Scale Invariant Feature Transform using Oriented Pattern

[1]Mohammad Baghery Daneshvar, [2]Massoud Babaie-Zadeh, [3]Seyed Ghorshi
[1,3]Sharif University of Technology, School of Science and Engineering, International Campus, Kish Island, Iran
[2]Sharif University of Technology, Department of Electrical Engineering, Tehran, Iran
[1] m_daneshvar@kish.sharif.edu, {[2]mbzadeh, [3]ghorshi}@sharif.edu

Abstract— **Image matching plays an important role in many aspects of computer vision. Our proposed method is based on Scale Invariant Feature Transform (SIFT) which is one of the popular image matching methods. The main ideas behind our method are removing the excess keypoints, adding oriented patterns to descriptor, and decreasing the size of the descriptors. By doing these changes to SIFT, we would have oriented patterns of keypoints. In addition, the numbers of keypoints have been reduced and the places of keypoints would be selected more accurately, and also the size of the descriptors has been reduced.**

Keywords— Image matching; keypoint; feature extraction; descriptor; oriented pattern.

I. Introduction

Our environments are filled with kinds of objects in their different poses and specification. The decision of the class detection should be irrelevant to these factors. However, we can add the illumination changes, background clutter and changing 3D viewpoint to these factors to make the process recondite. In this paper we focus on the matching phase of image classification process.

Two reliable methods have been developed for image matching: correlation based methods and feature based methods [1]. Correlation based methods involve all pixels of image but all of these pixels are grouped as certain sized windows. The algorithm computes the correlation of windows of new image and database's images. Recently many techniques have been developed for image matching where transformations are well known. Those techniques extract the features of the points of interest in the image. However, in correlation based methods, local descriptors are created to represent these features instead of image windows and then find the matching of these points in the database.

In the past few years, few region detecting approaches that are covariant to a class of transformations have been developed. First, Harris et. al [2] developed a derivative based detector for edge and corner detection by measuring the trace and determinant of the gradient distribution matrix around interest points. Mikolajczyk et. al [3, 4 , 5] used Harris et. al [2] function and Hessian Matrix to locate points in 2D and then obtain maxima as keypoints by Laplacian operator selection in multi-scale space. Regions detected by these methods were named Harris Laplace and Hessian Laplace regions. A similar idea was explored by Lowe [6, 7] who used Difference of Gaussian to approximate Laplacian of Gaussian. Lindeberg [8] designed a blob detector by using Laplacian of Gaussian and several other derivatives based operators. Also, [8] and [9] made the blob detector invariant to affine transformations by using affine adaptation process. Mikolajczyk et. al [4, 5] applied affine adaptation process to their Harris Laplace and Hessian Laplace detectors and created Harris Affine and Hessian Affine detectors that are affine invariant. Similar detectors were developed by Baumberg [10] and Schaffalitzky et. al [11]. In spite of using Gaussian derivatives, other detectors were developed based on edge and intensity extrema [8], entropy of pixel intensity [12] and stable extremal regions [13].

The rest of paper is organized as follows. In section 2 an improved SIFT using image oriented patterns (SIFTOP) is presented. Evaluation and experimental results are included in section 3 while section 4 concludes the paper.

II. An improved SIFT using image Oriented Patterns (SIFTOP)

In order to create SIFTOP the following changes should be added to SIFT.

1. In keypoint localization phase the excess keypoints need to be removed to decrease the time of the feature extraction process.
2. In feature description phase in addition to orientation histogram the oriented pattern is computed and must be added to the descriptor to improve the accuracy of the matching process.
3. In descriptor formation phase the size of the descriptor must be decreased to simplify the matching process.

The following sections describe the SIFTOP phases in detail.

A. Building Image Gaussian Pyramid

In this phase the algorithm identifies the keypoints which are stable with respect to image rotation, translation and those that are minimally affected by noise and small distortions. These keypoints are detected by comparing the points of different scales. In order to create Gaussian pyramid one needs to describe the Scale, Octave, and Difference of Gaussian according to [6].

Scale is defined in (1)

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y) \tag{1}$$

where * is 2D convolution operator, $I(x, y)$ is the input image, k is the Scale's order and

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

Octave is a set of Scales and to start each Octave the input image down sampled by factor of 2^n where n is the Octave's number and started by zero. Difference of Gaussian (DoG) is the difference between two Scales of an Octave. Our Gaussian pyramid has 4 Octaves and each Octave has 6 Scales and 5 DoGs.

B. *Keypoint Extraction*

If the DoG value of the under processed pixel (see Fig. 1) is strictly larger or smaller than the neighboring pixels' DoG values, the algorithm selects the pixel as a candidate keypoint [6]. The algorithm will remove candidate keypoints by thresholding the DoG values. In our experiments the candidate keypoints whose DoG values is smaller than 0.002 will be removed.

According to [6], for stability reasons, it is not sufficient to remove the low contrast candidate keypoints. In this case the algorithm should remove the candidate keypoints in the edge regions. To do this the following thresholding using Hessian matrix should be applied.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{3}$$

$$Tr(H) = D_{xx} + D_{yy} \tag{4}$$

$$Det(H) = D_{xx}D_{yy} - \left(D_{xy}\right)^2 \tag{5}$$

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{6}$$

where $r = 10$ for our exponents. It is noted that the SIFT keypoints have no limit in Euclidean distance [6]. In our proposed method if two or more keypoins are in a 3×3 local window, the keypoint which has the maximum DoG value will remain and the other will be removed. As the result, the excess keypoints will be removed. Executing this phase before the thresholding phase and eliminating edge responses phase may lead to having no keypoint in 3×3 under processed local window. Therefore, localization phase has been executed after the thresholding phase and eliminating edge responses phase.

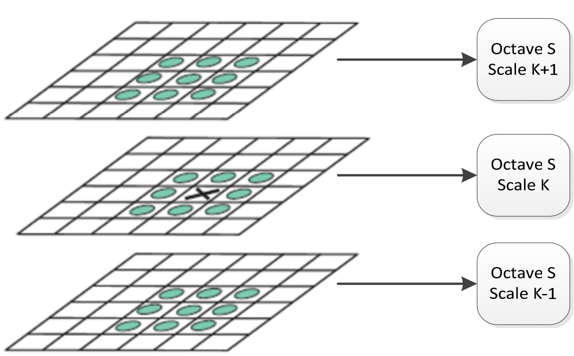

Fig. 1. Extrema detection, under processed pixel is shown by × and the neighboring pixels are shown by circles.

C. *Orientation Assignment*

The magnitude of gradient and orientation is defined in the following equations.

$$Magnitude = sqrt([I(x+1, y) - (x-1, y)]^2 + [I(x, y+1) - (x, y-1)]^2) \tag{7}$$

$$Orientation = tan^{-1}\left(\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}\right) \tag{8}$$

Fig. 2 illustrates the proposed method for orientation assignment. The orientation assignment will be executed faster using the proposed method described in Fig. 2.

D. *Orientation Histogram*

SIFTOP calculates dominant orientation. The following tasks are executed in each keypoints in this phase. The magnitudes in a 15×15 local window, around the keypoints are weighted by a 15×15 Gaussian window. Then, the dominant orientation is obtained by using the vector sum of the weighted Gradient vector as shown in (9).

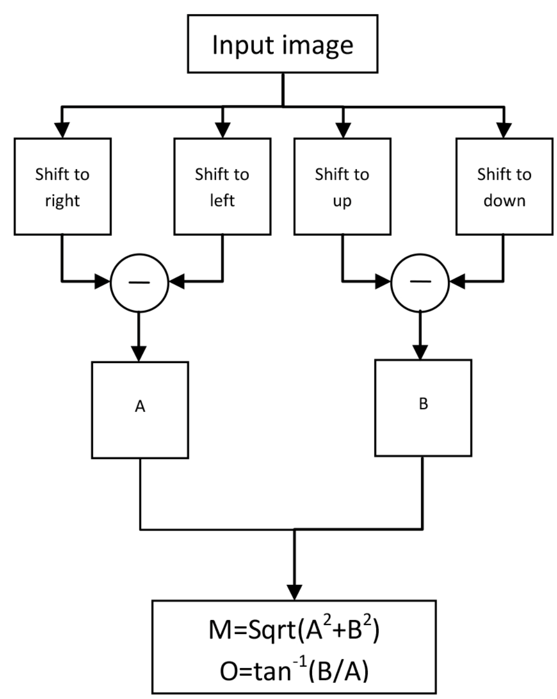

Fig. 2. The proposed method for orientation assignment. The magnitude and orientation of each pixel are computed after this phase.

$$\varphi = tan^{-1} \frac{\sum_{x=1}^{15}\sum_{y=1}^{15}(sin(M*O))}{\sum_{x=1}^{15}\sum_{y=1}^{15}(cos(M*O))} \tag{9}$$

where M is the weighted magnitude of pixel, O is the orientation of pixel and, x and y are the pixel location in 15×15 local window. When this task is done the algorithm rotates all orientation by the dominant orientation clockwise. Then, a histogram has been created by 8 bin orientation histogram which covers the 360 degrees, 45 degrees for each bin, and the magnitudes of each bin is the weighted magnitude of 15×15 local windows. Finally, the algorithm normalizes the obtained orientation histogram.

E. Oriented Pattern Extraction

In this section the oriented pattern extraction will be explained. In this phase the oriented pattern extracted in keypoint and the size of the pattern is 7×7.

At first SIFTOP creates a 7×7 logical mask, and sets all values to one. Then the algorithm rotates the mask by the dominant orientation counterclockwise. Because the size of the mask is 7×7 and hence the rotation smaller than 20 degrees has no significant effect on the mask. Therefore, we should change the dominant orientation to one of the integer multiples of 20 $(0, 20, 40, \ldots, 340)$. The rotated mask multiplies to the region around the keypoint and the rotated pattern is extracted in this way. Now the rotated pattern should be rotated clockwise by obtained orientation in this phase. So the zero value around the 7×7 pattern window should be removed. Finally, SIFTOP normalizes the gray values of the extracted pattern. Fig. 3 illustrates the oriented pattern extraction step by step.

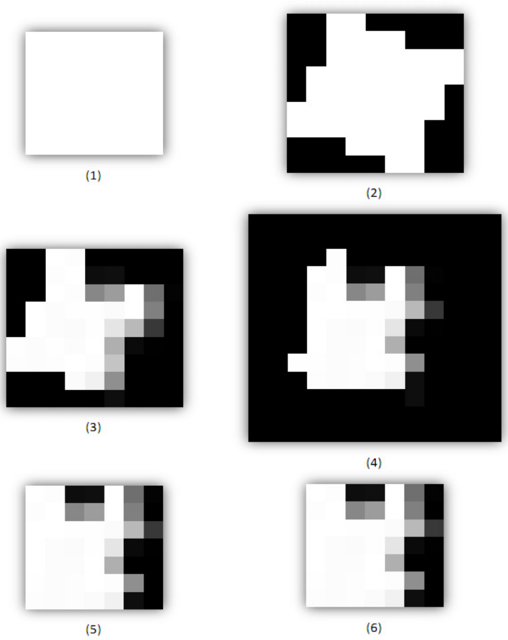

Fig. 3. Steps of pattern extraction.

1...49 Oriented Pattern	50...121 Orientation Histograms

Fig. 4. SIFTOP feature vector format.

F. Descriptor

Keypoint descriptor is the final output of SIFTOP. To create the descriptor, first the algorithm divides the 15×15 local window around the keypoint to 9, 5×5 windows. Then execute the orientation histogram phase for each 5×5 window. In this way, we would have 9, 8 bin histograms (72 values). At last the oriented pattern and orientation histograms are stored as feature vector as shown in Fig. 4.

III. Evaluation

The evaluation is carried out on real images under different transformations, including rotation, scaling, illumination change, additive Gaussian noise, and blurring. For every catalogue, a sequence of images is taken in the range from small image transformations to large ones and the transformations are significant enough to illustrate the features of SIFTOP. The details about how these image sets are created will be explained below.

To create a rotation image set, the picture of the object is taken by rotating the camera every 18 degrees, from 0 to 180 degrees. Scaling, illumination change, additive Gaussian noise, and blurring are created by using of MATLAB transformations.

We select 4 different finger prints for our objects and called them f1 to f4. Then we create different image sets by rotation, scaling, illumination change, additive Gaussian noise, and blurring image sets for each of these finger prints. After that we select f1 for our target object and execute our testing matching algorithm using Least Squares Error to find the matched object. In all tests SIFTOP can find the target object 100% correctly. Finally, we compute the correct matched keypoints percentage of target image and image sets as a correct ratio (CR) as (10)

$$CR = \frac{number\ of\ correct\ matched\ keypoints}{number\ of\ total\ keypoints} \times 100 \tag{10}$$

The results of our tests are shown in Figs. 5 to 9.

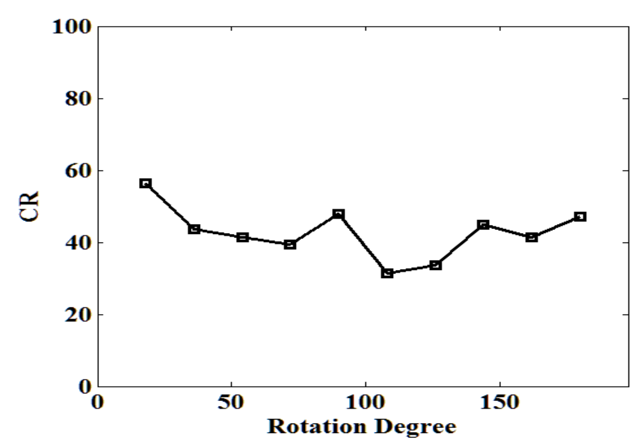

Fig. 5. Correct ratio per Rotation Degree.

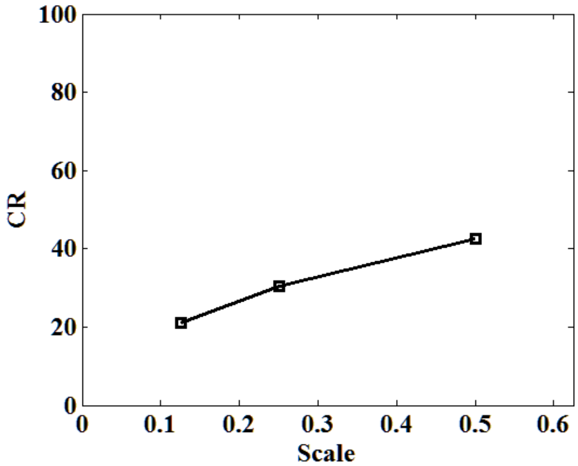

Fig. 6. Correct ratio per Scale.

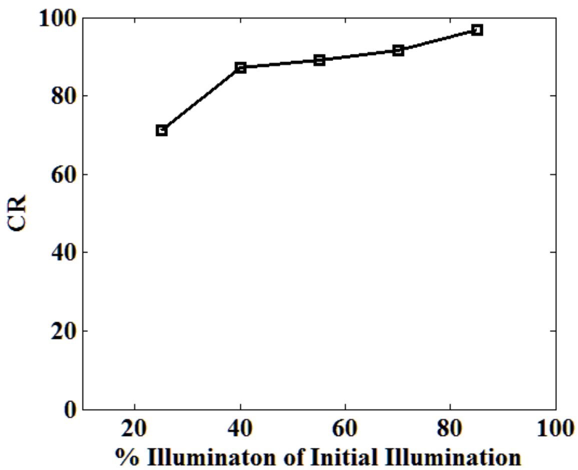

Fig. 7. Correct ratio per Illumination Changes.

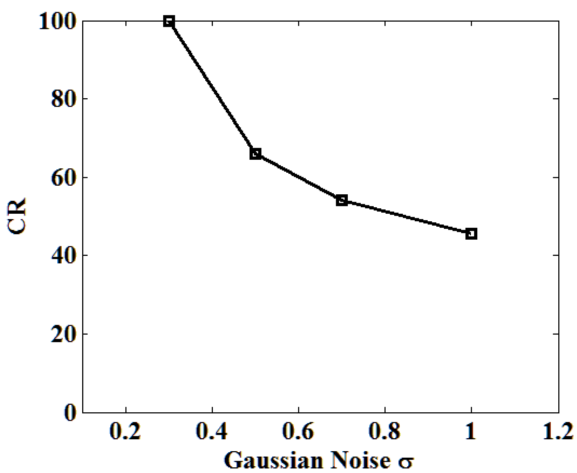

Fig. 8. Correct ratio per Gaussian Noise Sigma.

As it is shown in Figs. 5 to 9, SIFTOP correct ratio decreases by increasing the transformation range. It is happened because any transformations can change the keypoint's locations and patterns.

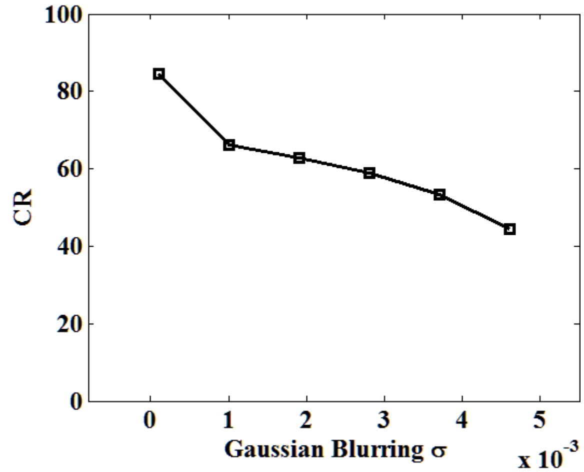

Fig. 9. Correct ratio per Gaussian Blurring Sigma

In addition, we have also developed an object recognition system to demonstrate the distinctive characteristics of SIFTOP in application, such as finger print recognition. As it is shown in Fig. 10(a), we select fingerprint of person 1 (P1) as target object. Then, we run the SIFTOP on this particular object. In Fig. 10(b) all 219 keypoints of P1 are shown. As it can be seen from Fig. 10(a) some of these keypoints cannot find the correct match (26 incorrect matched keypoints). Therefore, it is noted that the accuracy of detecting the correct object (P1) is almost 88%.

IV. Conclusion

Classification process has three major steps: feature extraction, image matching, classifying. We focused on the feature extraction step and proposed a feature extraction algorithm called SIFTOP. This algorithm is based on SIFT which is one of the popular feature extraction methods.

For classifying a new object, at first SIFTOP extracts feature vector of the new object. Then, the algorithm tries to find the matched feature vector in the database, so the feature extraction step is executed once and the matching step iterates over all data in database. Based on this method decreasing the time of the matching step is more efficient than decreasing the time of feature extraction step. Therefore, we changed SIFT to decrease the time and increase the accuracy of the matching step. SIFTOP can make these goals by adding the following changes to SIFT. SIFTOP removes excess keypoints in

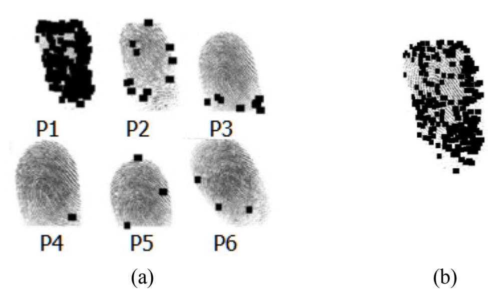

Fig. 10. SIFTOP Object recognition test.

localization phase, this task decreases the number of keypoints, so the matching step can execute faster. SIFTOP extracts the oriented pattern around the keypoints to increase the accuracy of the matching step. SIFTOP changes the descriptor formation phase. This can decrease the size of the feature vector, and the execution time of the matching step.

REFERENCES

[1] S. G. Domadia and T. Zaveri, "Comparative Analysis of Unsupervised and Supervised Image Classification Techniques," in Proceeding of National Conference on Recent Trends in Engineering & Technology, , pp 1-5, 2011.

[2] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in Alvey Vision Conference, pp. 147-151, 1988.

[3] K. Mikolajczyk, C. Schmid. "Indexing Based on Scale Invariant Interest Points," in Proceedings of the 8th International Conference of Computer Vision, Vancouver, Canada, pp. 525-531, 2001.

[4] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors," International Journal of Computer Vision, Vol. 1, No. 60, pp. 63-86, 2004.

[5] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," in Proceeding of 7th European Conference of Computer Vision, Copenhagen, Denmark, Vol. 1, No. 1, pp. 128-142, 2002.

[6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Interest Point Detectors," International Journal of Computer Vision, Vol. 2, No. 60, pp. 91-110, 2004.

[7] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in Proceeding of the 7th International Conference of Computer Vision, Kerkyra, Greece, pp. 1150-1157, 1999.

[8] T. Lindeberg, "Feature Detection with Automatic Scale Selection," International Journal of Computer Vision, Vol. 30, No. 2, pp. 279-116, 1998.

[9] T. Linderberg and J. Garding, "Shape-Adapted Smoothing in Estimation of 3-D Shape Cues from Affine Deformations of Local 2-D Brightness Structure," International Journal of Image and Vision Computing, Vol. 15, No. 6, pp. 415-434, 1997.

[10] A. Baumberg. "Reliable Feature Matching across Widely Separated Views," in Proceeding of Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA, pp. 774-781, 2000.

[11] F. Schaffalitzky and A. Zisseman. "Multi-View Matching for Unordered Image Sets," in Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, pp. 414-431, 2002.

[12] T. Kadir, M. Brady, and A. Zisserman, "An Affine Invariant Method for Selecting Salient Regions in Images," in Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic, pp. 345-457, 2004.

[13] J. Matas, O. Chum, M. Urban, and Y. Pajdle, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," in Proceedings of the 13th British Machine Vision Conference, Cardiff, UK, pp. 384-393, 2002.