Affine Invariant Recognition of Characters by Progressive Pruning

Akira Horimatsu
Osaka Pref. Univ., Japan

Ryo Niwa
Osaka Pref. Univ., Japan

Masakazu Iwamura
Osaka Pref. Univ., Japan

Koichi Kise
Osaka Pref. Univ., Japan

Seiichi Uchida
Kyushu Univ., Japan

Shinichiro Omachi
Tohoku Univ., Japan

Abstract

There are many problems to realize camera-based character recognition. One of the problems is that characters in scenes are often distorted by geometric transformations such as affine distortions. Although some methods that remove the affine distortions have been proposed, they cannot remove a rotation transformation of a character. Thus a skew angle of a character has to be determined by examining all the possible angles. However, this consumes quite a bit of time. In this paper, in order to reduce the processing time for an affine invariant recognition, we propose a set of affine invariant features and a new recognition scheme called "progressive pruning." The progressive pruning gradually prunes less feasible categories and skew angles using multiple classifiers. We confirmed the progressive pruning with the affine invariant features reduced the processing time at least less than half without decreasing the recognition rate.

1. Introduction

Recently, portable digital cameras and mobiles phone with built-in cameras have got very popular. Since some of them have over ten million pixels, without hesitation we can call them *general purpose capturing tools* with high potential and high portability. Due to the ability of the cameras, camera-based character recognition have been focused. Using the character recognition techniques at any time and any place, many convenient services can be realized. For example, a service which offers the related information to the things where the user is interested can be realized by just capturing them. For another example, by recognizing every character in a scene, a system may be able to offer selected important information to the user.

In order to realize such systems by camera-based character recognition, there are many problems to solve [1, 4]. From the problems, we focus on distortions because characters in scenes are often distorted by geometric transformations. One approach to resolve the problem is to remove distortions.

For affine distortions, Ohta et al. [2] and Leu [3] have proposed the same method which normalizes the affine distortions. However, this method has a drawback that this cannot remove a rotation transformation of the character. Thus a skew angle of a character has to be determined by examining all the possible angles, and this consumes quite a bit of time. For the problem, Wang et al. [7] have proposed a method to estimate the skew angle from moments. However, the method failed to estimate the angle because the estimated skew angle depends on the affine transformation that the character suffer (i.e., depends on the longer direction after the distortion) instead of the skew angle itself. In addition, although moment-based methods seem to be good because they can be derived from simple equations and calculated in short time, they are very weak for noise. Therefore, the task of estimating the skew angle with short processing time is still strongly demanded.

In this paper, in order to reduce the processing time for an affine invariant recognition, we propose a set of affine invariant features and a new recognition scheme called "progressive pruning." The features can be calculated under affine distortions and roughly distinguish characters. The progressive pruning gradually prunes less feasible categories and skew angles using multiple classifiers.

2. Basic Recognition Method by Affine Normalization

We explain a basic way of recognizing an affine distorted character image by applying an affine normalization [2, 3] and a Brute force search of the skew angle. This method is refereed as "conventional method" in the experiments in Sec. 4.

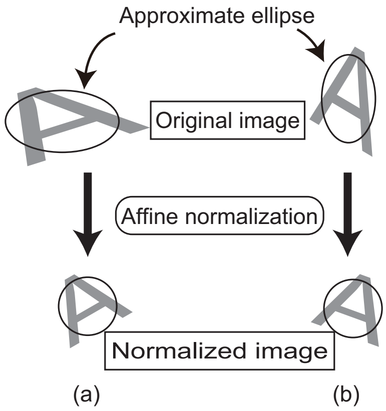


Figure 1. Affine normalization

2.1. Affine normalization

The methods which normalize an affine distorted image have been proposed [2, 3]. As shown in Fig. 1, the methods estimate an ellipse which approximates the character. We call the ellipse "approximate ellipse." Since a transformation of an ellipse into a circle is an affine transformation, we obtain a normalized image by applying the transformation which normalizes the estimated ellipse into a circle. The approximate ellipse is calculated using the covariance matrix of the character.

Let us go into greater detail. Since the approximate ellipse has the same covariance matrix as the character, we begin with the estimation of the covariance matrix $\boldsymbol{C}$ of the character. $\boldsymbol{C}$ is defined as

$$\boldsymbol{C} = \begin{bmatrix} m_{20}/m_{00} & m_{11}/m_{00} \\ m_{11}/m_{00} & m_{02}/m_{00} \end{bmatrix}, \quad (1)$$

where m_{00}, m_{02}, m_{11} and m_{20} are the central moments defined as

$$m_{ij} = \iint x^i y^j f(x, y) dx dy, \quad (2)$$

where

$$f(x, y) = \begin{cases} 1 & \text{if } (x, y) \in \text{character region} \\ 0 & \text{if } (x, y) \in \text{non-character region} \end{cases}$$

Then, we normalize the character using the inverse covariance matrix $\boldsymbol{C}^{-1}$ of the covariance matrix $\boldsymbol{C}$. Let (x_0, y_0) and (x_1, y_1) be the (x, y)-coordinates before and after transformation, respectively. If the center of mass $\begin{bmatrix} m_{10} \\ m_{01} \end{bmatrix}$ does not move by the affine transformation, we obtain

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \boldsymbol{C}^{-1} \begin{bmatrix} x_0 - m_{10} \\ y_0 - m_{01} \end{bmatrix} + \begin{bmatrix} m_{10} \\ m_{01} \end{bmatrix}. \quad (3)$$

Note that the normalized image obtained by the transformation is not the usual character image that we observe without distortions. For example, we can easily imagine that ellipsoidal characters such as "0" (zero) and "O" (in capital alphabet) are normalized into circumferential shape images. Similarly, vertically long rectangular characters such as "I" (in capital alphabet) in a Gothic font is normalized into a square shape image. Thus, the matching procedure has to be carried out using the normalized reference images and the normalized test image. We will explain the matching procedure in the next section.

2.2. Recognition after estimating skew angle

After the process described in Sec. 2.1, affine normalized character images are obtained. However, as shown in Figs. 1(a) and (b), the process does not normalize skew angles. Therefore, we have to search the best angle that offers the highest similarity between each normalized reference image and the normalized test image. The simplest way is to rotate the normalized test image by a certain angle (e.g., ten degrees) and find the skew angle that provides the highest similarity. This is achieved by transforming the coordinate system into the polar coordinate system denoted by (r, θ), and then creating an *angle histogram* whose bin width is θ_0. The matching of histograms finds approximately best angles by $2\pi/\theta_0$ matches. Note that if θ_0 is large, the processing time is small although the estimated angle contains large error. If θ_0 is small, the situation is the opposite. This is the trade-off relationship. The best value of θ_0 depends on the test image and cannot be determined in advance.

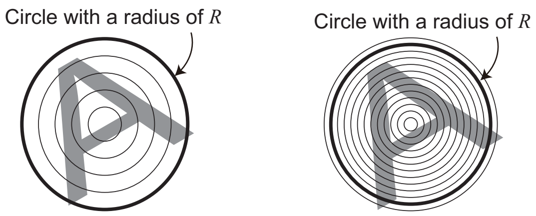
Finally, the normalized test image rotated by the candidate angles is recognized using a recognition method for OCR characters such as the normalized correlation (NC). This determines the corresponding reference character and the best angle simultaneously.

3. Proposed Method

We present the proposed method by improving the basic method described in Sec. 2. The proposed method has the same normalization process described in Sec. 2.1. However, angle estimation and recognition process are different from the one described in Sec. 2.2. Therefore we explain the latter process in this section.

3.1. Feature histograms

For the preparation of explaining angle estimation and recognition process, we explain two feature histograms used in the process.



(a) $k = 6$. (b) $k = 14$.

Figure 2. Distance histograms.

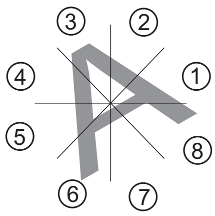


Figure 3. Angle histogram. $m = 8$**.**

3.1.1 Distance histogram

A distance histogram which represents the distribution of distances between the image center and pixels is created. Figs. 2(a) and 2(b) illustrate the distance histograms with 6 bins and 14 bins, respectively.

The distance histogram with k bins is constructed as follows. First of all, let R be the radius of the normalized approximate ellipse of a character after the affine normalization process in Sec. 2.1. Let n be the number of bins inside the circle. n is defined as

$$n = \begin{cases} n = k-1 & \text{if } k < 9 \\ n = k-2 & \text{if } k \geq 10. \end{cases} \tag{4}$$

Then, we introduce concentric circles whose radii are $\frac{i}{n}R$, $i = 1, \ldots, k-1$, respectively. The histogram is constructed based on the distance r between the image center and a pixel;

- For the most external circle, that is $i = k$, increase the bin value of the kth bin by one.
- Otherwise, increase the bin value of the ith bin by one, where i is the integer number satisfying $\frac{i-1}{n}R \leq r < \frac{i}{n}R$.

Finally, the histogram is normalized so that the sum of bins is one.

3.1.2 Angle histogram

An angle histogram which represents the distribution of pixel angles is created. Fig. 3 illustrates the angle histogram with eight bins.

The angle histogram with m bins is constructed as follows. First of all, a normalized character image is divided into m regions along the rotation direction. The histogram is constructed based on the angle θ of a pixel ($0 \leq \theta < 2\pi$); Increase the number of pixels by one in the ($\lceil \theta/m \rceil$)th bin. Finally, the histogram is normalized so that the sum of bins is one.

3.2. Progressive pruning

In this section, we present the process of the proposed method called "progressive pruning." It is illustrated in Fig. 4. It is carried out instead of the process of the conventional method described in Sec. 2.2. The basic idea of the progressive pruning is as follows. Though the conventional method takes care of only skew angles, actually we have two-dimensional searching space consists of a character class axis and a skew angle axis. Thus we couple them and gradually reduce candidate pairs by a multi-stage pruning. This enables us to reduce the processing time for the affine invariant recognition.

For convenience of explanation, we name the following three stages shown in Fig. 4; Pruning character classes is "stage A," pruning skew angles is "stage B," and a detailed matching is "stage C."

3.2.1 Pruning character classes (Stage A)

At the stage A, character classes are pruned using the distance histogram described in Sec. 3.1.1. We do not consider skew angles at the stage, however, this means to prune the candidates of all the rotation angles (from 0 to 360 degrees) for the pruned character classes.

We investigated the discrimination ability of the distance histogram. 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. As shown in Fig. 5, we prepared images whose sizes were 96×96 pixels and applied various affine transformations. For the sake of that, the affine transformation matrix $\boldsymbol{T} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ was decomposed into

$$\begin{aligned} \boldsymbol{T} &= L(\beta)R(\theta)S(\varphi)A(\alpha) \\ &= \begin{pmatrix} \beta & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \\ &\qquad \begin{pmatrix} 1 & \tan\varphi \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & 1/\alpha \end{pmatrix}, \end{aligned} \tag{5}$$

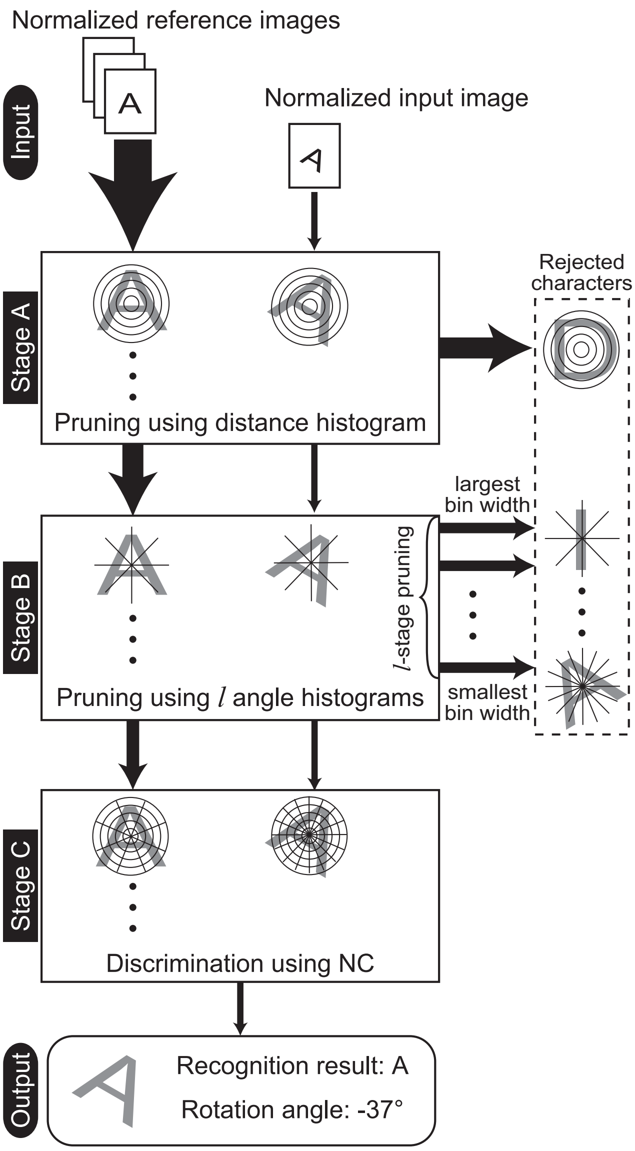


Figure 4. Progressive pruning. This gradually prunes candidate pairs each of which consists of a character class and a skew angle. At the stage A, character classes are pruned using the distance histogram. At the stage B, skew angles are pruned using the l different angle histograms (an angle histogram with the largest bin width is used first and that with the smallest one is last). At the stage C, a detailed matching is carried out. This enables us to reduce the processing time for the affine invariant recognition.

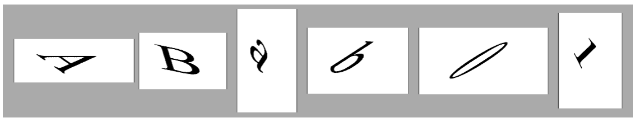


Figure 5. Parts of character images used for the investigation of the discrimination ability of the distance histogram, picked from 1260 affine distorted images per character.

where

$$\alpha = \pm\sqrt{\frac{a^2+c^2}{ad-bc}}, \tag{6}$$

$$\varphi = \tan^{-1}\frac{ab+cd}{ad-bc}, \tag{7}$$

$$\theta = \cos^{-1}\frac{\pm a}{\sqrt{a^2+c^2}}, \tag{8}$$

$$\beta = \pm\sqrt{ad-bc}. \tag{9}$$

Since β is the scale parameter, we changed the remaining three parameters α, φ and θ in the following ranges: $\alpha = \{1, 2, 3, 4\}$, $\varphi = \{-1.0, -0.9, \cdots, 1.0\}$, and $\theta = \{0.0, 0.1, \cdots, 1.4\}$. Thus, by combining them, we applied 1260 affine transformations for each character (1260 comes from $4 \times 21 \times 15$).

The distributions of the distance histogram in the case of $n = 5$ are shown in Fig. 6. The figure shows that (1) by comparing Fig. 6(a), (b) and (c), the distributions differ by bins, and (2) the distributions of the distance histogram vary depending on characters. Especially for (2), characters which take similar values in a bin do not take similar values in another bin. Therefore, we confirmed that the distance histogram has the discrimination ability enough to roughly distinguish characters.

3.2.2 Pruning skew angles (Stage B)

At the stage B, skew angles are pruned using the angle histograms described in Sec. 3.1.2. The difference between the conventional method and the proposed method is the number of histograms. The conventional method employs only one histogram. In contrast, the proposed method employs a certain number (say, l) of histograms. An angle histogram with the largest bin width is used first and that with the smallest one is last. This means that the proposed method gradually reduces candidate skew angles. Hereafter, the number of bins of ith angle histogram is denoted as m_i.

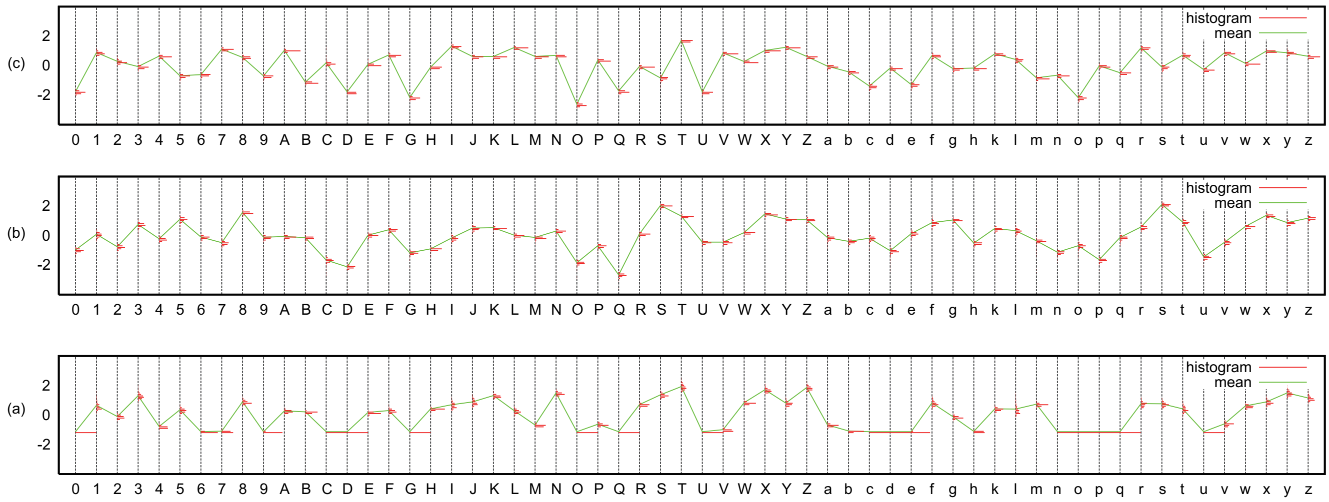


Figure 6. Distributions of bin values of the distance histogram. (a), (b) and (c) correspond to the first, the fourth, and the fifth bin from the image center, respectively. The histogram of the distribution for each character is plotted in red; Horizontal axis represents frequency. A green line connects the mean values of the distributions. The distributions are normalized so that the mean is zero and the standard variance is one.

3.2.3 Detailed matching (Stage C)

At the stage C, a detailed matching is carried out. The detailed matching finds the pair of the character class and the skew angle which offers the highest similarity to the normalized test image. Since less feasible pairs are pruned before the stage C, only feasible pairs are examined. In this paper, The normalized correlation (NC) is used for the detailed matching. The NC uses the combination of the distance histogram and the angle histogram with 360 bins (i.e., bin width is 1 degree).

3.3. Similarity measure

In this paper, a similarity measure proposed by Swain et al. [6] is used. Let H and M be the histograms of the test image and a reference image, respectively. Then, the similarity between H and M are given as

$$S_{HM} = \sum_{b=1}^{N} \min(H_b, M_b), \tag{10}$$

where N is the number of bins, and H_b and M_b are the bth bin value of H and M, respectively. Since both H and M are normalized histograms, S_{HM} satisfies $0 \leq S_{HM} \leq 1$. If the similarity is equal or less than a threshold, the corresponding character or skew angle is rejected. The thresholds of the distance histogram and the angle histogram should be determined in preliminary experiments.

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

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

4. Experiments

In order to confirm the effectiveness of the proposed method, we carried out three experiments.

4.1. Robustness and pruning effect for affine distorted character images

For affine distorted character images, we investigate the effect of the proposed method in both recognition rates and processing time by changing parameter values.

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 the experiments.

Reference character images were created in the following manner; Character images of 100 points Arial font were

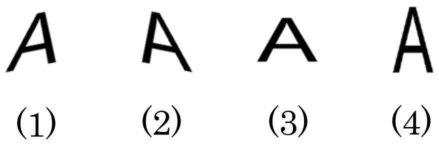


Figure 7. Affine distorted character images. (1) was created by translating the upper left and the lower right corners toward the image center by 25 pixels. (2) was created by the similar operation but the translated corners were the upper right and the lower left. (3) and (4) were created by reducing the height and the width to $2/3$ times, respectively.

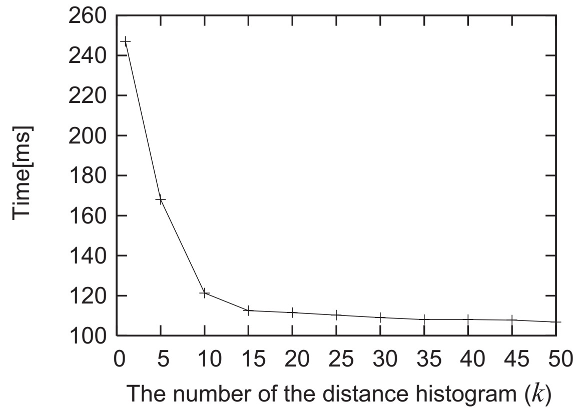


Figure 8. Relationship between k and the average processing time for the proposed method. $l = 1$ and $m_1 = 120$ were used.

created, and then scaled so that the sizes of "0" were 51×98 pixels. Test images were created as follows; Each reference image was put on the center of a white image whose sizes were 150×150 pixels, and then transformed by four affine transformation matrices. Examples of the test images are shown in Fig. 7.

Fig. 8 shows the relationship between k and the average processing time in the case of $l = 1$ and $m_1 = 120$. The definition of the average processing time is the average time to process a test image in CPU time on the PC with 2.8GHz AMD Opteron. The figure shows that as k increased, the average processing time decreased. The reason seems that since a distance histogram with a small bin width had high discriminant ability, a large number of less feasible candidates were rejected. The recognition rates were 100% for all values of k. The proposed method with $k = 1$ and $l = 1$ corresponds to the conventional method. Since the processing time was 248ms for $k = 1$ and 111ms for $k = 20$, we confirmed the proposed method outperformed the conventional method.

Fig. 9 shows the relationship between m_1 and the average processing time in the case of $k = 20$ and $l = 1$. The figure shows that (i) small values of m_1 marked very large processing time, (ii) $m_1 = 120$ marked the smallest processing time, 111ms, and (iii) the average processing time increased as $|m_1 - 120|$ increased. The reason of (i) seems that when m_1 was small, the discriminant ability at the stage B was small; Thus a small number of less feasible candidates were rejected at the stage B, and a very large number of candidates had to be processed at the stage C; This consumes quite a bit of time. To the contrary, if m_1 was large, a large number of candidates were rejected at the stage B; Although this did not consume much time at the stage C, the processing time at the stage B increased; Thus the processing time in total increased. To conclude the discussion above, the best value of m_1 seems to be determined by the trade-off relationship. This seems to be the reason for (ii) and (iii). The recognition rates were 100% for all values of m_1.

Fig. 10 shows the relationship between m_1 and the average processing time in the case of $k = 20$, $l = 2$ and $m_2 = 120$. The figure shows that (1) $m_1 = 12$ marked the smallest processing time, 108ms, and (2) the average processing time increased as $|m_1 - 12|$ increased, especially for a large m_1. For (2), when m_1 was large, the discriminant ability and the processing time for two angle histograms got similar because m_1 and m_2 were close. If two similar pruning processes ran, the second pruning rejected a very small number of less feasible candidates although the second consumed similar amount of time to the first. Thus the processing time seems to be large when m_1 was large. From the above discussion, the best value of m_1 also seems to be determined by the trade-off relationship. Thus $m_1 = 12$ seems to be the best balanced value. The recognition rates were 100% for all values of m_1.

The three experimental results show that the effect of the progressive pruning in both recognition rates and processing time for affine distorted character images. The best parameters were $k = 20$, $l = 2$, $m_1 = 12$ and $m_2 = 120$ [1] [2]. The proposed method reduced 44% of the processing time in the parameters (44% comes from $108/248$). We used the parameters in the following experiments.

[1]By preliminary experiments, we confirmed the processing time in the cases of $l \geq 3$ was larger than that of $l = 2$ due to the overhead of pruning. However, we believe the overhead can be reduced in future.

[2]As k increased, the processing time decreased. However, too large k can cause recognition error. Thus $k = 20$ was selected.

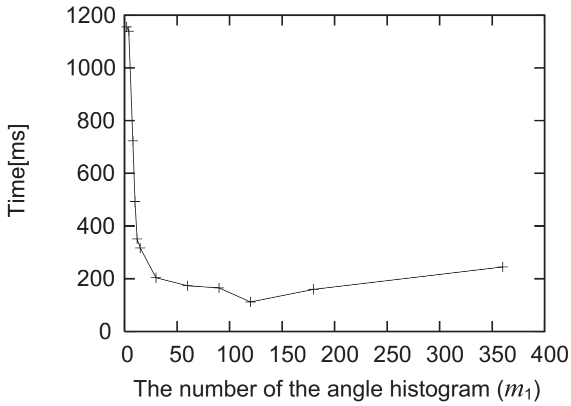


Figure 9. Relationship between m_1 and the average processing time for the proposed method. $k = 20$ and $l = 1$ were used.

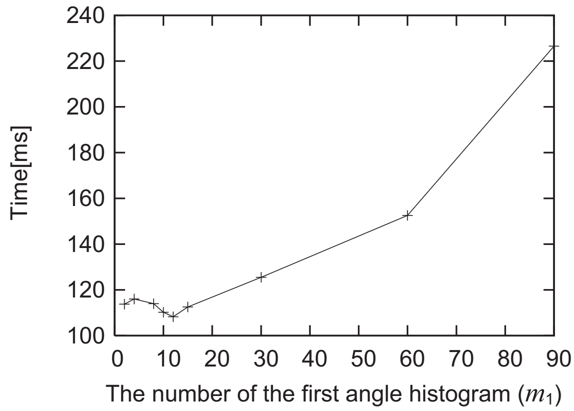


Figure 10. Relationship between m_1 and the average processing time for the proposed method. $k = 20$, $l = 2$ and $m_2 = 120$ were used.

4.2. Robustness for perspectively distorted character images created synthetically

We carried out an experiment to investigate the discriminant performance of the proposed method for perspectively distorted character images created synthetically.

Reference images of 100 points Arial font were created, and then scaled so that the sizes of "0" were 35×55 pixels. Test images were created as follows; Each reference image was put on the center of a white image whose sizes were 150×150 pixels, and then applied projective distortions on the image. The projective distortion was controlled by the (x, y)-displacement of four corners of a character image. For each corner, there were four variations of the (x, y)-displacement, $(0, 0)$, $(0, \delta)$, $(\delta, 0)$ and (δ, δ) toward inside, where δ denotes the displacement by pixel. Thus, for a fixed δ, $4^4 = 256$ test patterns were created from a single reference character image. We used $\delta = 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$. Fig. 11 shows several test patterns of "R".

The relationship between δ and a recognition rate is shown in Fig. 12. When $\delta \leq 10$, the recognition rate was 100%. As δ increased, the recognition rate decreased. However, the rate was higher than 90% when $\delta \leq 40$, and the rate was about 80% when $\delta = 50$. Thus, the proposed method was robust to light perspective distortion which can be approximated as an affine transformation.

4.3. Recognition performance for real scene images

Recognition performance of the proposed method was evaluated for real scene images.

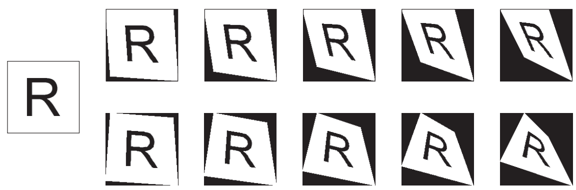


Figure 11. Parts of perspectively distorted character images used in Sec. 4.2.

Reference images of 100 points "MS Gothic" font were created, and then scaled so that the sizes of "0" were 114×203 pixels. Test images were created as follows; As shown in Fig. 13, the reference characters in 36, 48, 60 and 72 points were printed circularly on A4 size paper, and then the paper was captured from the point of 60cm distance and angled 45 degrees to the normal of the paper by a digital camera Canon EOS 5D. A captured image is shown in Fig. 14. Finally, character images were segmented using binalization.

The average recognition rate was 78.5%. Most confusing characters had very similar normalized shapes such as "7"-"L" and "0"-"Q." They seems to be affected by perspective distortions.

5. Conclusion

In this paper, in order to realize robust and quick recognition for affine distorted characters, we proposed a new recognition scheme called "progressive pruning." This gradually prunes less feasible categories and skew angles

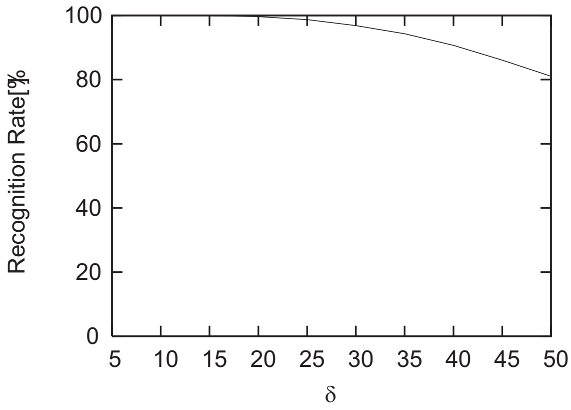


Figure 12. Recognition rates for perspectively distorted character images.

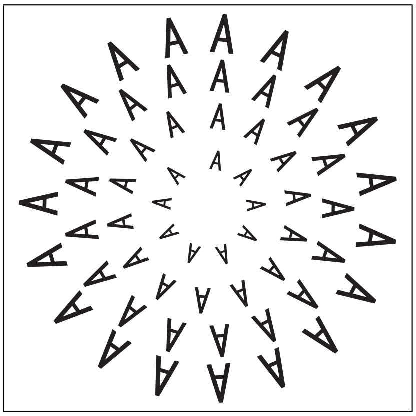


Figure 13. An image for printing.

using multiple classifiers. For the pruning, we used a distance histogram which is a set of affine invariant features and several angle histograms. We confirmed the progressive pruning with the affine invariant features reduced the processing time at least less than half without decreasing the recognition rate. We also confirmed that the proposed method can by applied to perspectively distorted characters.

Future work includes reducing the overhead of the pruning at the stage B. The most time consuming process is to calculate the similarity between the histograms of the normalized reference image and a normalized test image. This can be reduced by active search [5] which enables us to skip parts of the calculation. If the overhead is reduced, we expect the number of angle histograms increases and the processing time decreases.

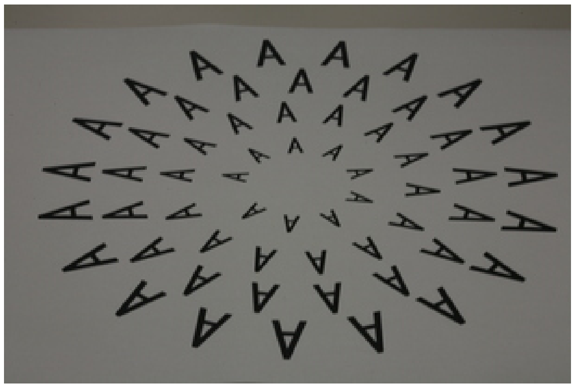


Figure 14. Printed image of Fig. 13 was captured by a digital camera for creating test images. The sizes of the image were 4368×2912.

References

[1] D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. pages 606–616, 2003.

[2] Y. ichi Ohta, K. Maenobu, and T. Sakai. Obtaining surface orientation from texels under perspective projection. In *Proc. of 7th International Conference on Artificial Intelligence*, pages 746–751, 1981.

[3] J.-G. Leu. Shape normalization through compacting. *Pattern Recognition Letters*, 10(4):243–250, 1989.

[4] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *International Journal on Document Analysis and Recognition*, 7(2–3):84–104, 2005.

[5] H. Murase and V. V. Vinod. Fast visual search using focused color matching - active search. *Systems and Computers in Japan*, 31(9):81–88, 2000. A translation of Trans. IEICE, J81-D2(9):2035–2042.

[6] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[7] W. Xiaohong and Z. Rongchun. A new method for image normalization. In *Proc. of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 356–359, May 2001.