Visual Self-Localization with Tiny Images

Marius Hofmeister, Sara Erhard and Andreas Zell

University of Tübingen, Department of Computer Science, Sand 1, 72076 Tübingen

Abstract. Self-localization of mobile robots is often performed visually, whereby the resolution of the images influences a lot the computation time. In this paper, we examine how a reduction of the image resolution affects localization accuracy. We downscale the images, preserving their aspect ratio, up to a tiny resolution of 15×11 and 20×15 pixels. Our results are based on extensive tests on different datasets that have been recorded indoors by a small differential drive robot and outdoors by a flying quadrocopter. Four well-known global image features and a pixelwise image comparison method are compared under realistic conditions such as illumination changes and translations. Our results show that even when reducing the image resolution down to the tiny resolutions above, accurate localization is achievable. In this way, we can speed up the localization process considerably.

1 Introduction

Mobile robots need to localize themselves in an environment to solve complex tasks. Positioning is often done visually, since cameras are inexpensive and flexible sensors and nowadays provide high resolutions. But even if computational power has increased significantly during the past decade, there are still fields in which it keeps restricted. For example, swarm robotics requires a large number of relatively simple agents that have to be reasonably priced at a small size. And even in case of unmanned aerial vehicles, processing power is often restricted due to the limited weight and battery power that the robots can carry.

Visual self-localization is often performed using image retrieval techniques that 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 leads then to an estimation 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 extraction time of those features depends mainly on the image resolution and thus could be decreased. However, by reducing the resolution of images we lose information that might be helpful for the localization task. Thus, the focus of this paper lies in the investigation to what extent a reduction of image data affects localization accuracy and computation time. We therefore examine the localization process on two different platforms: a small two-wheeled mobile robot indoors and a flying quadrocopter outdoors.

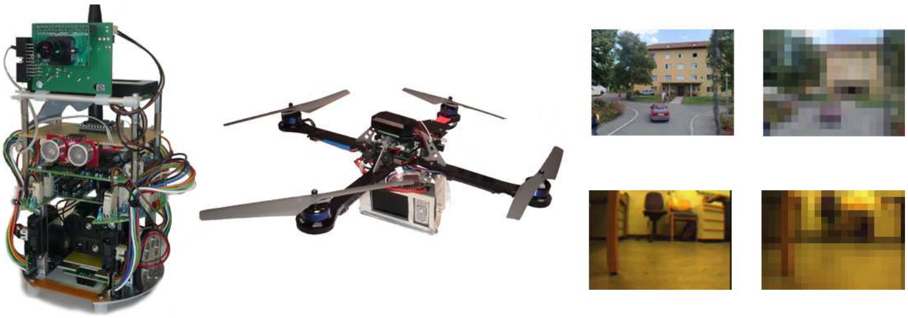


Fig. 1. Employed robots and example images at highest and lowest resolutions.

2 Related Work

Approaches to the visual self-localization problem mainly differ in the type of image features that are extracted from the images. We distinguish two kinds of image features: local and global ones. While local features, like the *Scale-Invariant Feature Transform* (SIFT) by Lowe [1], describe only patches around interest points in an image, global features describe the whole image as one single fixed-length vector.

Many local features are invariant to scale and rotation and robust to illumination changes [1]. As the number of local features in an image can be high, however, it may take a long time to find, match, and store them. Global image features have also shown a good localization accuracy [2,3,4], that is, however, lower. Their main advantage is their short computation time. As our applications require onboard image processing on microcontrollers with limited computation power, we decided to employ global image features in this work.

Ulrich and Nourbakhsh [5] established self-localization for 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 the topological map of the environment. Zhou et al. [6] extended this approach to the use of multidimensional histograms, taking into concern features like edges and texturedness. Wolf et al. [7] performed visual localization by combining an image retrieval system with Monte Carlo localization. They used local image features that are invariant to image rotations and limited scale [8] and that are also the basis for the global *Weighted Grid Integral Invariants*, which are employed in this paper.

Our approach to use tiny images was also inspired by Torralba et al. [9] who stored millions of images from the internet 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 [10]. He used images with a resolution of 64×48 pixels using measurement functions based on edges, gradients, and texturedness, but did not compare the localization rate and computation time to other image resolutions.

3 Robots

We conducted our experiments on two different robots: a small, two-wheeled *c't-Bot* (http://www.ct-bot.de), which was developed by the German computer magazine *c't*, and a quadrocopter *X3D-BL Hummingbird* distributed by *Ascending Technologies* [11] (see Fig. 1). The image processing is performed on separate modules: in case of the *c't-Bot* on a *POB-Eye* camera module equipped with a 60 MHz ARM7 microcontroller and in case of the quadrocopter on a *Nokia N95* mobile phone with a 332 MHz ARM11 processor. On the *c't-Bot*, feature vectors are saved on a SD card that is connected via I^{2}C, while on the quadrocopter feature vectors can be stored directly in the internal memory. On both systems, computation power is restricted and thus is a valuable resource.

4 Global Image Features

The selection of image features results from the limited processing power of our robots. Color and grayscale 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)*, which yielded good results in earlier research, especially under illumination changes [2,3,4].

All features are investigated for different resolutions. Therefore, we downscale the images preserving their aspect ratio up to a tiny resolution of 15×11 and 20×15 pixels by interpolating the pixel intensities in a bilinear fashion. This downscaling also permits a pixelwise image comparison in a reasonable computation time. 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 image resolutions and discovered that a 4×4 grid leads to the best results.

4.1 Weighted Gradient Orientation Histograms

Weighted Gradient Orientation Histograms (WGOH) were presented by Bradley et al. [2] and were intended for outdoor environments because of their robustness to illumination changes. Their design was inspired by SIFT features [1]. Bradley et al. first split the image into a 4×4 grid of subimages. On each subimage, they calculated 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 [3]. 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.

4.2 Weighted Grid Integral Invariants

The key idea of integral invariants was to design features which are invariant to Euclidean motion, i.e., rotation and translation [7,8]. In order to achieve that, all possible rotations and translations are applied to the image. In our case, two relational 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 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. [4] 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 (see Sect. 4.1) to make the vector more robust to translations. The output is a 2×8 histogram for each subimage and a 1×1024 histogram for the entire image.

4.3 Color/Grayscale Grid Histograms

For the color and grayscale 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 process the hue value of the HSV color space. This choice of space promises to be more robust to illumination changes. As stated above, we weight each pixel by a Gaussian to make the vector more robust to translations and normalize it afterwards.

4.4 Pixelwise Image Comparison

The reduction of the image resolution permits also to compare the image data in a pixelwise fashion rather than extracting first the features. In this way, computation time may be saved. Therefore, the image data is treated as a vector. To keep the data small, we only compare the normalized grayscale image and discard color information.

5 Localization Process

Our localization process consists of two steps, the mapping phase and the localization 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. In the localization phase, *test images* are recorded and features are again extracted. These features are subsequently compared to all other previously saved feature vectors. The mapped position of the vector with the highest similarity is then chosen to become the current position estimate of the robot.

To perform the 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* $\bigcap_{norm}(q, d)$:

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

Here, m is the number of histogram bins and q_k denotes bin k of histogram q. In [4], this method showed good results. 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^*:

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

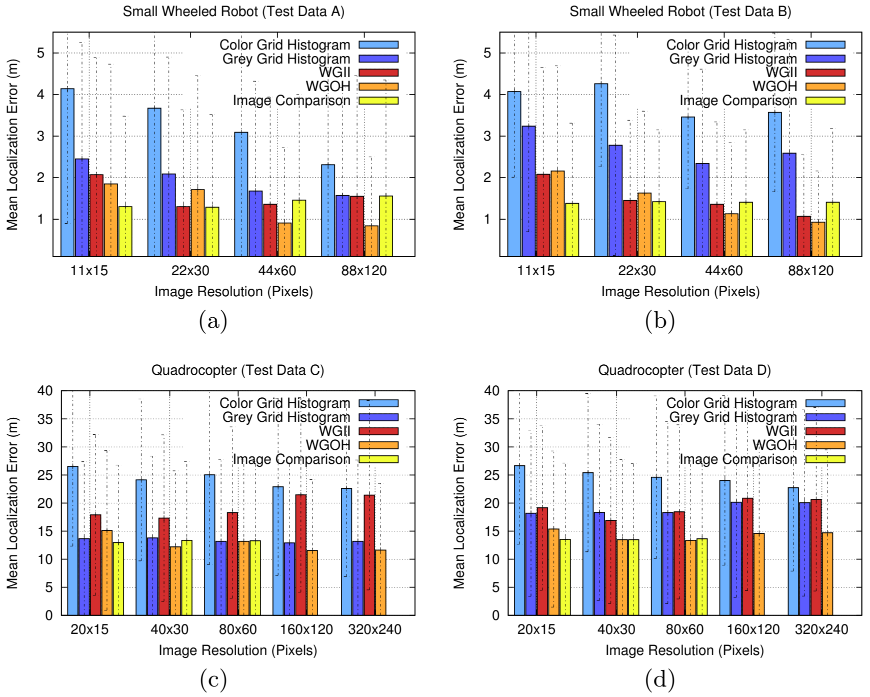
where r is the number of pixels and Q_k^* denotes pixel k of image Q^*. The similarity $sim(Q, D)$ of two images can now be computed as:

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

Note that in general $0 \leq L_1(Q^*, D^*) \leq 2$ (although $L_1(Q^*, D^*) > 1$ rarely happens for images).

6 Experimental Results

We conducted our experiments with the *c't-Bot* in an office environment. Since the robot does not have the ability to determine its ground truth position through GPS or other accurate sensors like laser scanners, we grabbed images every 0.5 m in an area of appr. 75 m^2. To limit possible viewpoints and thus faciliate the localization, we employed a compass. 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 furniture etc., the direction indicated by the compass was not always true but repeatable, thus it can be seen as a function of the position. The test data were grabbed at randomly chosen positions. 100 images, in the following called *test data A*, were grabbed at stable illumination. Another 100 images, *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 weak odometry and compass errors, the robot's rotation is affected by errors which appear approximately as translations in the images.



(a)

(b)

(c)

(d)

Fig. 2. Localization results in case of the *c't-Bot* (a,b) and quadrocopter (c,d). Shown are mean localization errors of the image features at different image resolutions. The pixelwise image comparison is referred to as image comparison. No measurements were taken for the pixelwise image comparison under high resolutions because of database restrictions.

To perform the experiments with the quadrocopter, we steered it at altitudes around eight meters, flying rounds of appr. 180 m in a courtyard. The view of the camera pointed in course direction. In total, we grabbed 1275 images in several rounds at a frequency of appr. one image per second. Each round consisted of a different number of images due to the different velocity of the quadrocopter. In the mapping phase we grabbed 348 images at dull daylight. For the localization phase we used two different datasets: *Test data C* consists of 588 images (four rounds) at similar lighting conditions, *test data D* of 369 images (three rounds) at sunny daylight.

Figure 2 shows the localization accuracies of the examined methods. The mean localization error is measured in 2D only; in case of the quadrocopter we tried to keep the flying altitude constant. The smallest mean localization error we obtain is in case of the *c't-Bot* 0.84 m and in case of the quadrocopter 11.55 m, using WGOH. Further experiments showed that by using a particle filter to compute a probabilistic position estimate, the localization errors can be reduced to about 0.50 m on the *c't-Bot* and about 6 m on the quadrocopter. However, in this paper we focus on feature extraction techniques.

Looking at the different methods in detail, we find out that in most cases WGOH lead to best results. The results of WGII are worse, even if it also computes differences of pixel intensities. It provides rotation invariance, but that comes with the cost of losing orientation information. The straightforward pixelwise image comparison yielded surprisingly high accuracies. This may be because the normalization helps to cope with illumination changes and the use of wide-angle lenses limits the influence of translations. A localization was not possible with the color grid histograms. The reason for this may be the poor color quality of our cameras and the lack of meaningful color information in the environments. As it could be expected, the overall accuracy on *test data B* and *test data D* was worse, due to the different lighting conditions. The grayscale grid histograms did not perform well here. Generally, despite the reduction of the image resolution we achieved a reasonable localization accuracy. While WGOH and WGII may have suffered from averaging over the subimages at small image resolutions, the pixelwise image comparison provided constant localization rates at all resolutions.

We also examined the computation times of the whole localization process on the training data (see Table 1). We chose WGOH since it revealed a good accuracy at a reasonable feature extraction time and the pixelwise image comparison on the small size images since it is the fastest method of all with respect to feature extraction time.

Table 1. Computation times of the localization process at different resolutions using WGOH and the pixelwise image comparison (referred to as Img. Comp.).

	WGOH (full res.)	WGOH (smallest res.)	Img. Comp. (smallest res.)
c't-Bot	4.812 s	3.841 s	4.002 s
Quad.	0.639 s	0.267 s	0.397 s

By the use of tiny images, we achieve a speed up of 20.2 % in case of the *c't-Bot* and in case of the quadrocopter of 58.2 %, comparing WGOH at the highest and the smallest resolution. The localization process can roughly be divided into feature extraction and feature matching.The pixelwise image comparison is not the fastest method since the corresponding vector has a higher dimensionality than the WGOH vector and thus requires more time to be compared. On the *c't-Bot*, the localization time is highly affected by the matching step and needs 3.841 s, which is quite long, but we should also keep in mind the computational limitations of small mobile robots. To further speed up the feature matching, different approaches could be employed in the future, e.g. a kd-tree for a more efficient search or a particle filter to limit the number of feature comparisons.

7 Conclusion

In this paper, we examined to what extent a reduction of the image resolution affects accuracy and computation time in the visual self-localization process. Therefore, we compared four well-known global image features in an indoor and outdoor scenario where computation power is restricted. The reduction of the image resolution made it also possible to apply a straightforward pixelwise image comparison. Generally, WGOH provided good performance with relatively low computation times.

We state that in our medium-sized indoor and outdoor test beds, tiny grayscale images of 15×11 and 20×15 pixels contained enough information to establish efficient self-localization at satisfactory accuracy. In this way, we could speed up the localization process considerably. These results are especially interesting for researchers working on systems with restricted computation power to achieve visual self-localization in similar environments in a fast and efficient way.

References

1. Lowe D.: Distinctive Image Features from Scale-Invariant Keypoints. Int. Journal of Computer Vision 60(2), pp. 91–110, 2004.
2. Bradley D. M., Patel R., Vandapel N., Thayer S. M.: Real-Time Image-Based Topological Localization in Large Outdoor Environments. Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Edmonton, Canada, pp. 3670–3677, 2005.
3. Weiss C., Masselli A., Zell A.: Fast Vision-based Localization for Outdoor Robots Using a Combination of Global Image Features. Proc. of the 6th Symposium on Intelligent Autonomous Vehicles (IAV), Toulouse, France, 2007.
4. Weiss C., Masselli A., Tamimi H., Zell A.: Fast Outdoor Robot Localization Using Integral Invariants. Proc. of the 5th Int. Conf. on Computer Vision Systems (ICVS), Bielefeld, Germany, 2007.
5. Ulrich I., Nourbakhsh I.: Appearance-Based Place Recognition for Topological Localization. Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), San Francisco, CA, USA, pp. 1023–1029, 2000.
6. Zhou C., Wei Y., Tan T.: Mobile Robot Self-Localization Based on Global Visual Appearance Features. Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Taipei, Taiwan, pp. 1271–1276, 2003.
7. Wolf J., Burgard W., Burkhardt H.: Robust Vision-based Localization by Combining an Image Retrieval System with Monte Carlo Localization. IEEE Transactions on Robotics, 21(2), pp. 208–216, 2005.
8. Siggelkow S.: Feature Histograms for Content-Based Image Retrieval. Ph.D. dissertation, Institute for Computer Science, University of Freiburg, Germany, 2002.
9. Torralba A., Fergus R., Freeman W. T.: 80 million tiny images: A large dataset for non-parametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(11), pp. 1958–1970, 2008.
10. Argamon-Engelson S.: Using Image Signatures for Place Recognition. Pattern Recognition Letters, 19(10), pp. 941–951, 1998.
11. Gurdan D., Stumpf J., Achtelik M., Doth K.-M., Hirzinger G., Rus D.: Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz. Proc. of the Int. Conf. on Robotics and Automation (ICRA), Rome, Italy, pp. 361–366, 2007.