Adaptive and Fast Scale Invariant Feature Extraction

Emanuele Frontoni and Primo Zingaretti

Universita' Politecnica delle Marche, Ancona, Italy

Abstract. The Scale Invariant Feature Transform, SIFT, has been successfully applied to robot vision, object recognition, motion estimation, etc. Still, the parameter settings are not fully investigated, especially when dealing with variable lighting conditions. In this work, we propose a SIFT improvement that allows feature extraction and matching between images taken under different illumination. Also an interesting approach to reduce the SIFT computational time is presented. Finally, results of robot vision based localization experiments using the proposed approach are presented.

1 Introduction

In computer vision features are at the basis of image data processing. The goal of computer vision is to extract information about the content of specific data, i.e., a 2D image [1–5]. The extraction process is then often defined in terms of image features of different levels of complexity, ranging from low-level features such as edges or lines via medium-level features such as corners or junctions to high-level features in terms of objects or living beings, and actions that they perform. However, the classification into low, medium and high level features is not standardized. In the field of computer vision and image processing a large number of features, with different complexity, have been defined, e.g.:

- point features, e.g., corners, line crossings, or general interest points;
- local boundary features, e.g., lines or edges and their orientation;
- shape features, e.g., curvature;
- region based features, e.g., color, texture or objects.

From a conceptual point of view, an image feature should be defined in terms of attributes related to the image data. Usually these features are used to perform some kind of scene geometric reconstruction or object recognition or, also, they are used to perform range measurements based on vision. In the appearance based approach the same features can be used to evaluate an image similarity, taking into account the whole image without any need of physic description, but with the only purpose of evaluate scene appearance.
The SIFT approach was introduced by David Lowe in [6, 7]. SIFT is a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different images of the same object or scene. Because of its computational efficiency and effectiveness in object recognition, the SIFT algorithm has led to

Frontoni E. and Zingaretti P. (2007).
Adaptive and Fast Scale Invariant Feature Extraction.
In *Robot Vision*, pages 117-125
DOI: 10.5220/0002066801170125

significant advances in computer vision. The goal of our project is essentially to clean up, simplify and improve Lowe's SIFT algorithm. We intend first to implement the algorithm roughly as Lowe has defined it and then to make changes to it, gauging their effectiveness in object recognition. Specifically, we intend to improve SIFT's robustness to illumination changes, which will be judged by recognition accuracy in various outdoor scenes. We hope in general to improve the effectiveness of SIFT recognition keys by experimenting with different keypoint-descriptor generation methods, trying to maximize recognition scores with varying cameras and illuminations. We are creating an image database along the way to test actual implementation and future changes to the algorithm.

One of the major problem with SIFT is that the algorithm is not crisply defined and has lots of free parameters; information provided by the Lowe's papers is sometimes vague, and thus leaves lots of implementation details to be filled in.

The SIFT is invariant to image translation, scaling and rotation. SIFT features are also partially invariant to illumination changes and affine 3D projections. These features have been widely used in the robot localization field as well as in many other computer vision fields. The SIFT algorithm has four major stages.

1. Scale-space extrema detection: the first stage searches over scale space using a Difference of Gaussian (DoG) function to identify potential interest points.

2. Key point localization: the location and scale of each candidate point are determined and key points are selected based on measures of stability.

3. Orientation assignment: one or more orientations are assigned to each key point based on local image gradients.

4. Key point descriptor: a descriptor is generated for each key point from information on local image gradients at the scale found in stage 2.

The first stage is clarified as follows. For each octave in the scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images. Adjacent Gaussian images are subtracted to produce the DoG images. After each octave, the Gaussian image is down-sampled by a factor of 2 and the process is repeated. For a more detailed discussion of the key point generation and factors involved see [6].

In a nutshell, Lowe's algorithm finds stable features over scale space by repeatedly smoothing and down sampling an input image and subtracting adjacent levels to create a pyramid of difference-of-Gaussian images. The features the SIFT algorithm detects represent minima and maxima in scale space of these difference-of-Gaussian images. At each of these minima and maxima, a detailed model is fit to determine location, scale and contrast, during which some features are discarded based on measures of their instability. Once a stable feature has been detected, its dominant gradient orientation is obtained, and a key point descriptor vector is formed from a grid of gradient histograms constructed from the gradients in the neighborhood of the feature. Key point matching between images is performed using a nearest-neighbor indexing method.

There are many points along the course of this algorithm where simplifications and potential improvements can be made. Our current goals, beyond implementing and testing Lowe's algorithm, are:

(1) simplify and clean up the algorithm as much as possible,

(2) improve lighting invariance by normalizing potential SIFT difference-of-Gaussian

points with the sum-of-Gaussians, and
(3) improve the general stability of key points.
Our approach is based on the Scale Invariant Feature Transform (SIFT). In particular we propose an improvement of this feature extraction method to deal with changing in lighting conditions. This kind of adaptive vision is necessary in various application fields of vision feature based techniques, e.g. outdoor robotics, surveillance, object recognition, etc. In general, the improvement is particular useful whenever the system needs to work in presence of strong lightness variations. Also, while invariant to scale and rotation and robust to other image transforms, the SIFT feature description of an image is typically large and slow to compute. To solve this matter an approach to reduce the SIFT computational time is also presented. The paper is organized as follow: next section provides a description of the SIFT algorithm and of the proposed improvements; results of mobile robot localization are discussed in section 3 and, finally, conclusions and references are given.

2 Extension to Reduced SIFT

SIFT features are distinctive and invariant features used to robustly describe and match digital image content between different views of a scene. Consequently, hese features have been widely used in the robot localization field as well as in many other computer vision fields.
As already said, the SIFT feature description of an image is typically large and slow to compute. For this reason we compute the image similarity in the innovation term using a reduced and optimized SIFT approach with 64 feature descriptors, and we introduced time saving improvements by the following two steps:
- adaptation of SIFT parameters to each sub-image in which the original image is splitted(Figure 1);
- extraction of a fixed number of key points.

In particular, the number of scales of original image is defined according to its dimensions and thus in some cases not all SIFT scales need to be computed.

The following threshold value (Tr) is also computed to define the contrast threshold value of the SIFT algorithm:

$$Tr = k \cdot \frac{\sum_{i,j=0}^{DimX,DimY} \left| I(x_i,y_j) - \bar{I}(x_i,y_j) \right|}{DimX \cdot DimY} \quad (1)$$

where k is a scale factor, DimX and DimY are the x and y image dimensions, I(x,y) is the intensity of the gray level image and (x,y) is the medium intensity value of the processed image. In the Lowe's SIFT implementation the contrast threshold is statically defined and low contrast key points are rejected because they are sensitive to noise. In our implementation this threshold is computed for each sub-image, sometime avoiding

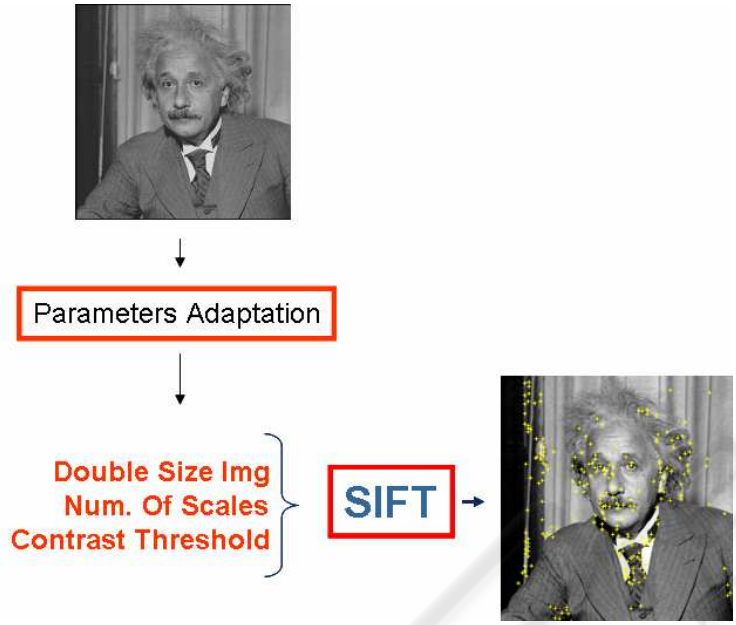

Fig. 1. Adaptive SIFT approach. Feature extraction parameters are adapted to the processed image and then the SIFT algorithm is performed.

Table 1. Average number of matched features between original images and dark images and average computational time.

	Matched features	Computational time (sec)
Lowe's SIFT	13	1,3
Improved SIFT	94	1,1

at all the time-consuming feature extraction process and in any case allowing to deal with different lighting conditions. Further details about this approach can be found in [8]. Results have been obtained using a data set of outdoor images. We artificially performed lighting and contrast variation to every image. Here below a comparison of the proposed feature extraction process with the classical Lowe's SIFT is reported. Figure 2 reports an example of feature matching between an original and a dark image.

Results showed in Tab. 1 demonstrate better performances obtained using the proposed approach. The key results of the experimental comparison are that the average number of matched features is drastically higher and the computational time of the feature extraction is lower than the standard SIFT implementation. We also want to reduce the number of key points and their corresponding extraction and matching time, while maintaining the same descriptor for each key point.
In the classical SIFT approach, key points are detected by testing each value in the DoG at each scale with the 8 surrounding values of the same scale as well as with 9 neighbouring values in the scale above and 9 neighbouring values in the scale below. The

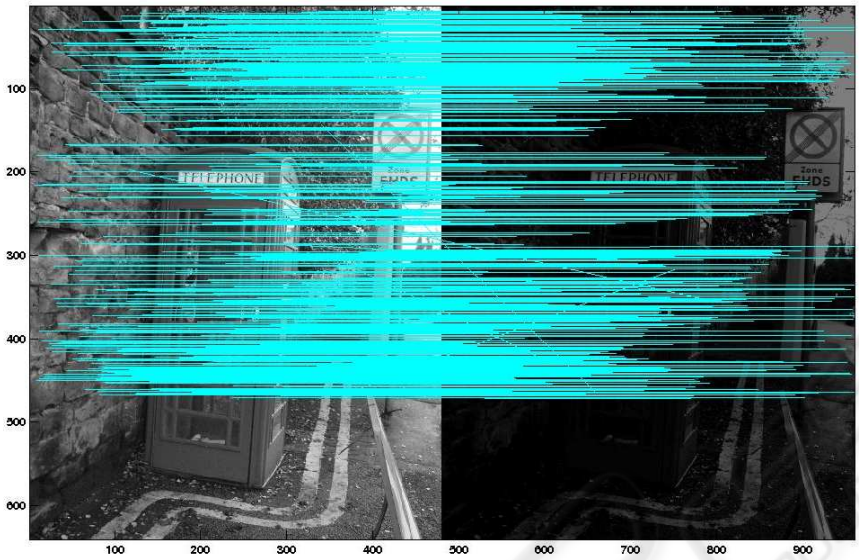

Fig. 2. An example of matched features between original and dark image. In this case 174 features are matched.

first and last DoG scales are not examined. This means 26mn comparisons for a DoG of size m n, taking into consideration that points around a given border of each DoG are not included in the key point detection.

Since SIFT establishes multiple scales in each octave, the above analysis is applied several times to each scale in each octave. Each octave has one quarter of the pixels of the previous one, so that key point detection in lower octaves requires more time than in higher ones. We aim to modify this exhaustive search into a sample based one.

In the proposed approach, the number of key points can be defined in advance. The process of finding the key points continues iteratively without the need for sequentially going through the whole scale space. This involves two phases. The first phase consists in randomly searching the scale space for local extrema. The random search is followed by an update phase only when the local extremum is more likely to be found. The theory behind this approach is mainly based on the assumption that local extrema points are located in a blob region, i.e. smooth wide two dimensional hills or valleys. In other words, blobs are regions in the image that are either significantly brighter or significantly darker than their surroundings. A local extremum cannot be located on a flat region and can hardly be found near it. Another possible location of local extrema are spikes, i.e. rapidly changing narrow regions. But since the scale space structure involves multiple smoothing operations on the image, only information on the coarse scale remains and the spikes are filtered out. With the above assumption we can say that our search mechanism involves dealing with only two cases when searching for a local extremum: detection or not of a blob region. When a blob region is detected an update phase handles the search for the position of the local extremum in that region.

The search ends either when the local extremum is found or when a given number of trials elapses. On the contrary, when a non-blob region is detected, the result of the search in this area is ignored and the search is started somewhere else.
More in particular, we first initialize, for each scale, a set of candidate key points (samples) by selecting random couples of numbers, each representing the coordinates of one point in the image. The samples are then verified and only those that have a value above the given threshold will be considered stable key points. This reflects our assumption that a value above the threshold is most probably a point that lies in a blob. A similar approach was introduced in [9]. The number of matched key points can be defined in advance and the computation time will result proportional to that number.

3 Application to Mobile Robotics

The application to mobile robotics is in the localization task. These features have been widely used in the robot localization field as well as many other computer vision fields. Here following we will apply the proposed approach to the vision based localization of a mobile robot in an indoor environment. Figure 3 reports the map of the used environment, which mainly consists of very similar corridors and offices. This makes the robot localization more difficult, due to perceptual aliasing (different places that look very similar).

Monte Carlo Localization (MCL) is the method used for estimating the position of the robot and the probability function for the robot position is approximated using a particle filter.
The generation of the weight that will be associated to each particle is a crucial aspect of the localization algorithm, as the robot position depends directly from particle weights. Each particle weight is assigned according to the difference between the actual sensor reading of the robot and the sensor reading that a robot would obtain from the position of the particle.
The MCL weight-update phase needs to consider that we have reference images only for a small number of reference positions over the environment; so the weight update takes into account the similarity between the actual and the reference image and the distance of the particle from the reference image position. For each particle j at every time step k the MCL innovation factor can be computed as the similarity measure between the current observation o_k and the reference observation o_j nearest (according to the Euclidean distance) to the particle j:

$$Innov_k^j = \left| \frac{Similarity(o_k, o_j)_k^j}{d_k} \right| , j = 1...nParticles \tag{2}$$

In our case observations are images and their similarity is computed using the proposed adaptive SIFT approach, but with only 64 feature descriptors (on the contrary of 128). In particular, the similarity for the whole image is computed by dividing the total number of features matched between the two images by the number of features

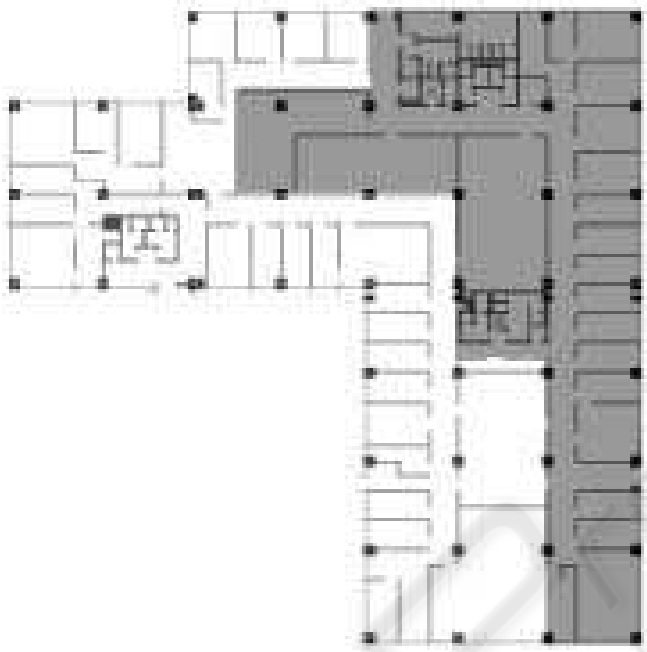

Fig. 3. Map of the environment.

extracted from the reference images. In the localization phase, the same set of SIFT features is extracted from the current observation, and for each particle the similarities with nearby nodes are interpolated to calculate the particle weight. These feature descriptors are then used to represent the environment appearance at the respective position and stored in a map. A preliminary visual tour is used to take some pictures of the environment and store their features and positions in a reference image database. The map is a graph of nodes, covering the two-dimensional environment, where each node contains the features extracted from the image at the respective position.
Two examples of feature extraction and matching between similar images are reported in Figure 5.

Results were obtained using an Active Media Pioneer3 DX robot, a differential drive robot with a high degree of mobility and the capability to climb over small obstacles. This robot is characterized by the following technical specifications: eight sonar sensors situated on the front part and characterized by a maximum range of 5 m and a visibility angle of 30; two incremental encoders; two wheels controlled by independent motors; serial connection RS232 and a low cost webcam.
Because of the partially random nature of Monte Carlo Localization, we executed 10 runs over the same data to receive significant results. The absolute position error for SIFT-based MCL, comparing the classical SIFT and the improved one is shown in Table 2. The computational time for every step is of about 0.4 seconds, with respect to the 1.4 needed for the classical SIFT approach.

Fig. 4. Sample images of the test environment.

Table 2. Results of visual MCL experiment. The table shows the difference between the classical SIFT and the proposed one.

	MCL ERROR (mm)	Time (sec)
Enanched SIFT	1110	0.4
SIFT	1070	1.4

4 Conclusions

In these experiments we tested a practical idea to improve and speed up the SIFT approach. The number of matched key points can be defined in advance and the computation time is proportional to that number. We also introduced the idea of parameter adaptation to avoid feature extraction from uniform regions of an image and to deal well with lightness changes. We applied the approach to a data set of outdoor images and we demonstrated that this approach is suitable since we need to deal with lightness changes. Even if results are preliminary the general idea applied to mobile robotics gives comfortable performances. The approach can be generally applied to any similar problem

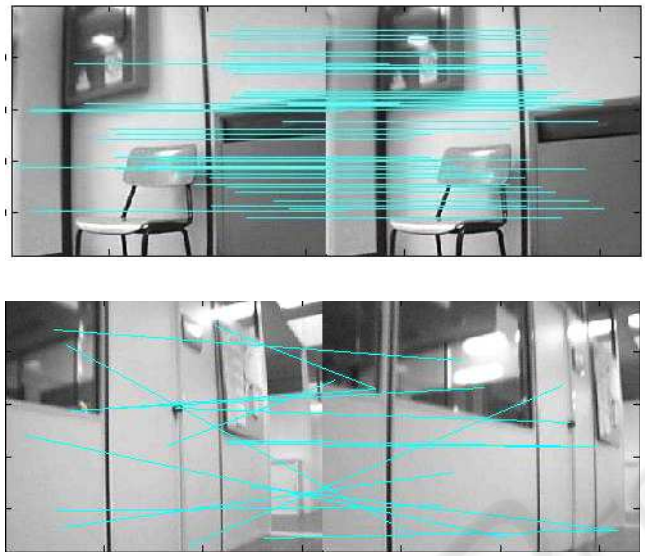

Fig. 5. Examples of feature extraction and matching between similar images.

and we plan to perform tests of mobile robot localization in outdoor environments. It is obvious that any further optimization to the original SIFT approach, such as in the key point descriptor or orientation assignment, may also be applied to this approach.

References

1. T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales." *Journal of Applied Statistics*, vol. 21, no. 2, pp. 224–270, 1994.
2. K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of International Conference on Computer Vision*, July 2001, pp. 525–531.
3. ——, "A performance evaluation of local descriptors," in *Proceedings of Computer Vision and Pattern Recognition*, June 2003.
4. C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors." *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.
5. C. Harris and M. Stephens, "A combined corner and edge detector." in *Proceedings of the Fourth Alvey Vision Conference*, Manchester, UK, 1988, pp. 148–151.
6. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
7. S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2001*, Seoul, Korea, May 2001, pp. 2051–2058.
8. E. Frontoni and P. Zingaretti, "Feature extraction under variable lighting conditions." in *Proceeding of CISI06 - Conferenza Italiana sui Sustemi Intelligenti*, Ancona, Italy, September 2006.
9. H. Tamimi, H. Andreasson, A. Treptow, T. Duckett, and A. Zell, "Localization of mobile robots with omnidirectional vision using particle filter and iterative sift," in *Proceeding of the 2nd European Conference on Mobile Robots*, Ancona, Italy, September 2005, pp. 2–8.