Real-Time Camera-Based Recognition of Characters and Pictograms

Masakazu Iwamura, Tomohiko Tsuji, Akira Horimatsu, and Koichi Kise
Osaka Prefecture University, 1-1 Gakuencho, Sakai, Osaka, 599-8531, Japan

Abstract

Camera-based character recognition systems should have the capability of quick operation and recognizing perspectively distorted texts in a complex layout. In this paper, in order to realize such a system, we propose a simple but efficient implementation of camera-based recognition of characters and pictograms. With help of new hashing and voting techniques, the proposed method runs well in real-time even on a laptop PC with a web camera.

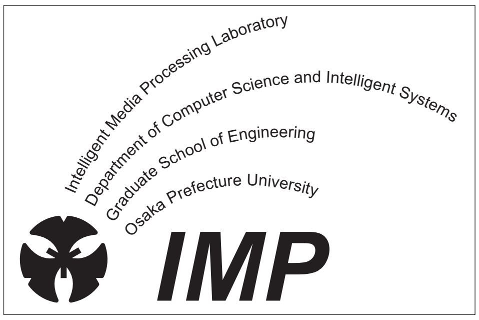

Intelligent Media Processing Laboratory
Department of Computer Science and Intelligent Systems
Graduate School of Engineering
Osaka Prefecture University
IMP

Figure 1. An example of a designed text.

1. Introduction

Camera-based character recognition has received considerable attention due to a wide variety of possible applications. One of convincing applications is "translation camera" which is a portable translating device integrated with a camera and an OCR system [9, 2]. As pointed out in [2], real-time processing is absolutely necessary because a late response hampers the user's convenience.

Such applications also require the capability to recognize characters suffering from a perspective distortion. Some camera-based text recognition techniques can recognize distorted texts [1, 8]. Especially, [8] works in real-time. Although these methods may be applicable to many texts in scene images, they are not usable for all characters in a scene because they have to extract text lines from a scene image to eliminate geometric distortions of the texts. To take an example, signboards often contain "isolated characters" which are characters existing solely or designed curved texts as shown in Fig. 1. Some other character recognition methods [5, 7] has the capability of recognizing the texts in the figure. Since they require much processing time, they are useless for a base system of real-time applications. Thus realizing real-time camera-based character recognition is still a big challenge.

In this paper, in order to realize such a system, we propose a simple but efficient implementation of real-time camera-based recognition of characters and pictograms. In order to realize real-time processing, the configuration of the proposed method is simple; adaptive binarization and contour extraction are used for segmentation and an improvement of geometric hashing (GH) like method whose computational cost is reduced from $O(P^4)$ to $O(P^2)$, where P is the number of feature points, is used for recognition. With help of new hashing and voting techniques, the proposed method runs well in real-time even on a laptop PC with a web camera. The proposed method enables us to recognize distorted characters and pictograms in a complex layout.

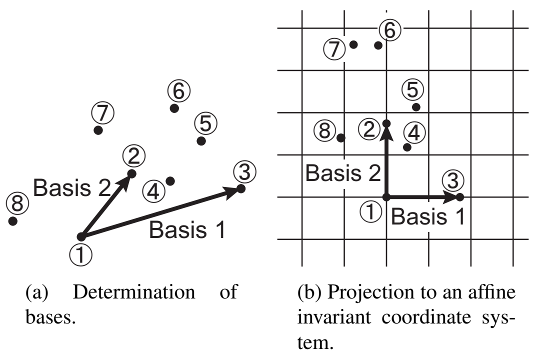
2. Improvement of Geometric Hashing

2.1. Geometric Hashing (GH) [6]

GH describes the image which has undergone a certain geometric transformation using invariant coordinate systems. In order to present our proposed recognition method outperforming GH, the storage and retrieval processes of GH are presented in the case of an affine transformation.

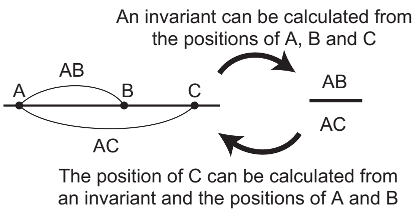
2.1.1 Storage process

Let us assume we have feature points extracted from a reference image. First of all, three of them are chosen and two bases are defined as shown in Fig. 2(a). Then, all the feature points are projected to the invariant coordinate system spanned by the bases. The invariant coordinate system is divided (quantized) into subregions in advance (each



⑥
⑦
⑤
②
Basis 2
④
③
⑧
Basis 1
①
(a) Determination of bases.
⑦
⑥
⑤
⑧
②
④
Basis 2
③
①
Basis 1
(b) Projection to an affine invariant coordinate system.

Figure 2. An invariant coordinate system of geometric hashing.



An invariant can be calculated from the positions of A, B and C
AB
A
B
C
AC
AB
AC
The position of C can be calculated from an invariant and the positions of A and B

Figure 3. The relationship between the positions of points and an affine invariant.

subregion is represented as a rectangle in Fig. 2(b)). Thus, the image ID of the reference image and a basis-set ID are stored into the corresponding subregion. The storage process finishes after the procedure above is carried out for all the invariant coordinate systems spanned by all sets of bases and for all the reference images to be stored.

2.1.2 Retrieval process

The initial phase of the retrieval process is almost the same as that of the storage one. Let us assume we have feature points extracted from a query image. Three of them are chosen and two bases are defined (Fig. 2(a)). Then, all the feature points are projected to the invariant coordinate system spanned by the bases (Fig. 2(b)). Each projected feature point corresponds to a subregion of the invariant coordinate system. The votes for the corresponding pairs of the image ID and the basis-set ID are cast. The procedure above is carried out for all sets of bases. Finally, the pair of the image ID and the basis-set ID with the highest vote is determined. The process can quit when the output image is obvious.

2.2. Our improvement

2.2.1 Computational cost reduction

The lack of GH is tremendous computational cost. GH requires $O(P^4)$ for retrieval (recognition), where P is the number of feature points. This means that if we have 100 points, it requires $O(100,000,000)$! Thus GH is never applicable to real-time applications. On the other hand, our proposed method reduces the computational cost down to $O(P^2)$ using uniqueness of arrangement of feature points.

The reason that GH requires tremendous computational cost is that GH cannot determine the corresponding set of bases in the storage and retrieval processes, though GH requires the correspondence for successful retrieval. This means that GH has to search the correspondence by examining all sets of bases. Thus if we can select the corresponding set of bases, the computational cost can be reduced. Before presenting our proposed method, note that in our method all the pixels on the external contour of a connected component are used for feature points. Thus we use the characteristic of the external contour of a connected component.

The strategy of our proposed method is as follows. Firstly we determine the first point uniquely using the nature that the center of gravity of a connected component does not change regardless of affine transformations. Next we have to select the second point out of P points. Finally, we can determine the third point almost uniquely by our proposed way of selecting a point. We present it below as simple as possible.

Usually geometric invariants are used as follows; From the arrangement of n points, we can calculate a geometric invariant. As long as we have the same n points, we can always calculate the same value of the invariant under a certain geometric transformation. In the case of Fig. 3, an invariant is calculated from the positions of A, B and C. On the other hand, we propose a different usage of invariants; If we have the value of the invariant and $n-1$ points out of the n points, we can determine the last point almost uniquely under some geometric transformations. In the case of Fig. 3, the position of C is calculated from an invariant and the positions of A and B. Finally, the computational cost is reduced from $O(P^4)$ to $O(P^2)$ since we determined two points uniquely.

2.2.2 Feature vector of a connected component

GH stores the pairs of the image ID and the basis-set ID into the database. Instead of the basis-set ID, our method stores a feature vector and coordinates of the three points to calculate the bases. Since we treat a character recognition problem hereafter, the "image ID" is called "character ID."

The reason why we use a feature vector instead of basis-set ID is that a feature vector is available and more descriptive. The purpose of GH is to match the arrangements of

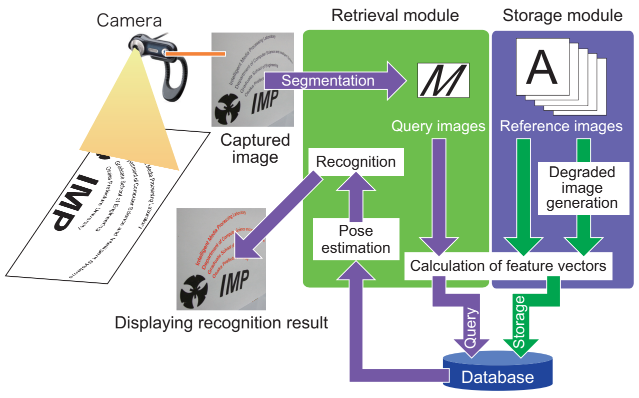


Figure 4. Overview of the proposed system.

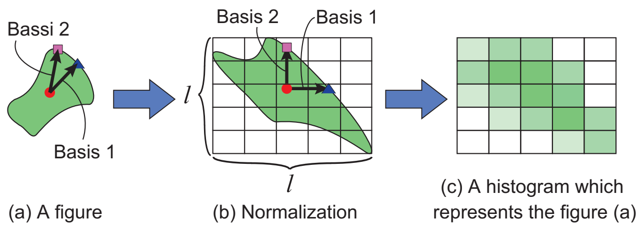


Figure 5. Calculation of a k-dimensional feature vector based on values of k subregions. $k = l \times l$. For convenience' sake, black pixels are represented in green.

points and the only information available is coordinates of points. On the other hand, our purpose is to match connected components. Thus we select a more descriptive feature. The reason why we store the coordinates of three points is that they are usable to estimate affine transformation matrices. This enhances recognition accuracy. We describe this in Sec. 3.2.

3. System Overview

The overview of the proposed system is shown in Fig. 4. The system consists of storage and retrieval modules.

3.1. Storage module

In the storage module, reference images are stored in the database. Three degraded images are created from each reference image using a Gaussian blurring model [3]. For calculation of feature vectors, an invariant coordinate system is calculated in the manner described in Sec. 2.2.1. As the result, a figure in Fig. 5(a) is normalized as shown in Fig. 5(b). Then, $k(= l \times l)$ uniform subregions are defined and a histogram of black pixels shown in Fig. 5(c) is calculated. Finally, the histogram is normalized to satisfy the sum of bins is 1 and a k-dimensional feature vector is obtained. The vector is an affine invariant in theory. Since we can calculate three feature vectors for an affine invariant coordinate system by selecting three points in different orders, a $(3k)$-dimensional feature vector is obtained by just concatenating the three feature vectors.

A set of character ID, a feature vector and coordinates of three points is stored into the hash table. The hash index of the hash table, H_{index}, is calculated as

$$H_{\text{index}} = \left(\sum_{i=1}^{3k} 3^{i-1} r_i \right) \bmod H_{\text{size}}, \tag{1}$$

where H_{size} is the size of the hash table ($2^{19} - 1$ was used) and r_i is the quantized value of the i-th element into D levels. $D = 2$ was used according to preliminary experiment results.

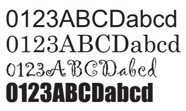
3.2. Retrieval module

A query image is captured by a digital camera or a web camera. The query image is adaptively thresholded into the binary image, and contour extraction is applied. Then, feature vectors are calculated in the same manner as in the storage module. Let S be the number of affine invariant coordinates calculated. This value is used in the experiments.

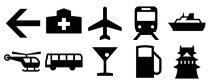
Then, parameter estimation and recognition are carried out. To begin with, by accessing the hash table, all the sets of character ID, a feature vector and coordinates of three points related to the connected components of the query image are retrieved. Then weighted voting with a weight $\frac{1}{\sqrt{P}}$ is cast for the corresponding character ID. The reason of using the weight is that a character having larger P is expected to have larger number of votes. Then, two groups of characters are defined. Let M be the number of the highest vote, and characters which have larger number of votes than $0.9M$ are grouped in "estimation group" and characters which have larger number of votes than $0.8M$ are grouped in "candidate group."

From each correspondence of three points in the query image and the databases, an affine transformation matrix is calculated. The matrix is decomposed into four parameters: scaling, rotation, shear and independent scaling. We assume all the characters in the query image are on a plane paper and have the same parameters of shear and independent scaling. Using affine transformation matrices of the estimation group, the best parameters of them are determined based on a densest point estimation (an approximate and faster version of [4]) in the 2D parameter space of shear and independent scaling.

Finally, from the affine transformation matrices of the



(a) Arial, Century, Gigi and Impact from the top.



(b) 10 pictograms used in the experiments.

Figure 6. Samples of fonts and pictograms.

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

0 O o	6 9	C c	I l	S s	u n
W w	X x	N Z z	p d	q b	7 L V v

candidate group, the closest point in the 2D parameter space is selected as the recognition result. If further recognition results are required, 2nd rank result is determined by the closest point of a different character and so on.

4. Experiments

4.1. Fonts

We evaluated the effectiveness of the proposed method using characters in various fonts, We employed 60 characters from numerals and alphabets; 10 figures, 24 lowercase alphabets except “i” and “j” which consists of two connected components, and 26 capital alphabets. Since some characters are difficult to distinguish under affine distortions, the characters in a cell in Table 1 were treated as the same class in all experiments. Samples of four fonts used in the experiments are shown in Fig. 6(a).

As recognition targets, we prepared 60 sheets of letters as shown in Fig. 7. Each of the sheet contained 108 characters evenly in nine conditions which were the combinations of three different sizes and three different rotation angles (0, 30 and 45 deg.). Each sheet was captured in three different angles (0, 30 and 45 deg.) by a digital camera with a resolution of 1024×768. $S = 200$ and $k = 25$ were used. A server with Opteron 2.4GHz was used for all experiments. The cumulative recognition rates and average processing time are shown in Fig. 8 and Table 2, respectively. Although the proposed method is affine invariant, the recognition results show that the proposed method recognized perspectively distorted images well. As the rank increased, the cumulative recognition rates of Arial, Century and Gigi increased and became flat around 6th rank. However, the recognition rate of Impact was very bad in the 1st rank and increased untill 20th rank.

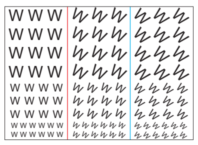


Figure 7. A sheet of recognition target.

Table 2. Average processing time for recognition of a letter.

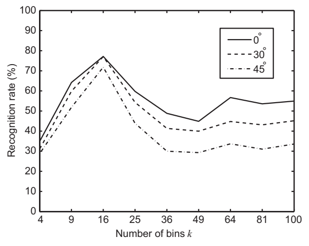
Font	Arial	Century	Gigi	Impact
Proc. time (ms)	32.4	24.7	24.0	81.0

4.2. Pictograms

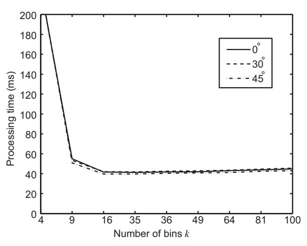
10 pictograms shown in Fig. 6(b) were captured in the same manner as in Sec. 4.1, and recognized using $S = 200$. The recognition rates and processing time are shown in Fig. 9. The highest recognition rate was achieved in the case of 16 bins. The number of bins almost did not change processing time except the case of four bins. The reason that in the case of four bins the recognition rate was the lowest and processing time was the largest was many collisions.

4.3. Designed text (Fig. 1)

Finally, the proposed method was applied to the designed text shown in Fig. 1. Camera-captured images of it in 0, 30 and 45 deg. are shown in Fig. 10. The figures contained 148 connected components of characters. However 18 of them were the parts of “i” and “j” and unable to recognize for the proposed method because the reference images of them were not stored in the database. Thus the recognition rates were calculated using 130 as the denominator. The

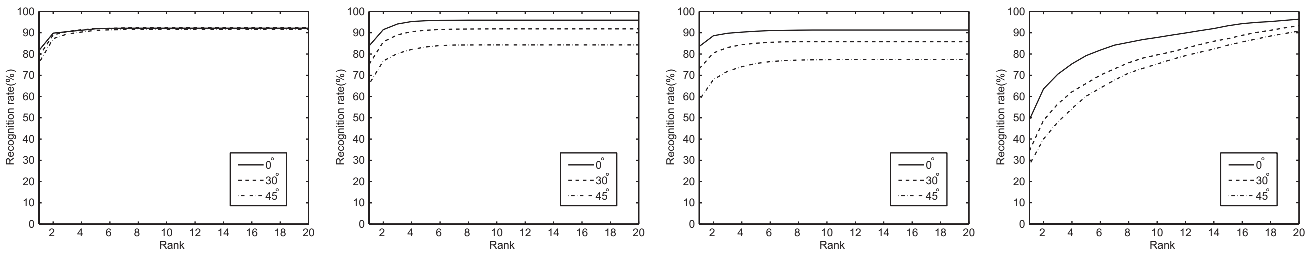


(a) Recognition rates.



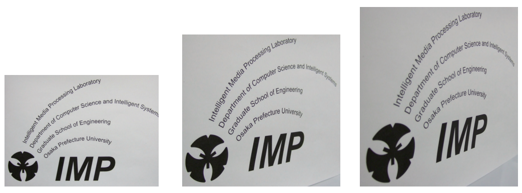
(b) Processing time.

Figure 9. Recognition rates and processing time for pictograms in different size of feature vectors.



(a) Arial. (b) Century. (c) Gigi. (d) Impact.

Figure 8. Cumulative recognition rates for various fonts.



(a) 0 deg. (b) 30 deg. (c) 45 deg.

Figure 10. Camera-captured images of the designed text in Fig. 1.

Table 3. Recognition rates and whole processing time for designed text in Fig. 1.

S	200			20		
Angle (deg.)	0	30	45	0	30	45
Rate (%)	91.5	93.1	86.9	90.0	86.2	83.9
Time (ms)	5400	5210	4790	740	710	640

recognition rates and processing time in the case of $k = 25$ is shown in Table 3. The table shows that $S = 200$ achieved higher recognition rates than $S = 20$. $S = 20$ was about 7 times faster than $S = 200$ though the recognition rates of them do not differ so much. The result of $S = 20$ supports that the proposed method is fast and robust.

5. Conclusions

In order to realize a camera-based character recognition system which has the capability of quick operation and recognizing perspectively distorted texts in a complex layout, we proposed a simple but efficient implementation of camera-based recognition. In order to realize real-time processing, adaptive binarization and contour extraction are used for segmentation and an improvement of geometric hashing like method whose computational cost is reduced from $O(P^4)$ to $O(P^2)$ is used for recognition. We confirmed the proposed method works well with a digital camera by experiments. With help of new hashing and voting techniques, the proposed method runs well in real-time even on a laptop PC with a web camera though the result is not shown in the paper.

Acknowledgement This research was supported by KAKENHI 19700177 and 21700202.

References

[1] X. Chen, J. Yang, and A. Waibel. Automatic detection and recognitionof signs from natural scenes. *IEEE Trans. Image Processing*, 13(1):87–99, Jan. 2004.

[2] H. Fujisawa, H. Sako, Y. Okada, and S.-W. Lee. Information capturing camera and developmental issues. In *Proc. ICDAR1999*, pages 205–208, Sept. 1999.

[3] H. Ishida, S. Yanadume, T. Takahashi, I. Ide, Y. Mekada, and H. Murase. Recognition of low-resolution characters by a generative learning method. In *Proc. CBDAR2005*, pages 45–51, 2005.

[4] M. Iwamura, R. Niwa, A. Horimatsu, K. Kise, S. Uchida, and S. Omachi. Layout-free dewarping of planar document images. In *Proc. DRR XVI*, 7247-36, Jan. 2009.

[5] Y. Kusachi, A. Suzuki, N. Ito, and K. Arakawa. Kanji recognition in scene images without detection of text fields—robust against variation of viewpoint, contrast, and background texture—. In *Proc. ICPR2004*, 2004.

[6] Y. Lamdan and H. J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *Proc. ICCV1988*, pages 238–249, 1988.

[7] L. Li and C. L. Tan. Character recognition under severe perspective distortion. In *Proc. ICPR2008*, 2008.

[8] G. K. Myers, R. C. Bolles, Q.-T. Luong, J. A. Herson, and H. B. Aradhye. Rectification and recognition of text in 3-d scenes. *IJDAR*, 7(2-3):147–158, 2004.

[9] Y. Watanabe, Y. Okada, Y.-B. Kim, and T. Takeda. Translation camera. In *Proc. ICPR1998*, pages 613–617, 1998.