# An Experimental Comparison of Localization Methods

Jens-Steffen Gutmann[1] Wolfram Burgard[2] Dieter Fox[2] Kurt Konolige[3]

[1]Institut für Informatik
Universität Freiburg
D-79110 Freiburg, Germany

[2]Institut für Informatik III
Universität Bonn
D-53117 Bonn, Germany

[3]SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

## Abstract

*Localization is the process of updating the pose of a robot in an environment, based on sensor readings. In this experimental study, we compare two recent methods for localization of indoor mobile robots: Markov localization, which uses a probability distribution across a grid of robot poses; and scan matching, which uses Kalman filtering techniques based on matching sensor scans. Both these techniques are* dense matching methods*, that is, they match dense sets of environment features to an a priori map. To arrive at results for a range of situations, we utilize several different types of environments, and add noise to both the dead-reckoning and the sensors. Analysis shows that, roughly, the scan-matching techniques are more efficient and accurate, but Markov localization is better able to cope with large amounts of noise. These results suggest hybrid methods that are efficient, accurate and robust to noise.*

## 1. Introduction

To carry out tasks, such as delivering objects, an indoor robot must be able to figure out where it is within its environment. A robot that goes through the wrong door or down the wrong corridor will be inefficient, even if it eventually recovers and determines its pose. And a robot that cannot position itself accurately is at risk from obstacles or dangerous areas that are in its map but which cannot be easily sensed. For these and other reasons, the problem of *localization* with respect to an internal map is an important one.

Generally speaking, localization techniques fall into three basic categories:

- Behavior-based approaches
- Landmarks
- Dense sensor matching

Behavioral approaches rely on the interaction of robot actions with the environment to navigate. For example, Connell's robot, Herbert, followed a right-hand rule to traverse an office environment, and found its way back by reversing the procedure [6]. More sophisticated systems learn internal structures that can be "played back" to redo or undo paths [1]. While behavioral approaches are useful for certain tasks, their ability to localize the robot geometrically is limited, because their navigation capability is implicit in their sensor/action history.

Landmark methods rely on the recognition of landmarks to keep the robot localized geometrically. Landmarks may be given *a priori* (for example, the satellites in GPS) or learned by the robot as it maps the environment (for example, sonar landmarks [11]). While landmark methods can achieve impressive geometric localization, they require either engineering the environment to provide a set of adequate landmarks, or efficient recognition of features to use as landmarks. In contrast, dense sensor methods [22, 12, 17, 9, 4] attempt to use whatever sensor information is available to update the robot's pose. They do this by matching dense sensor scans with a surface map of the environment, without extracting landmark features. Thus, dense sensor matching can take advantage of whatever surface features are present, without having to explicitly decide what constitutes a landmark.

In recent years several competing techniques for dense sensor matching have emerged. In this paper we undertake a systematic comparison of two such methods, concentrating on their ability to keep the robot localized in the presence of noise. We expect these results to be useful in determining the relative strengths and weaknesses of the methods, as well as giving guidelines for their application in typical environments. These experiments were motivated by the lack of experimental confirmation of the performance of localization methods, which makes it difficult to determine if a method that ran on one robot in one laboratory environment would be successful under any other conditions, with any other robot hardware.

In performing these experiments, we were interested in three questions:

1. Under what circumstances did the robots suffer catas-

trophic localization failure, that is, become completely lost?

2. How accurately did the techniques localize the robots under various conditions?

3. How well did the techniques tolerate ambiguity in the robot's pose, when there was insufficient information to accurately localize it?

## 2. Markov Localization and Scan-Matching Methods

In probabilistic terms, localization is the process of determining the likelihood of finding the robot at a pose $l$, given a history $s_1, \ldots s_n = S^n$ of sensor readings, and a history $a_1, \ldots a_n = A^n$ of position integration readings from the wheel encoders.[1] In practice, it is too difficult to determine the joint effect of all sensor and position integration readings; instead, a recursive approximation is assumed:

$$p(l \mid S^n, A^n) = \alpha \cdot \int p(l \mid s_n, a_n, l')p(l' \mid S^{n-1}, A^{n-1})\, dl'. \quad (1)$$

where $l'$ is the previous pose of the robot, and $\alpha$ is a normalizing factor ensuring that $p(l \mid S^n, A^n)$ sums up to one over all $l$. Thus, it is assumed that all information about the past history of the robot can be represented by the distribution $p(l' \mid S^{n-1}, A^{n-1})$.

Any localization method must decide the following questions:

1. How is the prior distribution to be represented?

2. How is the posterior distribution $p(l \mid s_n, a_n, l')$ to be calculated?

Markov localization makes the choice of an explicit, discreet representation for the prior probability, using a grid or topological graph to cover the space of robot poses, and keeping a probability for each element of this space. Scan matching, on the other hand, uses a simple Gaussian for the distribution. Given the divergence in representation, it is interesting that these methods both use the same general technique for calculating the posterior:

1. *Predict* the new robot pose $l$ and its associated uncertainty from the previous pose $l'$, given odometric information.

2. *Update* the robot pose (and uncertainty) $l$ using sensor information matched with the map.

[1] The map of the environment is a parameter to this process; we will describe its role in more detail in each of the methods.

The first step generally increases the uncertainty in the robot's pose, while the second generally reduces it. The prediction step is modeled by a conditional probability, denoted by $p(l \mid a_n, l')$ which denotes the probability that action $a_n$, when executed at $l'$, carries the robot to $l$. Upon robot motion, the pose is calculated as:

$$p(l) \quad \longleftarrow \quad \int p(l_u \mid a_n, l')p(l')\, dl' \;. \quad (2)$$

In the algorithms considered here, the possible errors of the odometry (i.e., $p(l \mid a_n, l')$) are modeled as normally distributed. Note that in the experiments, we deliberately use noise models that are *not* normal and violated this assumption, to test the robustness of the algorithms in a realistic environment.

In the update step, the new robot pose is calculated according to the Bayes formula:

$$p(l \mid s_n) = \alpha\, p(s_n \mid l)\, p(l) \;. \quad (3)$$

The sensor model $p(s_n \mid l)$ determines the likelihood of the sensor responding with $s_n$, given the robot at pose $l$.

### 2.1. Markov Localization

The key idea of Markov localization is to compute a discrete approximation of a probability distribution over all possible poses in the environment. This distribution evolves according to Equations 2 and 3. Different variants of Markov localization have been developed [15, 19, 10, 4, 20] and have been shown in experimental results to have several features:

- They are able to localize the robot when its initial pose is unknown. This property is essential for truly autonomous robots as it no longer requires that the initial pose of the robot is entered whenever it is switched on or gets lost.

- They are able to deal with noisy sensors such as ultrasonic sensors.

- They are able to represent ambiguities and can be extended to actively resolve ambiguities [5].

- Computationally, the technique is dominated by the dimensionality of the grid, and the size of its cells.

The existing methods can be distinguished according to the type of discretization they rely on. While [15, 19, 10, 20] use a topological discretization of the environment and detect landmarks to localize the robot, the system used in this paper computes a fine-grained grid-based approximation of the distribution [4]. To cope with the huge state space this technique includes several optimizations. In practice, usually only a small area around the robot is updated during localization.

Map information for Markov localization depends on the type of the state space discretization. The topological approaches [15, 19, 10, 20] use landmarks to detect locations. The fine-grained discretization applied in this paper in contrast uses metric maps of the environment. These can be hand-crafted CAD maps consisting of line segments representing vertical surfaces in the indoor environment, or learned occupancy grid maps [14]. In all approaches the map is used to compute what the sensor readings should be from a given cell in the state space. The closeness of the predicted readings to the actual ones give a measure of $p(s_n \mid l)$.

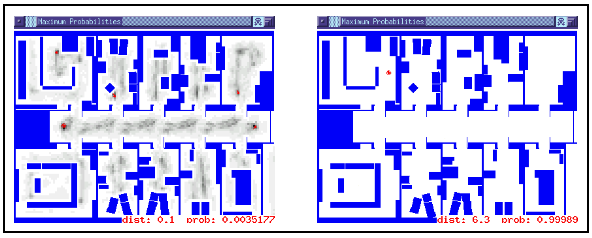

Fig. 1. Global position estimation using the grid-based Markov localization technique. Belief state after integrating the two sonar scans (left image). After 30 seconds and integrating 6 sonar scans the robot uniquely determined its position (right image).

As already mentioned, an important feature of Markov localization techniques is the ability to globally localize the robot within its environment. Figure 1 is a floor plan of a $27 \times 20\text{m}^2$ section of the Computer Science Department at the University of Bonn, in which some of the experiments reported here were carried out. It shows two density plots (dark positions are more likely) during global localization in the office environment. The robot started at position 1 and traveled to position 2 (refer to Figure 5). Initially the robot was completely uncertain about its position. The belief state after integrating two sonar scans of 24 ultrasound readings each is shown in the left image of Figure 1. After travelling 6.3 meter and incorporating 6 sonar scans the robot is absolutely certain about its position (see right image of Figure 1).

### 2.2. Scan Matching

Scan matching is the process of translating and rotating a range scan (obtained from a range device such as a laser range finder) in such a way that a maximum overlap between sensor readings and a priori map emerges. For matching a range scan with a map an initial estimate of the robot pose must be known and is usually derived from odometry information.

The robot pose and its update from scan matching are modeled as single Gaussian distributions. This has the advantage that robot poses can be calculated with high precision, and that an efficient method for computing the update step can be used, namely, Kalman filtering. Scan matching has the following properties:

- It can localize the robot precisely given good inputs, and in the linear case it is the optimal estimate of location.
- It cannot recover from catastrophic failures caused by bad matches or incorrect error models.
- Because its search is confined to small perturbations of the sensor scans, it is computationally efficient.

The extended Kalman filter method has the following form. For each time step $t$ the robot pose and error covariance are denoted by $l(t) = (x(t), y(t), \theta(t))^T$ and $\Sigma_l(t)$. On robot motion $a = (\delta, \alpha)^T$ the robot pose and covariance are updated according to:

$$\begin{aligned} l'(t+1) &= F(l(t), a) = \begin{pmatrix} x(t) + \delta \cos(\theta(t)) \\ y(t) + \delta \sin(\theta(t)) \\ \theta(t) + \alpha \end{pmatrix} \\ \Sigma'_l(t+1) &= \nabla F_l \Sigma_l(t) \nabla F_l^T + \nabla F_a \Sigma_a \nabla F_a^T \end{aligned}$$

From scan matching a pose update $l_s$ with an error covariance matrix $\Sigma_s$ is obtained and the robot pose and covariance are updated using the formulas:

$$\begin{aligned} l(t+1) &= (\Sigma_l'^{-1}(t+1) + \Sigma_s^{-1})^{-1} \cdot \\ &\qquad (\Sigma_l'^{-1}(t+1) l'(t+1) + \Sigma_s^{-1} l_s) \\ \Sigma_l(t+1) &= (\Sigma_l'^{-1}(t+1) + \Sigma_s^{-1})^{-1} \end{aligned}$$

These equations demonstrate that Kalman filter based self-localization can be implemented efficiently. As long as the error models are accurate, Kalman filtering will give a reasonable estimate of the robot pose (in the linear case, it will be an optimal estimate).

The success of the Kalman filter also depends heavily on the ability of scan matching to correct the robot pose. We use two matching methods, described in [9]. The first approach matches sensor readings with the line segments in a hand-crafted CAD map of the environment [7]. It assigns scan points to line segments based on closest neighborhood and then searches for a translation and rotation that minimizes the total squared distance between scan points and their target lines. For reasons of efficiency we modified the approach to extract the line segments from the CAD model that are visible from the current robot-position and discard the non-visible ones. This greatly reduces the number of line segments for the matching process and also avoids nonsensical assignments, e.g. assignments from a scan point to a line that corresponds to the backside of a wall.

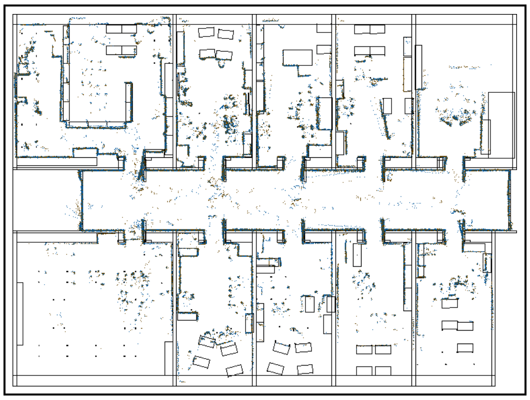

Fig. 2. Overlay of a hand-crafted CAD map of the office environment used for the experiments and the corresponding map learned by scan matching.

Scan matching can also be used with self-learned maps. We use a map of reference scans which have previously been obtained in an exploration run. As the scan positions in this run have been determined by dead-reckoning only and therefore contain errors, all positions have to be corrected first. This is done by the approach proposed in [13, 9] which computes a consistent map of the environment. Figure 2 shows an overlay of a learned and hand-crafted map of the Bonn Computer Science environment. This map was computed using two 180 degree laser-range finders. Obviously, scan matching produces extremely accurate maps.

For computing a position update, a range scan is matched with one of the reference scans, usually the one whose position is closest to the current robot position. For matching we use the approach proposed in [9] which is a combination of the line-segment matching method of the first approach, and a point-to-point match [12].

## 3. Localization Experiments

To compare both position estimation techniques we performed various experiments using the mobile robot RHINO [2, 21] (see Figure 3) in a typical structured office environment as well as in the rather unstructured environment of the "Deutsches Museum Bonn" during a six-days lasting deployment of the mobile robot RHINO as an interactive museum tour-guide [3] (see Figure 10). RHINO is an RWI B21 robot which is equipped with two laser-range finders covering 360° and a ring of 24 ultrasonic sensors each with an acceptance angle of 15 degrees.

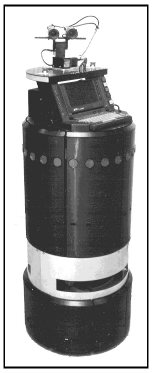

Fig. 3. Mobile robot RHINO used for the experiments.

### 3.1. Noise Models

There are several kinds of noise typically observed when robots operate in real-world environments. On one hand there is a typical Gaussian noise in the odometry and proximity sensors coming from the inherent inaccuracy of the sensors. On the other had there are non-Gaussian errors arising from robot colliding with obstacles, or from interference with the sensors.

In this paper, odometry errors coming from wheel-slippage, uneven floors, or different payloads are characterized according to the following three parameters (see left part of Figure 4).

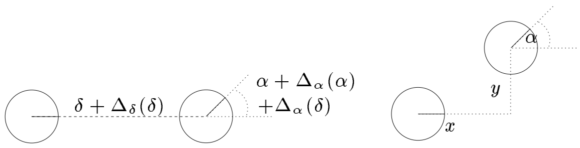


Fig. 4. Effect of adding noise $\langle \Delta_\delta(\delta), \Delta_\alpha(\alpha), \Delta_\alpha(\delta) \rangle$ (left) and bump noise $\langle x, y, \alpha \rangle$ (right) to the odometry.

**Range noise:** the error $\Delta_\delta(\delta)$ in range when the robot moves a certain distance $\delta$.

**Rotation noise:** the error $\Delta_\alpha(\alpha) + \Delta_\alpha(\delta)$ in rotation when the robot turns a certain angle $\alpha$ or moves a certain distance $\delta$.

There is another source of less frequent but much larger odometry errors coming from situations in which the robot bumps into obstacles. These abrupt errors can be characterized by the following parameters (see right part of Figure 4).

**Error of the odometry:** The error $x$, $y$, and $\alpha$ added to the odometry information.

**Frequency:** Probability that a bump occurs if the robot travels one meter. Throughout the experiments described below, this probability was set to $0.05$.

Finally, we will consider a type of noise is which the sensors do not return accurate information relative to the internal map (*map noise*). The source can be an inaccurate map, poorly-performing sensors, or the presence of dynamic objects such as people around the robot. This type of noise is very hard to characterize, and our experiments deal with a difficult case, sensor blocking by people.

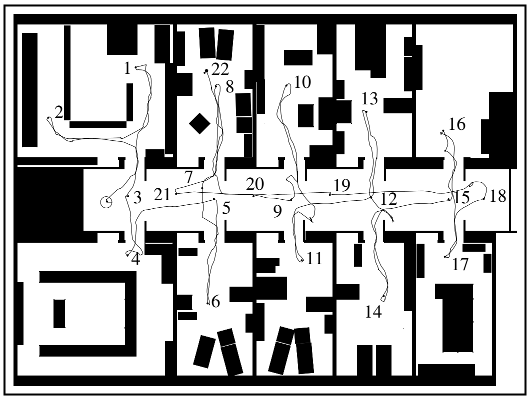


Fig. 5. Outline of the $27 \times 20m^2$ large office environment including the trajectory of the robot and the 22 reference positions.

### 3.2. Performance in the Office Environment

The office environment depicted in Figure 5 consists of a corridor and ten different offices. In this experiment we started the robot at the left side of the corridor, steered it through all offices, and measured the position of the robot at 22 different positions in the remainder denoted as reference positions.

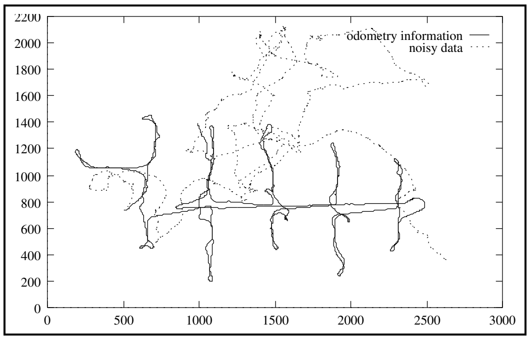


Fig. 6. Trajectory measured by the robot and typical trajectory obtained by adding large Gaussian noise with standard deviations $\langle 400, 20, 20 \rangle$ to these data.

We used the data recorded during this run of the robot and added different kinds of noise to the odometry information. Whenever the robot passed a reference position, which was detected given appropriate time stamps in the recorded data, we measured the distance between the estimated position and the corresponding reference position. We performed the same experiment 26 times with different seeds for each set of noise parameters. Figure 6 shows the trajectory measured by the robot's wheel encoders and a typical trajectory obtained by adding the maximum Gaussian noise $\langle 400, 20, 20 \rangle^2$.

We evaluated the scan matching technique by matching laser-range data with the self-learned map of the office environment and the CAD-model. The performance of the Markov localization technique was evaluated using ultrasound sensors and laser-range finders. In these experiments the grid resolution of Markov localization was adopted according to the noise ratio. It ranged from $15cm^2$ and $3°$ at the lowest noise level to $30cm^2$ and $10°$ at the highest noise level. For each method we measured the average distance between the estimated and the reference position. The distances were averaged over all situations in which the position was not lost. We used the threshold of $1m$ to determine whether or not the position of the robot was lost.

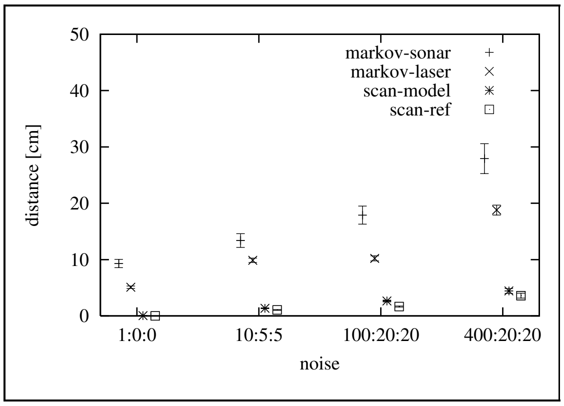


Fig. 7. Distances from reference positions in the office environment for different levels of Gaussian noise.

Figure 7 shows the average distances between the reference positions for four different levels of Gaussian noise. The value triples on the x-axis correspond to the standard deviation of the Gaussian noise $\langle \Delta_\delta(\delta), \Delta_\alpha(\alpha), \Delta_\alpha(\delta) \rangle$. In this and all following figures the error bars indicate the $95\%$ confidence interval of the average mean. This figure demonstrates that the scan matching technique is significantly more accurate than the Markov localization tech-

[2]The values correspond to the standard deviation of the Gaussian noise $\langle \Delta_\delta(\delta), \Delta_\alpha(\alpha), \Delta_\alpha(\delta) \rangle$ with the units mm/m, deg/$360°$, and deg/m.

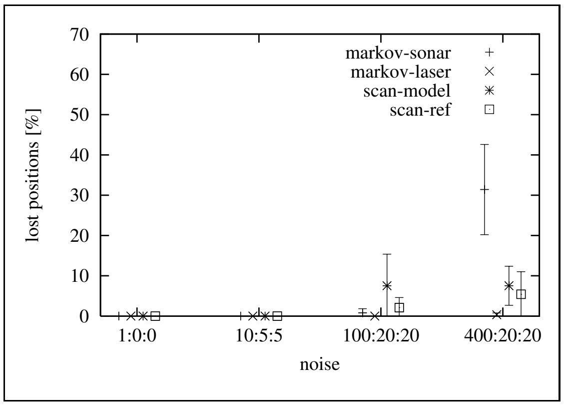


Fig. 8. Number of times where the positioning error was above $1m$ when the robot reached a reference position, for different levels of Gaussian noise in the office environment.

nique if the distance between the estimated and the reference position is smaller than $1m$.

However, scan matching turned out to be less robust against larger noise in the odometry. Figure 8 shows the number of times where the position of the robot was lost, whenever the robot reached a reference position, for four different noise parameters. As already mentioned, we regard the position to be lost, whenever the distance to the real position is at least $1m$. Under the conditions with the maximum Gaussian noise, Markov localization lost track of the robots position in 0.3% of all cases, which is significantly better than the scan matching technique. A further interesting fact is that Markov localization based on ultrasound sensors has a similar robustness to scan matching with laser-range finders in this experiment.

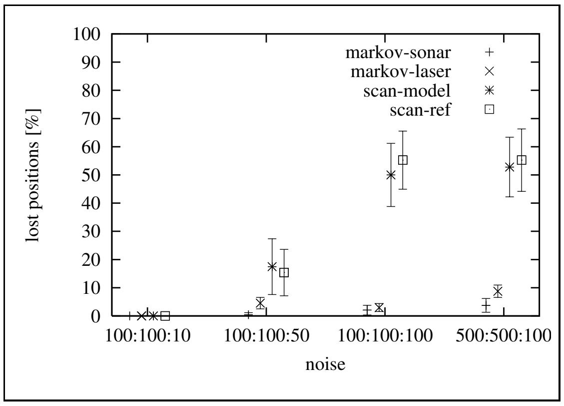


Fig. 9. Percentage of lost positions for different levels of bump noise in the office environment.

In the same manner, we investigated how the methods compare given simulated bump noise. For accuracy, the results were similar to the case of Gaussian noise. Scan matching again was significantly more accurate than Markov localization. Figure 9 shows the average number of reference positions at which the position of the robot was lost when bump noise was added to the odometry information. The labels at the x-axis again correspond to the bump-noise values $\langle x, y, \alpha \rangle$ used in this experiment. The scale of these values is mm for $x$ and $y$, and degrees for $\alpha$. In addition to these bumps occurring with probability 0.05, we applied a small Gaussian odometry error using the parameters $\langle 10, 5, 5 \rangle$. As shown in Figure 9, the Markov localization is significantly more robust than scan matching even when only ultrasound sensors are used for localization. The Gaussian distribution assumption of scan matching does not model bump noise well, while Markov localization can recover using its more robust assessment.

### 3.3. Performance in an Unstructured Populated Environment

We evaluated the performance of localization in an unstructured environment to test its sensitivity to map noise. This is important, since the application area of mobile robots is not restricted to structured office environments and since environments in which the robot operates cannot be assumed to be static, or perfectly described by the map.

Fig. 10. Typical situation in which several people surrounded the robot and lead to measurement errors.

The data for the experiments described in this section were recorded during the deployment of the mobile robot RHINO for several days as a personal tour-guide in the "Deutsches Museum Bonn," Germany [3]. A mobile robot moving in a real-world environment is quite challenging for localization techniques. In this case, the robot was often completely surrounded by visitors, and many sensor readings were shorter than expected. Figure 10 shows a typical situation in which RHINO gave a tour to visitors which block the sensors of the robot.

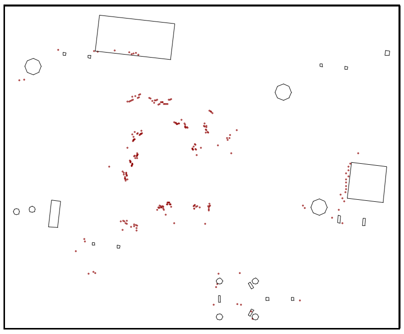

Fig. 11. Typical scan with many short readings in a situation like that shown in Figure 10. Rectangles and circles are objects contained in the map. The dots correspond to obstacles measured by the laser-range finder.

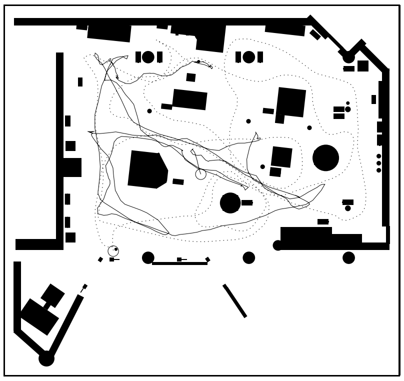

Fig. 12. Trajectories of the robot used in this experiment. The solid line indicates the trajectory of the robot while it was surrounded by people. The dotted line corresponds to the robot's path in the museum during closing hours.

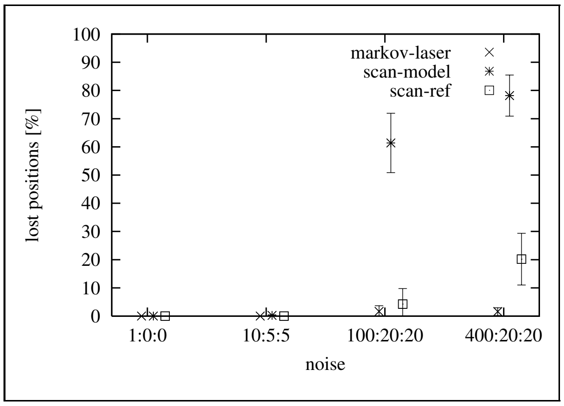


Fig. 13. Percentage of lost positions for different levels of Gaussian noise in the *empty* museum.

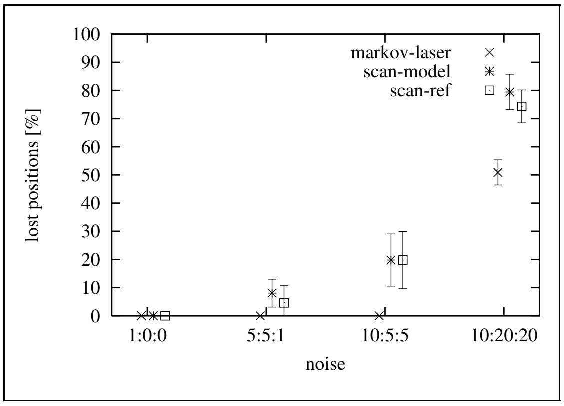


Fig. 14. Percentage of lost positions for different levels of Gaussian noise in the *crowded* museum.

We used two different data sets which are shown in Figure 12. While the solid trajectory was obtained during normal operation time of the robot, the dotted trajectory was recorded during a period of time where no visitors were present.

Figure 13 shows the relative time where the distance between the estimated position and a reference trajectory was above $1m$ for different levels of Gaussian noise. As in the previous examples in the structured office environment, scan matching was more accurate than Markov localization but less robust given larger noise[3]. Figure 14 contains similar plots for the trajectory recorded when people were present. Localization in this environment turned out to be much harder, which is illustrated by the fact that both methods failed earlier and even under small dead-reckoning noise. However, the relative performance of the methods was the same as in the empty museum, as well as the Bonn Computer Science environment. In this experiment, the Markov localization system used a sensor model that has been adopted to deal with the large number of too short readings.[4]

## 4. Discussion

This paper empirically compares two different and popular localization techniques for mobile robots: Markov localization, which represents arbitrary probability distributions across a grid of robot poses, and Kalman filtering which uses normal distributions together with scan-matching. Previous work reported in [18, 16, 9] largely focuses on the

[3]We omitted the results for ultrasound sensor based Markov localization since it already failed at lower degrees of Gaussian noise.

[4]Recently, an extension of Markov localization has been described [8] which is designed to filter out those measurements that are reflected by obstacles not contained in the map and thus shows a better performance than the version used in the experiments described here.

comparison of different matching strategies for Kalman filter based localization. Our work differs in that it compares different approaches to localization. While the two techniques analyzed here used similar Bayesian foundations, the choice of representation and subsequent algorithms differed significantly in their performance. The results of our empirical evaluation can be summarized broadly as follows.

- When sufficient information is available from the sensors, scan-matching and Kalman filtering are more accurate, sometimes by an order of magnitude.

- Markov localization is more robust, in that it potentially can keep track of the robot's position in an arbitrary probabilistic configuration. Having this position information is critical when the quality of information received from the sensors is degraded, and the odometry is unreliable.

The experimental evidence suggests combining these two techniques to produce a method that inherits the robustness of Markov localization and the efficiency and accuracy of Kalman filtering. Markov localization, at coarse grid spacing, could act as an overall check on the plausibility of scan matching: Whenever the position of the robot is uniquely determined, Kalman filtering is used to accurately estimate the position of the robot. As soon as Markov localization detects multiple positions where the robot is likely to be, Kalman filtering is no longer applied. As the Markov method converges on a single high-probability location, scan-matching could once again be invoked to produce high-accuracy results.

## References


[1] R. C. Arkin. Integrating behavioral, perceptual and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.

[2] J. Buhmann, W. Burgard, A. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot RHINO. *AI Magazine*, 16(2):31–38, Summer 1995.

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

[4] 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 Fourteenth National Conference on Artificial Intelligence*, pages 896–901, 1996.

[5] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proc. of the Fifteenth International Conference on Artificial Intelligence (IJCAI-97)*, 1997.

[6] J. Connell. *Minimalist Mobile Robotics: A Colony-style Architecture for an Artificial Creature*. Academic Press, 1990.

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

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

[9] 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.

[10] L. Kaelbling, A. Cassandra, and J. 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.

[11] J. Leonard, H. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. In *IROS*, pages 89–95, 1990.

[12] 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)*, pages 935–938, 1994.

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

[14] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 116–121, 1985.

[15] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, Summer 1995.

[16] 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.

[17] A. C. Schultz and W. Adams. Continuous localization using evidence grids. Technical Report AIC-96-007, Naval Center for Applied Research in Artificial Intelligence, 1996.

[18] G. Shaffer, J. Gonzalez, and A. Stentz. Comparison of two range-based estimators for a mobile robot. In *SPIE Conf. on Mobile Robots VII*, volume 1831, pages 661–667, 1992.

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

[20] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 1998. To appear.

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

[22] Z. Zhang and O. Faugeras. Estimation of displacements from two 3d frames obtained from stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):1141–1156, 1992.