A Comparison of SIFT, PCA-SIFT and SURF

Luo Juan qiuhehappy@hotmail.com
Computer Graphics Lab,
Chonbuk National University,
Jeonju 561-756, South Korea

Oubong Gwun obgwun@chonbuk.ac.kr
Computer Graphics Lab,
Chonbuk National University,
Jeonju 561-756, South Korea

Abstract

This paper summarizes the three robust feature detection methods: Scale Invariant Feature Transform (SIFT), Principal Component Analysis (PCA)–SIFT and Speeded Up Robust Features (SURF). This paper uses KNN (K-Nearest Neighbor) and Random Sample Consensus (RANSAC) to the three methods in order to analyze the results of the methods' application in recognition. KNN is used to find the matches, and RANSAC to reject inconsistent matches from which the inliers can take as correct matches. The performance of the robust feature detection methods are compared for scale changes, rotation, blur, illumination changes and affine transformations. All the experiments use repeatability measurement and the number of correct matches for the evaluation measurements. SIFT presents its stability in most situations although it's slow. SURF is the fastest one with good performance as the same as SIFT. PCA-SIFT show its advantages in rotation and illumination changes.

Keywords: SIFT, PCA-SIFT, SURF, KNN, RANSAC, robust detectors.

1. INTRODUCTION

Lowe (2004) presented SIFT for extracting distinctive invariant features from images that can be invariant to image scale and rotation. Then it was widely used in image mosaic, recognition, retrieval and etc. After Lowe, Ke and Sukthankar used PCA to normalize gradient patch instead of histograms [2]. They showed that PCA-based local descriptors were also distinctive and robust to image deformations. But the methods of extracting robust features were still very slow. Bay and Tuytelaars (2006) speeded up robust features and used integral images for image convolutions and Fast-Hessian detector [3]. Their experiments turned out that it was faster and it works well.

There are also many other feature detection methods; edge detection, corner detection and etc. Different method has its own advantages. This paper focuses on three robust feature detection methods which are invariant to image transformation or distortion. Furthermore, it applies the three methods in recognition and compares the recognition results by using KNN and RANSAC methods. To give an equality comparison, use the same KNN and RANSAC to the three methods. In the experiment, we use repeatability measurement to evaluate the performance of detection for each method [4]; the higher repeatability score is better than the lower one. When a method gives a stable detector and matching numbers we can say that it is a stable method and if we want to know how correct the method is, we need to use correct matches number that can be get from the RANSAC method.

The related work is presented in Section 2 while Section 3 discusses the overview of the method. In section 4 we can see the experiments and results. Section 5 tells the conclusions and future work of the paper.

2. RELATED WORK

In [1], Lowe did not only presented SIFT but also discussed the keypoint matching which is also needed to find the nearest neighbor. He gave an effective measurement to choose the neighbor which is obtained by comparing the distance of the closest neighbor to the second-closest neighbor. In my experiment compromising of the cost and match performance, the neighbor will be chosen when the distance ratio is smaller than 0.5 [1]. All the three methods use the same RANSAC model and parameters, which will explain more in the following.
K. Mikolajczyk and C. Schmid [6], compared the performance of many local descriptors which used recall and precision as the evaluation criterion. They gave experiments of comparison for affine transformations, scale changes, rotation, blur, compression, and illumination changes. In [7], they showed how to compute the repeatability measurement of affine region detectors also in [4] the image was characterized by a set of scale invariant points for indexing.

Some researches focused on the application of algorithms such as automatic image mosaic technique based on SIFT [9][11], stitching application of SIFT [10][15][12] and Traffic sign recognition based on SIFT [12]. Y. Ke [2] gave some comparisons of SIFT and PCA-SIFT. PCA is well-suited to represents keypoint patches but observed to be sensitive to the registration error. In [3], the author used Fast-Hessian detector which is faster and better than Hessian detector. Section 3 will show more details of the three methods and their differences.

3. OVERVIEW OF THE THREE METHODS

3.1 SIFT detector

SIFT consists of four major stages: *scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor.* The first stage used difference-of-Gaussian function to identify potential interest points [1], which were invariant to scale and orientation. DOG was used instead of Gaussian to improve the computation speed [1].

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \qquad (1)$$

In the keypoint localization step, they rejected the low contrast points and eliminated the edge response. Hessian matrix was used to compute the principal curvatures and eliminate the keypoints that have a ratio between the principal curvatures greater than the ratio. An orientation histogram was formed from the gradient orientations of sample points within a region around the keypoint in order to get an orientation assignment [1]. According to the paper's experiments, the best results were achieved with a 4 x 4 array of histograms with 8 orientation bins in each. So the descriptor of SIFT that was used is 4 x 4 x 8 = 128 dimensions.

3.2 PCA-SIFT detector

PCA is a standard technique for dimensionality reduction [2], which is well-suited to represent the keypoint patches and enables us to linearly-project high-dimensional samples into a low-dimensional feature space.

In other words, PCA-SIFT uses PCA instead of histogram to normalize gradient patch [2]. The feature vector is significantly smaller than the standard SIFT feature vector, and it can be used with the same matching algorithms. PCA-SIFT, like SIFT, also used Euclidean distance to determine whether the two vectors correspond to the same keypoint in different images. In PCA-SIFT, the input vector is created by concatenation of the horizontal and vertical gradient maps for the 41x41 patch centered to the keypoint, which has 2x39x39=3042 elements [2]. According to PCA-SIFT, fewer components requires less storage and will be resulting to a faster matching, they choose the dimensionality of the feature space, n = 20, which results to significant space benefits [2].

3.3 SURF detector

SIFT and SURF algorithms employ slightly different ways of detecting features [9]. SIFT builds an image pyramids, filtering each layer with Gaussians of increasing sigma values and taking the difference. On the other hand, SURF creates a “stack” without 2:1 down sampling for higher levels in the pyramid resulting in images of the same resolution [9]. Due to the use of integral images, SURF filters the stack using a box filter approximation of second-order Gaussian partial derivatives, since integral images allow the computation of rectangular box filters in near constant time [3].

In keypoint matching step, the nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. Lowe used a more effective measurement that obtained by comparing the distance of the closest neighbor to that second-closest neighbor [1] so the author of this paper decided to choose 0.5 as distance ratio like Lowe did in SIFT.

4. EXPERIMENTS & RESULTS

4.1 Evaluation measurement

The repeatability measurement is computed as a ratio between the number of point-to-point correspondences that can be established for detected points and the mean number of points detected in two images [4]:

$$r_{1,2} = \frac{C(I_1, I_2)}{mean(m_1, m_2)} \tag{2}$$

Where $C(I_1, I_2)$ denotes the number of corresponding couples, m_1 and m_2 means the numbers of the detector. This measurement represents the performance of finding matches.

Another evaluation measurement is RANSAC, which is used to reject inconsistent matches. The inlier is a point that has a correct match in the input image. Our goal is to obtain the inliers and reject outliers in the same time [4]. The probability that the algorithm never selects a set of m points which all are inliers is $1-p$:

$$1-p = (1-w^m)^k \tag{3}$$

Where m is the least number of points that needed for estimating a model, k is the number of samples required and w is the probability that the RANSAC algorithm selects inliers from the input data. The RANSAC repeatedly guess a set of mode of correspondences that are drawn randomly from the input set. We can think the inliers as the correct match numbers. In the following experiments, matches mean inliers.

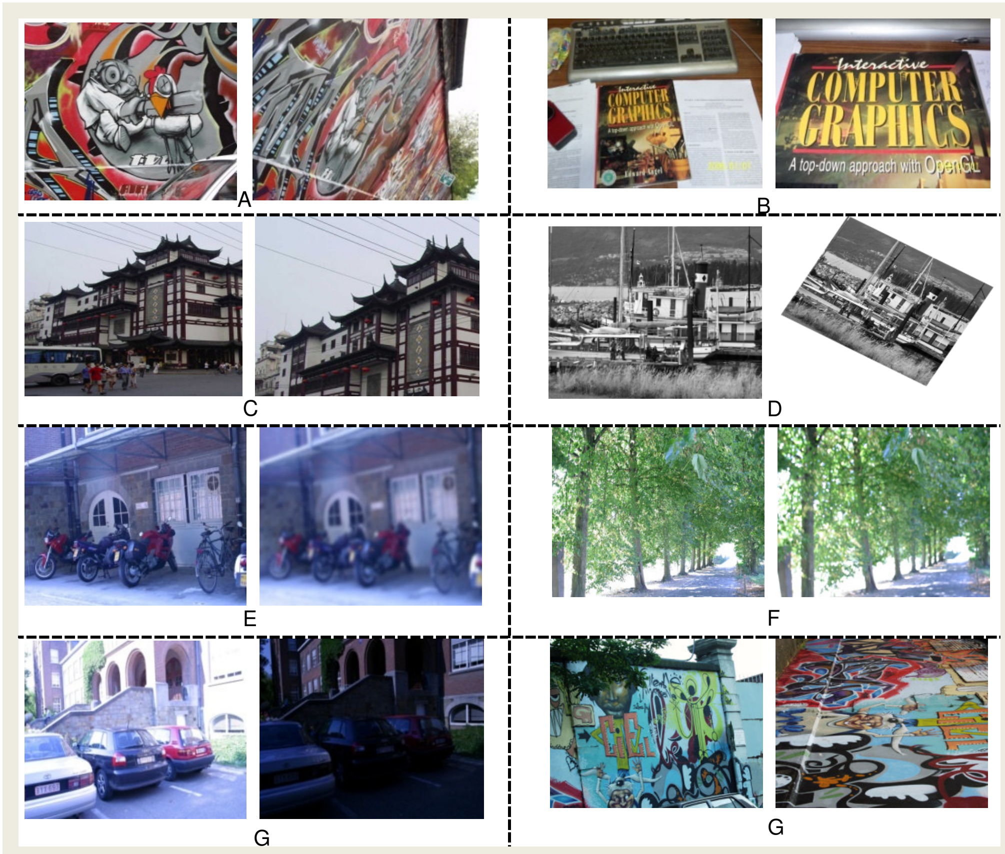

FIGURE 1: Part of test images. A and H are the affine transformed images, B and C are the scale changed images, D are the rotation images, E and F are the blurred images, G are the illumination changed images.

In this paper, the three methods that were used are all based on opencv. We use the same image dataset, which includes the general deformations, such as scale changes, view changes, illumination changes and rotation. As shown in figure 1. All the experiments work on PC AMD 3200+, 2.0G, and 1.0 GB RAM, with Windows XP as an operating system.

4.2 Processing Time

Time evaluation is a relative result, which only shows the tendency of the three methods' time cost. There are factors that influenced on the results such as the size and quality of the image, image types (e.g. scenery or texture), and the parameters of the algorithm (e.g. the distance ratio) [1]. The first part of the experiment uses Graffiti dataset as shown in group A of figure 1, whose sizes are all 300 x 240 pixels. The parameters of the three algorithms are the same settings according to the original paper [1] [2] [3]. Time is counted for the complete processing which includes feature detecting and matching. Table 1 show that SURF is the fastest one, SIFT is the slowest but it finds most matches.

Items	SIFT	PCA-SIFT	SURF
total matches	271	18	186
total time (ms)	2.15378e+007	2.13969e+007	3362.86
10 matches' time(ms)	2.14806e+007	2.09696e+007	3304.97

TABLE 1: Processing time comparison. Using group A of figure 1, total time is the time of finding all the matches, 10 matches' time is counted until the first 10 matches.

4.3 Scale Changes

The second experiment shows the performance of scale invariant, which uses group B and C of figure 1. Figure 2 shows some of the matching images, table 2 shows the matching number of the three methods. In order to see the matches clearly, we just show the first 10 matches [2]. When the scale change gets larger, SIFT or SURF is much better than PCA-SIFT, therefore the results shows that PCA-SIFT is not stable as SIFT and SURF to scale invariant and PCA-SIFT detects few matches.

FIGURE 2: Scale changes comparison. Using Group B, C of figure 1. The first result is SIFT (10/66matches), second is SURF (10/26matches). The last two images show the results of PCA-SIFT in scale changes.

Data	SIFT	PCA-SIFT	SURF
1-2	41	1	10
3-4	35	0	36
5-6	495	19	298
7-8	303	85	418

TABLE 2: Scale changes comparison. Using group B, C of figure 1, data represents the total number of matches for each method.

4.4 Image Rotation

The third experiment shows the influence of rotation on the three methods. As shown in group D of figure 1, the image rotates 5 or 10 degrees. SIFT is represented by the blue line that detects the most matches and stable to rotation. SURF doesn't work well, it finds the least matches and gets the least repeatability which also shown in figure 3 and 4. In addition with, PCA-SIFT found only one correct match of the first ten matches. Although it has one correct match, we can still improve the PCA-SIFT and this is better than SURF.

4.5 Image Blur

This fourth experiment uses Gaussian blur like the images in group E and F of figure 1. The radius of the blur changes from 0.5 to 9.0. As shown in figure 5, SURF and PCA-SIFT shows good performance, when the radius gets larger, mention the matching results in figure 6, SURF detects few matches and PCA-SIFT finds more matches but smaller correct number. In this case, SIFT shows its best performance here.

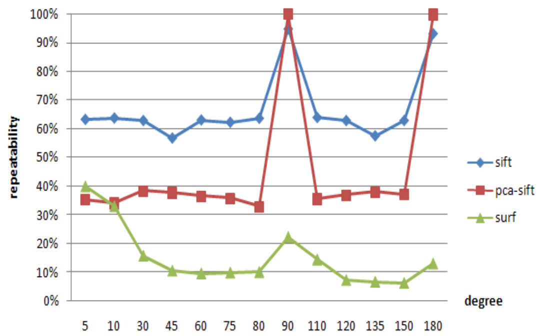

FIGURE 3: Rotation comparison. Using group D of figure 1, data represents the repeatability of rotation.

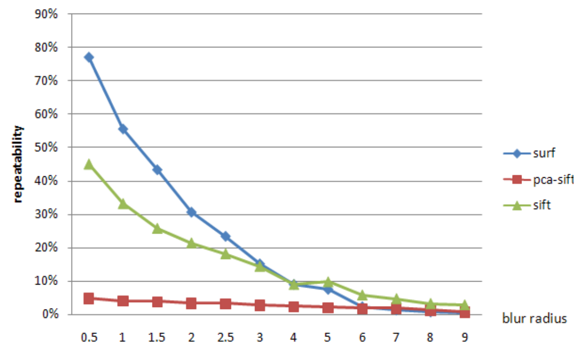

FIGURE 5: Blur comparison. Using group E of figure 1, data represents the repeatability of blur.

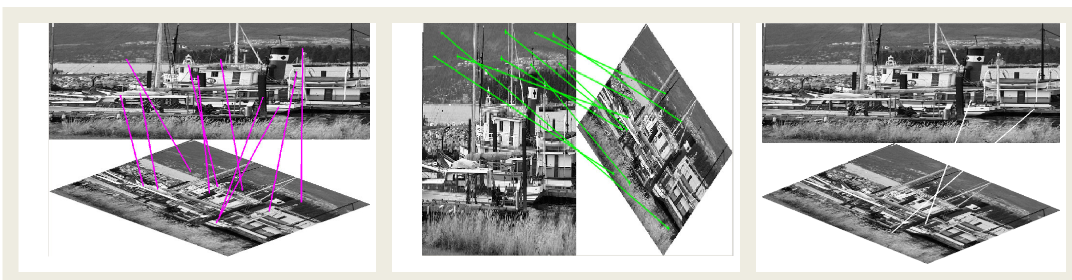

FIGURE 4: Rotation comparison. Using group D of figure 1, rotation degree is 45. The first two images show the first ten matches of SIFT (10 correct / 10) and PCA-SIFT (1 correct / 10), the result of SURF (2 correct / 2) is the total matches.

FIGURE 6: Blur comparison. Using group E and F of figure 1. From left to right, the results are: SIFT (10 correct / 10), SURF (3 correct / 3), PCA-SIFT (1 correct / 10), which blur radius is 9.

4.6 Illumination Changes

This fifth experiment shows the illumination effects of the methods. As shown in group G of figure 1, from data 1 to data 6, the brightness of the image gets lower and lower. Table 3 shows the repeatability of illumination changes. SURF has the largest repeatability, 31%, PCA-SIFT shows as good performance as SURF, which coincides with [2][3][6]. Look at one of the experiment result in figure 7, which shows the first 10 matches.

FIGURE 7: Illumination changes comparison. Using group G of figure 1, from left to right, SIFT (9 correct / 10), PCA-SIFT (6 correct / 10), SURF (10correct/10).

Data	SIFT	PCA-SIFT	SURF
1-2	43%	39%	70%
1-3	32%	34%	49%
1-4	18%	27%	25%
1-5	8%	18%	6%
1-6	2%	9%	5%
average	21%	25%	31%

TABLE 3: Illumination changes comparison. Using group G of figure 1, the data represents repeatability and the average of repeatability.

4.7 Affine Transformations

This sixth experiment evaluates the methods' stability of affine transformations (view transformation). Affine transformation is important in panorama stitching. This experiment uses images in group A and H of figure 1, the viewpoint change is approximately 50 degrees from data 1–6. Repeatability of affine transformation shows in table 4 and the matching results shown in figure 8. From the table and figure, SURF and SIFT have a good repeatability when the viewpoint change is small, but when the viewpoint change is larger than data 1–4, SURF detects 0 matches and PCA-SIFT shows better performance.

Data	SIFT	PCA-SIFT	SURF
1-2	47%	15%	54%
1-3	37%	12%	33%
1-4	22%	12%	12%
1-5	7%	10%	2%
1-6	0%	9%	0%

TABLE 4: Affine transformation comparison. Using group A, H of figure 1, the data represents the repeatability.

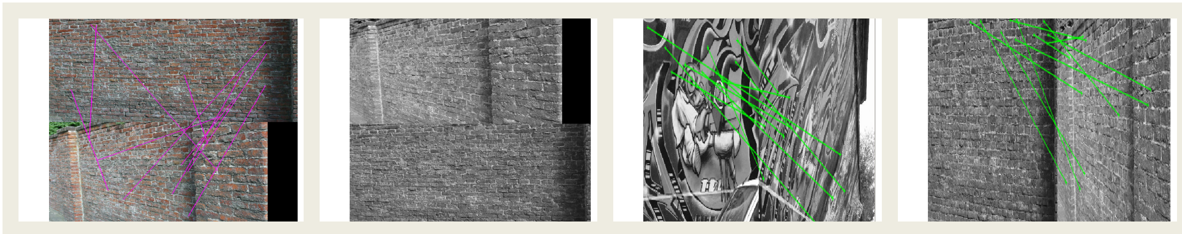

FIGURE 8: Affine transformations comparison. Using group A and H of figure 1, when overview change is large, the results shows from left to right: SIFT (5 correct / 10), SURF (0), PCA-SIFT (1), PCA-SIFT(2 / 10).

4.8 Discussion

Table 5 shows the results of all experiments. It also shows that there is no best method for all deformation. Hence, when choosing a feature detection method, make sure which most concerned performance is. The result of this experiment is not constant for all cases. Changes of an algorithm can get a new result, find the nearest neighbor instead of KNN or use an improved RANSAC.
SIFT's matching success attributes to that its feature representation has been carefully designed to be robust to localization error. As discussed by Y. Ke, PCA is known to be sensitive to registration error. Using a small number of dimensions provides significant benefits in storage space and matching speed [2]. SURF shows its stability and fast speed in the experiments. It is known that 'Fast-Hessian' detector that used in SURF is more than 3 times faster that DOG (which was used in SIFT) and 5 times faster than Hessian-Laplace [3]. From the following table, we know that PCA-SIFT need to improve blur and scale performances. SURF looks fast and good in most situations, but when the rotation is large, it also needs to improve this performance. SIFT shows its stability in all the experiments except for time, because it detects so many keypoints and finds so many matches, we can think about matching in some interest keypoints.

Method	Time	Scale	Rotation	Blur	Illumination	Affine
SIFT	common	best	best	best	common	good
PCA-SIFT	good	common	good	common	good	good
SURF	best	good	common	good	best	good

TABLE 5: Conclusion of all the experiments.

5 . CONCLUSIONS & FUTURE WORK

This paper has evaluated three feature detection methods for image deformation. SIFT is slow and not good at illumination changes, while it is invariant to rotation, scale changes and affine transformations.

SURF is fast and has good performance as the same as SIFT, but it is not stable to rotation and illumination changes. Choosing the method mainly depends on the application. Suppose the application concern is more than one performances compromising from the Table 5, choose a suitable algorithm and giving improvement according to the application. For example, PCA-SIFT is the best choice, but the application also concerns blur performance, so what needed is to give PCA-SIFT some improvement on blur performance. The future work is to improve the algorithm and apply these methods on single areas, such as image retrieval or stitching.

6. REFERENCES

1. **FOR JOURNALS**: D. Lowe.*"Distinctive Image Features from Scale-Invariant Keypoints", IJCV*, 60(2):91–110, 2004.

2. **FOR CONFERENCES**: Y. Ke and R. Sukthankar.PCA-SIFT: *"A More Distinctive Representation for Local Image Descriptors" ,Proc. Conf. Computer Vision and Pattern Recognition*, pp. 511-517, 2004.

3. **FOR CONFERENCES**: Bay,H,. Tuytelaars, T., &Van Gool, L.(2006). "*SURF: Speeded Up Robust Features*", *9th European Conference on Computer Vision*.

4. **FOR CONFERENCES:** K. Mikolajczyk and C. Schmid. "*Indexing Based on Scale Invariant Interest Points". Proc. Eighth Int'l Conf. Computer Vision*, pp. 525-531, 2001.

5. **FOR JOURNALS:** K.Kanatani. "*Geometric information criterion for model selection*", *IJCV, 26(3):*171-189,1998.

6. **FOR JOURNALS**: K. Mikolajzyk and C. Schmid. "*A Perforance Evaluation of Local Descriptors*", *IEEE,Trans. Pattern Analysis and Machine Intelligence*, vol.27, no.10, pp 1615-1630, October 2005.

7. **FOR JOURNALS:** K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool*." A Comparison of Affine Region Detectors*", *IJCV,* 65(1/2):43-72, 2005.

8. Eric Chu,Erin Hsu, Sandy Yu. *"Image-Guided Tours: Fast-Approximated SIFT with U-SURF Features*", Stanford University.

9. **FOR CONFERENCES**: Yang zhan-long and Guo bao-long. "*Image Mosaic Based On SIFT"*, International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp:1422-1425,2008.

10. **FOR CONFERENCES:** M. Brown and D. Lowe." *Recognizing Panoramas*". *Proc. Ninth Int'l Conf. Computer Vision*, pp. 1218-1227, 2003.

11. **FOR CONFERENCES:** Salgian, A.S. *"Using Multiple Patches for 3D Object Recognition***",** Computer Vision and Pattern Recognition, CVPR '07. pp:1-6, June 2007.

12. **FOR CONFERENCES:** Y. Heo, K. Lee, and S. Lee. "*Illumination and camera invariant stereo matching* ". In CVPR, pp:1–8, 2008.

13. **FOR SYMPOSIUM:** Kus, M.C.; Gokmen, M.; Etaner-Uyar, S. " *Traffic sign recognition using Scale Invariant Feature Transform and color classification*". ISCIS '08. pp: 1-6, Oct. 2008.

14. **FOR TRANSACTIONS**: Stokman, H; Gevers, T." *Selection and Fusion of Color Models for Image Feature Detection* ". Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 29, Issue 3, pp:371 – 381, March 2007.

15. **FOR CONFERENCES:** Cheng-Yuan Tang; Yi-Leh Wu; Maw-Kae Hor; Wen-Hung Wang. "Modified *sift descriptor for image matching under interference***".** Machine Learning and Cybernetics, 2008 International Conference on Volume 6, pp:3294 – 3300, July 2008.