Recognition of Affine Distorted Characters by Using Affine-Invariant Local Descriptors

Takuya KOBAYASHI†, Masakazu IWAMURA††, and Koichi KISE††

† School of Engineering, Osaka Prefecture University
†† Graduate School of Engineering, Osaka Prefecture University
Gakuen-cho 1-1, Naka, Sakai, Osaka, Japan
E-mail: †kobayashi@m.cs.osakafu-u.ac.jp, ††{masa, kise}@cs.osakafu-u.ac.jp

Abstract Camera-based character recognition has a possibility to realize a variety of applications and has been paid attention. However, character images are often taken from different angles and characters tend to be distorted. In such a case, recognition accuracy becomes worse. Therefore, we must find some methods to deal with this problem. In this paper, we experimentally evaluate some affine-invariant local descriptors by testing how well they can recognize affine distorted characters. As a result, affine-invariant local descriptors are robust to affine distorted characters to some extent.

Key words Character recognition , Affine-invariant , ASIFT , SIFT , Harris-Affine , Hessian-Affine

1. Introduction

Camera-based character recognition becomes more and more popular. This enables us to recognize characters automatically by using a portable camera. It has a possibility to realize a wide variety of applications. "Translation-camera" is a good example; in case you go abroad and find unknown words on a banner, you can take a picture of that and soon get the translated words. Such a system is very useful.

In order to realize the system, there are several problems. For example, character images are often taken from different angles and characters are distorted by projective transformation. In such a case, recognition accuracy becomes worse. Therefore, we must realize a robust recognition system for distorted characters.

In the field of object recognition, some local feature detectors and descriptors have been proposed. Feature detectors are used to extract discriminative features from an image. Extracted features are described as feature vectors by using a feature descriptor. By calculating the distance between each feature vector extracted from two different images, we can compute the similarity between two images. ASIFT [1], Harris-Affine and Hessian-Affine [2] are feature detectors and descriptors. The three methods are invariant to scale and affine transformation. They can recognize distorted images accurately. They utilize the fact that projective transformation can be approximated by affine transformation to some extent. However, whether they can recognize distorted characters is yet to be shown. Thus it is important to confirm the recognition accuracy of the methods for affine distorted characters.

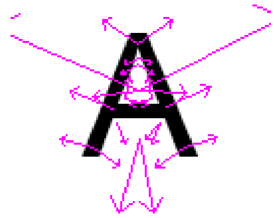

Figure 1 The result of feature detection by SIFT.

In this paper, we evaluate the robustness of the affine invariant features to affine distorted characters by testing how well these three methods can recognize them. Compared methods are above three and SIFT [3]. SIFT is also a local feature detector and descriptor but is not invariant to affine transformation.

2. Local feature detectors and descriptors

2.1 SIFT

Scale-Invariant Feature Transform (SIFT) is a local feature detector and descriptor proposed by David Lowe [3]. SIFT is used for matching of an image mosaic and object recognition [4]. The method extracts discriminative feature points from an image. The feature points are described as feature vectors by the SIFT descriptor. Figure 1 shows the result of feature detection. Extracted features are invariant to rotation, scale, and partially invariant to illumination and viewpoint

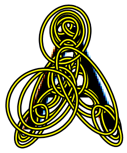

Figure 2 Circular regions extracted by Harris-Affine. Each region is invariant to scale and affine transformation.

changes. The process is mainly divided into two steps.

The first step is feature detection. In this step, SIFT decides the location and scale of feature points by scale-space extrema detection using Difference-of-Gaussian (DoG). Maximal or minimal pixel values are searched in the scale-space by comparing each pixel to its neighbors. Such points are treated as feature points. After that, we reject unstable feature points by considering the contrast of the image.

In the second step we obtain feature vectors computed from the feature points. SIFT calculates orientations of the feature points in order to normalize the direction of them. Then 128-dimensional feature vectors are computed based on the orientations. These feature vectors are normalized to the unit length in order to be invariant to change of illumination.

2.2 Harris-Affine

Harris-Affine is a local feature detector proposed by Mikolajczyk and Schmid [2]. The method is invariant to scale and affine transformation. In the method, corner points are extracted as feature points by using Harris corner detector. It utilizes a Harris matrix which considers the changes of image intensity. Corner points are searched through Gaussian scale-space in order to be invariant to scale changes. Then, a scale invariant region is computed from each corner point. The shape of the region is an ellipse. After that, each ellipse region is normalized to circle region by affine shape adaptation to be invariant to affine transformation. Figure 2 shows extracted regions which are invariant to scale and affine transformation. At last, from the information of the ellipses feature vectors are computed by using SIFT descriptor.

2.3 Hessian-Affine

Hessian-Affine is also a local feature detector proposed by Mikolajczyk and Schmid [2]. It consists of almost same process as Harris-Affine. The only difference from Harris-Affine is Hessian-Affine utilizes a Hessian matrix to compute corner points instead of using a Harris matrix.

2.4 ASIFT

ASIFT is an affine invariant local feature detector and descriptor proposed by Guoshen Yu et al. [1]. This method is based on SIFT and is invariant to affine transformation. While Harris-Affine and Hessian-Affine normalize all six parameters of affine transformation, this method simulates three out of six parameters. The three simulated parameters are the scale of an image and two axes that decide camera direction. The remaining three parameters, which decide rotation and translation are normalized by SIFT. In the process, first this method simulates the two axes of camera direction, and then simulates the scale and normalizes the translation of the camera parallel to its focal plane and the rotation of the camera around its optical axis by SIFT. At last, Feature vectors are described by SIFT descriptor.

3. Recognition methods

To recognize a query image, we compute the similarity between the query image and each reference image. The reference image which is most similar to the query image is the recognition result. We utilize the nearest neighbor search method to search for the most similar image. In order to compute the similarity between a query image and a reference image, we calculate the Euclidean distance between each feature vector extracted from the query image and that of the reference image. We apply the calculation to all the reference images. Before the process, we create a database which contains the feature vectors extracted from all the reference images. Each feature vector has an image ID of a reference image. Feature vectors extracted from the same image have the same image ID. After the calculation, for each feature vector of the query image we search for the feature vector from the database which has the shortest distance from each feature vector of the query image. Then, we cast a vote for the image ID of the nearest feature vector. We apply the voting process to all feature vectors of the query image. Finally, the image ID with the maximum votes is the recognition result.

4. Experiments

In order to evaluate the four methods above, we experimented how well they can recognize affine distorted characters. We employed 62 characters from numerals and alphabets; 10 figures, 26 capital alphabets, and 26 lowercase alphabets. The binary images of each character were treated as reference images. The font was "Arial" and the size was 60pt. Each reference image was put on the center of a white image whose sizes were 128×128 pixels. Since some characters were difficult to distinguish under affine distortions, the characters in a cell in Table 1 were treated as the same class in the experiments.

We prepared test images by applying various affine transformations to the reference images. For the sake of that, the affine transformation matrix $\boldsymbol{T} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ was decomposed into

(a) rotation (θ=9π/16) (b) independent scaling (α=3.0) (c) shearing (φ=-1.0)

Figure 3 Examples of affine distorted characters.

Table 1 List of similar characters. Characters in a cell were treated as the same class.

0 O o	6 9	I l	C c	S s	V v
W w	X x	N Z z	p d	q b	u n

$$T = L(\beta)R(\theta)S(\varphi)A(\alpha) \tag{1}$$

$$= \begin{pmatrix} \beta & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & \tan\varphi \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & 1/\alpha \end{pmatrix},$$

where

$$\alpha = \pm\sqrt{\frac{a^2+c^2}{ad-bc}}, \tag{2}$$

$$\varphi = \tan^{-1}\frac{ab+cd}{ad-bc}, \tag{3}$$

$$\theta = \cos^{-1}\frac{\pm a}{\sqrt{a^2+c^2}}, \tag{4}$$

$$\beta = \pm\sqrt{ad-bc}. \tag{5}$$

α, φ and θ represent the parameters about the independent scaling, the shearing and the rotation, respectively. Since β is the scale parameter, we changed the remaining three parameters α, φ and θ in the following ranges: $\alpha = \{1.0, 1.25, \cdots, 4.0\}$, $\varphi = \{-1.0, -0.9, \cdots, 1.0\}$, and $\theta = \{0, \pi/16, \cdots, \pi\}$. Only one parameter was changed at one time while the remaining two parameters were not changed. We applied this to all three parameters to create test images. Figure 3 shows examples of the test images.

Utilized source codes and binary files of the four methods are from the following websites. SIFT is from [5], Harris-Affine and Hessian-Affine are from [6] and ASIFT is from [7]. In the experiments, we did not measure the processing time of the four methods because the datasets of some methods were distributed only with binary files. Thus we could not measure the accurate recognition time through the whole process. Table 2 shows performance comparisons of the four methods. The number of feature points in the table were computed by extracting feature points from the character image of “0”. Since binary files of ASIFT could not extract features from small images, we magnified all the images three times in every experiment.

Table 2 Performance comparisons of the four methods. The number of feature points represents the number of extracted feature points from the character image of “0”.

	Affine transformation	Scale changes	The number of feature points
SIFT	×	○	20
Harris-Affine	○	○	8
Hessian-Affine	○	○	72
ASIFT	○	○	151

Table 3 Relationships between changes of the parameter of independent scaling and the number of feature points extracted by each method from the character image of “A”

	α=1.0	α=2.5	α=4.0
SIFT	28	8	4
Harris-Affine	76	14	5
Hessian-Affine	96	66	0
ASIFT	314	42	13

Figures 4 to 6 show the relationships between each parameter and the recognition rates of the methods. As shown in Fig. 4, all four methods could recognize rotated characters robustly. However because SIFT is not invariant to affine transformation, the recognition rates fell rapidly as the independent scaling and the shearing became serious as shown in Figs. 5 and 6. Also the remaining three methods were not so robust to independent scaling. The bad recognition rates might be caused by the shrinking of character images. In the experiment, character images shrank vertically by applying the independent scaling. Because of that, the number of extracted feature points became less and less as independent scaling got serious. Table 3 shows the relationships between changes of the parameter of independent scaling and the number of feature points extracted by each method. In all the methods the number of feature points decreased as independent scaling became serious. Therefore, it became more difficult to distinguish the characters. As for rotation and shearing, the number of feature points did not change so much as the transformation became serious. Because of the shrinking of characters, also geometric characteristics of each character like round shape or angulate shape might be destroyed. As a result, characters like “5” and “s” were sometimes difficult to distinguish.

Though Harris-Affine and Hessian-Affine are affine-invariant detectors the two methods were not so robust also to shearing. it might be caused by the way to normalize ellipse regions to circle regions. In the process, two axes of each ellipse were adjusted to change the ellipse to circle. However, this process cannot cover all of shearing transformation.

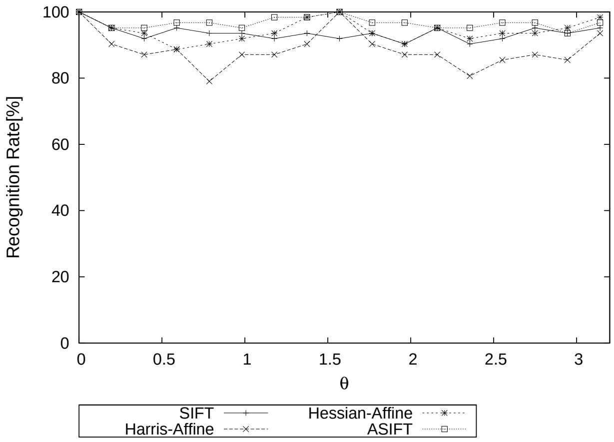

Figure 4 Relationship between the rotation(θ) and recognition rates of the four methods.

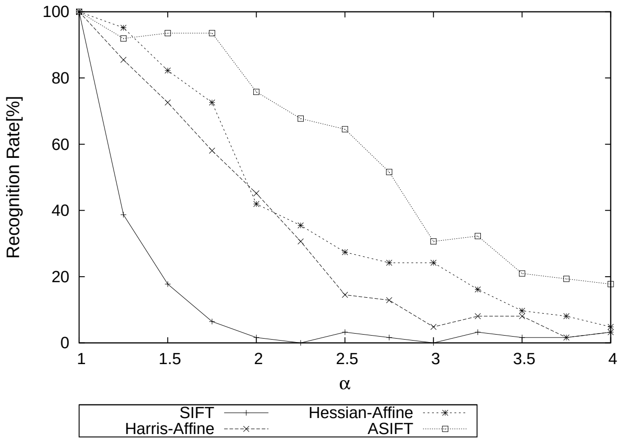

Figure 5 Relationship between the independent scaling(α) and recognition rates of the four methods.

Among the four methods, ASIFT had the best recognition rates in all the experiments. However, the result is still insufficient to realize the robust character recognition system. We consider what reduced the recognition rates as follows.

First, since character images usually have only two colors, black color for characters and white color for the background, the pixel values do not change except neighborhoods of contours of characters. Thus the variety of image intensity comes to monotonous compared with scenery images. Feature detectors can not extract discriminative feature points from character images according to the information of pixel values.

Second, characters which contain the shape of another character tend to fall into a false recognition. For example, "w" has the shape that double "v" are connected. Thus "w" has more feature vectors similar to those of "v", and "v" was sometimes recognized as "w" or "M". In order to solve the problem, it might be efficient to reject too similar pairs of feature vectors from an image.

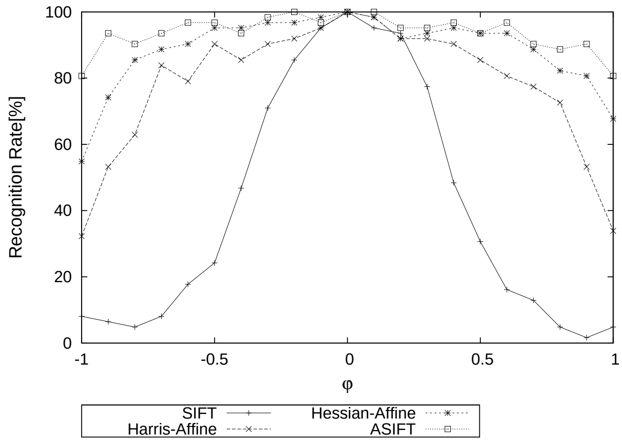

Figure 6 Relationship between the shearing(φ) and recognition rates of the four methods.

5. Conclusion

In this paper, we evaluated local feature descriptors in respect to the recognition accuracy of affine distorted characters. We experimented how well they can recognize affine distorted characters. Affine-invariant feature detectors and descriptors were robust to rotation and shearing of character images. However, they were not so robust to independent scaling because character images shrank heavily in a direction by independent scaling. It caused the decrease of the number of extracted feature points and the features were not discriminative.

As a result, character recognition by using affine-invariant local feature descriptors is robust to affine distorted characters to some extent. ASIFT was the best among the four methods.

In the experiments we did not measure the processing time of the local descriptors. To realize a real-time character recognition system, we must recognize characters quickly. Measuring the processing time is a future work.

Acknowledgment

This research was supported by KAKENHI 21700202.

References

[1] J.M. Morel and G.Yu, ASIFT: A New Framework for Fully Affine Invariant Image Comparison. SIAM Journal on Imaging Sciences, vol. 2, issue 2, 2009.

[2] K. Mikolajczyk and C. Schmid, Scale and Affine invariant interest point detectors. International Journal of Computer Vision, 60(1), pp. 63–86, 2004.

[3] David G. Lowe, Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60, 2, pp. 91–110, 2004.

[4] M. Brown and D. Lowe, Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision, 74(1), pp. 59–73, 2007.

[5] http://www.cs.ubc.ca/~lowe/keypoints/

[6] http://www.robots.ox.ac.uk/~vgg/research/affine/

[7] http://www.cmap.polytechnique.fr/ yu/research/ASIFT/demo.html