Using an Image Retrieval System for Vision-Based Mobile Robot Localization

Jürgen Wolf[1], Wolfram Burgard[2], and Hans Burkhardt[2]

[1] Department of Computer Science, University of Hamburg
2527 Hamburg, Germany
jwolf@informatik.uni-hamburg.de

[2] Department of Computer Science, University of Freiburg
79110 Freiburg, Germany
{burgard,burkhardt}@informatik.uni-freiburg.de

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, rotations, and limited scale. Since it furthermore uses local features, the system is robust against distortion and occlusions which is especially important in populated environments. The sample-based Monte-Carlo localization technique allows our robot to efficiently integrate multiple measurements over time. Both techniques are combined by extracting for each image a set of possible view-points using a two-dimensional map of the environment. Our technique has been implemented and tested extensively using data obtained with a real robot. We present several experiments demonstrating the reliability and robustness of our approach.

1 Introduction

Localization is one of the fundamental problems of mobile robots. The knowledge about its position allows a mobile robot to efficiently fulfill different useful tasks like, for example, office delivery. 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, which are used by a variety of successful robot systems, 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.

Over the past years, several vision-based localization systems have been developed. They mainly differ in the features they use to match images. For example, Basri and Rivlin [1] extract lines and edges from images and use this

M. S. Lew, N. Sebe, and J. P. Eakins (Eds.): CIVR 2002, LNCS 2383, pp. 108–119, 2002.

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, Zhang, and Sim [5,18] use a neural network to learn the position of the robot given a reference image. One advantage of this approach lies in the interpolation between the different positions from which the reference images were taken. Kortenkamp and Weymouth [10] extract vertical lines from camera images and combine this information with data obtained from ultrasound sensors to estimate the position of the robot. Paletta et al. as well as Winters et al. [14,22] consider trajectories in the Eigenspaces of features. A recent work presented by Se et al. [16] uses scale-invariant features to estimate the position of the robot within a small operational range. Furthermore, there are different approaches [11,12,20] that use techniques also applied for image-retrieval to identify the current position of the robot. Whereas all these approaches use sophisticated feature-matching techniques, they are not applying any filtering techniques to estimate the pose of the robot. The approach presented by Dellaert et al. [4] apply a probabilistic method for mobile robot pose estimation denoted as Monte-Carlo localization. Their system, however, requires an accurate ceiling mosaic of the robot's environment.

In this paper we present an approach that combines techniques from image retrieval with Monte-Carlo localization and thus leads to a robust vision-based mobile robot localization system. Our image retrieval system uses features that are invariant with respect to image translations, image rotations, and scale (up to a factor of two) in order to find the most similar matches. These features consist of histograms based on features of the local neighborhood of each pixel. This makes the localization system robust against occlusions and dynamics 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. 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.

This paper is organized as follows. In the following section we present 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 3 we briefly describe Monte-Carlo localization that is used by our system to represent the belief of the robot. In Section 4 we describe how we integrate the image retrieval system with the Monte-Carlo localization system. Finally, in Section 5 we present various experiments illustrating the reliability and robustness of the overall approach.

2 Image Retrieval Based on Invariant Features

In order to use an image database for mobile robot localization, one has to consider that the probability that the position of the robot at a certain point in time 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 [17,21], 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 considered without paying attention to its neighborhood. Our database, in contrast, exploits the local neighborhood of each pixel and therefore provides better search results [17,21].

In the remainder of this section we give a short description of the retrieval process for the case of grey-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 grey-value image, with $\mathbf{M}(i,j)$ representing the grey-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:

$$(g\mathbf{M})(i,j) = \mathbf{M}(k,l) \quad \text{with} \quad \begin{pmatrix} k \\ l \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} - \begin{pmatrix} t_0 \\ t_1 \end{pmatrix}, \quad (1)$$

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 grey-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 [15]. In particular, the features are constructed by generating a weighted histogram from a matrix $\mathbf{T}$ which is of the same size as $\mathbf{M}$ and is computed according to

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

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

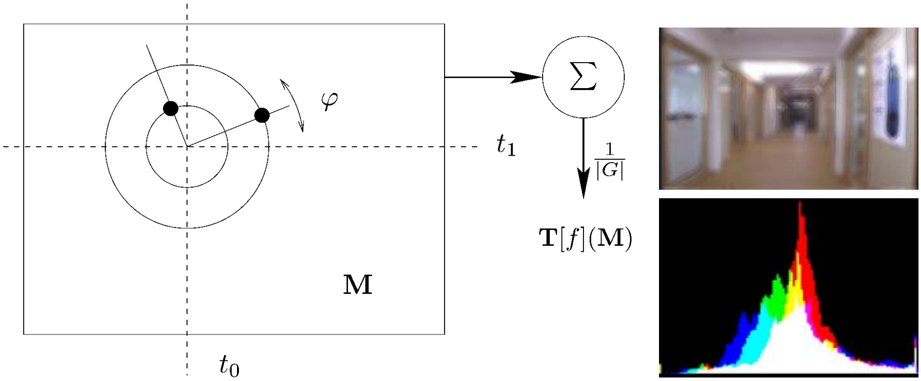

Fig. 1. Calculation of $\mathbf{T}[f](\mathbf{M})$ for $f = M(0,3) \cdot M(4,0))$, feature matrix (upper right image) and histogram (lower right image)

from a 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.

The kernel function defines the impact of surrounding pixels on the local feature of each coordinate. Obviously, for $f = M(0,0)$ the matrix of local features is given by $\mathbf{T}[f](\mathbf{M}) = \mathbf{M}$ and hence the global feature $F(\mathbf{M})$ simply is a grey-level histogram.

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. The corresponding grey-levels are multiplied and $(\mathbf{T}[f](\mathbf{M}))(t_0, t_1)$ is the average over all angles φ.

For each monomial $f \in \mathcal{F}$, 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. The upper resp. lower right image of Figure 1 shows the feature matrix resp. the resulting histogram for $f = M(0,3) \cdot M(4,0)$ extracted from the upper left image of Figure 2. In this figure each color channel has been calculated separately so that the chart contains an overlay of three histograms.

The 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 $\mathcal{F}$. Please note that the invariant features may be extracted with sub-linear complexity (based on a Monte-Carlo integration over the Euclidean motion) without the need of any feature extraction or segmentation.

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

$$\bigcap_{\mathbf{norm}} (F(\mathbf{Q}), F(\mathbf{D})) = \frac{\sum\limits_{k\in\{0,1,\ldots,m-1\}} \min(F(\mathbf{Q})_k, F(\mathbf{D})_k)}{\sum\limits_{k\in\{0,1,\ldots,m-1\}} F(\mathbf{Q})_k}, \tag{3}$$

where $F(\mathbf{M})_k$ denotes the value of the k-th (linear ordered) bin of the multidimensional histogram. 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. Figure 2 shows an example of a database query (upper left image) and the corresponding answer.

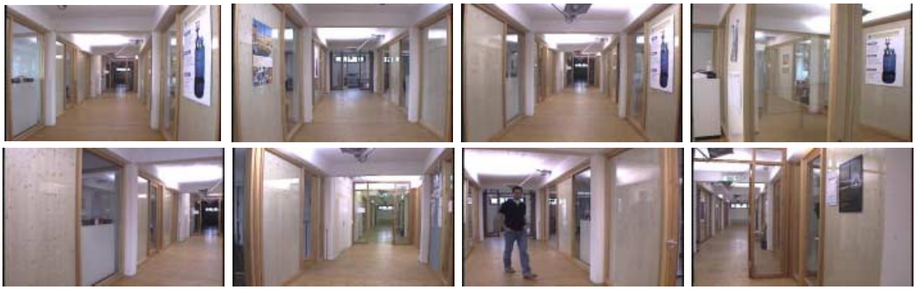

Fig. 2. Query image (upper left image) and the seven images with the highest similarity to it. The similarities from left to write, top-down are 81.67%, 80.18%, 77.49%, 77.44%, 77.43%, 77.19%, and 77.13%

3 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* [2] 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:

$$p(l \mid o, a) = \alpha \cdot p(o \mid l) \cdot \sum p(l \mid a, l') \cdot p(l'). \tag{4}$$

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 probability of making 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 [13,8,19,2,9]. In this paper, $p(o \mid l)$ is computed using the image retrieval system described in Section 2.

To represent the belief of the robot about its current position we apply a variant of Markov localization denoted as Monte-Carlo localization [4,6]. In Monte-Carlo localization, which is a variant of the well-known Condensation algorithm [7], the update of the belief generally is realized by 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.

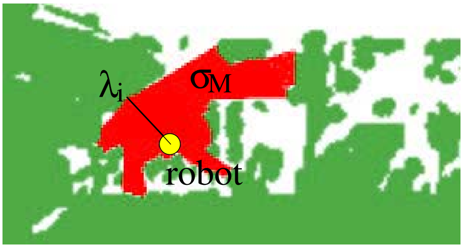

Fig. 3. Visibility area $\sigma_{\mathbf{M}}$ extracted for a reference image. The circle corresponds to the position of the robot when the image was grabbed in the environment depicted in Figure 4 (lower left portion). The position of the closest obstacle in the direction of the optical axis is indicated by λ_i

4 Using Retrieval Results for Robot Localization

The image retrieval system described above yields 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. In our current system, the visibility area of an image $\mathbf{M}$ corresponds to all positions in a given metric map of the environment from which the closest object in $\mathbf{M}$ in the direction of the optical axis is visible.

We represent each $\sigma_{\mathbf{M}}$ by a discrete grid of poses and proceed in two steps: First we apply ray-tracing to compute the position λ_i of the closest object on the optical axis according to the position of the robot when this image was grabbed. Then we use a constrained region growing technique to compute the visibility area $\sigma_{\mathbf{M}}$ for $\mathbf{M}$. Throughout this process only those points are expanded, from which λ_i is visible. Figure 3 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. In many systems this weight is chosen as the likelihood $p(o \mid l_i)$ [4,6] where l_i is the position represented by the sample and o is the measurement obtained by the robot. In the context of vision-based localization, however, $p(o \mid l_i)$ generally is hard to assess because of the high dimensionality of the image space. In our system, we use the similarity measure ξ_i of each image $\mathbf{M}_i$ to weight the samples in the corresponding visibility area $\sigma_{\mathbf{M}_i}$. Before we assign a similarity measure ξ_i to a sample, we need to check, whether the sample lies in the visibility area σ_i of image $\mathbf{M}_i$. At this point it is important to note, that each sample represents a possible pose of the robot, i.e., a three-dimensional state consisting of the $\langle x, y\rangle$-position and orientation ϕ. Thus, in order to appropriately weight the samples we also have to consider the orientation of that sample. For example, if the heading direction of pose represented by a sample is too far off, the image stored in the database cannot be visible for the robot.

In our system we compute the weight ω of a sample according to

$$\omega = \sum_{i=1}^{n} I(\langle x, y\rangle, \sigma_i) \cdot d(\psi) \cdot \xi_i, \tag{5}$$

where $\psi \in [-180; 180)$ is the deviation of the heading ϕ of the sample from the direction to λ_i. Furthermore, d is a function which computes a weight according to the angular distance ψ. Finally, $I(\langle x, y\rangle, \sigma_i)$ is an indicator function which is 1 if $\langle x, y\rangle$ lies in σ_i and 0, otherwise.

In our current implementation we use a step function so that only such areas are chosen, for which the angular distance $|\psi|$ does not exceed 5 degrees. Please note that Equation (5) rests on the assumption that the images in the database cover different aspects of the environment. For example, if the database contains two images taken from the same or a similar pose, then the weights of the samples lying in the intersection of both visibility areas would be weighted too high compared to other samples for which there is only one image. Although this independence assumption is not always justified, we did not observe any evidence in our experiments, that this made the robot overly confident in being at a certain position.

5 Experiments

The system described above has been implemented on our mobile robot Albert, an RWI B21 robot equipped with a stereo camera system, and tested intensively in real robot experiments. The image database used throughout the experiments contained 936 images. They were obtained by steering the robot

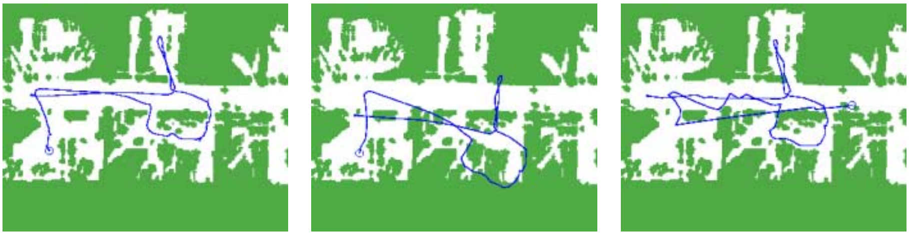

Fig. 4. Map of the office environment used to carry out the experiments and trajectory of the robot (ground truth) (left image). Trajectory of the robot according to the odometry data (center image). Positions of the robot estimated by our system during global localization (right image)

through the environment and grabbing sets of images from different positions in the environment. Our system is highly efficient since it only stores the histograms representing the global features. The overall space required for all 936 images therefore does not exceed 4MB. Furthermore, the retrieval process for one image usually takes less than .6 secs on an 800MHz Pentium III. Our current implementation (described in detail in [23]) updates the belief in each iteration in time $O(k^2 + n \cdot k)$, where k is the number of samples contained in the sample set and n is the number of reference images stored in the database.

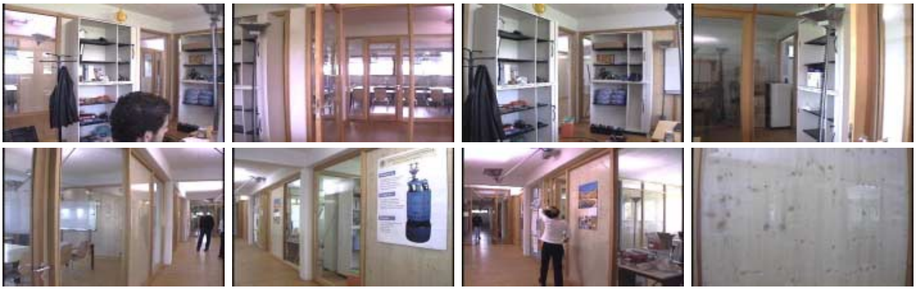

Fig. 5. Typical images captured by Albert during the global localization experiment

5.1 Global Localization

The experiment described in this section is designed to demonstrate that our system allows the robot to reliably estimate the global position of a mobile robot within its environment and to reliably keep track of it afterwards. During the experiment the robot was moving with speeds up to 30cm/sec through our office environment. The left image of Figure 4 shows a part of the map of the environment and the trajectory of the robot during this experiment. Also shown in green/grey is an outline of the environment. The significant error in the odometry obtained from the robot's wheel encoders is shown by the center image of the same figure. In this experiment we used a sample set consisting of 5000 samples that were initialized using a uniform distribution. Some of the images perceived by the robot during this experiment are depicted in Figure 5. The right image of Figure 4 shows the trajectory estimated by our system. Obviously, the system is able to quickly determine the position of the robot and to reliably keep track of it afterwards despite of the dynamic aspects. Please note that since we use the sample mean to estimate the robot's pose, in the beginning the estimated position is always in the center of the map, which is not shown entirely in this figure. One side-effect of using the sample mean is that the trajectories estimated by our system during global localization generally contain a line going from the center from the map to the true position of the robot. This corresponds to the situation in which the system has discovered the true position of the robot and happens after the integration of the fourth image in this particular example.

5.2 Effect of the Image Retrieval System

The visibility areas extracted for the reference images (see Section 4) 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 fact, it has already been demonstrated that these constraints can be sufficient to globally localize a robot [3].

The goal of the experiment described in this section therefore is to demonstrate that the localization capabilities significantly depend on the exploitation of the image retrieval results. Again, we 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 left image of Figure 6 shows a typical plot of the localization error if only the constraints imposed by the visibility regions are used (left image). 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 right image of Figure 6).

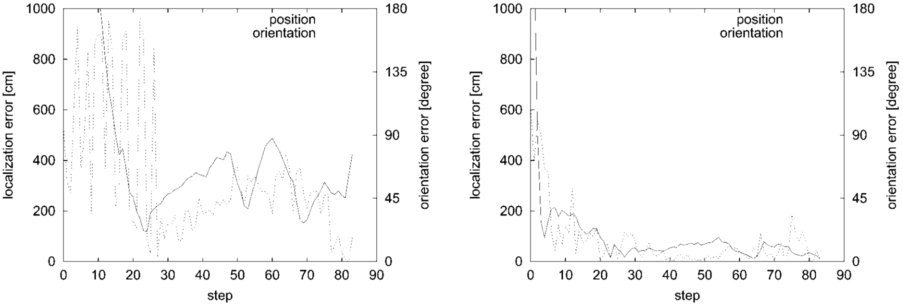

Fig. 6. Typical localization error during the *global localization* experiment affected by implied constraints (left picture) and additional using of retrieval results

6 Conclusions

In this paper we presented a new approach to vision-based localization of mobile robots. Our method uses an image retrieval system based on invariant features. These features are invariant with respect to translation, rotation, 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 previous approaches are lacking. It is able to globally estimate the position of the robot and to reliably keep track of it.

Acknowledgments

This work has partly been supported by the EC under contract number IST-2000-29456.

References

1. R. Basri and E. Rivlin. Localization and homing using combinations of model views. *Artificial Intelligence*, 78(1-2), 1995. 108
2. W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000. 112, 113

3. 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. 116
4. 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. 109, 113, 114
5. G. Dudek and C. Zhang. Vision-based robot localization without explicit object models. In *Proc. of the International Conference on Robotics & Automation (ICRA)*, 1996. 109
6. 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. 113, 114
7. M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of European Conference on Computer Vision*, pages 343–356, 1996. 113
8. L.P. Kaelbling, A. R. Cassandra, and J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 1996. 113
9. K. Konolige. Markov localization using correlation. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999. 113
10. 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. 109
11. B. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. In *Proc. of the International Conference on Robotics & Automation (ICRA)*, 1999. 109
12. Yoshio Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Visual navigation using omnidirectional view sequence. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, 1999. 109
13. I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2), 1995. 113
14. L. Paletta, S. Frintrop, and J. Hertzberg. Robust localization using context in omnidirectional imaging. In *Proc. of the International Conference on Robotics & Automation (ICRA)*, 2001. 109
15. H. Schulz-Mirbach. Invariant features for gray scale images. In G. Sagerer, S. Posch, and F. Kummert, editors, *17. DAGM - Symposium "Mustererkennung"*. Springer, 1995. 110
16. S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proc. of the International Conference on Robotics & Automation (ICRA)*, 2001. 109
17. 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. 110
18. R. Sim and G. Dudek. Learning visual landmarks for pose estimation. In *Proc. of the International Conference on Robotics & Automation (ICRA)*, 1999. 109
19. 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. 113

20. I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proc. of the International Conference on Robotics & Automation (ICRA)*, 2000. 109
21. R. Veltkamp, H. Burkhardt, and H.-P. Kriegel, editors. *State-of-the-Art in Content-Based Image and Video Retrieval*. Kluwer Academic Publishers, 2001. 110
22. 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. 109
23. J. Wolf. Bildbasierte Lokalisierung für mobile Roboter. Master's thesis, Department of Computer Science, University of Freiburg, Germany, 2001. In German. 115