A Hybrid Approach for Vision-based Outdoor Robot Localization Using Global and Local Image Features

Christian Weiss, Hashem Tamimi, Andreas Masselli and Andreas Zell

***Abstract*— Vision-based robot localization in outdoor environments is difficult because of changing illumination conditions. Another problem is the rough and cluttered environment which makes it hard to use visual features that are not rotation invariant. A popular method that is rotation invariant and relatively robust to changing illumination is the Scale Invariant Feature Transform (SIFT). However, due to the computationally intensive feature extraction and image matching, localization using SIFT is slow. On the other hand, techniques which use global image features are in general less robust and exact than SIFT, but are often much faster due to fast image matching. In this paper, we present a hybrid localization approach that switches between local and global image features. For most images, the hybrid approach uses fast global features. Only in difficult situations, e.g. containing strong illumination changes, the hybrid approach switches to local features. To decide which features to use for an image, we analyze the particle cloud of the particle filter that we use for position estimation. Experiments on outdoor images taken under varying illumination conditions show that the position estimates of the hybrid approach are about as exact as the estimates of SIFT alone. However, the average localization time using the hybrid approach is more than 3.5 times faster than using SIFT.**

I. INTRODUCTION

Vision-based localization in outdoor environments is a difficult task. In contrast to indoor environments, where illumination is often constant due to artificial lights, illumination in outdoor areas is highly dependent on the weather, season or time of day. Additionally, outdoor environments often are rough and cluttered which makes it hard to use visual features that are not rotation invariant. The most popular approach that is rotation invariant and relatively robust to illumination changes is the *Scale Invariant Feature Transform* (SIFT) developed by Lowe [1]. SIFT computes descriptors for local interest points in the image. These interest points are more dependent on structure than on illumination. However, as the number of interest points per image is usually large (about 420 for our 320×240 pixel images on average), extracting the features and matching two images is very slow. Approaches that use SIFT for indoor localization are for example [2], [3]. Outdoor localization using SIFT was presented, for example, in [4].

Another group of vision-based localization algorithms uses global image features. Usually, a single global feature vector represents an image. Well-known methods for robot localization are based on PCA [5]–[7] or on *Integral Invariants* [8]. Bradley *et al.* presented an approach that uses weighted gradient orientation histograms (WGOHs) [9]. In general, global features are more sensitive to illumination changes than local features. On the other hand, the global methods are faster, because matching two images by comparing two vectors is very efficient. An overview over local and global techniques for vision-based robot localization can be found in [10].

Artač *et al.* suggested a method that uses both global and local image features [11]. In the first step, they use global features to fastly select a small set of training images that are similar to the test image. In the second step, they calculate the position estimate by computing the local features (SIFT) only for the selected images and matching only these images to the test image. Another possibility to speed up a local approach like SIFT is to use a particle filter that can adapt the number of particles and with it the number of image matches. In difficult situations, the particle filter increases the number of particles, and in easy situations, it reduces the number of particles. A common method to determine the number of particles is KLD-sampling [12]. Heinemann *et al.* project the current sensor data onto the map, based on the position estimate [13]. They choose the number of particles based on the distance between the map data and the projected sensor data. However, the disadvantage of all these methods is that the local features of the test image must always be extracted.

In this paper, we propose a different hybrid approach that uses both global and local image features. The approach is inspired by the observation that in many situations, position estimates based on global features are comparable to position estimates based on local features. Only in difficult situations, the robustness of the local features leads to more reliable estimates. Thus, our approach tries to use local features only for difficult images, and global features for all other images. We use a particle filter for position estimation, and we base the decision whether to use local or global features for a certain image on the particle cloud. If the particle cloud indicates a reliable position estimate, we use global features. If the current position estimate seems to be uncertain, we use SIFT. The advantage over the methods mentioned above is that we do not have to extract the local features for each test image, but only for a few of them.

Experimental results on images of two different outdoor areas show that the localization errors of our hybrid approach are about the same as the errors of a SIFT approach.

C. Weiss, A. Masselli and A. Zell are with the Department of Computer Science, University of Tübingen, Sand 1, 72076 Tübingen, Germany {c.weiss, andreas.zell}@uni-tuebingen.de, andreas.masselli@web.de

H. Tamimi is with the College of Administrative Sciences and Informatics, Palestine Polytechnic University, P. O. Box 198, Hebron, Palestine hashem.tamimi@uni-tuebingen.de

However, the hybrid approach is more than 3.5 times faster than the SIFT approach and localization is possible two times per second on average.

The rest of the paper is organized as follows. Sections II and III describe the global and local image features we use in our hybrid approach. Section IV explains the particle filter and how it decides which features to use. Section V presents the experimental results, and Section VI concludes the paper and suggests future work.

II. GLOBAL IMAGE FEATURES

This section describes the global image features that we use in our hybrid method. We do not represent each image by a single global feature vector, but by two different ones, which makes the global method more robust. The first global feature vector consists of *Weighted Gradient Orientation Histograms* (WGOHs). For the second feature vector, we calculate weighted histograms of integral invariants, which we call *Weighted Grid Integral Invariant* (WGII) features.

A. WGOH

The Weighted Gradient Orientation Histograms were used by Bradley *et al.* for topological outdoor localization [9] and are similar to features used by Kosecka and Li for indoor localization [14]. The WGOHs are inspired by SIFT, as they also use weighted histograms of image gradient orientations. In contrast to SIFT, the histograms are not computed around local interest points, but on a grid of subimages. Bradley *et al.* obtained good results using WGOHs under different illumination conditions. This robustness is the reason why we also use WGOHs.

The extraction of WGOH features for an image works as follows. First, the image is split into a 4×4 grid of subimages. Then, an 8-bin gradient orientation histogram is calculated on each subimage, weighted by the magnitude of the gradient at each pixel. Additionally, pixels near the center of a subimage are assigned a higher weight than pixels near the borders, because under image translation or rotation, pixels near the centers are more likely to fall into the same subimage than pixels near the borders. In our implementation of WGOHs, we use 2D gaussians for weighting, centered at the centers of the subimages and with standard deviations equal to 0.5 times the width and the height of the subimage, respectively. To get the final 1×128 feature vector for an image, the histograms of the 16 subimages are concatenated.

B. WGII

Recently, we proposed a localization method based on Weighted Grid Integral Invariants (WGII) [15]. Therefore, this section only gives a short summary. A more detailed description of global integral invariants and the relational kernel can also be found in [16].

Global integral invariant features are invariant to euclidean motion, i.e. rotation and translation, and to some extent robust to illumination changes. The key idea is to apply all possible translations and rotations to the image and to calculate the features by averaging over all the transformed versions of the image.

To calculate the integral invariant features for a pixel, a kernel function is used that involves the local neighborhood of the pixel. This kernel function for example multiplies the intensities of two neighborhood pixels lying on circles with different radii around the pixel (*monomial kernel*). The final integral invariant feature is the mean of q kernel evaluations, for which the two neighborhood pixels are rotated by different angles around the pixel. For our purposes, the *relational kernel* works well, because it is robust to uniform illumination changes.

An image can be represented by a histogram of the integral invariants at the pixels. However, we experimentally found that global integral invariant feature histograms are not distinctive enough for our outdoor images. Thus, we modified the approach based on some ideas used for the WGOH features. First, we compute the integral invariant feature value for each pixel of the image. Then, we split the image into a 4×4 grid of subimages. On each subimage, we calculate a weighted 8-bin histogram of integral invariants. Similar to the WGOHs, we also assign a higher weight to pixels near the centers of subimages. For weighting, we use 2D gaussians centered at the centers of the subimages and with standard deviations equal to 0.25 times the width and the height of the subimages, respectively. After that, we concatenate the histograms to create the final 1×128 WGII feature vector.

To calculate the integral invariants, we use a relational kernel with pixel coordinates $p_1 = (10,0)$ and $p_2 = (0,20)$, set the parameter $\varepsilon = 0.098$ and use $q = 10$ rotations. We chose these parameters, because experimentally, they led to the best results.

C. Image Matching

For each image, we extract the WGOH and the WGII feature vector. To calculate the similarity between a test image $\mathbf{Q}$ and a training image $\mathbf{D}$, we first compare their WGOH and WGII feature vectors individually. For both comparisons, we use normalized histogram intersection. If $\mathbf{q}$ and $\mathbf{d}$ are $1\times m$-sized feature vectors of the test and training images,

$$\bigcap_{\mathbf{norm}}(\mathbf{q},\mathbf{d}) = \frac{\sum_{k\in\{0,1,\dots,m-1\}} \min(q_k,d_k)}{\sum_{k\in\{0,1,\dots,m-1\}} q_k}, \tag{1}$$

gives the similarity between the images. After having computed the image similarities s_{WGOH} and s_{WGII} individually, we multiply s_{WGOH} and s_{WGII} to get the final image similarity.

III. LOCAL IMAGE FEATURES

As local method in our hybrid localization approach we use SIFT [1]. In this approach, the most similar training image to a test image is the one which contains the highest number of local features that can be matched to the features of the test image, divided by the total number of features. As each feature of the test image must be compared to each feature of the training image, matching images is very time-consuming.

To reduce the time for matching, we reduce the number of SIFT features of the training and the test image. The idea is to delete "noisy" features, which are likely not to appear in more than one image. To reduce the number of features of the training images, we match each training image to the two nearest training images (based on the GPS positions of the images). We only keep the local features of the training image that can be matched to a feature of at least one of the two neighboring images. In the test phase, we match the current test image to the test image that was taken directly before. We again only keep the features that can also be found in the preceeding image. This method removes about 50 to 80% of the features, and accelerates image matching by a factor of 5 on average, without loss of accuracy.

IV. HYBRID APPROACH

To estimate the current position of the robot, we use a particle filter [17]. The particle filter updates the weights of its particles based on image similarities which are calculated using either the global or the local features described in the previous sections. We decide which features to use for a test image based on the particle cloud. If the particle cloud indicates that the current estimate is reliable, we use the fast global features. If the particle cloud indicates that the current estimate is uncertain, we use the local features. In the following, we describe the particle filter in more detail as well as the method that decides which features to use.

A. Position Estimation using the Particle Filter

Particle filters represent the belief of the robot about its position by a set of m particles. In our case, each particle consists of a position (x,y) and a non-negative weight or importance factor w. The weighted mean of all particles is the position estimate of the robot. For global robot localization, where the starting position of the robot is unknown, the particle filter starts with m particles that are randomly distributed over the robot's universe. The initial weight of each particle is $\frac{1}{m}$. The weights and the positions of the particles are updated for a new test image at time step t as follows:

1) Randomly draw m particles from the particle cloud according to the particle weights at time $t-1$.
2) Update the position of each particle according to a motion model. As we do not use a motion model from odometry, we randomly update the position of a particle according to a 2D gaussian centered at the particle and with standard deviations of 4 m. Additionally, we move each particle a short distance d towards the position of its nearest training image, where d is 0.2 times the distance between the particle and its nearest training image.
3) Compute new weights for the particles based on the similarity between the current test image and the training images. For each particle, we first search the nearest training image. Then, we match this training image to the test image using either SIFT or global features to get the new weight of the particle. Additionally, we multiply the weight of a particle by a factor that decreases with the distance of the particle to the nearest training image. In the case of the global method, we also potentiate the weight by 20, because the differences between the matching scores are low (but still distinctive at that low level). Finally, we normalize the weights.

At this point, we calculate the estimated position of the robot. Then we decide if we use local or global image features for the next image. Section IV-B explains this decision in more detail. As a last step, we replace the worst 5% of the particles by new particles with random positions and weight $w=\frac{1}{m}$, and renormalize the weights. The random insertion insures that the robot can fastly recover its position if the position was lost or the robot was kidnapped.

To speed up the calculation of the weights, we save for each particle the matching result to the test image. If another particle has the same nearest training image, we can use the saved value. In the case of SIFT, this method speeds up the estimation of a new position by a factor of about 5. For the global features, we only get a slight speedup.

B. Feature Selection

In general, localization based on SIFT features is relatively accurate and robust, but takes about 1.7 s per image on our robot. Position estimates based on our global features are only slightly worse in most cases, but can be computed in about 0.4 s. However, there are situations in which the error created by the global features is much larger than the error created by SIFT. Thus, the goal of the hybrid approach is to use global features in as many situations as possible and to use SIFT features only when necessary.

The particle cloud represents a probability distribution for the position of the robot. Thus, analyzing the particle cloud provides a way to evaluate the quality of the current estimate. The robot is relatively certain about its position if most of the particles are concentrated around a single spot. This corresponds to a single high peak in the probability distribution. If the particles are relatively wide-spread, the position estimate is not reliable. If there is more than one high peak in the probability distribution, i.e. the particles are divided into groups which concentrate around different spots, the position estimate is also uncertain, because it is not clear which of the peaks corresponds to the true position. Moreover, the weighted mean of the particles will be located somewhere between the peaks.

We found that the standard deviations in x and y direction can characterize these situations well. The standard deviation σ_x in x direction is given by

$$\sigma_x = \sqrt{\sum_{i=1}^{m}(x_i-\mu_x)^2 w_i}, \qquad (2)$$

where m is the number of particles, x_i is the x-position of the i-th particle, μ_x is the weighted mean of the x-positions

Fig. 1. Our RWI ATRV-JR outdoor robot “Arthur”.

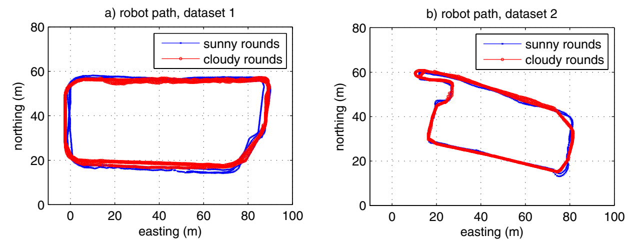

Fig. 2. GPS data. a) Dataset 1. b) Dataset 2.

of the particles and w_i is the weight of the i-th particle. The calculation of σ_y is analogous to (2).

The algorithm that decides which features we use in a particle filter step works as follows. In the initial step, we always use SIFT features, because the position of the robot is completely unknown. At the end of each step, we calculate the standard deviations σ_x and σ_y of the particle cloud. Now, there a two possible cases, depending on whether we used SIFT or global features for the current step:

1) If we used SIFT features, we switch to global features for the next image if both σ_x and σ_y are below a threshold d_{SIFT}. If $\sigma_x > d_{SIFT}$ or $\sigma_y > d_{SIFT}$, we keep on using SIFT.
2) If we used global features, we switch to SIFT for the next image if at least one of σ_x and σ_y is above a threshold d_{global}. If $\sigma_x < d_{global}$ and $\sigma_y < d_{global}$, we keep on using global features.

The remaining question is the choice of the thresholds d_{SIFT} and d_{global}. The larger d_{global} and d_{SIFT}, the more steps will use global features. The smaller d_{global} and d_{SIFT}, the more steps will use SIFT. Good thresholds result in a small number of SIFT steps while also creating a low error. Our particle filter randomly updates the position of a particle according to a 2D gaussian with standard deviations of 4 m. Thus, if the particles were all perfectly located at a single spot before the update (indicating a very certain estimate), they will be distributed according to a 2D gaussian with standard deviations of 4 m after the update. But as we move each particle towards its nearest training image and particles near the mean of the cloud will possibly get high weights, the final standard deviation is likely to be < 4 m. In our experiments, about 50% of the particle clouds have standard deviations < 2.5 m. To accept not only very good but also good estimates, we set $d_{global} = 5.5$ m. To support a fast switch from local to global features, we set d_{SIFT} even higher to 7.0 m. For other motion models of the particle filter, d_{global} and d_{SIFT} must be chosen differently.

V. EXPERIMENTAL RESULTS

In our experiments, we used an RWI ATRV-JR outdoor robot (Fig. 1). The robot is equipped with a stereo camera, of which we only used the left camera. The robot took one 320×240 pixel gray-scale image per second while we moved it around with a constant velocity of about 0.6 m/s.

To get ground truth information about the positions of the images, we used a differential GPS (DGPS). Under ideal conditions, the position error of the DGPS is below 0.5 m. However, due to distorted GPS signals, the GPS path sometimes significantly deviated from the true path of the robot. Another problem are missing GPS signals due to occlusion by trees and buildings, which results in gaps in the GPS path. As we always used a constant velocity, we filled gaps in the GPS path by linearly interpolating between the positions before and after the gap. We also manually corrected wrong GPS values that significantly deviated from the true path.

The mentioned problems show that the ground truth information itself is not very exact. Thus, the experimental errors may also be inexact. However, as we mainly use the errors to compare different approaches instead of focussing on the absolute errors, the ground truth information is sufficient, because every approach uses the same ground truth data.

We collected image datasets in two different environments. Dataset 1 consists of six rounds around a big building. Each round is about 260 m long and consists of about 400 images. We took the first three rounds on a sunny day. However, there are some short sections (5 to 10 s long) during which the sun was covered by clouds. Six weeks later, we collected the other three rounds on a cloudy day. The images mainly show artificial objects like streets, buildings and cars, but also some bushes and trees. Additionally, the images show some dynamic objects like cars and people passing by.

Dataset 2 consists of four rounds on a meadow. The images mainly show vegetation like grass, trees and bushes. Each round is about 220 m long and consists of about 350 images. We collected the first two rounds in the early afternoon, in which the sun was shining brightly. In the evening, we took the images of the third and fourth round. The sun was completely covered by clouds and it was starting to get dark. Fig. 2 shows the GPS ground truth data for dataset 1 and 2. Fig. 3 shows example images under different illumination.

For evaluation, we calculated the mean error of all possible training/test combinations of rounds using $m = 300$ particles. Additionally, we repeated each experiment n times, where n is the number of test images. For each of these experiments, we used a different test image as starting image for the localization. Then we calculated the mean error of all experiments that are similar, e.g. all experiments in which we used the

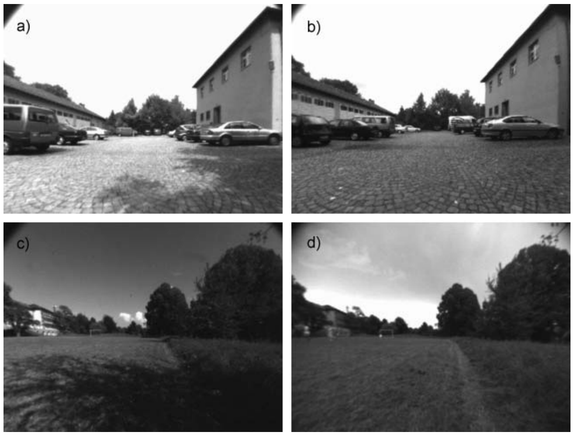

Fig. 3. Example images. a) Dataset 1, sunny. b) Dataset 1, cloudy. c) Dataset 2, sunny. d) Dataset 2, cloudy.

TABLE I

MEAN LOCALIZATION ERRORS $\pm$ STANDARD DEVIATION (M)

set	training	test	WGII + WGOH	SIFT	Hybrid
1	sunny	sunny	3.15 ± 1.20	2.15 ± 0.29	2.03 ± 0.22
	cloudy	cloudy	1.60 ± 0.26	2.06 ± 0.56	1.59 ± 0.27
2	sunny	sunny	1.09 ± 0.06	1.78 ± 0.05	1.09 ± 0.05
	cloudy	cloudy	2.07 ± 0.76	2.10 ± 0.14	1.80 ± 0.35
1	sunny	cloudy	3.95 ± 0.54	3.27 ± 0.27	3.17 ± 0.32
	cloudy	sunny	3.85 ± 0.79	2.52 ± 0.17	3.07 ± 0.36
2	sunny	cloudy	3.64 ± 0.80	2.88 ± 0.20	2.92 ± 0.23
	cloudy	sunny	3.45 ± 0.32	2.74 ± 0.24	2.78 ± 0.22

sunny images of dataset 1 for training and the cloudy images for testing.

Tab. I compares the localization errors of the global and local methods to the errors of the hybrid approach. The hybrid method uses the thresholds $\sigma_{global} = 5.5$ m and $\sigma_{SIFT} = 7.0$ m. Only in one case (dataset 1, cloudy vs. sunny), the error of the hybrid approach is significantly larger than the error of the SIFT approach. In all other cases, the error of the hybrid approach is similar to or smaller than the error of SIFT. On average, the localization error of the hybrid approach is 0.13 m smaller than the error of the SIFT approach. When only regarding the experiments with changing illumination, the average error of the hybrid approach is 0.13 m larger than the error of the SIFT approach. Fig. 4 a) to d) compare the localization errors of the three methods over time. The plots are mean curves for similar experiments. For all methods, the error decreases rapidly after a few images.

The experiments show that the localization accuracy of our hybrid approach is similar to the accuracy of the SIFT approach. However, the advantage of the hybrid approach over SIFT is the much lower average localization time per image. On our robot's PC (1.8 GHz Pentium M, 1 GB RAM), the SIFT approach needs about 1.70 s for one image and the global method needs about 0.39 s. Thus, the maximal speed-up of the hybrid method compared to SIFT would be 4.36. In this case, the global features would be used for all images. Tab. II and Fig. 5 compare the hybrid approach to the local

TABLE II

LOCALIZATION OF ONE IMAGE USING THE HYBRID METHOD

dataset	SIFT steps (%)	time (s)	speed-up
1, constant illumination	2.57	0.43	3.92
2, constant illumination	1.72	0.42	4.02
1, changing illumination	6.37	0.48	3.52
2, changing illumination	8.36	0.51	3.34
average	4.75	0.46	3.70

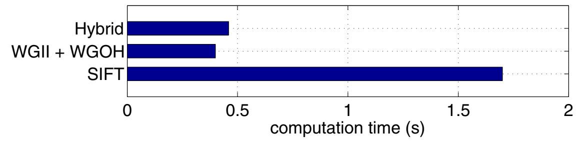

Fig. 5. Average localization time per image.

and global approaches. As expected, the percentage of SIFT steps is higher for experiments with changing illumination than for experiments with constant illumination. On average, SIFT is used for less than 5% of the images. This results in an average localization time per image of 0.46 s. The 2.9 ms for the decision which features to use in the next step are included. The average speed-up of the hybrid method compared to SIFT is 3.70.

Fig. 4 e) shows the mean error curves for different numbers of particles. There is only a very small difference between 300 and 500 particles. The difference between the errors for 300 and 100 particles is about 0.15 m on average. The difference is more significant when using SIFT (0.81 m) or the global features (0.42 m) alone. However, 100 particles are too few to reliably cover environments of our size or larger ones.

Finally, Fig. 4 f) shows the influence of the thresholds d_{local} and d_{SIFT} on the localization error and the computation time. The figure shows that there is a trade-off between accuracy and speed: Lower thresholds lead to smaller errors, but also to a higher percentage of SIFT steps and hence to reduced speed. On the other hand, higher thresholds lead to higher speed, but also to larger localization errors.

VI. CONCLUSION

We presented a new hybrid method for vision-based outdoor localization that combines local and global image features. Before each new test image, the method selects whether to use the local or the global image features for the test image. The selection is based on the particle cloud that represents the belief of the robot about its position. In situations in which the robot is relatively certain about its position, it uses the fast global features. In situations in which the robot is uncertain about its position, it uses the slow, but more exact and robust local image features.

Experiments on images of outdoor environments show that the localization errors of the hybrid method are about the same as the errors of a pure SIFT approach. However, the hybrid method is more than 3.5 times faster than the SIFT method.

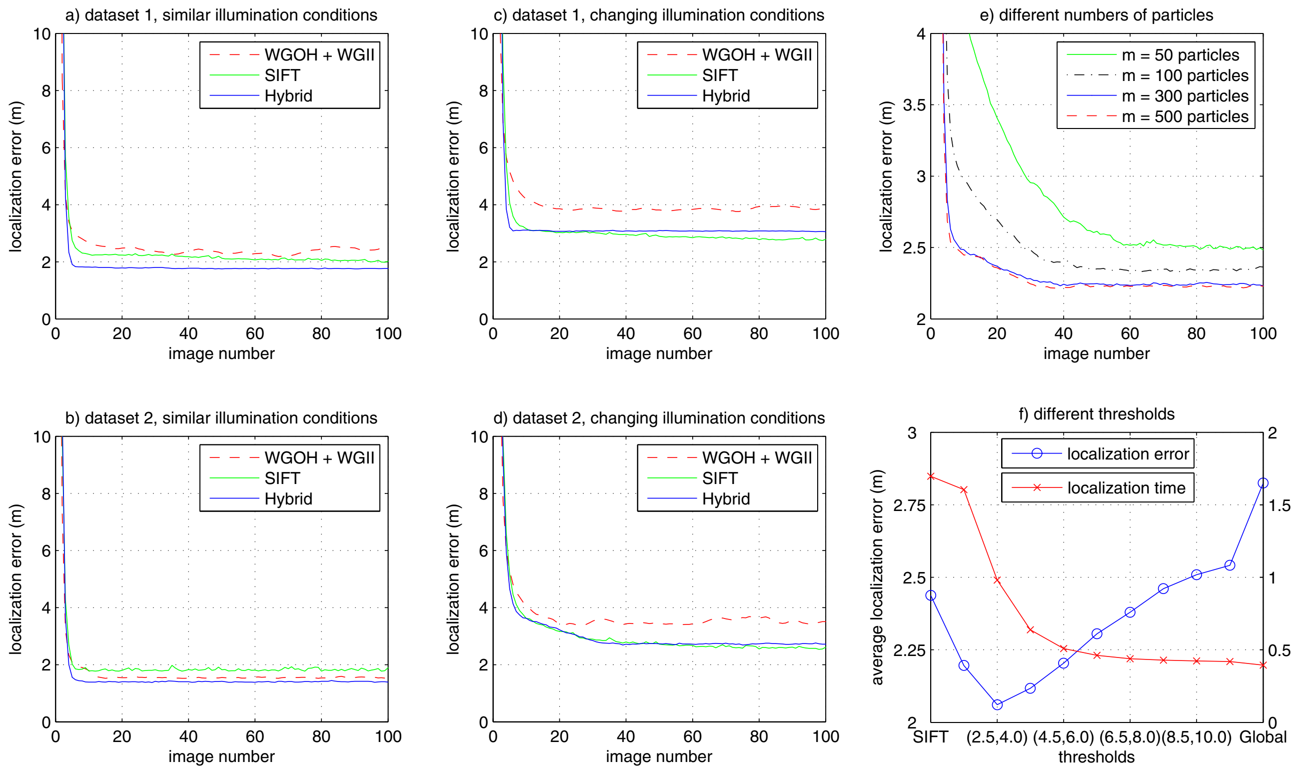

Fig. 4. a) - d) Mean errors for particle filter experiments. There is no significant change after image 100. The mean initial error is about 36 m for dataset 1 and about 26 m for dataset 2. e) Mean errors for different numbers of particles. f) Influence of the thresholds d_{global} and d_{SIFT} on the localization error and the computation time. The thresholds are given as pair (d_{global}, d_{SIFT}). Both values are incremented by 1 for each step on the x-axis.

Of course the local and global image features we used in this paper (SIFT and WGII + WGOH) are only examples of features that can be used in the hybrid approach. In future work, we will examine if other combinations of features are more exact and/or faster. Another issue is the method that decides whether to use the global or the local image features for the next image. We will try to find other techniques that further minimize both the localization error and the number of steps that use local features.

REFERENCES

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] S. Se, D. Lowe, and J. Little, "Local and global localization for mobile robots using visual landmarks," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Maui, Hawaii, October 2001, pp. 414 – 420.

[3] H. Tamimi and A. Zell, "Global robot localization using iterative scale invariant feature transform," in *Proc. 36th International Symposium on Robotics (ISR)*, Tokyo, Japan, 2005.

[4] T. D. Barfoot, "Online visual motion estimation using fastslam with sift features," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, 2005, pp. 3076 – 3082.

[5] M. Artac, M. Jogan, and A. Leonardis, "Mobile robot localization using an incremental eigenspace model," in *Proc. IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, Washington, D.C., 2002, pp. 1025–1030.

[6] M. Jogan, M. Artac, D. Skocaj, and A. Leonardis, "A framework for robust and incremental self-localization," in *Proc. 3rd International Conference on Computer Vision Systems (ICVS)*, Graz, Austria, 2003, pp. 460–469.

[7] M. Jogan, A. Leonardis, H. Wildenauer, and H. Bischof, "Mobile robot localization under varying illumination," in *Proc. International Conference on Pattern Recognition (ICPR)*, Quebec, Canada, 2002, pp. 741–744.

[8] 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*, vol. 21, no. 2, pp. 208–216, 2005.

[9] D. M. Bradley, R. Patel, N. Vandapel, and S. M. Thayer, "Real-time image-based topological localization in large outdoor environments," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Canada, 2005, pp. 3062 – 3069.

[10] H. Tamimi, "Vision-based features for mobile robot localization," Ph.D. dissertation, University of Tübingen, Tübingen, Germany, 2006.

[11] M. Artač and A. Leonardis, "Outdoor mobile robot localisation using global and local features," in *Proc. 9th Computer Vision Winter Workshop (CVWW)*, 2004.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. Cambridge, MA, USA: MIT Press, 2005.

[13] P. Heinemann, J. Haase, and A. Zell, "A novel approach to efficient monte-carlo localization in robocup," in *RoboCup 2006: Robot Soccer World Cup X*, 2006.

[14] J. Kosecka and F. Li, "Vision based topological markov localization," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, USA, 2004, pp. 1481–1486.

[15] C. Weiss, A. Masselli, H. Tamimi, and A. Zell, "Fast outdoor robot localization using integral invariants," in *Proc. 5th International Conference on Computer Vision Systems (ICVS)*, Bielefeld, Germany, March 2007.

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

[17] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.