Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

Vision-based Monte Carlo Self-localization for a Mobile Service Robot Acting as Shopping Assistant in a Home Store *

H.-M. Gross, A. Koenig, H.-J. Boehme, and Ch. Schroeter

Department of Neuroinformatics, Ilmenau Technical University, 98684 Ilmenau, Germany
Horst-Michael.Gross@tu-ilmenau.de

Abstract

We present a novel omnivision-based robot localization approach which utilizes the Monte Carlo Localization (MCL) [2], a Bayesian filtering technique based on a density representation by means of particles. The capability of this method to approximate arbitrary likelihood densities is a crucial property for dealing with highly ambiguous localization hypotheses as are typical for real-world environments. We show how omidirectional imaging can be combined with the MCL-algorithm to globally localize and track a mobile robot given a taught graph-based representation of the operation area. In contrast to other approaches, the nodes of our graph are labeled with both visual feature vectors extracted from the omnidirectional image, and odometric data about the pose of the robot at the moment of the node insertion (position and heading direction). To demonstrate the reliability of our approach, we present first experimental results in the context of a challenging robotics application, the self-localization of a mobile service robot acting as shopping assistant in a very regularly structured, maze-like and crowded environment, a home store.

1 Introduction and motivation

An interactive mobile service robot, e.g., a shopping assistant, should be able to actively observe its operation area, to detect, localize, and contact potential users, to interact with them continuously, and to adequately offer its specific services. Typical service tasks we want to solve in our PERSES (PERsonal SErvice System) project are to guide the user to desired areas or articles within a home store *(guidance function)* or to follow him as a mobile information kiosk while continuously observing the user and his behavior *(companion function)* (see [3]). To accommodate the challenges that arise from the specifics of our interaction-oriented scenario and the characteristics of the operation area, a very regularly structured, maze-like and crowded environment, we place special emphasis on vision-based methods for both human-robot interaction and robot navigation. The motivation for this is outlined in the following:

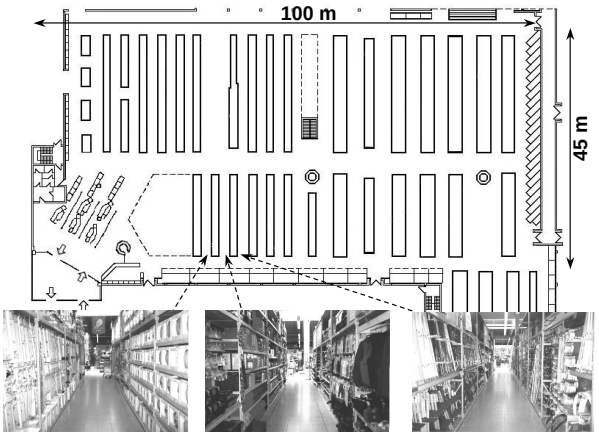

Figure 1: *(Top) Location plan of our experimental area, a large home store in Erfurt (toom BauMarkt). The topology of the store is characterized by many similar, long hallways of equal width. Because of their very regular structure, most of the hallways can be distinguished only visually. (Bottom) exemplary appearance of three hallways which can not be distinguished by distance sensors (sonar, laser) because of identical geometrical features. The hallways and racks, however, show very characteristic views, which allow a vision-based self-localization.*

Functional and economical advantages: Meanwhile, vision systems have become available as very powerful universal sensor systems with a good price-performance ratio such that they can be successfully utilized in a great number of robotics tasks - both in human-robot interaction and autonomous navigation. Therefore, our low-cost prototype of a mobile and interactive shopping assistant currently under development will be equipped with an universally usable omnidirectional vision-system instead of an expensive laser rangefinder, which shows a number of limitations in human-robot interaction and naviga-

*Supported by a Thuringian Ministry of Science, Research, and Art Grant (PERSES & SERROKON-Projects)

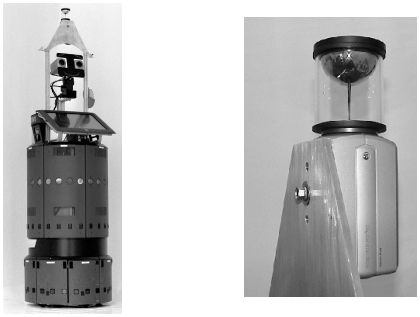

Figure 2: *(Left) Experimental platform PERSES, an extended version of a B21 robot, equipped with an omnidirectional imaging system on top. (Right) The omnidirectional vision system consists of a vertically oriented color camera (digital Sony CCD camera DFW-VL 500 with IEEE 1394 connection) and a middle-size spherical mirror (from Accowle).*

tion (see below). Omnidirectional imaging recently has become an effective basis for vision-based robot navigation and human-robot interaction. An excellent overview of the designs and principles of existing omnidirectional cameras and their applications in fields of autonomous robot navigation, telepresence, and remote surveillance is given in [11].

Specifics of the human-robot interaction: In the context of an interactive mobile shopping assistant, we have to cope with a number of demanding human-robot interaction tasks which require image processing to be solved: (1) fast and robust detection of users willing to interact with the shopping assistant, (2) fast learning of a specific appearance-based model of the current user allowing his tracking and re-detection if lost from view, and also preventing a confusion with other customers, (3) estimation of age and gender of the person willing to interact, (4) robust tracking of the current user both while standing still and during self-movement of the robot, (5) continuous estimation of the heading direction of the current user to estimate his actual interest in interaction (6) recognition of gesticulated user instructions, (7) telepresence for consultancy service, and remote surveillance.

Characteristics of the operation area: The topology of the operation area, a typical home store, is characterized by many similar, long hallways of equal length, width and geometrical structure (see Fig. 1) with a constant ceiling structure. Because of this regular, maze-like topology, self-localization methods based on distance sensors (laser or sonar) can produce numerous ambiguities complicating a quick self-localization or re-localization in case of a complete loss of positioning. Moreover, 2D-distance sensors only operate at certain planes of the 3D space. Therefore, goods racks cleared out show blank space at the respective height which can be misinterpreted by the navigation system as free space. In contrast, vision-based systems do not show these limitations, but supply a much greater wealth of information about the 3D-structure of the hallways and racks. The filling of the racks with different articles gives the hallways a characteristic appearance, especially with respect to color or texture. Because of this, we expected to defuse the localization problem drastically by development of an approach for view-based robot self-localization that combines omnidirectional imaging with the probabilistic Monte Carlo Localization (MCL) [2] which is based on the Condensation algorithm [4].

The robot PERSES we use as experimental platform is a standard B21 robot additionally equipped with an omnidirectional imaging system for vision-based navigation and human-robot interaction (Fig. 2). The omnivision system consists of a vertically oriented color camera and a middle-size spherical mirror mounted in front of the lens. The camera is mounted on top of the mobile platform with its axis placed coincidentally to the platform's rotation axis. The spherical mirror yields an image of the environment around the robot, and its field of view is the widest among sensors with convex mirrors [11]. Such a wide view is very useful for locating the robot along its route, as we are interested in.

2 Omnivision-based MCL

In recent years, several omnivision-based self-localization methods have been developed [10, 9, 6, 7]. Most of them use topological maps as environmental representation. During operation, commonly the input image or a describing feature vector is compared to all reference images or feature vectors of the topological map. While in many approaches, e.g. in [9], the location whose reference image best matches the input image is considered to be the location currently taken by the robot, in our approach, we use a distributed probabilistic location estimation. The reason is, that in uncertain and maze-like environments, localization on the basis of a crisp mapping from observations $\underline{\mathbf{o}}$ to states $\underline{\mathbf{x}}$ becomes unreliable. Because of this, soft computing methods are required to quantify the ambiguity by means of beliefs for multiple pose hypotheses [8, 6, 7].

The Monte Carlo Localization (MCL) method underlying our omnivision-based localization approach is a version of Markov localization [8], a family of probabilistic approaches for approximating a multi-modal probability density distribution coding the robot's belief $Bel(\underline{\mathbf{x}}_t)$ for being in state $\underline{\mathbf{x}}_t = (x, y, \varphi)_t$ in its state space. x and y are the robot's position coordinates in a world-centered Cartesian reference frame,

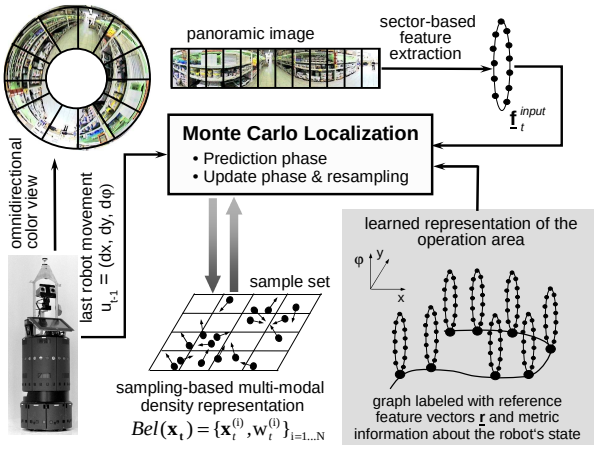

Figure 3: *General idea of our view-based Monte Carlo Localization. The approach is based on a graph-based representation of the operation area. The nodes of the graph are labeled with both view-based visual features and metric information about the pose of the robot (position and heading direction in a world-centered reference frame) at the moment of the node insertion during teaching.*

and φ is the robot's heading direction. MCL applies sampling techniques to represent the belief $Bel(\underline{\mathbf{x}}_t)$ for being in the current state $\underline{\mathbf{x}}_t$ by a set S_t of N weighted samples or particles distributed according to $Bel(\underline{\mathbf{x}}_t)$: $S_t = \{\underline{\mathbf{x}}_t^{(i)}, w_t^{(i)}\}_{i=1..N}$. Here each $\underline{\mathbf{x}}_t^{(i)}$ is a sample, and the $w_t^{(i)}$ are non-negative numerical weighting factors called importance factors. Because the sample set constitutes a discrete approximation of the continuous probability distribution, the MCL approach is computationally efficient, it places computation just "where needed". Additionally, it is more accurate than Markov localization with a fixed cell size, as state represented in samples is not discretized [2]. This allows a self-localization and position tracking with sub-grid accuracy.

2.1 Map building

We use a graph-based representation of the operation area by a set of visual reference vectors $\underline{\mathbf{r}}(x, y, \varphi)$ extracted from the respective panoramic views at positions x, y in heading direction φ (Fig. 3, bottom right). To train the system, it only needs to be led through the environment once. This training run can be performed under human control or by an autonomous exploration system. Presently, the graph is constructed when manually joy-sticking the robot through the hallways of the store. During training, omnidirectional images are captured from the environment and associated with the corresponding locations. In addition to the feature vectors extracted from the omnidirectional images (see Section 2.2), the nodes of the graph are labeled with metric information about the pose $\underline{\mathbf{x}} = (x, y, \varphi)$ of the robot at the moment of the node insertion. A new node (reference point) with importance for the representation is inserted, when the Euclidian distance to the positions of the reference points in a local Ω-vicinity is larger than $d_{distance}$ or if the norm of the difference vector between the current feature vector $\underline{\mathbf{f}}_t^{input}$ and the feature vectors $\underline{\mathbf{r}}_\Omega(x, y, \varphi)$ of these reference points is larger than $d_{feature}$.

However, the labeling of the graph nodes with odometric data about the pose of the robot necessitates an efficient correction of odometry because of the increasing error over time, especially concerning the orientation angle. This problem is well known and leads to the fact that a global map generated along a closed-loop course cannot be really closed without additional efforts (see [3]). To attenuate this effect, we utilize a specific feature of our market floor (ground). It shows a very regular structure caused by tiles that are uniquely oriented across the whole market area. The general idea of our vision-based odometry correction is illustrated in Fig. 4. A top-down oriented on-board camera acquires images of the floor in front of the robot. By continuously estimating the dominant orientations within these images, we can determine the accurate orientation of the robot and, therefore, substitute the rotation angle supplied by odometry by the orientation determined visually. Hence, it is possible to eliminate the orientation error, and subsequently, the position error. Under the assumption that the initial position and heading direction of the robot are known during training, this method allows an accurate, iterative position and orientation tracking as required for graph building. Earlier experiments (see [3]) already illustrated the efficiency of this specific correction method for building of large-scale occupancy maps based on sonar data. Of course, this correction

Figure 4: *General idea of our vision-based odometry correction considering the regular structure of the market floor: (from left to right) a) image of the floor in front of the robot, b) local orientation or structure tensors [5] (orientations coded as gray values), c) confidences of local orientations (low-black, high-white), d) histogram of confidence-weighted local orientations (interval $0^o - 90^o$). The dominant orientation (center of gravity) is a significant measure for the accurate orientation of the robot with respect to the global orientation of the tiles.*

method is not able to solve the localization problem in general because the translation error of the odometry can not be corrected by this approach and, therefore, is accumulated by the navigation system. However, because the total distance to be travelled during training is limited to about 2000 meters, the obtainable accuracy is sufficient. We utilized this odometry correction method for learning a graph representation of the operation area as shown in Fig. 5 and achieved a relatively low localization error of about $60cm$ after a total distance of 1000 meters. All dots represent nodes labeled with the pose of the robot and the visual feature vector extracted from the corresponding panoramic image.

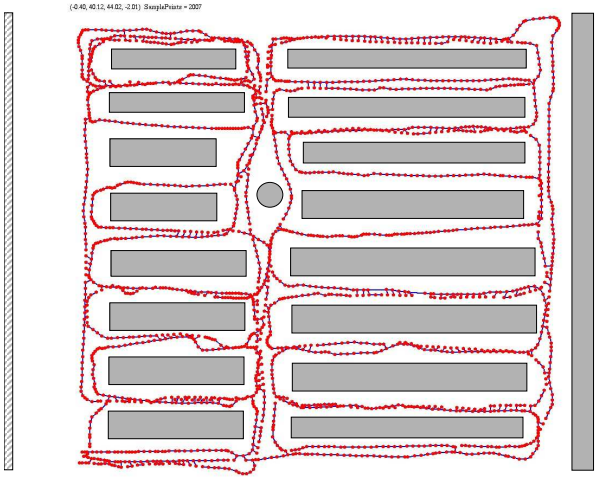

Figure 5: *Topological map of the operation area in the home store. The size of the area is* 42×45 *meters, the graph consists of 2007 reference points labeled with visual feature vectors and odometric data about the pose (position and orientation) of the robot at the moment of node insertion. The total distance travelled to learn this map was about 1000 meters.*

2.2 Iris control and feature extraction

To realize a constant image quality within the omnidirectional annulus of the camera image, we switched off the integrated automatic iris-control of the camera and implemented a simple but specific iris-control that realizes a constant average brightness only within this interesting part of the image. Because of this on-line control of the digital camera, in all experiments the image quality proved to be very constant despite extreme changes in illumination caused by shop windows, lamps, air shafts, and dark regions. Both during map-building and self-localization, in a first step, the omnidirectional image is transformed into a panoramic image (see Fig. 3, top). Each panoramic image is first partitioned into a fixed number of non-overlapping sectors (typically 10) each covering 36^o of the panoramic field of view. As a second step, features are extracted from these sectors.

The following criteria determined the selection of appropriate features to describe the present scene: 1) To allow a sufficient performance of the MCL-method as a whole (on-line localization), the calculation of the features should be as easy and efficient as possible. 2) The features should include the orientation of the robot, since based on it, the estimation of the x, y-position may be complemented by an estimation of the heading direction of the robot. 3) The feature description should allow for an easy generation of expected observations for unknown positions and orientations of the robot in its operation area, e.g. by interpolation between known reference positions. 4) The features should be largely insensitive against partial occlusion of the environment, such as caused by people in the direct vicinity of the robot.

Considering these criteria and the requirements and results of the omnivision-based localization approaches published in recent years [10, 9, 6, 7], we decided to implement the simplest feature extraction method possible. For each sector of the panoramic image an average color value in the RGB color space is determined (see Fig. 3, top). Note, that investigations comparing RGB to other color spaces, like HSI, did not show any advantages neither in accuracy nor in speed of the method. Our model of the operation area simply stores a reference feature vector $\underline{\mathbf{r}}(x, y, \varphi)$ consisting of mean RGB-values for each reference point in the graph. This provides a very efficient coding of the respective situation in the environment. The influence of size and number of segments on the localization accuracy is explained in Section 3 (Fig. 8).

Since the feature vectors are extracted from the segmented panoramic image, their organization is cyclic (see Fig. 3). This cyclic organization makes the calculation of the feature distribution of any desired orientation very easy: the reference vectors just need to be rotated by the appropriate angle, which is realized by linear interpolation between the discrete components of the feature vector. Moreover, we investigated whether it were possible to generate feature vectors for positions between reference points, i.e. completely unknown observations. We found that up to a distance of $90cm$, a linear interpolation yielded sufficiently accurate interpolation results.

2.3 The localization algorithm

In analogy to the MCL algorithm presented in [2], our omniview-based MCL proceeds in two phases, the *Prediction phase* and the *Update phase*.

Prediction phase (robot motion): In this phase, the sample set computed in the previous iteration

(or during random initialization) is moved according to the last movement of the robot u_{t-1} (Fig. 3, left). The *motion model* $p(x_t|x_{t-1}, u_{t-1})$ describes how the position of the samples changes using information u_{t-1} from the odometry. This way, MCL generates N new samples that approximate the expected probability density of the robot's position after the movement u_{t-1}. As described above, we use a graph-based model of the operation area labeled with both visual reference vectors $\underline{\mathbf{r}}(x, y, \varphi)$ and the corresponding pose data (Fig. 3). Because of this discrete graph representation and the chosen feature coding, our approach requires interpolations both in state and feature space to determine the expected feature vectors $\underline{\mathbf{f}}_t^{(i)}$ of the samples moved to new states $\underline{\mathbf{x}}_t^{(i)}$ within the continuous 3D state space. For each sample $s^{(i)}$, we first interpolate linearly between the two reference feature vectors $\underline{\mathbf{r}}(x, y, \varphi)$ of those reference nodes closest to the respective sample position $\underline{\mathbf{x}}_t^{(i)}$. After this, the resulting feature vector is rotated according to the expected new orientation $\varphi_t^{(i)}$ of the sample $s^{(i)}$. Since the feature vector only has a discrete number of components, for this step we use a linear interpolation between the features of neighboring segments. Thus, we obtain a set of N feature vectors $\underline{\mathbf{f}}_t^{(i)}(x, y, \varphi)$ describing the expected observations of the moved samples in the new states $\underline{\mathbf{x}}_t^{(i)}$.

Update phase (new observation): In this phase, the actual panoramic view at the new robot position has to be taken into account in order to re-weight the sample set S_t. For this, the importance factor $w_t^{(i)}$ of each sample $s^{(i)}$ is computed. It describes the probability that the robot is located in the state $\underline{\mathbf{x}}_t^{(i)}$ of the sample. We determine the similarity $E_t^{(i)}$ between the current input feature vector $\underline{\mathbf{f}}_t^{input}$ extracted from the panoramic view at the new robot position and each of the expected feature vectors $\underline{\mathbf{f}}_t^{(i)}$ of each sample $s^{(i)}$ simply by computing the angle between both normalized vectors applying a simple Gaussian-like *observation model*. Now $w_t^{(i)} = 1 - \alpha E_t^{(i)}$ can be determined, where α is a normalization constant that enforces $\sum_{j=1}^{N} w_t^{(j)} = 1$.

For the next iteration, the final sample set S_t is obtained by *re-sampling* from this weighted set. The re-sampling selects those samples with higher probability that have a high importance factor $w_t^{(i)}$. Samples with low importance factors are removed and randomly placed in the state-neighborhood of samples with high factors. After that, both phases are repeated recursively.

To allow for a faster self-localization or a re-localization of the robot, we extended the common MCL algorithm and inserted a new type of samples with fixed positions and orientations, so called grounded or "surviving" samples. These surviving samples are uniformly distributed within the state space and act as nuclei of crystallization for the freely movable regular samples in all cases, where these samples are already localized in a local region of the state space (convergent state), but a new localization is required, for example, because of a false localization or a "kidnapping" of the robot. Without these surviving samples a quick self-localization or re-localization in case of a complete loss of positioning is rarely possible. The influence of the number of surviving samples on the localization accuracy is analyzed experimentally in the next section.

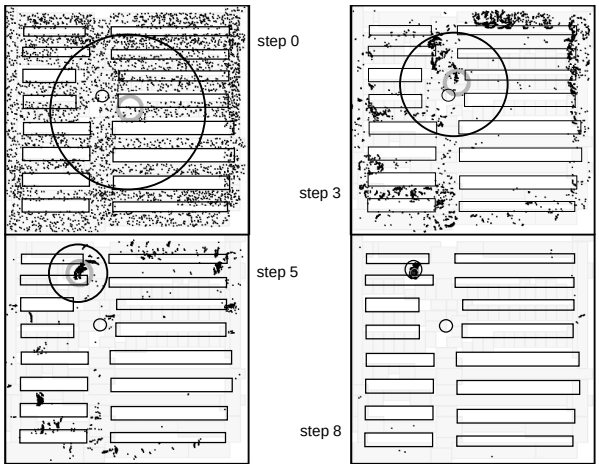

Figure 6: *Self-localization and tracking experiment executed in a large section ($42 \times 45m^2$) of the home store covering 18 hallways, 3 main passages, and 16 goods racks. The sequence depicts the temporal condensation dynamics of the samples (initial distribution, after 3 step, 5 steps, and 8 steps) - as result of local robot movements and the sampling/importance re-sampling cycle. In the beginning, the robot is globally uncertain, the particles are spread uniformly throughout the free space. The variance of the 10% of the samples with the highest importance factors is marked as circle. Already after 8 movements (about 5 meters), MCL has disambiguated the robot's position - the majority of samples is now centered tightly around the correct position, the variance is drastically reduced.*

3 Experimental results

All experiments were carried out in the 'toom' home store Erfurt with our experimental platform PERSES (see Fig. 2, left). The experiments were performed as off-line cross-validation tests on different sequences of images acquired in the home store. All images were labeled with the corresponding cor-

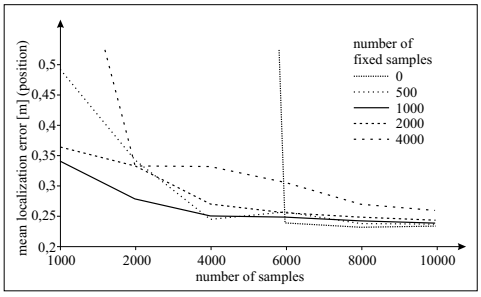

Figure 7: *Influence of the number of regular and fixed samples on the localization error.*

rect position and orientation of the robot. One of the sequences is used as training data (about 2000 pose-labeled panoramic images), while the other one is used as test data (about 5000 pose-labeled images) to determine the localization error. Every localization experiment has a typical length of 190 movements, this corresponds to a path length of about 130 meters. The average localization error is determined per experiment. Because of the temporal dynamics and the estimation error of the sample distribution at the beginning of each localization experiment, the estimations of the first 10 movements are not considered. Every experiment was repeated 20 times, and the localization errors were averaged.

It is to note that, in all cases, we studied the worst-case scenario: our robot had no prior information about its initial position and orientation - this is a typical kidnapped robot problem. All tests can be judged as being very successful, as our localization system was able to find and continually track the location of the robot. Fig. 6 illustrates the typical course of a view-based self-localization and position tracking experiment executed in a section of the store ($42 \times 45m^2$). Despite the geometrical uniformity of the selected hallways and the coarse graph-structure (2007 nodes), our omniview-based MCL yields very precise localization results already after a few robot movements. For example, after 8 movements and observations, which corresponds to a travelled distance of about 5 meters, the difference between estimated and correct position of the robot was lower than 40 cm. The average localization error of our test set is even smaller than 25 cm.

Fig. 7 illustrates the influence of the number of regular and surviving samples on the localization error. For the given localization problem, the best results were achieved with 4.000 to 10.000 regular and 500 to 1.000 surviving samples. Without surviving samples satisfactory localization results (mean error lower than 25 cm) can be obtained only if more than 6.000 regular samples are used. In contrast to this result, the error of the orientation estimation is largely independent of the number of regular samples. The reason, why the localization error is slightly higher compared to earlier experiments, is that the surviving samples added to the MCL algorithm can produce relatively high estimation errors due to their fixed positions that worsen the average error and even dominate it if less than 2000 regular samples are used. The advantage of the surviving samples, however, is the faster and more flexible self-localization and re-localization. Note, that in all following experiments a constant number of 4.000 regular samples and 500 surviving samples is used.

The time required for computation of the MCL algorithm directly depends on the total number of samples. With the current on-board equipment (1500 MHz AMD Athlon), for example, the sampling/importance re-sampling for 1000 samples requires about 12 ms, for 10.000 samples about 120 ms. The time for image transformation and feature extraction takes 25 ms per image. Therefore, our localization system enables real-time localization leaving a good amount of processing time for other navigation modules.

The influence of the number of panoramic image sectors used for feature extraction on the accuracy of the pose estimation is illustrated in Fig. 8. The function showing the localization error (left) and the orientation error (right) have an optimum for 10 segments used for training and localization. In this case, the mean localization error covers the range from 20 to 35 cm with a mean value of 25 cm. The average error of the orientation estimation (right) is largely independent from the number of segments and covers a range from 3.5 to 7.5 degrees for a number of 6 to 20 segments. If the feature extraction uses less than 6 segments, the localization and orientation errors increase drastically.

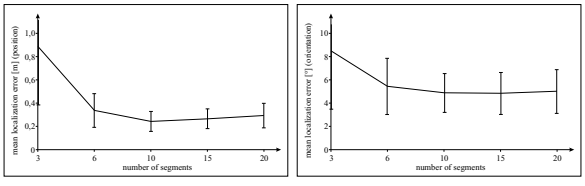

Figure 8: *Influence of the number of sectors for feature extraction on the localization error (left) and the orientation estimation (right).*

Dealing with occlusions: It is clear that we have to cope with occlusions in the scene, such as, for example, people walking by, other objects being moved around the environment, etc. Due to its wide visual field, occlusion of the entire panoramic view becomes very unlikely. For example, in Fig. 9 the two people standing as close as possible to the robot occlude

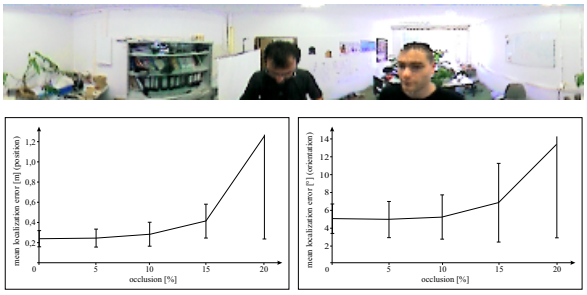

Figure 9: *(Top) Occlusion example: two people are standing as close as possible to the robot and occlude about 10% of the visual field. (Bottom) Result of experiments investigating the influence of local occlusions on the localization error (left) and and the orientation estimation (right).*

no more than 10% of the visual field. We expected that this occlusion is not sufficiently large so as to cause the robot to misinterpret its position. To test the robustness of the localization algorithm, the test images were occluded by artificial gray-colored segments. The impact of occlusion effects was gradually controlled by the percentage of image content covered by the artificial image. Fig. 9 (bottom) depicts the results w.r.t. localization accuracy and various degrees of occlusion. For 0% occlusion, the mean error of the localization is 25 cm and covers a range between 15 and 30 cm. The mean localization error remains relatively low until 15% occlusion. Thereafter, the error vigorously increases since the image is affected by severe occlusions. To cope with higher occlusions, the approach presented in [7] seems to be sufficiently powerful. However, due to the geometry of robot and vision system, it is not possible to place more than three or four people directly around the robot. Therefore, the maximum occlusion by people cannot be larger than 20%. Moreover, the particle dynamics of the MCL-algorithm realizes a temporal self-stabilization of the estimation result, therefore, the influence of large but short occlusions can be largely neglected.

4 Conclusions and future work

In this paper, we have shown that particle filters in combination with a graph-based representation of the operation area by local panoramic views can be used to perform omniview-based self-localization of a mobile service robot in a challenging real-world application, the navigation of a mobile service robot in a home store with a maze-like topology. The results of the executed experiments confirm the accuracy and robustness of our omniview-based self-localization method. Our localization system uses color omni-vision, works in real-time, and can easily be trained in new operation areas by joy-sticking.

Currently, theoretical and experimental studies are carried out to further improve our omniview-based MCL-system. For example, we are investigating the impact of the motion and observation models on the pose estimation and are studying the influence of a new mechanism adaptively controlling the sample rate on the localization accuracy and computing time. Other running experiments are dealing with the impact of appearance variations at the reference points in the learned graph, for example, as result of a changed filling of the goods racks or modifications in the market topology. The view-based localization approach presented here seems to be a computationally efficient and robust localization algorithm, however, it still has to demonstrate its capabilities scaling up to the whole market area with a size of $100 \times 60m^2$ over a longer period of operation.

References

[1] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the Condensation Algorithm for Robust Vision-based Mobile Robot Localization. In: *Proc. IEEE Int. Conf. on Comp. Vision and Pattern Recogn. 1999*

[2] D. Fox, W. Burgard, F. Dellaert, S. Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In: *Proc. AAAI-99*, 1999

[3] H.-M. Gross, H.-J. Boehme. PERSES - a Vision-based Interactive Mobile Shopping Assistant. in: *Proc. IEEE Intern. Conf. on Systems, Man and Cybernetics, 2000*, pp. 80-85

[4] M. Isard, A. Blake. CONDENSATION - conditional density propagation for visual tracking. *Int. Journal on Computer Vision*, 29 (1): 5-28, 1998

[5] B. Jaehne. *Practical Handbook on Image Processing for Scientific Applications*, CRC Press LLC, 1997

[6] B. Kroese, N. Vlassis, R. Bunschoten and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19 (6) pp. 381-391, 2001

[7] L. Paletta, S. Frintrop, and J. Hertzberg. Robust Localization Using Context in Omnidirectional Imaging. *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA 2001)*, pp. 2072-2077, 2001.

[8] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artif. Intell.*, 99 (1998) 21-71

[9] I. Ulrich and I. Nourbakhsh. Appearance-based Place Recognition for Topological Localization. *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA 2000)*, pp.1023-1029, 2000

[10] N. Winters, J. Gaspar, G. Lacey, J. Santos-Victor. Omnidirectional Vision for Robot Navigation. *Proc. IEEE Workshop on Omnidirectional Vision*, 2000

[11] Y. Yagi. Omnidirectional Sensing and Its Applications. *IEICE Trans Inf. & Syst.*, pp. 568-579, vol. E82-D, no. 3, 1999