A Comparison of Efficient Global Image Features for Localizing Small Mobile Robots

Marius Hofmeister, Philipp Vorst and Andreas Zell
Computer Science Department, University of Tübingen, Tübingen, Germany

Abstract

Global image features are well-suited for the visual self-localization of mobile robots. They are fast to compute, to compare and do not require much storage space. Especially when using small mobile robots with limited processing capabilities and low-resolution cameras, global features can be preferred to local features. In this paper, we compare the accuracy and computation times of different global image features when localizing small mobile robots. We test the methods under realistic conditions, taking illumination changes and translations into account. By employing a particle filter and reducing the image resolution, we speed up the localization process considerably.

1 Introduction

Self-localization is a key ability of mobile robots. It is often done visually, since cameras are inexpensive and flexible sensors. In this paper, we compare different efficient global image features for the visual self-localization of small mobile robots. We here use a *c't-Bot* developed by the German computer magazine *c't* [1]. This size class of robots has been shown to be useful in many cases: The well-known *Khepera* robot has been widely used in a variety of tasks, e.g., [2, 12]. The *e-puck* robot serves for educational purposes [7]; and also in swarm robotics, small mobile robots nowadays play a major role [8, 13].

Yet, the use of small mobile robots is challenging due to their limited processing power and restricted sensing capabilities. If cameras are used, only low image resolutions can be processed. Thus, the focus of this paper lies in the investigation of how small mobile robots can be enabled to localize themselves visually indoors in an efficient manner.

Vision-based positioning is often performed using image retrieval techniques, which store images in a database. For localization, a new image is taken and compared to all or a subset of previously recorded images. The computed similarity then leads to an estimate of the robot's position. Mostly, this task is performed by extracting features from the images. Such features often promise to be robust to changes in the environment and the viewpoint of the observer.

The existing approaches to visual self-localization often differ in the type of features they extract from images. Local features, like the *Scale-Invariant Feature Transform (SIFT)* [11] or the *Speeded-up Robust Features (SURF)* [4], describe patches around interest points in an image, while global features describe the whole image as one single fixed-length vector. This implies the advantages and drawbacks of the two approaches: As the number of local features in an image can be large, it may take a long time to find, match and store these features. Many kinds of local features are invariant to scale and rotation, which global features can hardly provide. By contrast, global image features are fast to compute and have also shown good localization accuracy [6, 18, 21]. Because of the limited processing capabilities of small mobile robots, we decided to use global image features in this work.

Our paper is organized as follows: In the next section, we present related research. In Sect. 3, we introduce the employed robot. Then, we explain the compared image features in Sect. 4. The localization process, which is composed of image matching and particle filtering, is described in Sect. 5. In Sect. 6, we present the setup of our experiments and the corresponding results. Section 7 concludes the paper and recapitulates our contribution.

2 Related Work

In the following, we present approaches to visual self-localization relying on image-retrieval techniques. Ulrich and Nourbakhsh [17] established place recognition using color histograms. They applied a nearest-neighbor algorithm to all color bands and combined it with a simple voting scheme based on a topological map of the environment. Zhou et al. [21] extended this approach to multidimensional histograms, taking features such as edges and texturedness into account. Wolf et al. [20] performed visual localization by combining an image retrieval system with a particle filter. They used local image features which are invariant to image rotations and limited scale [14]. These features are also the basis for the global *Weighted Grid Integral Invariants*, which are employed in this paper.

In former work, we investigated the self-localization with tiny images on two different platforms: a small wheeled robot and a flying quadrocopter [9]. This approach to reduce the image resolution was inspired by Torralba et al. [16], who built on psychophysical results showing a remarkable tolerance of the human visual system to degradation in image resolution. They stored millions of images in a size of 32×32 pixels and performed object and scene recognition on this dataset. Self-localization with small images was earlier performed by Argamon-Engelson [3]. He used images with a resolution of 64×48 pixels and applied measurement functions based on edges, gradients, and texturedness. As described in his paper, he simplified the localization process by recognizing topological places only.

The computation time of our localization process can roughly be divided into the extraction of the image features and the feature comparison. To speed up the feature extraction, we reduce the image size and investigate to what extent this reduction influences the localization accuracy. To speed up the feature matching, different approaches have been proposed. In [5], Beis and Lowe presented the best-bin-first algorithm as a variant of the kd-tree for efficient search in high-dimensional spaces. Jegou et al. improved large scale image searches based on weak geometric consistency constraints in the Hamming space [10]. In this work, we decided to use a particle filter to limit the number of feature comparisons as in [18], since this approach includes a number of other advantages: Particle filters compute a probabilistic estimate of the robot's position by including specific models for perception and motion. In this way, arbitrary probability distributions can be handled that make the estimation robust, since they allow recovering from possible localization failures. Furthermore, particle filters are easy to implement and to adapt for small mobile robots.

3 Hardware Components

We use one of 13 *c't-Bots* (see **Fig. 1**) of our lab to perform the experiments. These robots have a diameter of 12 cm, are 19 cm high and were developed by the German computer magazine *c't* [1]. They are equipped with an ATmega644 microprocessor with 64 KB flash program memory, a clock frequency of 16 MHz and 4 KB SRAM. The most capable sensor is a *POB-Eye* color camera that includes an image processing module. This module permits to perform all image processing directly on it and to send the extracted image features to the robot via I^2C. The camera provides a resolution of 120×88 pixels and possesses an ARM7TDMI processor at 60 MHz with 64 KB RAM. Furthermore, the robot has a WLAN interface to send data to a PC. For localization, we additionally use a Devantech CMPS03 compass with a specified accuracy of 3-4° (sic), and a low-cost SD card to store the image features. Our algorithm runs on the robot itself. The WLAN interface is only used for debugging and monitoring purposes. Since the *c't-Bot* is part of a swarm of 13 identical such robots, having all robots sent their images via WLAN to an external server is not desirable.

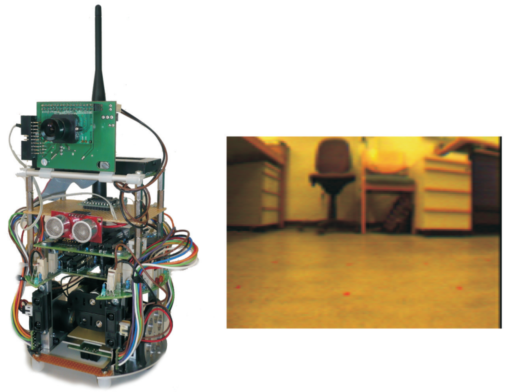


Figure 1: c't-Bot and example image taken by the onboard camera.

4 Efficient Global Image Features

The selection of image features results from the computational limitations of small mobile robots. Color and greyscale histograms are simple and fast methods for computing the feature vectors. More complex methods are *Weighted Gradient Orientation Histograms (WGOH)* and *Weighted Grid Integral Invariants (WGII)*. They yielded good results in earlier research, especially under illumination changes [18, 19]. Additionally, we employ the pixelwise comparison of images.

All selected features, except the pixelwise image comparison, are based on a grid which divides the image into a number of subimages. This makes the features more distinctive through adding local information. Changes within one subimage only influence a small part of the feature vector. We tested the methods at different grid sizes and experimentally found out that a 4×4 grid is a good tradeoff between efficiency and retrieval accuracy, even when reducing the image resolution, as mentioned in Sect. 4.4.

4.1 Color/Greyscale Grid Histograms

For the color and greyscale histograms, we use eight bins for each subimage. Through concatenation we get a 1×128 feature vector of the 16 subimages. In case of the color histogram we use the hue value of the HSV color space. This choice of space promises to be robust to illumination changes.

4.2 Weighted Gradient Orientation Histograms

Weighted Gradient Orientation Histograms (WGOH) were proposed by Bradley et al. [6] and were originally intended for outdoor environments because of their robustness to illumination changes. They were inspired by SIFT features [11].

Bradley et al. first split the image into a 4×4 grid of subimages. On each subimage, they calculate an 8-bin histogram of gradient orientations, weighted by the magnitude of the gradient at each point and by the distance to the center of the subimage. In our implementation of WGOH, we use a 2D Gaussian for weighting, where the mean corresponds to the center of the subimage and the standard deviations correspond to half the width and the height of the subimage, respectively [19]. This choice is similar to SIFT, where a Gaussian with half the width of the descriptor window is used for weighting. The 16 histograms are concatenated to a 1×128 feature vector, which is normalized subsequently. To reduce the dependency on particular regions or some strong gradients, the elements of the feature vector are limited to 0.2, and the feature vector is normalized again.

4.3 Weighted Grid Integral Invariants

The key idea of integral invariants is to design features which are invariant to Euclidean motion, i.e., rotation and translation [14, 20]. Therefore, all possible rotations and translations are applied to the image. In our case, two relative kernel functions are applied to each pixel. These functions compute the difference between the intensities of two pixels p_1 and p_2 lying on different radii and phases around the center pixel. The described procedure is repeated several times, where p_1 and p_2 are rotated around the center up to a full rotation while the phase shift is preserved. By averaging over the resulting differences we get one value for each pixel and kernel. We experimentally found out that the following radii for p_1 and p_2 lead to the best results: radii 2 and 3 for kernel one and radii 5 and 10 for kernel two, each with a phase shift of 90°. One rotation is performed in ten 36° steps.

Weiss et al. [18] extended the basic algorithm by dividing the image into a set of subimages to add local information. Each pixel is then weighted by a Gaussian as with WGOH to make the vector more robust to translations. The result is a 2×8 histogram for each subimage and a 1×256 histogram for the whole image.

4.4 Downscaled Images and Pixelwise Image Comparison

The resolution of an image has a large effect on the computation time of its feature vector. We downscale the images, preserving their aspect ratio, to a tiny resolution of 15×11 pixels by interpolating the pixel intensities. This allows also for comparing the image data in a pixelwise fashion rather than extracting first the features. Therefore, the image data are treated as a vector. To keep the amount of data small, we only compare the normalized greyscale image and discard color information.

5 Localization Process

5.1 Overview

Our localization process consists of two steps, the mapping phase and the retrieval phase. In the mapping phase, *training images* are recorded and feature vectors are extracted. These vectors are saved together with their current global position coordinates, which are manually measured. In the retrieval phase, *test images* are recorded and features are again extracted. These features are subsequently compared to all other previously saved feature vectors as it is described in Sect. 5.2. By using a particle filter, the test images are only compared to a subset of previously saved features. This procedure is presented in Sect. 5.3.

5.2 Image Comparison

We calculate the similarity $sim(Q, D)$ of two images Q and D from their corresponding normalized feature histograms q and d through the *normalized histogram intersection* $\bigcap_{norm}(q, d)$:

$$sim(Q, D) = \bigcap_{norm}(q, d) = \sum_{k=0}^{m-1} \min(q_k, d_k). \tag{1}$$

Here, m is the number of histogram bins and q_k denotes bin k of histogram q. The advantage of normalized histogram intersection is its short computation time as compared to other similarity measures such as cosine similarity or dissimilarity measures such as Jeffrey divergence.

For the pixelwise image comparison, the normalized histogram intersection did not yield satisfactory results. In this case, we use the L_1-norm with the normalized images Q^* and D^*:

$$L_1(Q^*, D^*) = \sum_{k=0}^{r-1} |Q^*_k - D^*_k|, \tag{2}$$

where r is the number of pixels and Q^*_k denotes pixel k of image Q^*. The similarity $sim(Q, D)$ of two images can now be computed as:

$$sim(Q, D) = 1 - min(1, L_1(Q^*, D^*)). \tag{3}$$

Note that in general $0 \leq L_1(Q^*, D^*) \leq 2$ (although $L_1(Q^*, D^*) > 1$ rarely happens for images). The image with highest similarity is then the best match.

5.3 Combination with a Particle Filter

If the number of training images is large, the matching step can be time-consuming. Especially with small robots, it may take a long time to compare the test image against all training images. To limit the number of image comparisons and speed up this process, we use a particle filter for self-localization [15].

Particle filters approximate the belief $Bel(\mathbf{x}_t)$ of the robot about its position $\mathbf{x}_t$ by a set of m particles. Each particle consists of a position (x, y) together with a nonnegative weight, its *importance factor*. The estimated position of the robot is given by the weighted mean of all particles. The inital belief is represented by particles which are randomly distributed over the robot's global coordinate system. All importance factors are set to $\frac{1}{m}$. The particles are updated for each test image iteratively, according to the following three steps:

1. *Resampling:* After the first iteration, m random particles $\mathbf{x}_{t-1}^{(i)}$ are drawn from $Bel(\mathbf{x}_{t-1})$ according to the importance factors $w_{t-1}^{(i)}$ at time $t-1$. This step is only performed if the estimate $\widetilde{n}_{eff} = 1/(\sum_{i=1}^{n}(w_t^{(i)})^2)$ of the effective sample size falls below the threshold $m/2$.

2. *Prediction:* The sample $\mathbf{x}_{t-1}^{(i)}$ is updated to sample $\mathbf{x}_t^{(i)}$ according to an action u_{t-1}. In our case, we update the particles according to odometry and compass measurements. The odometry values measure the translation δ between two image recordings. The direction θ of the straight line motion is determined by the compass.

   For each particle, Gaussian noise is added to δ with zero mean and standard deviation $\sigma_{trans} = \delta \cdot 0.1$. Additionally, Gaussian noise is added to θ with zero mean and $\sigma_{rot} = 21°$. We assigned a relatively high value to σ_{rot} due to the imprecise rotations of our robot, the limited compass accuracy and the magnetic deflections that appear indoors. Finally, each particle is moved according to δ and θ.

3. *Correction:* The sample $\mathbf{x}_t^{(i)}$ is weighted by the observation model $p(y_t|\mathbf{x}_t^{(i)})$, i.e., the likelihood of the measurement y_t, given the sample $\mathbf{x}_t^{(i)}$. In our method, we first search the nearest training image $D(x_t^{(i)})$ to each particle. The current test image Q and the training image $D(x_t^{(i)})$ are then matched by means of one of the abovementioned methods.

   The new weight is then computed as $w_t^{(i)} = w_{t-1}^{(i)} \cdot sim(Q, D(x_t^{(i)}))$. This method is referred to as *standard weighting* in the following.

   An alternative way of updating the particle's weight is $w_t^{(i)} = w_{t-1}^{(i)} \cdot sim(Q, D(x_t^{(i)}))^{\alpha}$, with $\alpha > 1$. This method is referred to as *alternative weighting* in the following. By potentiating the original similarity, the differences between the weights become more distinctive [18]. This makes the particle cloud converge faster and focus on the particles with the largest weights. In our implementation, we set $\alpha = 20$.

After the correction step, we normalize the importance factors and calculate the estimated position.

6 Experimental Results

6.1 Setup

We conducted several experiments in an office environment. Since the robot did not have the ability to determine its ground truth position through GPS or other accurate sensors like laser scanners, we manually grabbed images every 0.5 m in an area of appr. 75 m^2. Our dataset consists of 190 training images, that were grabbed facing west (determined by the compass) with a manually oriented robot. Due to magnetic deflections of, for instance, furniture, the direction indicated by the compass was not always true but repeatable, thus it can be seen as a function of the position.

6.2 Comparison of Image Features

To compare the accuracy of the various image features, we grabbed 200 test images at randomly chosen positions under different conditions. 100 images, in the following called *test data A*, were grabbed at stable illumination. Another 100 images, in the following referred to as *test data B*, were grabbed at different lighting conditions with and without ceiling lights, at shining sun or dull daylight. In both datasets, the robot rotated autonomously towards west by means of the compass. Because of the weak odometry, the robot's rotation is affected by errors which appear approximately as horizontal translations in the images.

Figure 2 shows the localization accuracy of the examined methods at different image resolutions. The localization error is the distance between the actual recording position of the test image and the corresponding best match, that is, the image with the highest similarity. To compare the accuracy of the image features, we calculated the median localization error, since it is less affected by outliers than the mean localization error. We use the latter in combination with the particle filter in the next section.

The smallest median localization errors we obtained are 0.42 m on test data A and 0.50 m on test data B using WGOH. Generally, the results of WGOH and WGII are mostly similar and outperform the common color and grey grid histograms, especially under illumination changes. Having a look at the pixelwise image comparison, we find that in our scenario this straightforward approach lead to surprisingly high accuracies in all cases. Even under the illumination changes of test data B WGOH and WGII,

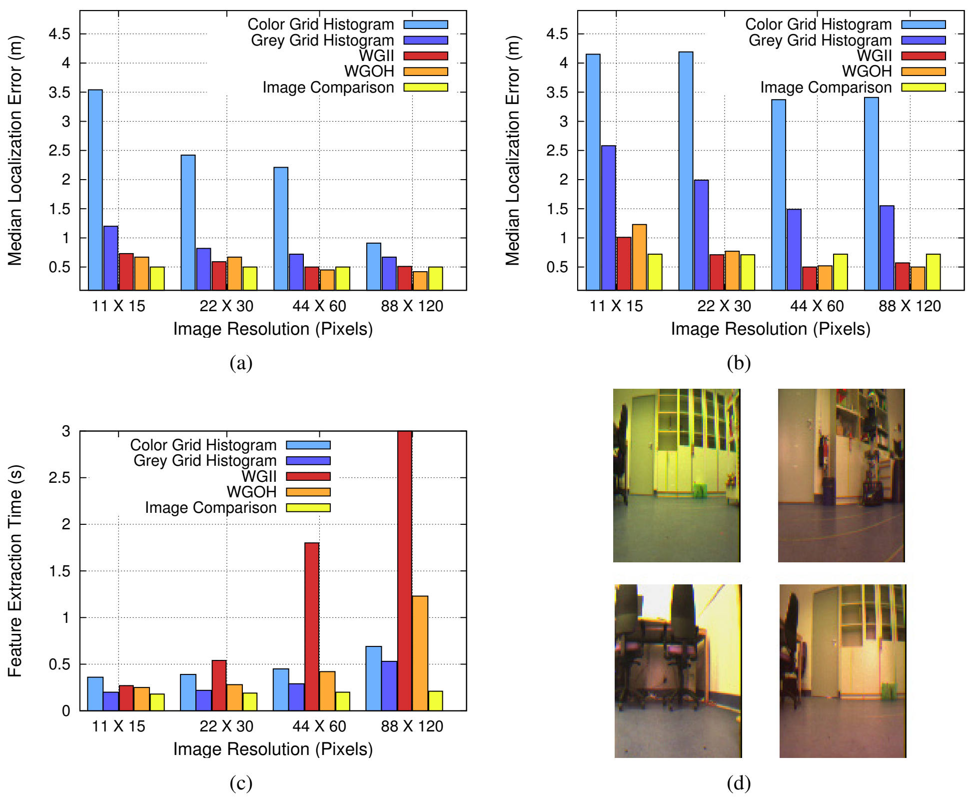


Figure 2: Median localization errors of the image features on test data A and B at different image resolutions are shown in (a),(b), respectively. The pixelwise image comparison is referred to as image comparison. The time which was required to extract the features of the images is shown in (c). The extraction time of WGII at 120×88 pixels is 7.03 s and had to be cut off for better visibility. (d) shows test images at different illumination conditions.

which are expected to be more robust, could hardly outperform the pixelwise image comparison. Only at resolutions of 44×60 and 88×120 pixels, they achieved a smaller localization error (0.22 m at most). At smaller image resolutions, the pixelwise image comparison provided equal or better results. A localization was not possible with the color grid histograms. The reason for this may be the poor color quality of the camera and the lack of meaningful color information in the environment.

Another unexpected result was that the reduction of the image resolution has only little influence on the localization error. By reducing the resolution to up to 22×30 pixels, the localization error is only increasing slowly: 0.14 m in case of WGII and 0.27 m in case of WGOH (round 2). In case of the pixelwise image comparison, no change of localization accuracy was observed at all. Further open research questions are the influence of changing environments and therewith occlusions in the images on the localization accuracy.

When working with small mobile robots, computation times are an important issue besides the accuracy of the localization process. Figure 2 (c) depicts the required time to extract the features. The reduction of the image resolution speeds up the process especially in case of WGOH and WGII. This is because both methods perform more complex computations on each pixel than simple histograms. The values that are denoted for the pixelwise image comparison are composed of the time to grab a frame, to convert it to greyscale and to resize it. These computations are performed in all methods, except the greyscale conversion in case of the color histograms, and are included in the measured computation times of Fig. 2 (c).

Furthermore, the use of a compass attested to be an adequate way for localizing the small robots despite their relatively large rotation error.

6.3 Localization with a Particle Filter

To test the particle filter presented in Sect. 5.3, we steered the robot arbitrarily on two rounds through the environment as depicted in Fig. 3 (f). To grab an image, the robot was rotated facing west by means of the compass and was afterwards rotated back to continue its path. Between the image recordings, the robot was steered straight ahead. Round 1 consists of $n = 97$ images and was conducted at stable illumination conditions. Round 2 consists of $n = 62$ images and was taken at varying illumination. Then, we ran the particle filter on these rounds, processing

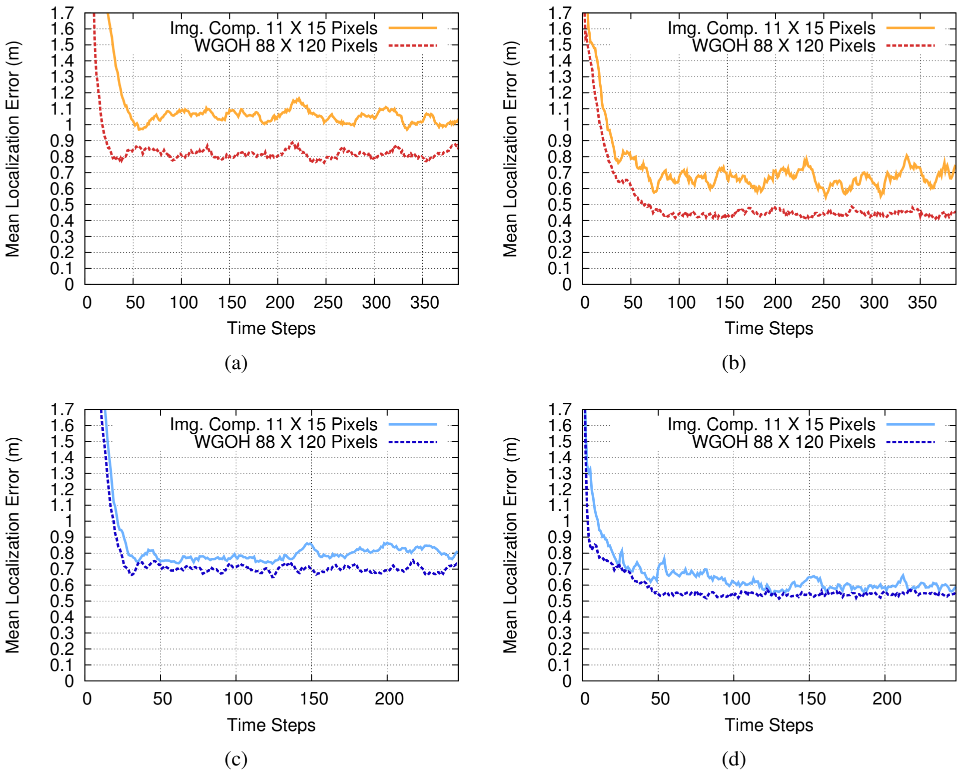


Figure 3: Rounds 1 (a),(b) and 2 (c),(d) and their mean localization errors using a particle filter with 40 particles. In (a),(c) we applied the standard weighting to the particle filter, in (b),(d) we applied the alternative weighting (potentiating the similarity with 20) referring to Sect. 5.3. The pixelwise image comparison is referred to as Img. Comp..

each round four times. To get the mean localization error over time, we conducted this experiment n times; each cycle started at a different test image.

Since our aim in using the particle filter was to keep the matching time short, we compared two different methods for this experiment: the pixelwise image comparison on the 15×11 images and WGOH on the full-size images. We chose the pixelwise image comparison with small-size images, because it was the fastest method according to the feature extraction time while providing good accuracy.

To compare the results, we used WGOH since it revealed the smallest localization error and a feature extraction time that was smaller than WGII. We had to limit the number of particles to $m = 40$ because of the restricted memory of the *c't-Bots*.

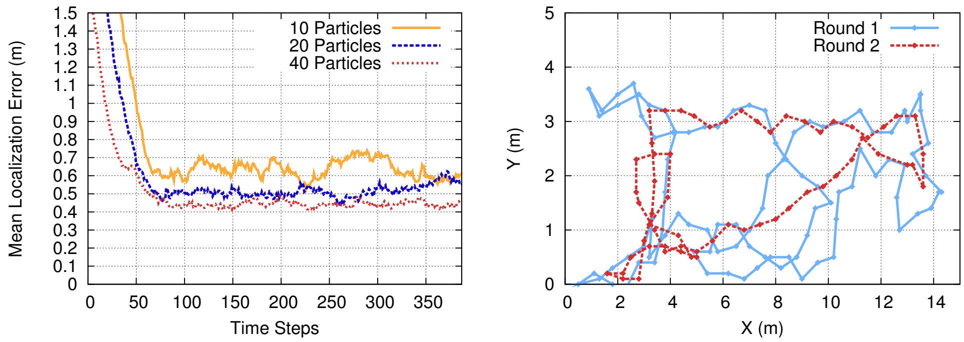
Figure 3 shows the mean localization errors for the two rounds over the cycles with the different particle weighting methods (referring to Sect. 5.3). The alternative weighting (potentiating the similarity with 20) achieved a higher localization accuracy than the standard weighting. By using the pixelwise image comparison at 15×11 pixels, we achieved a reasonable localization accuracy, even if it was always slightly worse than WGOH at full resolution.

The mean localization error ± standard deviation over the 388 (248) images in case of WGOH is 0.52 m±0.27 m (0.60 m±0.23 m) and with the pixelwise image comparison 0.74 m±0.27 m (0.67 m±0.24 m), in rounds 1 and 2 respectively, using the alternative weighting. The overall localization error by using the particle filter is in case of WGOH 0.32 m (0.06 m) smaller and in case of the pixelwise image comparison 0.56 m (0.54 m) smaller than matching all images and using only the best match, in rounds 1 and 2 respectively. **Figure 4** reveals the influence of the number of particles on the mean localization error and depicts the trajectories of the two rounds.

	WGOH 120×88	Img. Comp. 15×11
50 Images	3.76 s	2.89 s
100 Images	4.96 s	4.14 s
190 Images	7.13 s	6.56 s
PF 20	3.87 s	2.63 s
PF 40	4.81 s	3.84 s

Table 1: Computation times of the localization process by matching different numbers of images and by using a particle filter with 190 training images and 20 (PF 20), 40 (PF 40) particles.

The computation times of the matching process with and without the particle filter are shown in **Tab. 1**. While the matching step of one test image to all 190 training images needs 6.56 s in case of the pixelwise image comparison at



(a) (b)

Figure 4: (a) reveals the influence of the number of particles on the mean localization error in round 1, using WGOH at a resolution of 88×120 pixels with the alternative weighting method. (b) depicts the true trajectories of the two rounds.

15×11 pixels, it can be speeded up to 3.84 s by using the particle filter. This is still quite slow, but we also have to keep in mind the limitations of small mobile robots. Further speedup can be achieved by reducing the number of particles.

7 Conclusion

In this paper, we compared different global image features for localizing small mobile robots with limited computation and sensing capabilities. We investigated the algorithms with respect to localization accuracy and computation time at different image resolutions. Best results could be achieved employing WGOH, but even the simplest method, the pixelwise image comparison, lead to reasonable results at shortest computation times. This method became feasible by reducing the image resolution.

In our medium-sized indoor test bed, the image resolution had only little influence on localization accuracy. Tiny greyscale images of 15×11 pixels contained enough information to provide an accurate self-localization and helped saving computation time.

Additionally, a particle filter attested to be a good extension in our scenario. It enhanced localization accuracy and reduced computation time. By varying the number of particles, the approach could easily be adapted to robots that are further miniaturized.

Acknowledgment

The first author would like to acknowledge the contribution by Maria Liebsch in the scope of her student project and the financial support by the Friedrich-Ebert-Stiftung (FES) of his Ph.D. scholarship at the University of Tübingen.

References

[1] c't-Bot project homepage. URL: http://www.ct-bot.de, January 2010.

[2] G. Antonelli, F. Arrichiello, S. Chakraborti, and S. Chiaverini. Experiences of formation control of multi-robot systems with the null-space-based behavioral control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1068–1073, Rome, Italy, April 2007.

[3] S. Argamon-Engelson. Using image signatures for place recognition. *Pattern Recognition Letters*, 19(10):941–951, August 1998.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, June 2008.

[5] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in highdimensional spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1000–1006, San Juan, Puerto Rico, June 1997.

[6] D. M. Bradley, R. Patel, N. Vandapel, and S. M. Thayer. Real-time image-based topological localization in large outdoor environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3670–3677, Edmonton, Canada, August 2005.

[7] C. Cianci, X. Raemy, J. Pugh, and A. Martinoli. Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics. In *Proceedings of Simulation of Adaptive Behavior (SAB), Swarm Robotics Workshop*, pages 103–115, 2007.

[8] N. Correll and A. Martinoli. Towards multi-robot inspection of industrial machinery - from distributed coverage algorithms to experiments with miniature robotic swarms. *IEEE Robotics and Automation Magazine*, 16(1):103–112, March 2008.

[9] M. Hofmeister, S. Erhard, and A. Zell. Visual self-localization with tiny images. In *Proceedings of the 21. Fachgespräch Autonome Mobile Systeme (AMS)*, pages 177–184, Karlsruhe, Germany, December 2009.

[10] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision (ECCV)*, pages 304–317, Marseille, France, October 2008.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60, pages 91–110, November 2004.

[12] A. Martinoli and F. Mondada. Collective and cooperative group behaviours: Biologically inspired experiments in robotics. In *Proceedings of the 4th International Symposium on Experimental Robotics (ISER)*, pages 3–10, Stanford, USA, June 1995.

[13] F. Mondada, A. Guignard, M. Bonani, D. Bar, M. Lauria, and D. Floreano. SWARM-BOT: From concept to implementation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1626–1631, 2003.

[14] S. Siggelkow. *Feature Histograms for Content-Based Image Retrieval*. PhD thesis, Institute for Computer Science, University of Freiburg, Germany, December 2002.

[15] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. In *Artificial Intelligence*, volume 1-2, pages 99–141, 2000.

[16] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 30, pages 1958–1970, November 2008.

[17] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1023–1029, San Francisco, USA, April 2000.

[18] C. Weiss, A. Masselli, H. Tamimi, and A. Zell. Fast outdoor robot localization using integral invariants. In *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS)*, Bielefeld, Germany, March 2007.

[19] C. Weiss, A. Masselli, and A. Zell. Fast vision-based localization for outdoor robots using a combination of global image features. In *Proceedings of the 6th Symposium on Intelligent Autonomous Vehicles (IAV)*, Toulouse, France, September 2007.

[20] 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*, 21(2):208–216, April 2005.

[21] C. Zhou, Y. Wei, and T. Tan. Mobile robot self-localization based on global visual appearance features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1276, Taipei, Taiwan, September 2003.