurements of standard sensors with the given model of the environment. The advantage of map-matching techniques is that they do not require any modifications of the environment and expensive special purpose sensors. However, there are several problems, these methods have to deal with. First, they must be able to deal with uncertain information coming from the inherent noise of sensory data. Second, models of the environment are generally approximative especially if the environment is populated. Finally, the methods have to deal with ambiguous situations which frequently arise for example in office environments with long corridors.

The position estimation techniques developed so far can be distinguished according to the type of problem they attack. *Tracking* or *local* techniques aim at compensating odometric errors occurring during robot navigation. They require, however, that the initial location of the robot is known and occasionally fail if they lost track of the robot's position. On the opposite side are the so called *global* techniques which are designed to estimate the position of the robot even under global uncertainty. Techniques of this type solve the so-called wake-up resp. initialization and kidnapped robot problems. They can estimate the position of the robot without any prior knowledge about it, and they can recover from situations in which the robot is exposed serious positioning errors coming for example from bumping into an object.

In this paper we present Markov localization as a technique for globally estimating the position of the robot given a model of the environment. It uses a probabilistic framework and estimates a position probability density over the whole state space of the robot. This allows the technique to globally estimate the robot's position. In the beginning this technique starts with a uniform distribution over the whole three-dimensional state-space of the robot. By integrating sensory input this method keeps track of multiple hypotheses and incrementally refines the density until it ends up with a uni-modal distribution. It furthermore can re-localize the robot in the case of localization failures. Both properties are a basic precondition for truly autonomous robots which are designed to operate autonomously over longer periods of time. Furthermore, Markov localization can deal with uncertain information. This is important because sensors such as ultrasound sensors as well as models of the environment such as occupancy grid maps are generally imperfect. Our method uses a fine-grained and metric discretization of the state space. This approach has several advantages. First, it provides accurate position estimates which allow a mobile robot to efficiently perform tasks like office delivery. Second, the method can integrate raw sensory input such as a single beam of an ultrasound sensor. Most approaches for global position estimation, in contrast to that, rely on assumptions about the nature of the environment such as the orthogonality or the types of landmarks found in that environment. Therefore, such techniques are prone to fail if the environment does not align well with these assumptions. Typical examples of such environments can be found in the experimental results section of this paper. Furthermore, our Markov localization technique includes a method for filtering sensory input, which is designed to increase the robustness of the position estimation process especially in densely populated environments such as museums or exhibitions. This way, our technique can even be applied in dynamic environments in which most of the robot's sensor readings do not correspond to the expected measurements, since the senor beams are reflected by people surrounding the robot or by other un-modelled objects. The technique has been implemented and proven robust in several long-term and real-world applications of mobile robots in populated environments.

The paper is organized as follows. In the next section we will describe the mathematical framework of Markov localization. Section 3 introduces the grid-based representation of the position probability density. It furthermore presents techniques for efficiently updating these densities in real-time and also introduces two filtering schemes to deal with obstacles in the robot's environment that are not contained in the map. Section 4 presents successful applications of Markov localization in real-world deployments of mobile robots. Finally, we relate our approach to previous work in Section 5.

2 Markov Localization

The basic idea of Markov localization is to maintain a probability density over the whole state space of the robot within its environment [26,29,7,19]. Markov localization assigns to each possible pose in the (x, y, θ)-space of the robot in the given environment the probability that the robot is at that particular position and has the corresponding orientation. Let L_t denote the random variable representing the state in the (x, y, θ) space of the robot at time t. Thus, $P(L_t = l)$ denotes the robot's belief that it was at location l at time t. The state L_t is updated whenever new sensory input is received or the robot moves. Without loss of generality we assume that at every discrete point t in time first a measurement s_t is perceived and then a movement action a_t is performed. Then, given a sensory input s_t Markov localization updates the belief for each location l in the following way:

$$P(L_t = l|s_t) \leftarrow \alpha_t \cdot P(s_t \mid l) \cdot P(L_t = l) \tag{1}$$

In this equation the term $P(s_t \mid l)$ is denoted as the *perception model* since it describes the probability of measuring s_t at location l. The constant α_t simply is a normalizer ensuring that the left-hand side sums up to one over all l. Upon executing action a_t Markov localization applies the following formula coming from the domain of Markov chains to update the belief:

$$P(L_{t+1} = l) \leftarrow \sum_{l'} P(L_{t+1} = l \mid L_t = l', a_t) \cdot P(L_t = l' \mid s_t) \tag{2}$$

The term $P(L_{t+1} = l \mid L_t = l', a_t)$ is also called *action model* since it describes the probability that the robot is at location l upon executing action a_t at a position l'.

The belief $P(L_0)$ at time $t = 0$ reflects the knowledge about the starting position of the robot. If the position of the robot relative to its map is entirely unknown, $P(L_0)$ corresponds to a uniform distribution. If the initial position of the robot is known with absolute certainty, then $P(L_0)$ is a Dirac distribution centered at this position.

3 Grid-based Markov Localization

Grid-based Markov localization — in contrast to other variants of Markov localization which use a topological discretization — uses a fine-grained geometric discretization to represent the position of the robot (see [7]). More specifically, L is represented by a three-dimensional fine-grained, regularly spaced grid, where the spatial resolution is usually between 10 and 40 cm and the angular resolution is usually 2 or 5 degrees. This approach has several desirable advantages. First, it provides very precise estimates for the position of a mobile robot. Second, the high resolution grids allow the integration of raw (proximity) sensor readings. Thus, the grid-based technique is able to exploit arbitrary geometric features of the environment such as the size of objects or the width and length of corridors resp. rooms. It furthermore does not require the definition of abstract features such as openings, doorways or other types of landmarks as it has to be done for the techniques based on topological discretizations.

A disadvantage of the fine-grained discretization, however, lies in the huge state space which has to be maintained. For a mid-size environment of size $30 \times 30\text{m}^2$, an angular grid resolution of $2°$, and a cell size of $15 \times 15\text{cm}^2$ the state space consists of $7,200,000$ states. The basic Markov localization algorithm updates each of these states for each sensory input and each movement operation of the robot. To efficiently update such large state spaces our system includes two techniques which are described in the remainder of this section. The first optimization is the selective update strategy which focuses the computation on the relevant part of the state space. The second method is our sensor model which has been designed especially for proximity sensors and allows the computation of the likelihood $P(s \mid l)$ by two look-up operations. Based on these two techniques, grid-based Markov localization can be applied in real-time to estimate the position of a mobile robot during its operation.

Global localization techniques, which are based on matching sensor readings with a fixed model of the environment assume that the map reflects the true state of the world and therefore are prone to fail in highly dynamic environments in which crowds of people cover the robots sensors for longer periods of time. To deal with such situations, our system also includes a filtering technique that analyzes sensor readings according to whether or not they come from dynamic and un-modelled aspects of the environment. Based on this technique the robot can robustly operate even in populated environments such as a museum.

3.1 Selective Update

An obvious disadvantage of the grid-based discretization comes from the size of the state space. To integrate a single measurement s_t into the belief

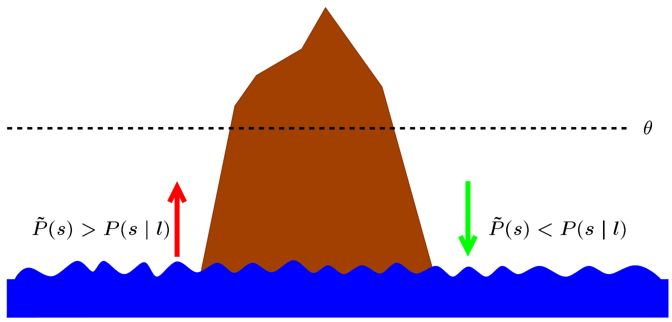


Figure1. Basic idea of the selective update technique

state all cells have to be updated. In this section we therefore describe a technique which allows a *selective* update of the belief state. The key idea of this approach is to exclude unlikely positions from being updated. For this purpose, we introduce a threshold θ and approximate $P(s_t \mid l)$ for cells with $P(L_{t-1} = l) \leq \theta$ by the probability $\tilde{P}(s_t)$. The quantity $\tilde{P}(s_t)$ is given by the average probability of measuring the feature s_t given a uniform distribution over all possible locations. This leads us to the following update rule for a sensor measurement s_t:

$$P(L_t = l | s_t) \longleftarrow \begin{cases} \alpha_t \cdot P(s_t \mid l) \cdot P(L_t = l) & \text{if } P(L_t = l) > \theta \\ \alpha_t \cdot \tilde{P}(s_t) \cdot P(L_t = l) & \text{otherwise} \end{cases} \tag{3}$$

Since α_t is a normalizing constant ensuring that $P(L_t = l \mid s_t)$ sums up to one over all l, this is equivalent to

$$P(L_t = l \mid s_t) \longleftarrow \begin{cases} \tilde{\alpha}_t \cdot \frac{P(s_t|l)}{\tilde{P}(s_t)} \cdot P(L_t = l) & \text{if } P(L_t = l) > \theta \\ \tilde{\alpha}_t \cdot P(L_t = l) & \text{otherwise} \end{cases} \tag{4}$$

Thus, all positions with a probability less or equal θ only have to be updated with the normalizing constant $\tilde{\alpha}_t$ which can simply be computed by summing up over all likely cells and adding to that $n \cdot \tilde{P}(s_t) \cdot \max\{P(L_t) | P(L_t) \leq \theta\}$ which is an approximation of the probability mass contained in the unlikely cells. In this term n is the number of states with $P(L_t) \leq \theta$. Since all these states are multiplied with the same value, it suffices to update only one variable instead of all of unlikely cells (see [5]).

At this point it should be noted that the approximation of $P(s \mid l)$ by $\tilde{P}(s)$ for a measurement s is a conservative approximation, since $P(s \mid l)$ is usually below the average $\tilde{P}(s)$ at unlikely positions l.

Figure 1 illustrates the key idea of the selective update scheme. In this example the belief is concentrated on a single peak which sticks out of the sea. The water level represents the maximum probability of all unlikely states.

Whenever the estimated position corresponds to the true location of the robot, the probability $P(s \mid l)$ exceeds $\tilde{P}(s)$ and the belief is confirmed so that the level of the sea goes down. However, if the system loses track of the robot's position, then the obtained measurements no longer match the expected measurements and $P(s \mid l) < \tilde{P}(s)$. Thus the robot's certainty decreases and the level of the sea increases. As soon as the sea level exceeds θ, the so far unlikely states are updated again.

In extensive experimental tests we did not observe evidence that the selective update scheme impacts the robot's behavior in any noticeable way. In general, the probabilities of the active locations sum up to at least 0.99. During global localization, the certainty of the robot permanently increases so that the density quickly concentrates on the true position of the robot. As soon as the position of the robot has been determined with high certainty, a scan comprising 180 laser measurements is typically processed in less than 0.1 seconds using a 200MHz Intel Pentium Pro.

The advantage of the selective update scheme is that the computation time required to update the belief state adapts automatically to the certainty of the robot. This way, our system is able to efficiently track the position of a robot once its position has been determined. Thereby, Markov localization keeps the ability to detect localization failures and to re-localize the robot. The only disadvantage lies in the fixed representation of the grid which has the undesirable effect that the space requirement in our current implementation stays constant even if only a minor part of the state space is updated. In this context we would like to mention that recently promising techniques have been presented to overcome this disadvantage by applying alternative and dynamic representations of the state space [6,5,12].

3.2 The Model of Proximity Sensors

As mentioned above, the likelihood $P(s \mid l)$ that a sensor reading s is measured at position l has to be computed for all positions l in each update cycle. Therefore, it is crucial for on-line position estimation that this quantity can be computed very efficiently. In [25] Moravec proposed a method to compute a generally non-Gaussian probability density function $P(s \mid l)$ over a discrete set of possible distances measured by the sensor for each location l. In a first implementation of our approach [7] we used a similar method, which unfortunately turned out to be computationally too expensive for on-line position estimation. Furthermore, the pre-computation of all these densities is not feasible since it requires an enormous amount of memory.

To overcome these disadvantages, we developed a perception model which allows us to compute $P(s \mid l)$ by two look-up operations. The key idea is to store for each location l in the (x, y, θ) space the distance o_l to the next obstacle in the map. This distance can be extracted by ray-tracing from

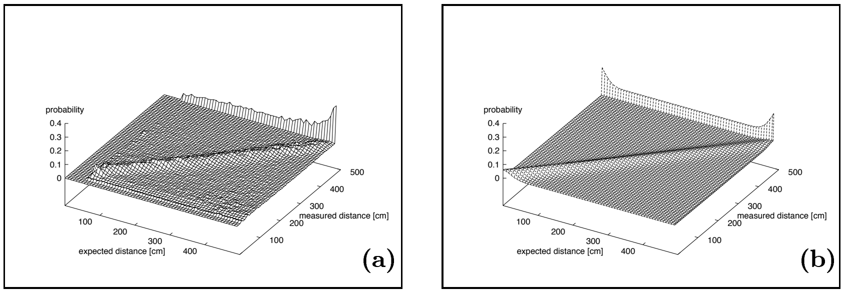


Figure 2. Measured (a) and approximated (b) densities for an ultrasound sensor.

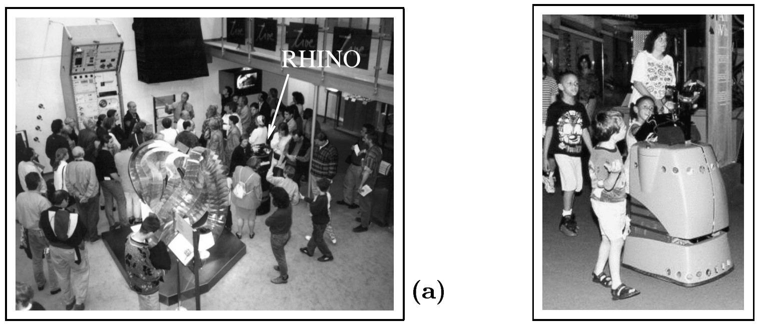
occupancy grid maps or CAD-models of the environment. In order to store these distances compactly, we use a discretization $d_0, \ldots, d_{n-1}$ of possible distances measured by a proximity sensor and store for each location only the index of the expected distance. Thus, to compute the probability $P(s_t \mid l)$ it remains to determine the probability $P(s_t \mid o_l)$ of measuring the value s_t given the expected distance o_l.

In our approach, the density $P(s \mid o_l)$ is defined as a mixture of a Gaussian density centered around the expected distance o_l and a geometric distribution [17,11]. The geometric distribution is designed to allow the system to deal with a certain amount of un-modelled objects such as people walking by. In order to determine the parameters of this model we collected several million data pairs consisting of the expected distance o_l and the measured distance d_i during the typical operation of the robot. Based on these data we adopted the parameters of the sensor model so as to best fit the measured data. The measured data and resulting densities for ultrasound sensors are depicted in Figure 2. The similarity between the measured and the approximated distributions shows that our sensor model yields a good approximation of the data.

Based on these densities we now can compute $P(s \mid l)$ simply by two nested look-up operations. After retrieving the expected distance o_l for the currently considered location l, we compute $P(s \mid o_l)$ by a further look-up operation in the corresponding table.

3.3 Filtering Techniques for Dynamic Environments

The perception model described in the previous section assigns a fixed probability to every pair of measured and expected distances. Although it also incorporates a certain amount of un-modelled objects, it is only capable to model such noise *on average*. While this approach showed to reliably deal with occasional sensor blockage, it is not sufficient in situations where a large amount of all sensor readings are corrupted. In general, localization tech-



(a) (b)

Figure 3. The mobile robots Rhino (a) and Minerva (b) acting as interactive museum tour-guides.

niques based on map-matching are prone to fail if the assumption that the map represents the state of the world is severely violated [14]. Consider, for example, a mobile robot which operates in an environment in which large groups of people permanently cover the robots sensors and thus lead to unexpected measurements. The mobile robots Rhino and Minerva which were deployed as interactive museum tour-guides in the *Deutsches Museum Bonn*, Germany and in the *National Museum of American History, Washington DC* respectively [4,31] were permanently faced with such a situation. Figure 3 shows cases in which the robots were surrounded by many visitors while giving a tour through the museum. To deal with such situations, we developed two filtering techniques which are designed to detect whether a certain sensor reading is corrupted or not. The basic idea of these filters is to sort all measurements into two buckets, one that contains all readings believed to reflect the true state of the world and another one containing all readings believed to be caused by un-modelled obstacles surrounding the robot. Compared to a fixed integration in the perception model, these filters have the advantage that they select the maximum amount of information from the obtained readings and automatically adopt their behaviour according to whether the environment is empty or highly populated.

Figure 4 contains two typical laser scans obtained in such situations. Obviously, the readings are to a large extend corrupted by the people in the museum which are not contained in the static world model. The different shading of the beams indicates the two classes they belong to: the black lines correspond to static obstacles that are part of the map, whereas the grey-shaded lines are those beams reflected by visitors in the Museum (readings above 5m are excluded for the sake of clarity). Since people standing close usually increase the robot's belief of being close to modelled obstacles, the robot quickly loses track of its position when taking all incoming sensor data seriously.

In the remainder of this section we introduce two different kinds of filters. The first one is called *entropy filter* which can be applied to arbitrary sensors,

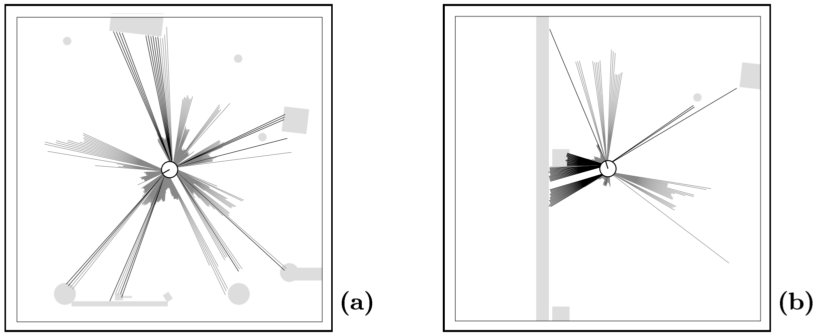


Figure 4. Typical laser scans obtained when Rhino is surrounded by visitors.

since it filters a reading according to its effect on the belief $P(L)$. The second filter is the *distance filter* which selects the readings according to how much shorter they are than the expected value. It therefore is especially designed for proximity sensors.

The Entropy Filter The entropy $H(L)$ of the belief over L is defined as

$$H(L) = -\sum_{l} P(L = l)\ \log P(L = l) \tag{5}$$

and is a measure of uncertainty about the outcome of the random variable L [8]: The higher the entropy, the higher the robot's uncertainty as to where it is. The *entropy filter* measures the relative change of entropy upon incorporating a sensor reading into the belief $P(L)$. More specifically, let s denote the measurement of a sensor (in our case a single range measurement). The change of the entropy of $P(L)$ given s is defined as:

$$\Delta H(L \mid s) := H(L) - H(L \mid s) \tag{6}$$

Here, $H(L \mid s)$ is the entropy of the belief $P(L \mid s)$. While a negative change of entropy indicates that after incorporating s, the robot is less certain about its position, a positive change indicates an increase in certainty. The selection scheme of the entropy filter is:

Exclude all sensor measurements s with $\Delta H(L \mid s) < 0$.

Thus, the entropy filter makes robot perception highly selective, in that it considers only sensor readings confirming the robot's current belief.

The Distance Filter The advantage of the entropy filter is that it makes no assumptions about the nature of the sensor data and the kind of disturbances occurring in dynamic environments. In the case of proximity sensors, however, un-modelled obstacles produce readings that are shorter than the distance that can be expected given the map. The *distance filter* which selects sensor readings based on their distance relative to the distance to the closest obstacle in the map removes those sensor measurements s which with probability higher than θ (which we generally set to 0.99) are shorter than expected, i.e. caused by an un-modelled object.

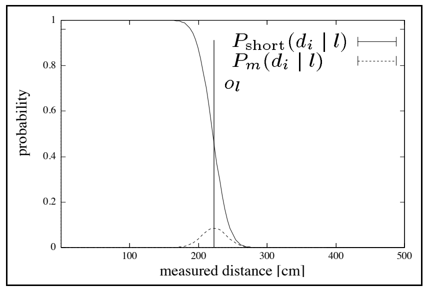


Figure 5. Probability $P_m(d_i \mid l)$ of expected measurement and probability $P_{\text{short}}(d_i \mid l)$ that a distance d_i is shorter than the expected measurement.

Consider a discrete set of possible distances $d_1, \ldots, d_n$ measured by a proximity sensor. Let $P_m(d_i \mid l) = P(d_i \mid o_l)$ denote the probability of measuring distance d_i if the robot is at position l and the sensor detects the next obstacle in the map. The distribution P_m describes the sensor measurement *expected* from the map. This distribution is assumed to be Gaussian with mean at the distance o_l to the next obstacle. The dashed line in Figure 5 represents P_m for a laser-range finder and a distance o_l of 230cm. Given P_m we now can define the probability $P_{\text{short}}(d_i \mid l)$ that a measured distance d_i is *shorter* than the expected one given the robot is at position l. This probability is obviously equivalent to the probability that the expected measurement is longer than the o_l and can be computed as:

$$P_{\text{short}}(d_i \mid l) = \sum_{j>i} P_m(d_j \mid l). \tag{7}$$

$P_{\text{short}}(d_i \mid l)$ is the probability that a measurement d_i is shorter than expected given the robot is at location l. In practice, however, we are interested in the probability $P_{\text{short}}(d_i)$ that d_i is shorter than expected given the current belief of the robot. Thus, we have to average over all possible positions of the robot:

$$P_{\text{short}}(d_i) = \sum_{l} P_{\text{short}}(d_i \mid l) P(L = l) \tag{8}$$

Based on $P_{\text{short}}(d_i)$ we now can define the distance filter as:

Exclude all sensor measurements d_i with $P_{short}(d_i) > \theta$.

4 Applications of Markov Localization in Real-world Environments

The grid-based Markov localization technique including the filter extension has been implemented and proven robust in various environments. In this section we illustrate the application of Markov localization in the context of the deployments of the mobile robots Rhino [4] and Minerva [31] as interactive museum tour-guide robots (see Figure 3). Rhino [3,32] has a ring of 24 ultrasound sensors each with an opening angle of 15 degrees. Both, Rhino and Minerva are equipped with two laser-range finders covering 360 degrees of their surrounding.

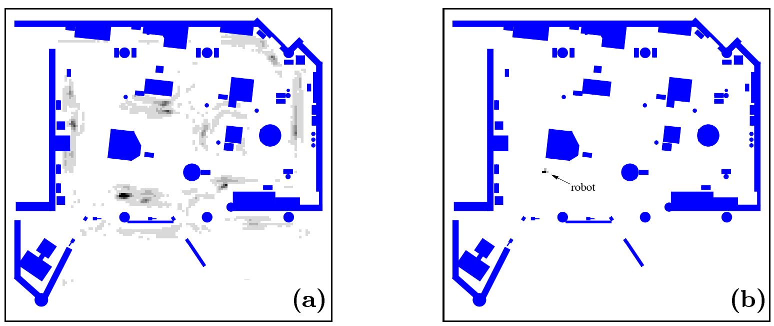


Figure 6. Global localization in the Deutsches Museum Bonn. The left image (a) shows the belief state after incorporating one laser scan. After incorporating two more scans, the robot uniquely determined its position (b).

4.1 Reliable Position Estimation in Populated Environments

During the deployment of the mobile robots Rhino and Minerva in the Deutsches Museum Bonn resp. the National Museum of American History (NMAH) in Washington DC the reliability of the localization system was a precondition for the success of the entire mission. Although both robots used the laser-range finder sensors for (global) localization, they had to deal with situations in which more than 50% of all sensor readings were corrupted because large crowds of people surrounded the robot for longer periods of time [14].

Rhino used the entropy filter to identify sensor readings that were corrupted by the presence of people. Since this filter integrates only readings which do not increase the uncertainty, it is restricted to situations in which the robot knows its approximate location. Unfortunately, when using the entropy filter, the robot is not able to recover from situations in which the robot loses its position entirely. To prevent this problem, Rhino additionally

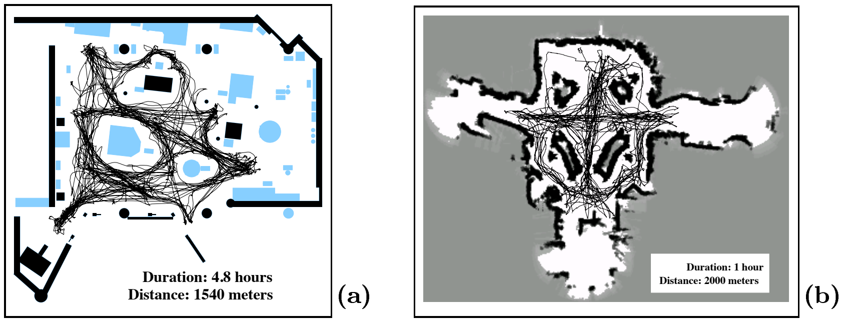


Figure 7. Typical trajectories of the robot Rhino in the Deutsches Museum Bonn (and) Minerva in the National Museum of American History (b).

incorporated a small number of randomly chosen sensor readings into the belief state. Based on this technique, Rhino's localization module was able to (1) globally localize the robot in the morning when the robot was switched on and (2) to reliably and accurately keep track of the robot's position. Figure 6 shows the process of global localization in the Deutsches Museum Bonn. RHINO is started with a uniform distribution over its belief state. The probability distribution given after integrating the first sensor scan is shown in Figure 6(a). After incorporating two more sensor scans, the robot knows its position with high certainty (see Figure 6(b)). Based on the selective update mechanism, the robot could efficiently keep track of the robot's position after determining it uniquely. Figure 7(a) shows a typical trajectory of the robot Rhino in the museum in Bonn. In the entire six-day deployment period Rhino travelled over 18km. Its maximum speed was over 80cm/sec and the average speed was 37cm/sec. Rhino completed 2394 out of 2400 requests which corresponds to a success rate of 99.75%.

Figure 7(b) shows a trajectory of 2 km of the robot Minerva in the National Museum of American History. Minerva used the distance filter to identify readings reflected by un-modelled objects. Based on this technique, Minerva was able to operate reliably over a period of 13 days. During that time over 50.000 people were in the museum and watched or interacted with the robot. At all Minerva travelled 44km with a maximum speed of 1.63m/sec.

In an extensive experimental comparison [14,11] it has been demonstrated that both filters significantly improve the robustness of the localization process especially in the presence of large amounts of sensor noise.

4.2 Probabilistic Integration of Map Information into a Reactive Collision Avoidance System

During the deployments of Rhino and Minerva as interactive museum tour-guides, safe and collision-free navigation was of uttermost importance. In the Deutsches Museum Bonn, the collision avoidance was complicated by the fact

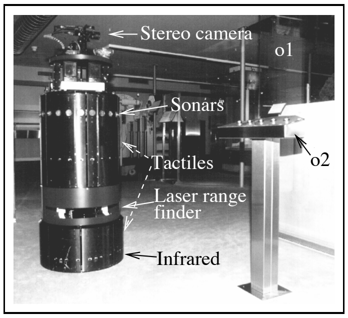


Figure 8. The robot Rhino and its sensors.

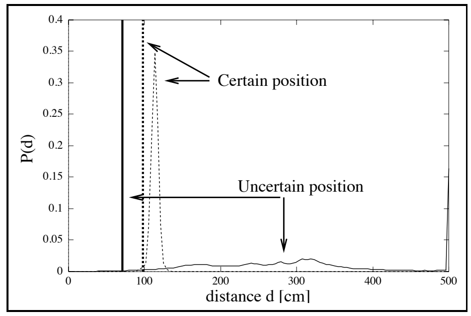


Figure 9. Probability densities $P(S_m = s)$ and selected measurement s^*.

that many of the obstacles and exhibits could not be sensed by the robot's sensors although Rhino possessed five state-of-the-art sensor systems (vision, laser, sonar, infrared, and tactile). Invisible obstacles included glass cages put up to protect exhibits, metal bars at various heights, and small podiums or metal plates on which exhibits were placed (c.f., the glass cage labelled "o1" and the control panel label "o2" in Figure 8). Furthermore, the area in which the robots were intended to operate was limited due to un-modelled regions or dangerous areas such as staircases. Therefore, both robots needed a technique which forces the robots not to leave their area of operation.

The collision avoidance system of the robots is based on the dynamic window algorithm (DWA) [13], a purely sensor-based approach designed to quickly react to obstacles blocking the robot's path. The basic DWA, just like any other sensor-based collision avoidance approach, does not allow for preventing collisions with obstacles which are "invisible" to the robot's sensors. To avoid collisions with those, the robot had to consult its map. Of course, obstacles in the map are specified in world coordinates, whereas reactive collision avoidance techniques such as DWA require the location of those obstacles in robo-centric coordinates.

At first glance, one might be inclined to use the maximum likelihood position estimate produced by Rhino's localization module, to convert world coordinates to robo-centric coordinates, thereby determining the location of "invisible" obstacles relative to the robot. However, such a methodology would be too brittle in situations where the localization module assigns high probability to multiple poses, which we observed quite frequently especially when the robot moves at high speeds. Our approach is more conservative. Based on the map, it generates "virtual" sensor readings that underestimate the true distance to the next obstacle in the map with probability 0.99. More specifically, let S_m be the proximity measurement that one would expect if all invisible obstacles were actually detectable. Then

$$P(S_m = s) = \sum_l P(S_m = s \mid L_t = l)\ P(L_t = l) \tag{9}$$

is the probability distribution that s is the distance to the next obstacle in the map given the current belief state of the robot. Now the measurement s^* which underestimates the true distance to the next obstacle in the map with probability θ, is obtained as

$$s^* := \max\{s \mid P(S_m > s) \geq \theta\} \tag{10}$$

where

$$P(S_m > s) := \sum_{s' > s} P(S_m = s'). \tag{11}$$

Figure 9 depicts two different densities $P(S_m = s)$ based on two different belief states. Whereas the solid line corresponds to a situation, in which the robot is highly uncertain about its position, the dashed line comes from a typical belief state representing high certainty. This figure illustrates, that the robot conservatively chooses a very small distance s^* if it is uncertain about its position and a distance close to the true distance in the case of high certainty.

Our extension of the DWA worked well in the museum. Figure 7 shows a 1.6 km-long trajectory of the robot Rhino in the Deutsches Museum Bonn. The location of the "invisible" obstacles are indicated by the gray-shaded areas, which the robot safely avoided. By adding appropriate obstacles to the map, the same mechanism was used to limit the operational range of the robots. With it, Rhino and Minerva successfully avoided to enter any terrain that was out of their intended operational range.

4.3 Reacting to People Blocking the Path of the Robot

In the museum, people were not always cooperative. A typical behaviour of visitors was to intentionally block the robot's path for longer periods of time. One example of such a situation with Rhino in the Deutsches Museum Bonn is shown in Figure 10. Here a group of visitors tries to challenge the robot by forming a u-shaped obstacle. To deal with such situations, the robot needed the capability to detect that un-modelled obstacles in the environment are blocking its path. Rhino and Minerva applied the distance filter to detect such situations, since people standing close to the robot lead to readings that are shorter than expected. If the path of the robot was blocked for a certain amount of time, then Rhino blew its horn to ask for clearance [4]. Minerva used a more sophisticated way of interaction [31]. It possessed a face (see

Figure 11) allowing it to express different moods according to the duration of the blockage. This face was mounted on a pan-tilt head and pointed to the person standing in the direction the robot was supposed to move. Minerva changed the mimic of the face according to the duration of the blocking. The moods ranged from "happy" (Figure 11(a)) to "angry" (Figure 11(b)). Minerva additionally used different phrases of pre-recorded texts to express the desire for free space.

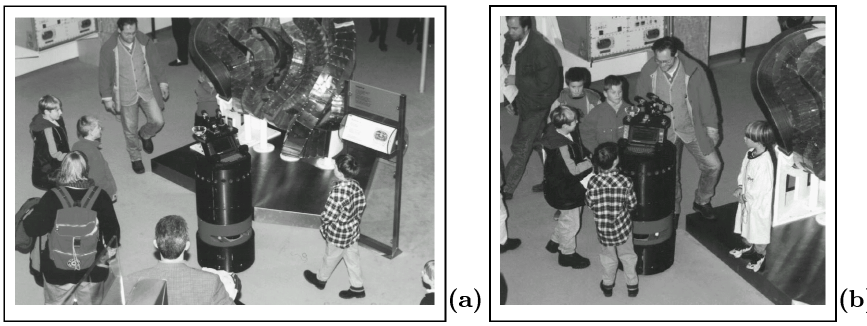


Figure 10. Typical situation in which visitors try to challenge the robot by intentionally blocking its path.

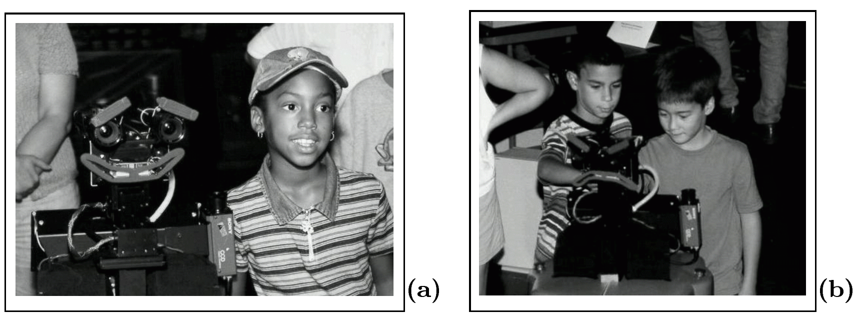


Figure 11. The different moods of the robot Minerva according to its progress ranged from happy (a) to angry (b).

Rhino's and Minerva's ability to react directly to people was among the most entertaining aspects, which contributed enormously to its popularity and success. Many visitors were amazed by the fact that the robots acknowledged their presence by blowing its horn, changing the mimic or explicitely asking to stay behind the robot. One effect was that they repeatedly stepped in its way to get the acoustic "reward." Nevertheless, the ability to detect such situations and to interactively express their intentions allowed the robots to reliably reach their goals.

5 Related Work

Most of the techniques for position estimation of mobile robots developed so far belong to the class of local approaches resp. tracking techniques which are designed to compensate odometric error occurring during navigation given that the initial position of the robot is known (see [2] for a comprehensive overview). Weiß et. al. [33] store angle histograms derived from laser-range finder scans taken at different locations in the environment. The position and orientation of the robot is calculated by maximizing the correlation between the stored histograms and laser-range scans obtained while the robot moves through the environment. The estimated position together with the odometry information is then used to predict the position of the robot and to select the histogram used for the next match. Yamauchi [34] applies a similar technique, but uses hill-climbing to match local maps built from ultrasonic sensors against a given occupancy grid map. As in [33], the location of the robot is represented by the position yielding the best match.

A very popular mathematical framework for position tracking are *Kalman filters* [24], a technique that was introduced by Kalman in 1960 [20]. Kalman filter based methods represent their belief about the position of the robot by a unimodal Gaussian distribution over the three-dimensional state-space of the robot. The existing applications of Kalman filtering to position estimation for mobile robots are similar in how they model the motion of the robot. They differ mostly in how they update the Gaussian according to new sensory input. Leonard and Whyte [22] match beacons extracted from sonar scans with beacons predicted from a geometric map of the environment. These beacons consist of planes, cylinders, and corners. To update the current estimate of the robot's position, Cox [9] matches distances measured by infrared range finders against a line segment description of the environment. Schiele and Crowley [27] compare different strategies to track the robots position based on occupancy grid maps and ultrasonic sensors. They show that matching local occupancy grid maps against a global grid map results in a similar localization performance as if the matching is based on features that are extracted from both maps. Shaffer et. al. [28] compare the robustness of two different matching techniques against different sources of noise. They suggest a combination of map-matching and feature-based techniques in order to inherit the benefits of both. Gutmann, Lu, and Milios [16,23] use a scan-matching technique to precisely estimate the position of the robot based on laser-range finder scans and learned models of the environment. Arras and Vestli [1] use a similar technique to compute the position of the robot with a very high accuracy. All these variants, however, rest on the assumption that the position of the robot can be represented by a single Gaussian distribution. The advantage of this approach lies in the high accuracy that can be obtained. The assumption of a unimodal Gaussian distribution, however, is not justified if the position of a robot has to be estimated from scratch, i.e. without knowledge about the starting position of the robot.

To overcome these disadvantages, recently different variants of Markov localization have been developed and employed successfully [26,29,19,7,18,30]. The basic idea of Markov localization is to maintain an arbitrary and not necessarily Gaussian position probability density over the whole three-dimensional $\langle x, y, \theta \rangle$ state space of the robot in its environment. The different variants of this technique can be roughly distinguished by the type of discretization used for the representation of the state space. In [26,29,19,18,30] Markov localization is used for landmark-based corridor navigation and the state space is organized according to the topological structure of the environment. Based on an orthogonality assumption [26,29,19] consider only four possible headings of the robot. Our fine-grained and grid-based approach of Markov localization has the advantage that it provides accurate position estimates and that it can be applied in arbitrary unstructured environments. The disadvantage of this technique, however, lies in its computational complexity and space requirements. In order to represent the whole state space of the robot within its environment, usually several million states have to be represented and updated. In this paper we therefore present different techniques to overcome these disadvantages. The result is an efficient and accurate position estimation technique for mobile robots. Our implementation of Markov localization is able to globally estimate the position of the robot from scratch and to efficiently keep track of the robots position once it has been determined. Simultaneously, it allows the robot to recover from localization failures. In a recent experimental comparison it has been demonstrated that Kalman filter based tracking techniques provide highly accurate position estimates but are less robust than Markov localization since they lack the ability to globally localize the robot and to recover from localization errors [15]. Our technique furthermore includes an approach to filter out readings coming from objects that are not contained in the map. This way our approach, in contrast to other Markov localization techniques, is able to reliably estimate the position of the robot even in densely populated and highly dynamic environments [4,11,14,31].

6 Discussion

In this paper we presented Markov localization as a robust technique for estimating the position of a mobile robot. The key idea of Markov localization is to maintain a position probability density over a fine-grained discretization of the robot's state space within the environment. This density is updated whenever new sensory input is received and whenever the robot moves.

We introduced two techniques for efficiently updating the density. First, we use a selective update scheme which focuses the computation on the relevant parts of the state space. Second, we apply a sensor model that allows to efficiently compute the necessary quantities by two look-up operations. Based on these two approaches, the belief state can be updated in real-time. Our approach to Markov localization furthermore includes filtering techniques significantly increasing the robustness of the position estimation process even in densely populated or highly dynamic environments. We presented two kinds of filters. The entropy filter is a general filter, that can be applied to arbitrary sensors including cameras. The distance filter has been developed especially for proximity sensors and therefore provides better results with sensors like a laser-range finder.

Our technique has been implemented and evaluated in several experiments at various sites. During the deployments of the mobile robots Rhino in the Deutsches Museum Bonn, Germany, and Minerva in the National Museum of American History, Washington DC, our system was successfully applied. The accuracy obtained by the grid-based and fine-grained discretization of the state space turned out to be high enough to avoid even obstacles that could not be sensed by the robot's sensors. It furthermore provided the basis for detecting situations, in which people intentionally block the path of the robot.

There are several aspects which are objectives of future research. First, the current implementation of Markov localization uses a fixed discretization of the whole state space which is always kept in memory. To overcome this disadvantage, recently different alternative representations of the density have been developed and suggested [6,5,15]. Whereas [6] uses a local grid for efficient position tracking, we introduced an Octree-based representation of the robot's state space in [5] which allows to dynamically adopt the required memory as well as the resolution of the discretization. [15] suggest a combination of Markov localization with Kalman filtering which should lead to a system inheriting the advantages of both approaches. A further restriction of the current system is that the model of the environment is assumed to be static. In this context the problem to combine map building and localization on-line is an interesting topic for future research.

References

1. K.O. Arras and S.J. Vestli. Hybrid, high-precision localization for the mail distributing mobile robot system MOPS. In *Proc. of the IEEE International Conference on Robotics and Automation*, 1998.
2. J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.
3. J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot Rhino. *AI Magazine*, 16(2), 1995.
4. 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.

5. 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*, 1998.
6. W. Burgard, D. Fox, and D. Hennig. Fast grid-based position tracking for mobile robots. In *Proc. of the 21st German Conference on Artificial Intelligence, Germany*. Springer Verlag, 1997.
7. 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.
8. T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 1991.
9. 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, 4 1991.
10. I.J. Cox and G.T. Wilfong, editors. *Autonomous Robot Vehicles*. Springer Verlag, 1990.
11. D. Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation*. PhD thesis, Dept of Computer Science, Univ. of Bonn, Germany, 1998.
12. 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*, Orlando, Fl, 1999.
13. D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation*, 4(1), 1997.
14. 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.
15. J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.
16. 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.
17. D. Hennig. Globale und lokale Positionierung mobiler Roboter mittels Wahrscheinlichkeitsgittern. Master's thesis, Department of Computer Science, University of Bonn, Germany, 1997. In German.
18. J. Hertzberg and F. Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proc. of the First Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.
19. 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.
20. R.E. Kalman. A new approach to linear filtering and prediction problems. *Tansaction of the ASME – Journal of basic engineering*, pages 35–45, March 1960.
21. D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. MIT/AAAI Press, Cambridge, MA, 1998.

22. J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.
23. F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 1994.
24. P.S. Maybeck. Autonomous robot vehicles. In Cox and Wilfong [10].
25. H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, Summer 1988.
26. I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2), Summer 1995.
27. 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*, 1994.
28. G. Shaffer, J. Gonzalez, and A. Stentz. Comparison of two range-based estimators for a mobile robot. In *SPIE Conf. on Mobile Robots VII*, pages 661–667, 1992.
29. R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the International Joint Conference on Artificial Intelligence*, 1995.
30. S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1), 1998.
31. S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second-generation museum tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.
32. S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. In Kortenkamp et al. [21].
33. 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*, 1994.
34. B. Yamauchi. Mobile robot localization in dynamic environments using dead reckoning and evidence grids. In *Proc. of the 1996 IEEE International Conference on Robotics and Automation*, 1996.