Metric Localization with Scale-Invariant Visual Features using a Single Perspective Camera

Maren Bennewitz, Cyrill Stachniss, Wolfram Burgard, and Sven Behnke

University of Freiburg, Computer Science Institute, D-79110 Freiburg, Germany

Abstract. The Scale Invariant Feature Transform (SIFT) has become a popular feature extractor for vision-based applications. It has been successfully applied to metric localization and mapping using stereo vision and omnivision. In this paper, we present an approach to Monte-Carlo localization using SIFT features for mobile robots equipped with a single perspective camera. First, we acquire a 2D grid map of the environment that contains the visual features. To come up with a compact environmental model, we appropriately down-sample the number of features in the final map. During localization, we cluster close-by particles and estimate for each cluster the set of potentially visible features in the map using ray-casting. These relevant map features are then compared to the features extracted from the current image. The observation model used to evaluate the individual particles considers the difference between the measured and the expected angle of similar features. In real-world experiments, we demonstrate that our technique is able to accurately track the position of a mobile robot. Moreover, we present experiments illustrating that a robot equipped with a different type of camera can use the same map of SIFT features for localization.

1 Introduction

Self-localization is one of the fundamental problems in mobile robotics. The topic was studied intensively in the past. Many approaches exist that use distance information provided by a proximity sensor for localizing a robot in the environment. However, for some types of robots, proximity sensors are not the appropriate choice because they do not agree with their design principle. Humanoid robots, for example, which are constructed to resemble a human, are typically equipped with vision sensors and lack proximity sensors like laser scanners. Therefore, it is natural to equip these robots with the ability of vision-based localization.

In this paper, we present an approach to vision-based mobile robot localization that uses a single perspective camera. We apply the well-known Monte-Carlo localization (MCL) technique [5] to estimate the robot's position. MCL uses a set of random samples, also called particles, to represent the belief of the robot about its pose. To locate features in the camera images, we use the Scale Invariant Feature Transform (SIFT) developed by Lowe [15]. SIFT features are invariant to image translation, scale, and rotation. Additionally, they are partially invariant to illumination changes and affine or 3D projection. These properties make SIFT features particularly suitable for mobile robots since, as the robots move around, they typically observe landmarks from different angles and distances, and with a different illumination.

Whereas existing systems, that perform metric localization and mapping using SIFT features, apply stereo vision in order to compute the 3D position of the features [20, 7, 21, 2], we rely on a single camera only during localization. Since we want to concentrate on the localization aspect, we facilitate the map acquisition process by using a robot equipped with a camera and a proximity sensor. During mapping, we create a 2D grid model of the environment. In each cell of the grid, we store those features that are supposed to be at that 2D grid position. Since the number of observed SIFT features is typically high, we appropriately down-sample the number of features in the final map. During MCL, we then rely on a single perspective camera and do not use any proximity information. Our approach estimates for clusters of particles the set of potentially visible features using ray-casting on the 2D grid. We then compare those features to the features extracted from the current image. In the observation model of the particle filter, we consider the difference between the measured and the expected angle of similar features. By applying the ray-casting technique, we avoid comparing the features extracted out of the current image to the whole database of features (as the above mentioned approaches do), which can lead to serious errors in the data association. As we demonstrate in practical experiments with a mobile robot in an office environment, our technique is able to reliably track the position of the robot. We also present experiments illustrating that the same map of SIFT features can be used for self-localization by different types of robots equipped with a single camera only and without proximity sensors.

This paper is organized as follows. After discussing related work in the following section, we describe the Monte-Carlo localization technique that is applied to estimate the robot's position. In Section 4, we explain how we acquire 2D grid maps of SIFT features. In Section 5, we present the observation model used for MCL. Finally, in Section 6, we show experimental results illustrating the accuracy of our approach to estimate the robot's position.

2 Related Work

Monte-Carlo methods are widely used for vision-based localization and have been shown to yield quite robust estimates of the robot's position. Several localization approaches are image-based, which means that they store a set of reference images taken at various locations that are used for localization. Some of the image-based methods rely on an omnidirectional camera in order to localize a mobile robot. The advantages of omnidirectional images are the circular field of view and thus, the knowledge about the appearance of the environment in all possible gaze directions. Recent techniques were for example presented by Andreasson et al. [1] who developed a method to match SIFT features extracted from local interest points in panoramic images, by Menegatti et al. [16] who use Fourier coefficients for feature matching in omnidirectional images, and by Gross et al. [9] who compare the panoramic images using color histograms. Wolf et al. [23] apply a combination of MCL and an image retrieval system in order to localize a robot equipped with a perspective camera. The systems presented by Ledwich and Williams [12] and by Kŏsécka and Li [11] perform Markov localization within a topological map. They use the SIFT feature descriptor to match the current view to the reference images. Whenever using those image-based methods, care has to be taken in deciding at which positions to collect the reference images in order to ensure a complete coverage of the space the robot can travel in. In contrast to this, our approach stores features at the positions where they are located in the environment and not for all possible poses the robot can be in.

Additionally, localization techniques have been presented that use a database of observed visual landmarks. SIFT features have become very popular for metric localization as well as for SLAM (simultaneous localization and mapping, [21, 2]). Se et al. [20] were the first who performed localization using SIFT features in a restricted area. They did not apply a technique to track the position of the robot over time. Recently, Elinas and Little [7] presented a system that uses MCL in combination with a database of SIFT features learned in the same restricted environment. All these approaches use stereo vision to compute the 3D position of a landmark and match the visual features in the current view to all those in the database to find correspondences. To avoid matching the observations to the whole database of features, we present a system that determines the sets of visible features for clusters of particles. These relevant features are then matched to the features in the current image. This way, the number of ambiguities, which can occur in larger environments, is reduced. The relevant features are determined by applying a ray-casting technique in the map of features. The main difference to existing metric localization systems using SIFT features is however that our approach is applicable to robots that are equipped with a single perspective camera only, whereas the other approaches require stereo vision or omnivision.

Note that Davison et al. [3] and Lemaire et al. [13] presented approaches to feature-based SLAM using a single camera. These authors use extended Kalman filters for state estimation. Both approaches have only been applied to robots moving within a relatively small operational range.

Vision-based MCL was first introduced by Dellaert et al. [4]. The authors constructed a global ceiling mosaic and use simple features extracted out of images obtained with a camera pointing to the ceiling for localization. Systems that apply vision-based MCL are also popular in the RoboCup domain. In this scenario, the robots use environment-specific objects as features [19, 22].

3 Monte-Carlo Localization

To estimate the pose x_t (position and orientation) of the robot at time t, we apply the well-known Monte-Carlo localization (MCL) technique [5], which is a variant of Markov localization. MCL recursively estimates the posterior about the robot's pose:

$$\begin{aligned} & p(x_t \mid z_{1:t}, u_{0:t-1}) \\ & \quad = \eta \cdot p(z_t \mid x_t) \cdot \int_{x_{t-1}} p(x_t \mid x_{t-1}, u_{t-1}) \cdot p(x_{t-1} \mid z_{1:t-1}, u_{0:t-2})\, dx_{t-1} \end{aligned} \tag{1}$$

Here, η is a normalization constant resulting from Bayes' rule, $u_{0:t-1}$ denotes the sequence of all motion commands executed by the robot up to time $t-1$, and $z_{0:t}$ is the sequence of all observations. The term $p(x_t \mid x_{t-1}, u_{t-1})$ is called motion model and denotes the probability that the robot ends up in state x_t given it executes the motion command u_{t-1} in state x_{t-1}. The observation model $p(z_t \mid x_t)$ denotes the likelihood of making the observation z_t given the robot's current pose is x_t. To determine the observation likelihood, our approach compares SIFT features in the current view to those SIFT features in the map that are supposed to be visible (see Section 5).

MCL uses a set of random samples to represent the belief of the robot about its state at time t. Each sample consists of the state vector $x_t^{(i)}$ and a weighting factor $\omega_t^{(i)}$ that is proportional to the likelihood that the robot is in the corresponding state. The update of the belief, also called particle filtering, is typically carried out as follows. First, the particle states are predicted according to the motion model. For each particle a new pose is drawn given the executed motion command since the previous update. In the second step, new individual importance weights are assigned to the particles. Particle i is weighted according to the likelihood $p(z_t \mid x_t^{(i)})$. Finally, a new particle set is created by resampling from the old set according to the particle weights. Each particle survives with a probability proportional to its importance weight.

Due to spurious observations it is possible that good particles vanish because they have temporarily a low likelihood. Therefore, we follow the approach proposed by Doucet [6] that uses the so-called number of effective particles [14] to decide when to perform a resampling step

$$N_{\mathit{eff}} = \frac{1}{\sum_{i=1}^{N} \left(w^{(i)}\right)^2}, \tag{2}$$

where N is the number of particles. N_{eff} estimates how well the current particle set represents the true posterior. Whenever N_{eff} is close to its maximum value N, the particle set is a good approximation of the true posterior. Its minimal value 1 is obtained in the situation in which a single particle has all the probability mass contained in its state.

We do not resample in each iteration, instead, we only resample each time N_{eff} drops below a given threshold (here set to $\frac{N}{2}$). In this way, the risk of replacing good particles is drastically reduced.

4 Acquiring 2D Maps of Scale-Invariant Features

We use maps of visual landmarks for localization. To detect features, we use the Scale Invariant Feature Transform (SIFT). Each image feature is described by a vector $\langle p, s, r, f \rangle$ where p is the subpixel location, s is the scale, r is the orientation, and f is a descriptor vector, generated from local image gradients. The SIFT descriptor is invariant to image translation, scaling, and rotation and also partially invariant to illumination changes and affine or 3D projection. Lowe presented results illustrating robust matching of SIFT descriptors under various image transformations [15]. Mikolajczyk and Schmid compared SIFT and other image descriptors and showed that SIFT yields the highest matching accuracy [17].

Ke and Sukthankar [10] presented an approach to compute a more compact representation for SIFT features, called PCA-SIFT. They apply principal components analysis (PCA) to determine the most distinctive components of the feature vector. As shown in their work, the PCA-based descriptor is more distinctive and more robust than the standard SIFT descriptor. We therefore use that representation in our current approach. As suggested by Ke and Sukthankar, we apply a 36 dimensional descriptor vector resulting from PCA.

To acquire a 2D map of SIFT features, we used a B21r robot equipped with a perspective camera and a SICK laser range finder. We steered the robot through the environment to obtain image data as well as proximity and odometry measurements. The robot was moving with a speed of $40cm/s$ and collected images at a rate of $3Hz$. To be able to compute the positions of features and to obtain ground truth data, we used an approach to grid-based SLAM with Rao-Blackwellized particle filters [8]. Using the information about the robot's pose and extracted SIFT features out of the current camera image, we can estimate the positions of the features in the map. More specifically, we use the distance measurement of the laser beam that corresponds to the horizontal angle of the detected feature and the robot's pose to calculate the 2D position of the feature. Thus, we assume that the features are located on the obstacles detected by the laser range finder. In the office environment in which we performed our experiments, this assumption leads to quite robust estimates even if there certainly exist features that are not correctly mapped. In each 2D grid cell, we store the set of features that are supposed to be at that 2D grid position. Currently, we use a grid resolution of 10 by $10cm$. In the first stage of mapping, we store all observed features.

After the robot moved through the environment, the number of observed SIFT features is extremely high. Typically, we have 150-500 features extracted per image with a resolution of 320 by 240 pixels. This results in around 600,000 observed features after the robot traveled for $212m$ in a typical office environment. After map acquisition, we down-sample a reduced set of features that is used for localization. For each grid cell, we randomly draw features. A drawn feature is rejected if there is already a similar feature within the cell. We determine similar features by comparing their PCA-SIFT vectors (see below). We sample a maximum of 20 features for each grid cell. Using the sampling process, features that were observed more often have a higher chance to be selected and features that were detected only once (due to failure observations or noise) are eliminated. The goal of this sampling process is to reduce computational resources and at the same time obtain a representative subset of features. To choose good representatives for the features, a clustering based on the descriptor vectors can be carried out.

The left image of Figure 3 shows a 2D grid map of SIFT features of an office environment that was acquired by the described method. The final map contains approximately 100,000 features. Note that also a stereo camera system, which was not available in our case, would be an appropriate solution for map building. The presented map acquisition approach is not restricted to robots equipped with a laser range finder.

5 Observation Model for SIFT Features

In the previous section, we described how to built a map of SIFT features using a robot equipped with a camera and a proximity sensor. In this section, we describe how a robot without a proximity sensor can use this environmental model for localization with a single perspective camera.

Sensor observations are used to compute the weight of each particle by estimating the likelihood of the observation given the pose of the particle in the map. Thus, we have to specify how to compute $p(z_t \mid x_t)$. In our case, an observation z_t consists of the SIFT features in the current image: $z_t = \{o_1, \ldots, o_M\}$ where M is the number of features in the current image. To determine the likelihood of an observation given a pose in the map, we compare the observed features with the features in the map by computing the Euclidean distance of their PCA-SIFT descriptor vectors.

In order to avoid comparing the features in the current image to the whole set of features contained in the map, we determine the potentially visible features. This helps to cope with an environment that contains similar landmarks at different locations (e.g. several similar rooms). In case one matches the current observation against the whole set of features, this leads to serious errors in the data association.

To compute the relevant features, we group close-by particles to a cluster. We determine for each particle cluster the set of features that are potentially visible from these locations. This is done using ray-casting on the feature grid map. To speed-up the process of finding relevant features, one could also precompute for each grid cell the set of features that are visible. However, in our experiments, computing the similarity of the feature vectors took substantially longer than the ray-casting operations. Typically, we have 150-500 features per image.

In order to compare two SIFT vectors, we use a distance function based on the Euclidian distance. The likelihood that the two PCA-SIFT vectors f and f' belong to the same feature is computed as

$$p(f = f') = exp\Big(- \frac{\|f - f'\|}{2 \cdot \sigma_1^2} \Big), \tag{3}$$

where σ_1 is the variance of the Gaussian.

In general, one could use Eq. (3) to determine the most likely correspondence between an observed feature and the map features. However, since it is possible that different landmarks exist that have a similar descriptor vector, we do not determine a unique corresponding map feature for each observed feature. In order to avoid misassignments, we instead consider all pairs of observed features and relevant map features. This set of pairs of features is denoted as C. For each pair of features in C we use Eq. (3) to compute the likelihood that the corresponding PCA-SIFT vectors belong to the same feature.

This information is than used to compute the likelihood $p(z_t \mid x_t^{(i)})$ of an observation given the pose $x_t^{(i)}$ of particle i, which is required for MCL. Since a single perspective camera does not provide depth information, we can use only the angular information to compute this likelihood. We therefore consider the difference between the horizontal angles of the currently observed features in the image and the features in the map to compute $p(z_t \mid x_t^{(i)})$. More specifically, we compute the distribution over the angular displacement of a particle given the observation and the map. For each particle, we compute a histogram over the angular differences between the observed features and the map features. The x-values in that histogram represent the angular displacement and the y-values its likelihood. The histogram is computed using the pairs of features in C evaluated using Eq. (3).

In particular, we compute for each pair $(o, l) \in C$ the difference between the horizontal angle at which the feature was observed and the angle at which the feature should be located according to the map and the particle pose. We add the likelihood that these features are equal, which is given by Eq. (3), to the corresponding bin of the histogram. As a result, we obtain a distribution about the angular error of the particle.

In mathematical terms, the value $h(b)$ of a bin b (representing the interval of angular differences from $\alpha^-(b)$ to $\alpha^+(b)$) in the histogram is given by

$$h(b) = \beta + \sum_{\left\{(o,l)\in C \,\middle|\, \alpha^-(b) \leq \alpha(o) - \alpha(l) < \alpha^+(b)\right\}} p(f_o = f_l), \qquad (4)$$

where $\alpha(\cdot)$ is the function that computes the horizontal angle of a feature for a given pose of the robot, f_o is the PCA-SIFT descriptor of feature o, and f_l of feature l accordingly. β is a constant greater that zero ensuring that no angular displacement has zero probability.

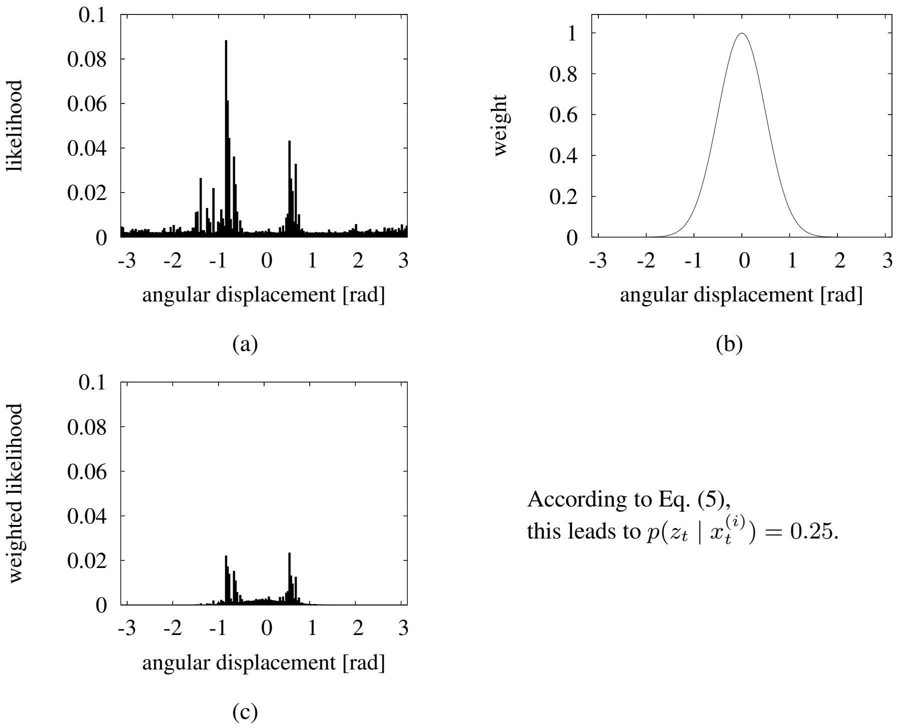
The histograms of particles that are close to the correct pose of the robot have high values around zero. In case that there are several similar features in the environment, the histogram has multiple modes.

One finally needs to compute the observation likelihood of a particle. So far, we computed the distribution about the horizontal angular displacement, not its actual value. In case of a uni-modal or Gaussian distribution it would be sufficient to consider only the distance of the mean from zero taking into account the variance. However, in real-world situations, it is likely that one obtains multi-modal distributions.

Each bin of that histogram stores the probability mass of the corresponding angular displacement of the particle. Therefore, we compute the observation likelihood given we have the angular displacement of that bin and multiply it with the value stored in that bin. The observation likelihood given the histogram is then computed by the sum over these values

$$p(z_t \mid x_t^{(i)}) = \sum_b h(b) \cdot exp\Big(- \frac{1}{2 \cdot \sigma_2^2} \cdot \Big[\frac{\alpha^+(b) + \alpha^-(b)}{2} \Big]^2 \Big), \tag{5}$$

where σ_2 is the variance of a Gaussian describing the likelihood of a particle depending on the angular displacement. Figure 1 illustrates the whole process of computing the observation likelihood for a single particle.



(a)

(b)

(c)

According to Eq. (5),
this leads to $p(z_t \mid x_t^{(i)}) = 0.25$.

Fig. 1. Image (a) shows the distribution about the horizontal angular displacement for a particular particle computed according to Eq. (4). The plot shown in (b) depicts the Gaussian that is used to compute the weight of a sample depending on the displacement. Finally, image (c) shows the resulting histogram in which each bin of the histogram (a) is multiplied by the corresponding value of the Gaussian. Summing up the bins leads to an observation likelihood of 0.25.

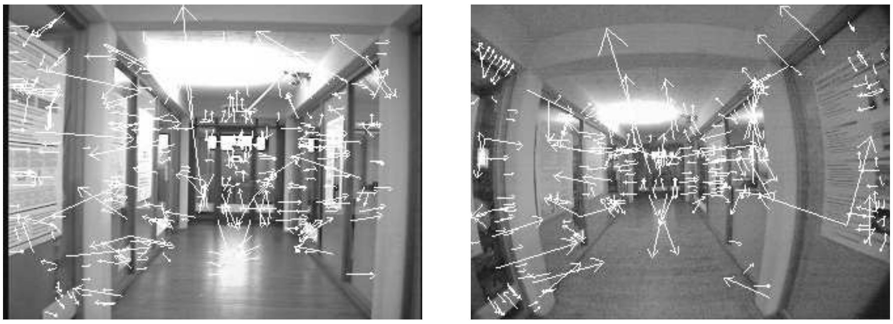


Fig. 2. Example images with generated SIFT features. The images were obtained from two different cameras used in the experiments. The standard camera (left) was used for map acquisition as well as for localization and a low-cost wide-angle camera (right) for further evaluation of our localization approach.

Note that a further improvement of the sensor model can be obtained by using the joint compatibility test between pairs of feature as proposed by Neira and Tardós [18] and not considering all possible data associations.

6 Experimental Results

To evaluate our approach to estimate the pose of the robot equipped with a single perspective camera, we carried out a series of real-world experiments with wheeled and humanoid robots in an office environment. The B21r robot that performed the mapping task carries a standard camera with an opening angle of approximately $65°$. In order to show that the acquired feature map can be used by robots equipped with different cameras, we performed the localization experiments using a low-cost wide-angle camera (with an opening angle of about $130°$). The difference between typical images of both cameras can be seen in Figure 2. The arrows indicate the location, orientation, and scale of the generated SIFT features. The acquired map is depicted in Figure 3.

6.1 Localization Accuracy

In this experiment, the wheeled robot traveled a distance of approximately $20m$. Figure 3 shows the estimated trajectory as well as the true pose of the robot during this experiment. The ground truth has been determined using laser range data. The evolution of the particle filter is illustrated in Figure 4. It shows the particle clouds as well as the true position and the pose estimate provided by odometry.

A more quantitative analysis showing the localization error over time can be found in Figure 5. Between time step 40 and 50, the error in the pose of the vehicle was comparably high. This is because we used the weighted mean of the samples

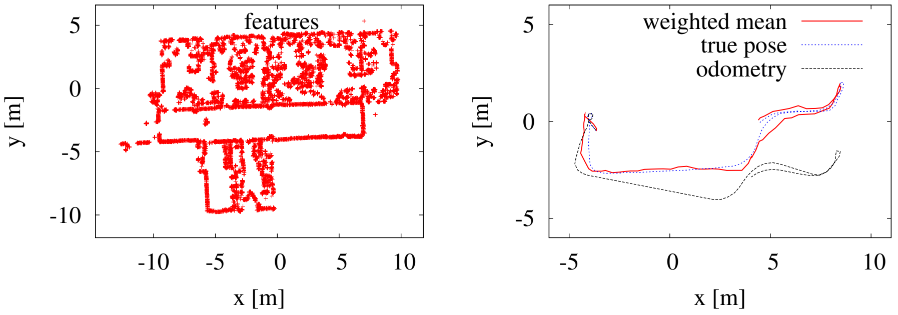


Fig. 3. The left image shows the 2D map acquired in a typical office environment. Each cross represents the estimated 2D position of a SIFT feature. The right image depicts the estimated trajectory as well as the ground truth of a localization experiment. As can be seen, the weighted mean of the particles is close to the true pose of the robot.

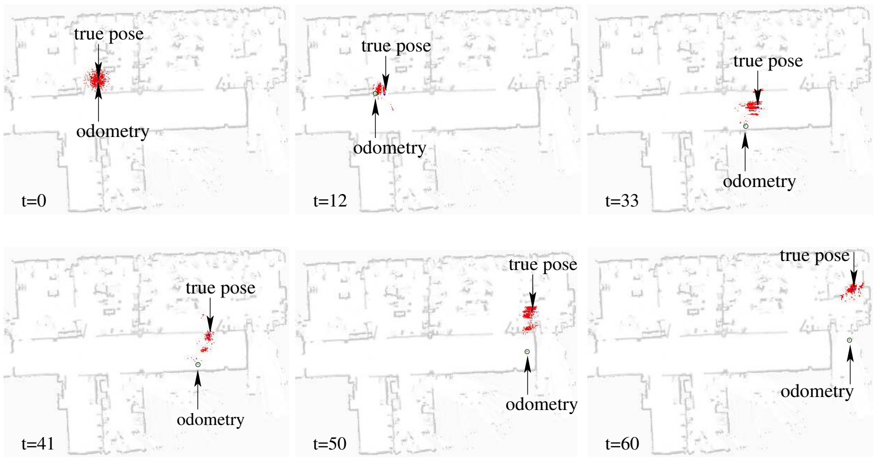


Fig. 4. The particle set during localization. The two arrows indicate the pose resulting from odometry information as well as the true pose of the robot. The true pose of the vehicle was determined by using a laser range finder that was mounted on the robot for this purpose. The occupancy grid map is only shown for a better illustration and was not used for localization.

for the error computation and because the belief was temporarily multi-modal. This fact can be observed in the snapshots depicted in Figure 4. As this experiment illustrates, our technique is able to accurately estimate the pose of the robot. The average error in the x/y-position was $39cm$. The average error in the orientation of the vehicle was $4.5°$. We got comparable localization results when using different cameras with a more constrained field of view like the one which was used for map acquisition. During our experiments, we used 800 particles in our particle filter, which were initialized with a Gaussian centered at the starting pose of the robot.

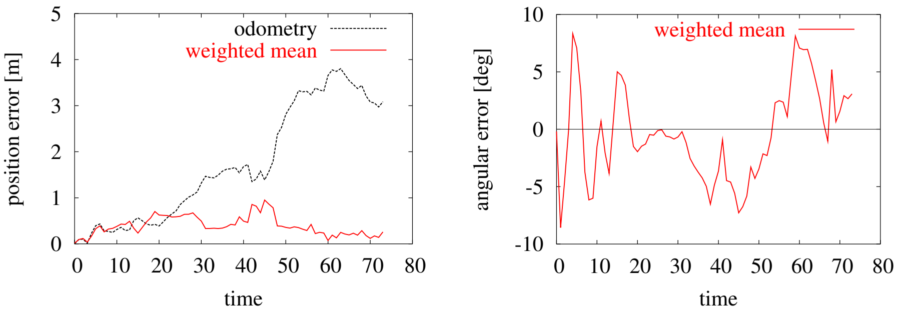


Fig. 5. Evolution of the error during the localization experiment depicted in Figure 3.

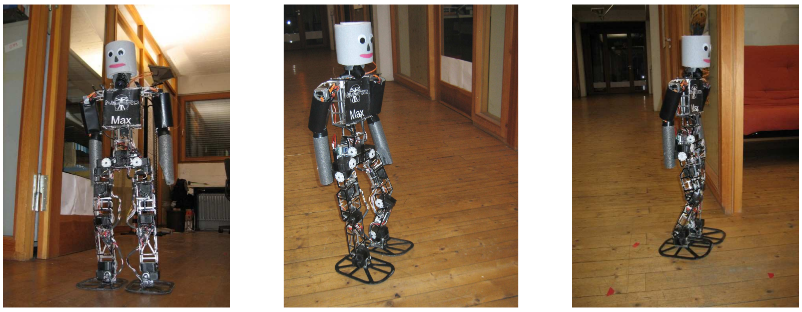


Fig. 6. The humanoid robot Max.

6.2 Tracking the Pose of a Humanoid Robot

To further evaluate our approach, we applied our localization technique to the humanoid robot depicted in Figure 6. To estimate the pose of the robot based on executed motion commands, we perform dead reckoning. The control input consists of the gait target vector that controls the lateral, sagittal, and the rotational speed of omnidirectional walking. The estimated velocities are integrated to determine the relative movement. Compared to a wheeled robot equipped with odometry sensors, this leads to a noisy pose estimate. Furthermore, due to the design of the humanoid robot, the camera images are often blurred because of vibrations.

In this experiment, the robot Max traveled along the trajectory shown in Figure 7. The red circles correspond to position where an observation was made. The particle clouds obtained in this experiment are given in Figure 8. In case no sensor information is integrated, the pose estimate has a high uncertainty as can be seen in the first row of that figure. In contrast to this, the use of our vision-based localization technique reduces the uncertainty and enables to localize the humanoid. Note that due to unstable motion of the humanoid, missing odometry sensors, vibrations, and the shaking camera, the localization is less robust compared to a wheeled robot.

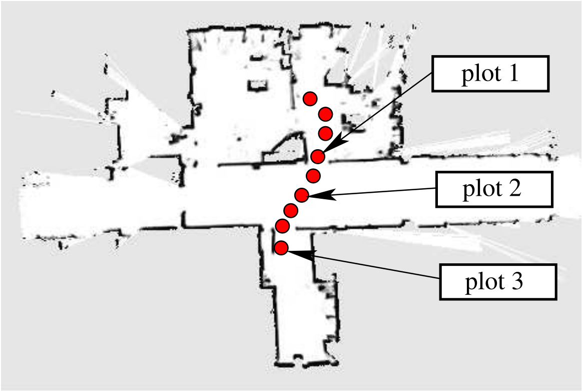


Fig. 7. The trajectory of Max. The red circles indicate the positions where observations were made. The corresponding plots of the particle clouds are shown in Figure 8.

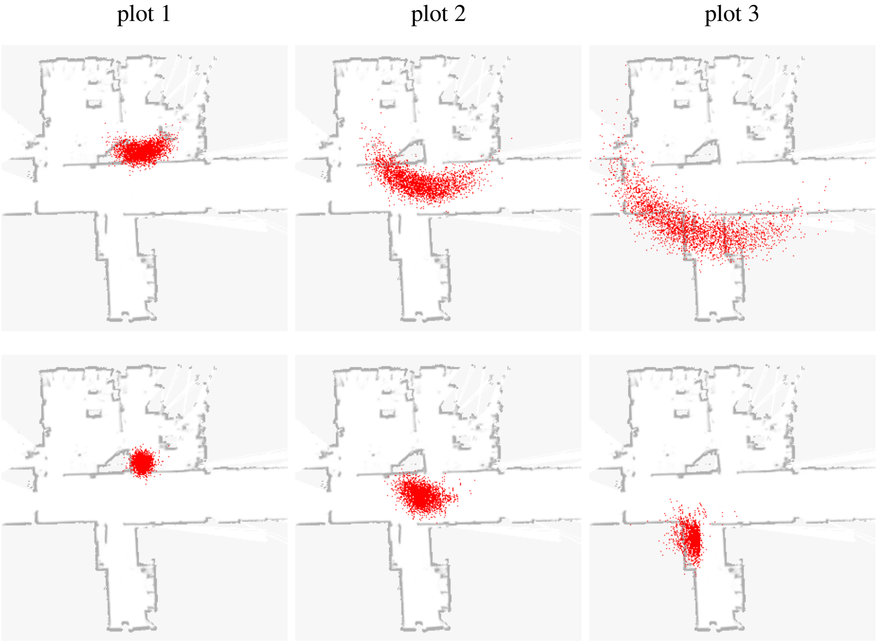


Fig. 8. Vision-based localization of a humanoid robot. The images in the first row depict the evolution of the particles in case no sensor information is used. The high uncertainty in the particle cloud results from the poor motion estimate resulting from dead reckoning. The images in the second row show the result of our localization approach. As can be seen, the visual information allows to accurately estimate the pose of the humanoid robot.

7 Conclusions

In this paper, we presented an approach to mobile robot localization that relies on a single perspective camera. Our technique is based on Monte-Carlo localization and uses SIFT features extracted from camera images. In the observation model of our particle filter, we compare descriptor vectors of features in the current image to the set of potentially visible map features given the pose of the particles. Based on this information, we compute a distribution about the angular displacement for each sample given the current observation. The evaluation of potential correspondences between features is done efficiently by performing the necessary computations for clusters of particles. By using only the relevant features in the vicinity of the particles in the observation model, we reduce the number of data association failures. As we demonstrate in real-world experiments carried out with a wheeled as well as with a humanoid robot, our system provides an accurate metric pose estimate for a mobile robot without requiring proximity sensors, omnivision, or a stereo camera.

Acknowledgment

This project is partially supported by the German Research Foundation (DFG), grant BE 2556/2-1 and SFB/TR-8 (A3). We would like to thank D. Lowe for providing his software to detect SIFT features and Y. Ke for his PCA-SIFT implementation. Further thanks to J. Stückler and M. Schreiber for helping us carrying out the experiments with the humanoid robot.

References

1. H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
2. T.D. Barfoot. Online visual motion estimation using FastSLAM with SIFT features. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.
3. A.J. Davison, Y. González Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *IFAC/EURON Symposium on Intelligent Autonomous Vehicles (IAV)*, 2004.
4. F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the Condensation algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1999.
5. F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1998.
6. A Doucet. On sequential simulation-based methods for bayesian filtering. Technical report, Signal Processing Group, Departement of Engeneering, University of Cambridge, 1998.
7. P. Elinas and J.J. Little. σMCL: Monte-Carlo localization for mobile robots with stereo vision. In *Proc. of Robotics: Science and Systems (RSS)*, 2005.

8. G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
9. H.-M. Gross, A. Köning, C. Schröter, and H.-J. Böhme. Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.
10. Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.
11. J. Kǒsécka and L. Li. Vision based topological Markov localization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2004.
12. L. Ledwich and S. Williams. Reduced SIFT features for image retrieval and indoor localization. In *Australian Conf. on Robotics and Automation (ACRA)*, 2004.
13. T. Lemaire, S. Lacroix, and J. Solà. A practical 3D bearing-only SLAM algorithm. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.
14. J.S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statist. Comput.*, 6:113–119, 1996.
15. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 1999.
16. E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro. Image-based Monte-Carlo localisation with omnidirectional images. *Robotics & Autonomous Systems*, 48(1):17–30, 2004.
17. K. Mikolajczk and C. Schmid. A performance evaluation of local descriptors. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.
18. J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.
19. T. Röfer and M. Jüngel. Vision-based fast and reactive Monte-Carlo Localization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.
20. S. Se, D.G. Lowe, and J.J. Little. Global localization using distinctive visual features. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.
21. R. Sim, P. Elinas, M. Griffin, and J.J. Little. Vision-based SLAM using the Rao-Blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, 2005.
22. M. Sridharan, G. Kuhlmann, and P. Stone. Practical vision-based Monte Carlo localization on a legged robot. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
23. J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization by combining an image retrieval system with Monte Carlo Localization. *IEEE Transactions on Robotics and Automation*, 21(2):208–216, 2005.