IEEE Transactions on Robotics, 21(2), 2005.

Robust Vision-based Localization by Combining an Image Retrieval System with Monte Carlo Localization

Jürgen Wolf[†] Wolfram Burgard[‡] Hans Burkhardt[‡]

[†]Department of Computer Science, University of Hamburg, 22527 Hamburg, Germany
[‡]Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany

***Abstract*— In this paper we present a vision-based approach to mobile robot localization, that integrates an image retrieval system with Monte-Carlo localization. The image retrieval process is based on features that are invariant with respect to image translations and limited scale. Since it furthermore uses local features, the system is robust against distortion and occlusions which is especially important in populated environments. To integrate this approach with the sample-based Monte-Carlo localization technique we extract for each image in the database a set of possible view-points using a two-dimensional map of the environment. Our technique has been implemented and tested extensively. We present practical experiments illustrating that our approach is able to globally localize a mobile robot, to reliably keep track of the robot's position, and to recover from localization failures. We furthermore present experiments designed to analyze the reliability and robustness of our approach with respect to larger errors in the odometry.**

I. Introduction

Localization is one of the fundamental problems of mobile robots. The knowledge about the position of a robot is useful in different tasks such as office delivery, for example. In the past, a variety of approaches for mobile robot localization has been developed. They mainly differ in the techniques used to represent the belief of the robot about its current position and according to the type of sensor information that is used for localization. In this paper we consider the problem of vision-based mobile robot localization. Compared to proximity sensors cameras have several desirable properties. They are low-cost sensors that provide a huge amount of information and they are passive so that vision-based navigation systems do not suffer from the interferences often observed when using active sound- or light-based proximity sensors. Moreover, if robots are deployed in populated environments, it makes sense to base the perceptional skills used for localization on vision like humans do.

In principle, one can use a variety of different techniques from computer vision to facilitate the registration of perceived images with images stored in a database. Popular approaches are pixel-based techniques that compute the correlation between images or feature-based methods that exploit typical properties of environments such as linear structures. Image retrieval techniques can be regarded as a more general approach to view-registration. The goal of image retrieval is to find images in a given database that look similar to the given query image. Instead of relying on a specific kind of feature like lines or colors, image retrieval systems usually use a combination of different similarity measures that have been proven to yield accurate results in a wide variety of domains. This, in fact, makes image retrieval systems also attractive to mobile robot localization. Autonomous mobile robots must be applicable in a variety of different environments. Accordingly, they must be able to localize themselves in a wide range of situations. If the robots rely on vision for localization, they must possess a similarity measure that allows the retrieval of similar images for a huge variety of environments. However, they also need a technique to associate images in the database with locations in the environment.

In this paper we present an approach that combines an image retrieval system with the sample-based Monte-Carlo localization technique. The image retrieval system we use relies on features that are invariant with respect to image translations and scale (up to a factor of two) in order to find the most similar matches. Each feature consists of a set of histograms computed from the local neighborhood of each individual pixel. This makes the approach robust against occlusions and dynamic aspects such as people walking by. To incorporate sequences of images and to deal with the motions of the robot our system applies Monte-Carlo localization which uses a sample-based representation of the robot's belief about its position. During the filtering process the weights of the samples are computed based on the similarity values generated by the retrieval system and according to the visibility area computed for each reference image using a given map of the environment. Compared to other appearance-based techniques, the advantage of our approach is that the system is able to globally estimate the position of the robot and to recover from possible localization failures.

Our system has been implemented and tested on a real robot system in a dynamic office environment. In different experiments it has been shown to be able to globally estimate the position of the robot and to accurately keep track of it. We furthermore present experiments illustrating that our system is able to estimate the position of the robot even in situations in which the odometry suffers from serious noise.

This paper is organized as follows. After discussing related work in the following section we briefly describe Monte-Carlo localization that is used by our system to represent the belief of the robot. Section IV presents the techniques of the image-retrieval system used to compare the images grabbed with the robot's cameras with the reference images stored in the database. In Section V we describe how we integrate the image retrieval system with the Monte-Carlo localization system. Finally, in Section VI we present various experiments illustrating the reliability and robustness of the overall approach.

II. Related Work

Over the past years, several vision-based localization systems have been developed. They mainly differ in the features they use to match images. Horswill [1] extracts several kinds of environment-specific features like openings, walls or doors from images to realize a navigation system for a mobile robot. Basri and Rivlin [2] extract lines and edges from images and use this information to assign a geometric model to every reference image. Then they determine a rough estimate of the robots position by applying geometric transformations to fit the data extracted from the most recent image to the models assigned to the reference images. Dudek and Zhang [3] apply a neural network to learn the position of the robot. One advantage of this approach lies in the interpolation between the different positions from which the reference images were taken. Kortenkamp and Weymouth [4] extract vertical lines from camera images and combine this information with data obtained from ultrasound sensors to estimate the position of the robot. Whereas Dudek and Sim [5] apply a principal component analysis to learn landmarks, Paletta et al. [6] as well as Winters et al. [7] consider trajectories in the Eigenspaces of features. Dodds and Hager [8] use a heuristic color interest operator over color histograms to identify landmarks that are useful for navigation. A recent work presented by Se et al. [9] uses scale-invariant features to estimate the position of the robot within a small operational range. Olson [10] extracts depth information from stereo images in a probabilistic approach to mobile robot localization.

Additionally, there are approaches that rely on image-retrieval techniques to identify the current position of the robot. Kröse and Bunschoten [11] describe an appearance-based localization method which uses a principal component analysis on images recorded at different locations. Ulrich and Nourbakhsh [12] developed a system that uses color histograms for appearance-based localizations. As described in this paper, such an approach is quite efficient but suffers from the fact that pure color histograms cannot represent local relationships between pixels in the images.

Furthermore, there has been work in the context of the RoboCup in which camera data is used for mobile robot localization [13], [14], [15]. These techniques exploit given information about the environment (colors, lines, e.g.) and compare the images obtained from the robot with this model.

Dellaert et al. [16], [17] match images obtained with a camera pointed to the ceiling to a large ceiling mosaic covering the whole operational space of the robot. The mosaic has to be constructed in advance, which involves a complex state estimation problem. Finally, Thrun [18] developed an approach to learn landmarks that are useful for robot localization depending on the uncertainty of the robot in its current pose.

The techniques described above either use sophisticated feature-matching techniques or rely on simple features like lines and colors and use probabilistic state estimation or learning techniques to localize the robot. The goal of this paper is to illustrate that by combining a standard image retrieval system, that has been designed for a variety of different application domains [19], with sophisticated state estimation techniques one obtains a robust approach to vision-based robot localization. We describe how both approaches can be integrated by introducing a visibility area for each database image. In practical experiments we demonstrate that our approach is able to reliably keep track of the position of a mobile robot, to globally localize it, and to recover from potential localization failures.

III. Monte-Carlo Localization

To estimate the pose $l \in L$ of the robot in its environment, we apply a Bayesian filtering technique also denoted as *Markov localization* [20] which has successfully been applied in a variety of successful robot systems. The key idea of Markov localization is to maintain the probability density of the robot's own location $p(l)$. It uses a combination of the recursive Bayesian update formula to integrate measurements o and of the well-known formula coming from the domain of Markov chains to update the belief $p(l)$ whenever the robot performs a movement action a:

$$p(l \mid o, a) \quad = \quad \alpha \cdot p(o \mid l) \cdot \sum p(l \mid a, l') \cdot p(l') \qquad (1)$$

Here α is a normalization constant ensuring that the $p(l \mid o, a)$ sum up to one over all l. The term $p(l \mid a, l')$ describes the probability that the robot is at position l given it executed the movement a at position l'. Furthermore, the quantity $p(o \mid l)$ denotes the likelihood of the observation o given the robot's current location is l. It highly depends on the information the robot possesses about the environment and the sensors used. Different kinds of realizations can be found in [21], [22], [23], [20], [24]. In this paper, $p(o \mid l)$ is computed using the image retrieval system described in Section IV.

To represent the belief of the robot about its current position we apply a variant of Markov localization denoted as Monte-Carlo localization [17], [25]. In Monte-Carlo localization, the belief of the robot is represented by a set of random samples [26]. Each sample consists of a state vector of the underlying system, which is the pose l of the robot in our case, and a weighing factor ω. The latter is used to store the importance of the corresponding particle. The posterior is represented by the distribution of the samples and their importance factors. In the past a variety of different particle filter algorithms have been developed and many variants have been applied with great success to various application domains [27], [28], [29], [30], [31], [32], [33], [17]. The particle filter algorithm used by our system is also known as *sequential importance sampling* [26]. It updates the belief about the pose of the robot according to the following two alternating steps:

1) In the **prediction step**, we draw for each sample a new sample according to the weight of the sample

and according to the model $p(l \mid a, l')$ of the robot's dynamics given the action a executed since the previous update.

2) In the **correction step**, the new observation o is integrated into the sample set. This is done by bootstrap resampling, where each sample is weighted according to the likelihood $p(o \mid l)$ of making observation o given sample l is the current state of the system.

Particle filters have been demonstrated to be a robust technique for global position estimation and position tracking. To achieve re-localization in cases of localization errors several approaches have been proposed. They range from the insertion of random samples [25] to techniques that use the most recent observations to more intelligently insert samples at potential positions of the robot [15], [34].

IV. Image Retrieval Based on Invariant Features

In this section we briefly describe our method for comparing color images obtained with the robot's cameras with the images stored in the image database. In order to use an image database for mobile robot localization, one has to consider that the probability that the position of the robot exactly matches the position of an image in the database is virtually zero. Accordingly, one cannot expect to find an image that exactly matches the search pattern. In our case, we therefore are interested in obtaining similar images together with a measure of similarity between retrieved images and the search pattern.

Our image retrieval system simultaneously fulfills both requirements. The key idea of this approach, which is described in more detail in [35], [36], [19], is to compute features that are invariant with respect to image rotations, translations, and limited scale (up to a factor of two). To compare a search pattern with the images in the database it uses a histogram of local features. Accordingly, if there are local variations, only the features of some points of the image are disturbed, so that there is only a small change in the histogram shape. An alternative approach might be to use color histograms. However, this approach suffers from the fact that all structural information of the image is lost, as each pixel is assigned without paying attention to its neighborhood. Our database, in contrast, exploits the local neighborhood of each pixel and therefore provides better search results [35], [36].

In the remainder of this section we give a short description of the retrieval process for the case of gray-value images. To apply this approach to color images, one simply considers the different channels independently. Let $\mathbf{M} = \{\mathbf{M}(x_0, x_1), 0 \leq x_0 < N_0, 0 \leq x_1 < N_1\}$ be a gray-value image, with $\mathbf{M}(i,j)$ representing the gray-value at the pixel-coordinate (i,j). Furthermore let G be a transformation group with elements $g \in G$ acting on the images. For an image $\mathbf{M}$ and an element $g \in G$ the transformed image is denoted by $g\mathbf{M}$. Throughout this paper we consider the group of Euclidean motions:

$$(g\mathbf{M})(i,j) \quad = \quad \mathbf{M}(k,l) \tag{2}$$

with

$$\begin{pmatrix} k \\ l \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} - \begin{pmatrix} t_0 \\ t_1 \end{pmatrix}, \tag{3}$$

```
for all $f \in \mathcal{F}$ do
  for $x_0 = 0, \ldots, N_0 - 1, x_1 = 0, \ldots, N_1 - 1$ do
    $(\mathbf{T}[f](\mathbf{M}))(x_0, x_1) \leftarrow$
        $\frac{1}{P} \sum_{p=0}^{P-1} f(g(t_0 = x_0, t_1 = x_1, \varphi = p\frac{2\pi}{P})\mathbf{M})$
  end for
end for
```

Algorithm 1: Computation of a global Feature $\mathbf{F}(\mathbf{M})$ for an image M.

where all indices are understood modulo N_0 resp. N_1.

In the context of mobile robot localization we are especially interested in features $F(\mathbf{M})$ that are invariant under image transformations, i.e., $F(g\mathbf{M}) = F(\mathbf{M}) \forall g \in G$. For a given gray-value image $\mathbf{M}$ and a complex valued function $f(\mathbf{M})$ we can construct such a feature by integrating over the transformation group G [37]. In particular, the features are constructed by generating a histogram from a matrix $\mathbf{T}$ which is of the same size as $\mathbf{M}$ and is computed according to

$$(\mathbf{T}[f](\mathbf{M}))(x_0, x_1) = \frac{1}{P} \sum_{p=0}^{P-1} f\left(g(t_0 = x_0, t_1 = x_1, \varphi = p\frac{2\pi}{P})\mathbf{M}\right). \tag{4}$$

Since we want to exploit the local neighborhood of each pixel, we are interested in functions f that have a local support, i.e., that only use image values from the local neighborhood. Our system uses a set of different functions $\mathcal{F}$ with $f(\mathbf{M}) = \mathbf{M}(0,0)\mathbf{M}(0,1)$ as one member. For each such monomial, we generate a weighted histogram over $\mathbf{T}[f](\mathbf{M})$. These histograms are invariant with respect to image translations and rotations and robust against distortion and overlapping and therefore well-suited for mobile robot localization based on images stored in a database. Due to the fact that the kernel function f has local support we obtain invariance (or robustness) not only with respect to global Euclidean motion of the whole scene but also with respect to independent Euclidean motion of individual objects and to different appearances of articulated objects in the scene. Therefore, the results typically are rather stable, e.g. for a structured background with people moving independently in the foreground. The finally considered global feature $F(\mathbf{M})$ of an image $\mathbf{M}$ consists of a multi-dimensional histogram constructed out of all histograms computed for the individual features $\mathbf{T}[f](\mathbf{M})$ for all functions in $\mathcal{F}$.

Algorithm 1 describes precisely how $\mathbf{F}(\mathbf{M})$ is calculated given the individual kernel functions $f \in \mathcal{F}$. Figure 1 illustrates the calculation of $\mathbf{T}[f](\mathbf{M})$ for the kernel function $f = M(0,3) \cdot M(4,0)$. This function considers for each pixel (t_0, t_1) all neighboring pixels with distance 3 and 4 and with a phase shift of $\pi/2$ in polar representation relative to this pixel. The corresponding gray-levels are multiplied and $(\mathbf{T}[f](\mathbf{M}))(t_0, t_1)$ is the average over all angles φ. To evaluate f for a given angle φ the system uses bilinear interpolation. Figure 3 shows the feature matrix obtained for the color image shown in Figure 2 using this kernel function. Finally, Figure 4 depicts the histograms obtained for the three different color channels (red, green, and blue) of the image. Please

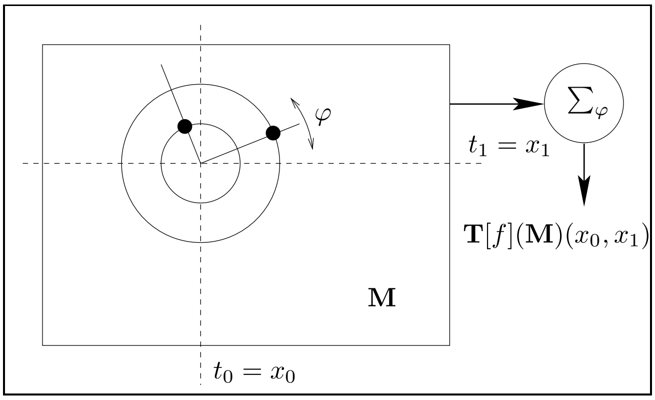

Fig. 1. Calculation of $\mathbf{T}[f](\mathbf{M})$ for $f = M(0,3) \cdot M(4,0)$.

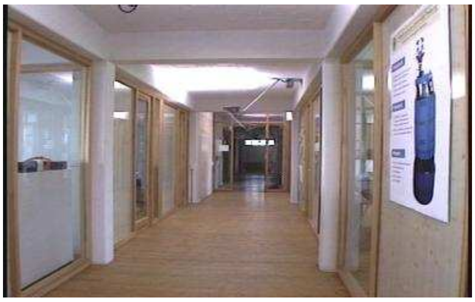

Fig. 2. Query image

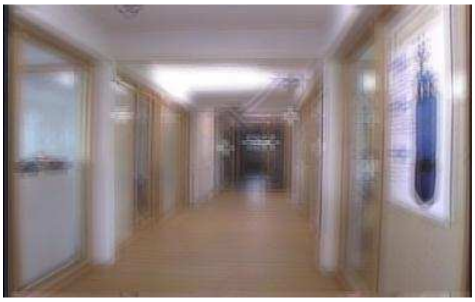

Fig. 3. Feature matrix obtained for the image shown in Figure 2 and the kernel depicted in Figure 1.

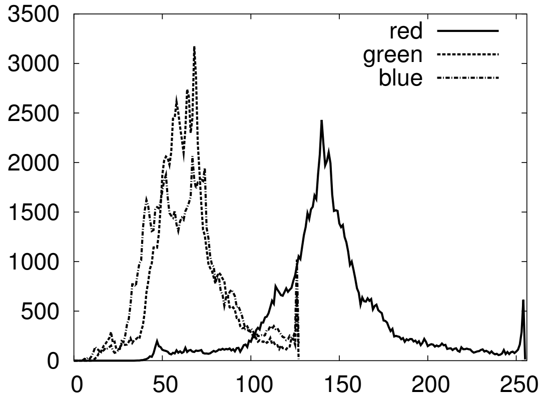

Fig. 4. Histogram obtained for the feature matrix depicted in Figure 3.

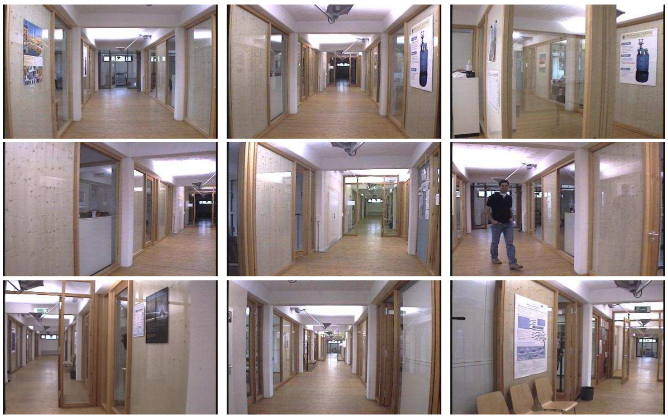

Fig. 5. The nine images with the highest similarity to the query image. The similarities from left to right, top-down are 81.67%, 80.18%, 77.49%, 77.44%, 77.43%, 77.19%, 77.13%, 77.06%, and 76.42%.

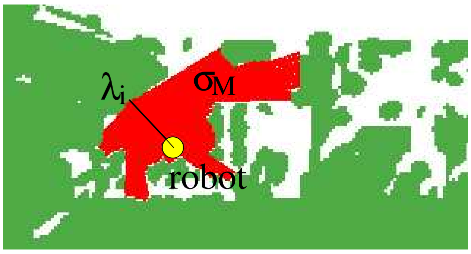

Fig. 6. Visibility area $\sigma_{\mathbf{M}}$ extracted for a reference image. The circle corresponds to the position of the robot when the image was grabbed in the environment depicted in Figure 8 (lower left portion). The position of the closest occupied grid cell in the direction of the optical axis is indicated by λ_i.

note that the individual features can be computed with sub-linear complexity (based on a Monte-Carlo integration over the Euclidean motion). Additionally, during the integration over the angle φ weighted means can be computed to deal with potential discretization errors.

The similarity between the global feature $\mathbf{q}$ of a query image and the global feature $\mathbf{d}$ of a database image is then computed using the intersection-operator normalized by the sum over all m histogram bins of the query image:

$$\bigcap_{\textbf{norm}} (\mathbf{q}, \mathbf{d}) = \frac{\sum\limits_{k \in \{0,1,\ldots,m-1\}} \min(q_k, d_k)}{\sum\limits_{k \in \{0,1,\ldots,m-1\}} q_k} \qquad (5)$$

Compared to other operators, the normalized intersection has the major advantage that it also allows to match partial views of a scene with an image covering a larger fraction. To achieve the invariance with respect to limited scale the image retrieval system also stores global features for scaled variants (up to a factor of two) of the individual kernel functions.

Figures 2 and 5 show an example of a database query and the corresponding answer. All images were recorded by our mobile robot in our department. The images in the answer are ordered by their similarity with the query image.

V. Using Retrieval Results for Robot Localization

The image retrieval system described above yields such images that are most similar to a given sample. In order to integrate this system with a Monte-Carlo localization approach, we need a technique to weight the samples according to the results of the image retrieval process. The key idea of our approach is to extract a visibility region $\sigma_{\mathbf{M}}$ for each image $\mathbf{M}$ in the image database. To determine the visibility regions for the individual images we use an occupancy grid map that is computed beforehand using the system developed by Hähnel et al. [38]. Given such a map we compute the visibility area of an image $\mathbf{M}$ corresponds as all positions in that map from which the closest occupied cell λ_i along the optical axis of $\mathbf{M}$.

We represent each $\sigma_{\mathbf{M}}$ by a discrete grid of poses and proceed in two steps: First we apply ray-casting to compute λ_i. Then we use a constrained region growing technique to determine the free grid cells in the occupancy grid map from which λ_i is visible. Figure 6 shows a typical example of the visibility area for one of the images stored in our database.

In Monte-Carlo localization one of the crucial aspects is the computation of the weight ω_i of each sample. Typically this weight corresponds to the likelihood $p(o \mid l_i)$ [17], [25] where l_i is the position represented by the sample and o is the measurement obtained by the robot. If we apply the law of total probability, we can compute $p(o \mid l_i)$ according to

$$p(o \mid l_i) \quad = \quad \sum_{j=1}^{n} p(o \mid l_i, \mathbf{M}_j) \cdot p(\mathbf{M}_j \mid l_i) \qquad (6)$$

where $\mathbf{M}_j, j = 1, \ldots, n$, are the images stored in the database. In our system we compute $p(o \mid l_i, \mathbf{M}_j)$ as the degree of similarity (see Equation (5)) denoted by ξ_j between the image $\mathbf{M}_j$ and the observation o. To determine the quantity $p(\mathbf{M}_j \mid l_i)$ we consider whether the location l_i of sample i lies in the visibility area σ_i of image $\mathbf{M}_i$. Since each sample represents a possible pose of the robot, i.e., a three-dimensional state consisting of the position $\langle x_i, y_i \rangle$ and orientation ϕ_i, we also have to incorporate ϕ_i to compute $p(\mathbf{M}_j \mid l_i)$. For example, if ϕ_i differs largely from the direction towards λ_i, the image stored in the database cannot be visible for the robot. Accordingly, the likelihood $p(o \mid l_i)$ turns into

$$p(o \mid l_i) = \frac{1}{K_i} \sum_{j=1}^{n} \xi_j \cdot I(\langle x_i, y_i \rangle, \sigma_j) \cdot d(\psi_i), \qquad (7)$$

where

$$K_i = \sum_{j=1}^{n} I(\langle x_i, y_i \rangle, \sigma_j) \cdot d(\psi_i). \qquad (8)$$

In these equations $\psi_i \in [-180; 180)$ is the deviation of the heading ϕ_i of the sample from the direction to λ_j. Furthermore, d is a function which computes a weight according to the angular distance ψ_i. Finally, $I(\langle x_i, y_i \rangle, \sigma_j)$ is an indicator function which is 1 if $\langle x_i, y_i \rangle$ lies in σ_j and 0, otherwise. In our current implementation we use a step function for $d(\psi_i)$ such that only such areas are chosen, for which the angular distance $|\psi|$ does not exceed 5 degrees. Note that Equation (7) computes the average likelihood of the current image o over all images stored in the database. If no database image is visible from the position of a sample, in which case K_i turns out to be 0, we assume $p(o \mid l_i)$ to be equal to the prior probability of observations o. This quantity corresponds to the average similarity of perceived images to the images in the database.

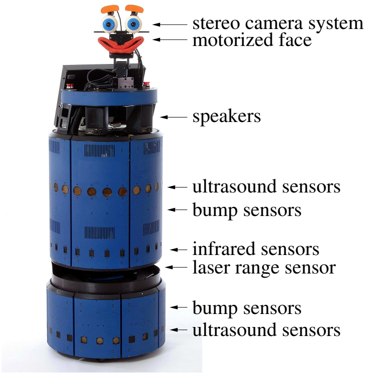

Fig. 7. The mobile robot Albert, a B21r robot equipped with Sony XC-999 cameras and standard TV cards used for image acquisition.

VI. Experiments

The system described above has been implemented on our mobile robot Albert and tested intensively in real robot experiments as well as in off-line runs using recorded data. Albert (see Figure 7) is an RWI B21 robot equipped with a stereo camera system. The image database used throughout the experiments contained 936 images. They were obtained by steering the robot through the environment and grabbing sets of images from different positions in the environment. The positions were determined with a localization system that uses laser range data [39]. The corresponding visibility areas covered approximately 80% of the state space that can be attained by the robot in this environment. Figure 5 shows 9 typical images stored in the database. Our system is highly efficient since it only stores the histograms representing the global features. The overall space used for all 936 images therefore does not exceed 4MB. Furthermore, the entire retrieval process for one query image usually takes less than .6 secs on an 800MHz Pentium III [40]. Please note that each update of the belief can be realized in $O(n \cdot k)$, where k is the number of samples contained in the sample set and n is the number of reference images stored in the database.

The goal of the experiments described in the remainder of this section is to demonstrate that our system allows the robot to reliably estimate the pose of a mobile robot. Furthermore, we present a simulation experiment carried out with recorded data that illustrates the robustness of our approach against large noise in the odometry.

A. *Tracking Capability*

The first experiment was carried out to analyze the ability to keep track of a robot's pose while it is moving with speeds up

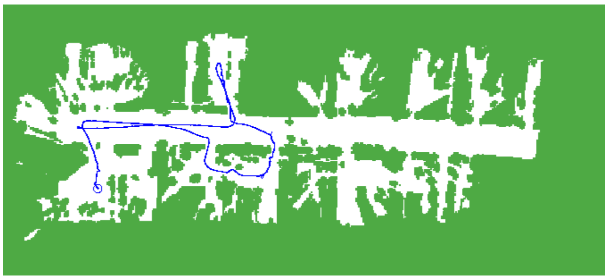

Fig. 8. Map of the office environment used to carry out the experiments and trajectory of the robot (ground truth). The size of the environment is 37 m times 14 m.

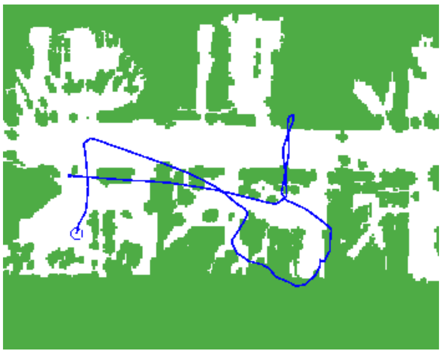

Fig. 9. Trajectory of the robot according to the odometry data.

to 30cm/sec through our office environment. In this experiment we steered the robot through the corridor and several rooms of our department. Figure 8 shows a part of the map of the environment and the trajectory of the robot during this experiment. This trajectory has been determined using laser range data. The accuracy of this localization procedure is below 5 cm. Also shown in green/gray is an outline of the environment. The significant error in the odometry obtained from the robot's wheel encoders is shown in Figure 9. Figure 10 shows the first 16 images captured by the robot. As can be seen from the figure, the lighting conditions are different at different places in the environment. Furthermore, the images contain dynamic objects such as doors as well as students present in the lab.

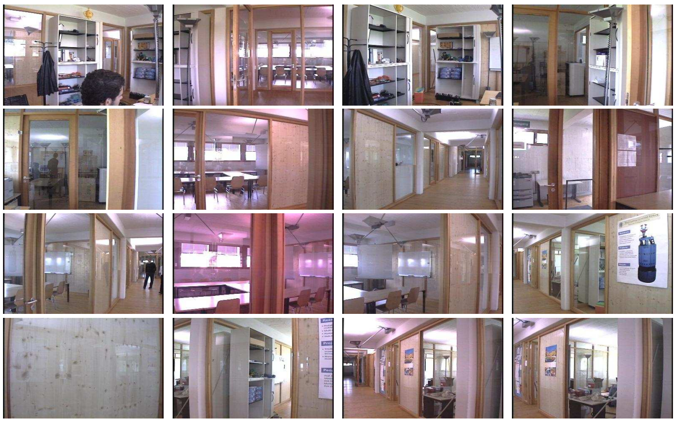

Fig. 10. Images captured by Albert during the experiment

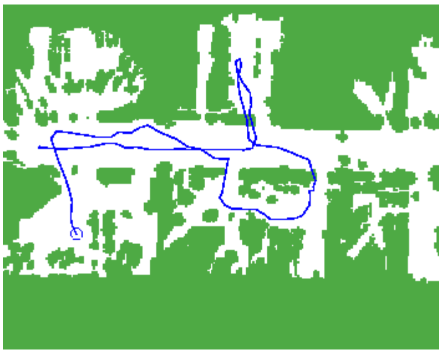

Fig. 11. Trajectory obtained by tracking the position of the robot using our system.

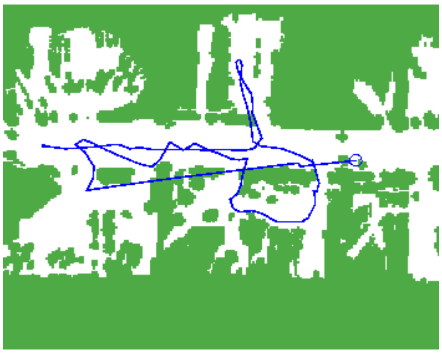

Fig. 13. Positions of the robot estimated by our system during global localization.

We initialized the sample set consisting of 5000 samples with a Gaussian centered at the starting pose of the robot. The trajectory estimated by our system is shown in Figure 11. As this figure illustrates, the system is able to correct the errors in odometry and to keep track of the position of the robot despite of the dynamic aspects. In this experiment the maximum pose error was less than 82 cm and 17 degrees.

B. *Global Localization*

The next experiment is designed to demonstrate the ability of the system to globally estimate the position of the robot. In this case we used the data obtained in the previous experiment and initialized the sample set, which again consisted of 5000 samples, with a uniform distribution. Figure 12 shows how the samples converge during the global localization process. In the beginning they are randomly distributed over the environment. After integrating four images the samples have almost concentrated on the true position of the robot (center image). The right image shows a typical sample set observed when the system has uniquely determined the position of the robot.

Figure 13 shows the trajectory estimated by our system. As can be seen, the system is able to quickly determine the position of the robot and to reliably keep track of it afterwards. Please note that we currently use the sample mean to estimate the robot's pose, so that, in the beginning, the estimated position is always close to the center of the map, which is not shown entirely in this figure. Because of the

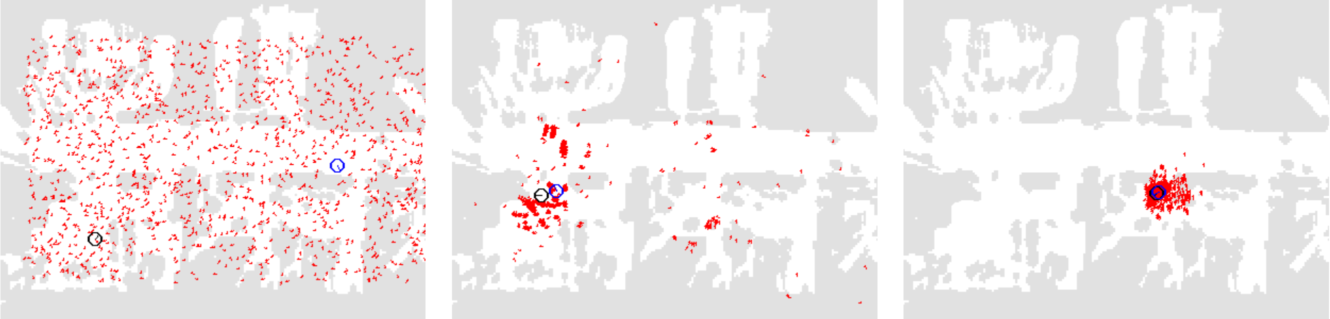

Fig. 12. Typical sample sets during global localization: At the beginning (left), after integrating 4 (center) and 35 (right) images.

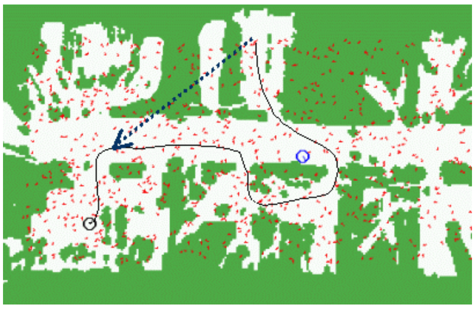

Fig. 14. Trajectory of the robot and kidnaping operation during the *kidnapped robot* experiment.

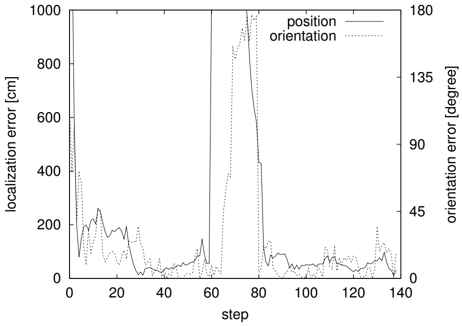

Fig. 15. Typical localization error during a *kidnapped robot* experiment.

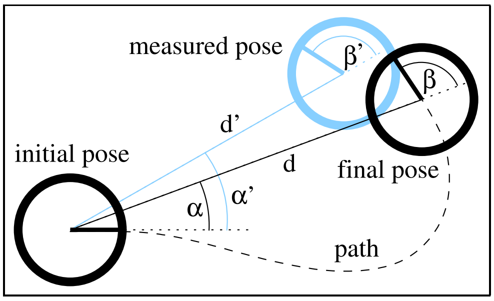

Fig. 16. Parameters of the probabilistic motion model.

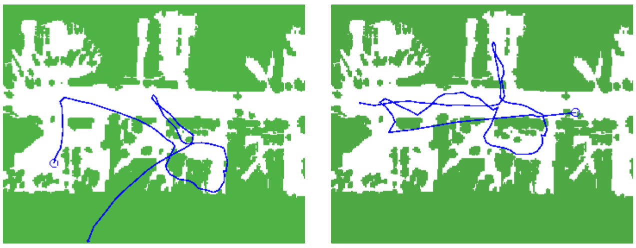

Fig. 17. Trajectory obtained after applying the noise according to $\langle 10, 5, 5\rangle$ to the odometry data (left image) and trajectory obtained after using our system to global localization (right image).

high uncertainty in the robot pose which typically results in multi-modal beliefs during global localization, the estimated trajectory of the systems often contain lines leading towards the current hypothesis about the robot's pose. Figure 13 only contains one such line leading from the center of the robot to the true position of the robot. After the integration of the fourth image in this particular run the belief consisted of a single mode.

C. Kidnapped Robot

The third experiment demonstrates the ability of our system to recover from localization failures. We initialized and started this experiment like the global localization experiment described above. After integrating 60 images, when the system already had determined the robot's position, we provided data corresponding to a completely different location, which corresponds to kidnaping the robot and taking it to a different place in the environment. Thus, the system had to re-localize itself. The trajectory and the kidnaping operation are indicated in Figure 14. To enable the system to deal with such situations, we randomly inserted 50 samples in each iteration. This approach has previously also been applied by Fox et al. [41]. More sophisticated schemes for mobile robot re-localization have recently been developed by Thrun et al. [34] as well as Lenser and Veloso [15].

Figure 15 shows the localization errors of one typical run. As can be seen, the system recovers the position approximately 20 steps after being kidnapped. We repeated this experiment 20 times and in all cases our system was able to re-localize the robot.

D. Robustness

The previous three experiments illustrate situations, in which the system is able to reliably estimate the position of the robot. To obtain a more quantitative assessment of

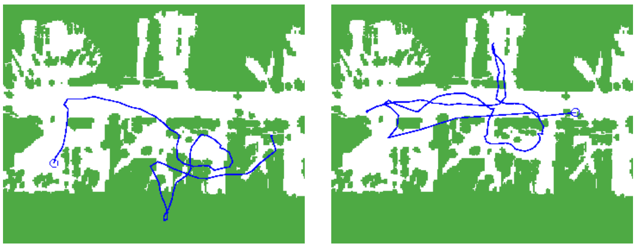

Fig. 18. Trajectories obtained by adding noise according to $\langle 20, 20, 20\rangle$ to the input data (left image) and trajectory obtained with our system after global localization (right image).

the performance of our approach, we performed a series of experiments using the data recorded in the tracking experiment. In each experiment we artificially distorted the odometry data by adding different amounts of noise to it. The model for odometry errors we used is similar to that used by Gutmann et al. [39] and is depicted in Figure 16. For each incremental movement carried out by the robot, we introduced a rotational error $\alpha' - \alpha$ at the beginning of the movement, a translational error $d' - d$ to the measured distance d between the final location and the starting position, and a rotational error $\beta' - \beta$ at the end of the movement. Each individual error was normally distributed . Two typical trajectories that resulted from this process are depicted as left images in Figures 17 and 18. The trajectories estimated by our system are shown as right images in the corresponding figures. As can be seen, the system is able to globally localize the robot and to reliably keep track of its position even in the case of large noise in odometry.

For different parameter sets we generated 20 different trajectories and for each resulting trajectory we used our system to estimate the pose of the vehicle. Then we counted the number of cases in which the pose error was below 2m and 35 degrees. Figure 19 shows the resulting statistics for nine different noise values. As the figure demonstrates, our system is robust against even large amounts of noise. Only for very large noise values, the success rate starts to drop. Please note, that we did not obtain a success-rate of 100%, because the system always had to perform a global localization in the beginning of each experiment.

E. Effect of the Image Retrieval System

The visibility areas extracted for the reference images (see Section V) introduce constraints on the possible locations of the robot while it is moving through the environment. In principle, the visibility areas characterize the free space in the environment. Therefore, just by knowing the odometry information one can often infer the position of the system. Since a robot cannot move through obstacles, the trajectory that minimizes the tradeoff between the deviation from the odometry data and the number of times the robot moves through obstacles corresponds to the most likely path of the robot and thus indicates the most likely position of the vehicle. In the past it has already been demonstrated that these constraints can be sufficient to globally localize a robot [42].

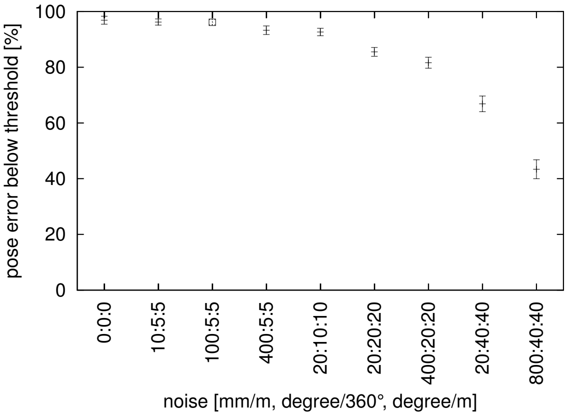

Fig. 19. Number of times when the pose error was not larger than 2m and 35 degrees.

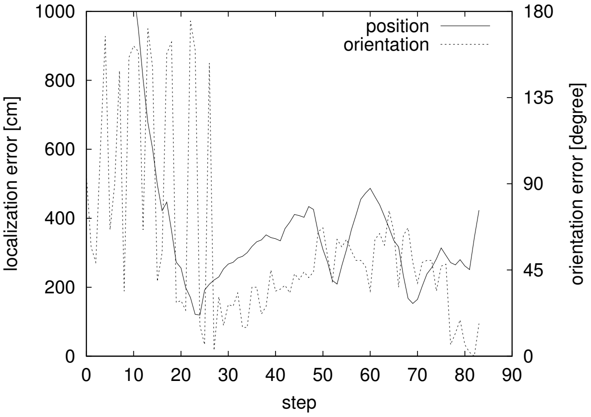

Fig. 20. Typical localization error during the *global localization* experiment using the constraints implied by the visibility areas.

The goal of the experiment described in this section therefore is to demonstrate that the localization capabilities presented in this paper significantly depend on the exploitation of the image retrieval results.

To evaluate the contribution of the image retrieval system we again evaluated the global localization capabilites but without utilizing the results from image retrieval process. More specifically, all samples obtained the same weight as long as the were located within an arbitrary visibility region. Accordingly, the outcome resulted only from the odometry data and from the constraints introduced by the visibility areas. The image depicted in Figure 20 shows a typical plot of the localization error if only the constraints imposed by the visibility regions are used. As can be seen from the figure, the system is unable to localize the robot solely based on this information. However, if the image retrieval results are used, the localization accuracy is quite high and the robot is quickly able to determine its absolute position in the environment (see Figure 21).

VII. Conclusions

In this paper we presented a new approach for vision-based localization of mobile robots. Our method uses an image

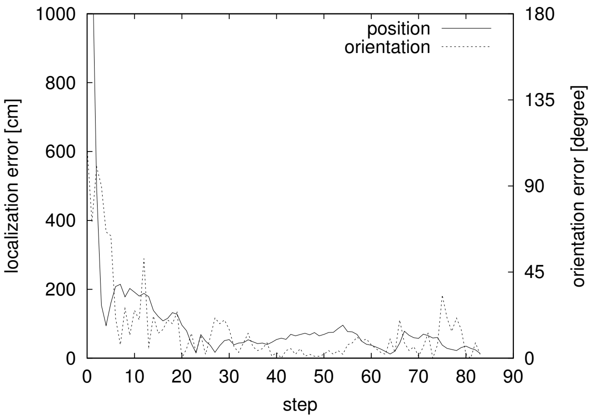

Fig. 21. Localization error during the *global localization* experiment if additionally the retrieval results are used.

retrieval system based on invariant features. These features are invariant with respect to translation and scale (up to a factor of two) so that the system is able to retrieve similar images even if only a small part of the corresponding scene is seen in the current image. This approach is particularly useful in the context of mobile robots, since a robot often observes the same scene from different view-points. Furthermore, the system uses local features and therefore is robust to changes in the scene. To represent the belief of the robot about its pose, our system uses a probabilistic approach denoted as Monte-Carlo localization. The combination of both techniques yields a robust vision-based localization system with several desirable properties. It is able to globally estimate the position of the robot and to reliably keep track of it and to recover from localization failures. Additionally, our system can deal with dynamic aspects in the scenes such as people walking by as well as with large amounts of noise in the odometry data. In extensive experiments carried out on real robots and in an unmodified office environment we have demonstrated the general applicability of our technique.

References

[1] I. Horswill, "Polly: A vision-based artificial agent," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1993, pp. 824–829.

[2] R. Basri and E. Rivlin, "Localization and homing using combinations of model views," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 327–354, 1995.

[3] G. Dudek and C. Zhang, "Vision-based robot localization without explicit object models," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1996, pp. 76–82.

[4] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1994, pp. 1972–1978.

[5] R. Sim and G. Dudek, "Learning visual landmarks for pose estimation," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.

[6] L. Paletta, S. Frintrop, and J. Hertzberg, "Robust localization using context in omnidirectional imaging," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001, pp. 2072–2077.

[7] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, "Omni-directional vision for robot navigation," in *Proc. IEEE Workshop on Omnidirectional Vision, South Carolina*, 2000.

[8] Z. Dodds and G. Hager, "A color interest operator for landmark-based navigation," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1997, pp. 55–660.

[9] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.

[10] C. Olson, "Probabilistic self-localization for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 1, pp. 55–66, 2000.

[11] B. Kröse and R. Bunschoten, "Probabilistic localization by appearance models and active vision," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999, pp. 2255–2260.

[12] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000, pp. 1023–1029.

[13] T. Schmitt, R. Hanek, M. Beetz, S. Buck, and B. Radig, "Cooperative probabilistic state estimation for vision-based autonomous soccer robots," *IEEE Transactions on Robotics and Automation*, pp. 670–684, 2002.

[14] C. Marques and P. Lima, "Vision-based selflocalization for soccer robots," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2000, pp. 1193–1198.

[15] S. Lenser and M. Veloso, "Sensor resetting localization for poorly modelled mobile robots," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000, pp. 1225–1230.

[16] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "MINERVA: A second generation mobile tour-guide robot," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.

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

[18] S. Thrun, "Bayesian landmark learning for mobile robot localization," *Machine Learning*, vol. 33, no. 1, pp. 41–76, 1998.

[19] S. Siggelkow, "Feature histograms for content-based image retrieval," Ph.D. dissertation, University of Freiburg, Department of Computer Science, 2002.

[20] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, pp. 3–55, 2000.

[21] I. Nourbakhsh, R. Powers, and S. Birchfield, "DERVISH an office-navigating robot," *AI Magazine*, vol. 16, no. 2, pp. 53–60, 1995.

[22] L. Kaelbling, A. Cassandra, and J. Kurien, "Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 1996, pp. 963–972.

[23] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan, "A layered architecture for office delivery robots," in *Proc. of the First International Conference on Autonomous Agents*, Marina del Rey, CA, 1997, pp. 245–252.

[24] K. Konolige, "Markov localization using correlation," in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999, pp. 1154–1159.

[25] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: Efficient position estimation for mobile robots," in *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1999, pp. 343–349.

[26] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[27] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[28] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[29] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Procedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[30] J. Carpenter, P. Clifford, and P. Fernhead, "An improved particle filter for non-linear problems," *IEE Proceedings on Radar and Sonar Navigation*, vol. 146, no. 2-7, 1999.

[31] K. Kanazawa, D. Koller, and S. Russell, "Stochastic simulation algorithms for dynamic probabilistic networks," in *Proc. of the 11th Annual Conference on Uncertainty in AI*, 1995, pp. 346–351.

[32] D. Fox, S. Thrun, F. Dellaert, and W. Burgard, "Particle filters for mobile robot localization," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer Verlag, 2000.

[33] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.

[34] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence Journal*, vol. 128, pp. 99–141, 2001.

[35] S. Siggelkow and H. Burkhardt, "Image retrieval based on local invariant features," in *Proceeding of the IASTED International Conference on Signal and Image Processing*, 1998, pp. 369–373.

[36] R. Veltkamp, H. Burkhardt, and H.-P. Kriegel, Eds., *State-of-the-Art in Content-Based Image and Video Retrieval.* Kluwer Academic Publishers, 2001.

[37] H. Schulz-Mirbach, "Invariant features for gray scale images," in *17. DAGM - Symposium "Mustererkennung"*, G. Sagerer, S. Posch, and F. Kummert, Eds. Springer, 1995.

[38] D. Hähnel, D. Schulz, and W. Burgard, "Map building with mobile robots in populated environments," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 2002, pp. 496–501.

[39] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige, "An experimental comparison of localization methods," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 1998, pp. 726–743.

[40] J. Wolf, "Bildbasierte Lokalisierung für mobile Roboter," Master's thesis, Department of Computer Science, University of Freiburg, Germany, 2001, in German.

[41] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proc. of the National Conference on Artificial Intelligence*, 1999, pp. 343–349.

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