

Accelerating Large Character Set Recognition using Pivots

Yiping Yang , Ondrej Velek, Masaki Nakagawa

Graduate School of Technology, Tokyo University of Agriculture and Technology
2-24-16 Naka-cho Koganei-shi, Tokyo, 184-8588, Japan

E-mail: yangyiping@yahoo.com, ovelek@hands.ei.tuat.ac.jp, nakagawa@cc.tuat.ac.jp

Abstract

This paper proposes a method to accelerate character recognition of a large character set by employing pivots into the search space. We divide the feature space of character categories into smaller clusters and derive the centroid of each cluster as a pivot. Given an input pattern, it is compared with all the pivots and only a limited number of clusters whose pivots have higher similarities (or smaller distances) to the input pattern are searched for with the result that we can accelerate the recognition speed. This is based on the assumption that the search space is a distance space. The method has been applied to pre-classification of a practical off-line Japanese character recognizer with the result that the pre-classification time is reduced to 61 % while keeping its pre-classification recognition rate up to 40 candidates as the same as the original 99.6% and the total recognition time is reduced to 70% of the original time without sacrificing the recognition rate at all. If we sacrifice the pre-classification rate from 99.6% to 97.7%, then its time is reduced to 35% and the total recognition time is reduced to 51.5% with recognition rate as 96.3% from 98.3%.

1. Introduction

Japanese, Chinese or Korean have a large character set with several thousands different categories, so that their character recognition takes more time than Western alphabet or numeral recognition. Therefore, to accelerate their recognition has been studied with practical importance. The two-stage architecture of candidates selection and final classification has been employed in many practical systems [1]-[6]. Candidate selection methods should be significantly faster than the final classification and select a limited number of candidates robustly with the effect that template patterns for the final classification to match with a input pattern is reduced from several thousands to several hundreds or even less. Consequently, the whole recognition process is accelerated. Candidate selection is made during recognition so it is a dynamic process.

The other possible method is to structure the search space so that the search for candidates can be made only to some portion in the search space. This can also accelerate the recognition. Since this is made before the recognition process, so it is a static process and we distinguish it from

the above-mentioned dynamic process.

In this paper, we take the latter approach and structure the search space of a large character set in order to speed up the recognition of handwritten Japanese characters.

Section 2 of this paper introduces the structuring the search space. Section 3 describes the off-line recognizer based on which the proposed method is evaluated. Section 4 presents the detailed design for implementing the method. Section 5 describes experiments and Section 6 analyzes the result. Section 7 concludes the paper.

2. Structuring the search space

For the large character set recognition, we consider to divide the feature space of the character categories into smaller clusters with regarding the centroid of each cluster as a pivot. Fig. 1 is a conception figure of the feature space drawn in two-dimensional space (although typical feature space for large character set recognition takes 256 or 512 dimensions). Note that each cluster is made up of different character categories rather than multiple templates of a single category.

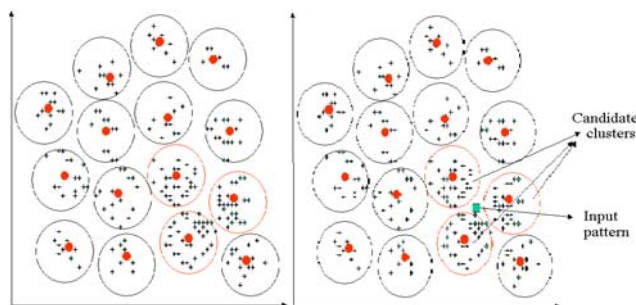


Fig. 1 Conception figure of structuring the search space.

Given an input pattern, it is compared with all the pivots and only the limited number of clusters whose pivots have higher similarities (or smaller distances) to the input pattern are searched for with the result that we can accelerate the recognition speed. This is based on the assumption that the search space is a distance space.

Problem remains here in selecting the number of clusters, i.e., the number of pivots and in selecting the clustering algorithm. They will influence the efficiency and accuracy.

3. Off-line recognizer

The off-line classifier used for this research represents each character as a 256-dimensional feature vector. It scales every input pattern to a 64x64 grid by non-linear normalization [7]. Then, it decomposes the normalized image into 4 contour sub-patterns representing directional feature of the 4 main orientations. Finally, it extracts a 64-dimensional feature vector for each contour pattern from the convolution with a blurring mask (Gaussian filter).

A pre-classification step precedes the actual final classification. Pre-classification selects 40 candidates with the shortest Euclidian distances between the categories' mean vectors and the test pattern.

The final classification employs a modified quadratic discriminant function (MQDF2) developed by Kimura [8] from traditional QDF. It was trained by training set composed of three databases, namely Nakayosi database [9, 10] (with 1,695,689 patterns of 4,438 categories), ETL9b database (607,200 patterns of 3,036 categories), and HP-JEITA database (1,917,480 patterns of 3,214 categories). All together these databases consist of 4,443 different categories, including digits, western characters, symbols, katakana, hiragana and Japanese Kanji characters.

While off-line databases ETL9b and HP-JEITA can be immediately used for training an off-line recognizer, the on-line database Nakayosi must be transformed to off-line format (bitmap images) first. We have employed a unique method for generating realistic Kanji character images from on-line patterns [11]. This method combines on-line patterns with a calligraphic stroke shape library, which contains genuine off-line patterns written with different writing tools. Since the artificially generated off-line images are combinations of on-line and actual off-line patterns they look very natural and realistic.

4. Detailed design

4.1 Details of clustering

We employ the LGB algorithm [12] for clustering since it is one of the simplest and effective methods. In order to start the algorithm, however, we have to define initial clusters or centroids, which somehow determines the result. We have tried the following three methods for selecting initial clusters, but our experiment results reveal that the influence of different initial centroids is very subtle. Therefore, we employed the simplest method (3), which is detailed here to some extent.

- (1) Select initial clusters according to the distance d between a template pattern and the origin point $(0, \dots, 0)$.
- (2) Select initial clusters according to only one axis of 256-dimensional vector space.
- (3) Select initial clusters according to the order of

characters in the dictionary. The detailed approach is:

- a) To divide all of patterns into "n" clusters equally according to the order.
- b) To set the centroid of each cluster as the initial.

4.2 Selection of candidate clusters

Given an input pattern, and compared with all the pivots, the next question is how to select candidate clusters to search. There are two ways to select them:

(1) We set a constant for the number of candidate clusters, and select l clusters whose centroids have from the shortest up to the l -th shortest Euclidian distances to the input pattern (Method-I).

(2) We find the nearest centroid with the distance d_{min} , and then we set a multiplying coefficient m ($m > 1$) so that all clusters with the distances less than $m * d_{min}$ become candidate clusters (Method-II).

4.3 An optimal number of clusters

It is difficult to find out an optimal number of clusters through inference. In the dictionary of our Japanese off-line recognizer, there are 4,443 template patterns and they can be divided into from 2 to 4,442 clusters. The optimal number of clusters may depend on the method for selecting original centroids (section 4.1) and on the method for selecting candidate clusters (section 4.2). We have to make experiments to determine the optimal number. They follow in section 5.

4.4 Management of clusters

After all the template patterns are divided into small clusters, we manage them so as to speed up the search. We store the following information in a file as the total number of clusters, the number of templates and the centroid of each cluster, indexes to all the templates in each cluster and the dimension of feature vectors. Thus when the search process is carried out, all the pre-calculated information is loaded from the file.

5. Preparation for experiments

5.1 Testing set and environment

We have created a testing set from the Kuchibue database, which is an established benchmark for Japanese handwritten characters. The set has 4,443 testing patterns with one randomly selected from each category.

We made experiments on a PC with an Intel Pentium3 CPU of 1GHz and 512M RAM employing Microsoft Windows NT.

5.2 Original state of the recognizer

Before introducing the structuring of the search space, the pre-classification produced 40 candidates with the pre-classification rate (the rate that the correct answer is within the candidates) as 99.6% in 150 sec. and the final recognition produced 98.3% in 220 sec.

6. Experiments

This section considers the appropriate number of candidate clusters and the method for selecting candidate clusters described in 4.2 through recognition experiments.

And then we will find out the best method for selecting original centroids, for selecting candidate clusters and we will investigate an optimal ratio between a number of all clusters and candidate clusters. Because their relationships are too complex, it is difficult to analyze them clearly at once, so the analysis is divided to several steps as following:

6.1 Fixed number of candidate clusters

This section considers the performances when the number of candidate clusters is fixed and determined according to Method-I described in 4.2.

6.1.1 Optimal number of candidate clusters from all

Table 1 shows performances of the pre-classifier and the total recognizer with respect to the various number of candidate clusters when the template vectors are divided into 500 clusters. Fig. 2 presents the relation between the rate and the time of the pre-classification. Here, pre-classification rate denotes the rate that a correct answer is included in the candidate character categories passed to the final classifier. Pre-classification time and recognition time is the time to process all the test set.

The pre-classification rate grows up significantly up to about 97.5% as the number of candidate clusters increases up to about 35, then grows up gradually up to the best rate of 99.6% as the number of candidate clusters increases up to 150. This is less than 1/3 of the total cluster number of

Table 2. Performances when template vectors are clustered into 500.

No. of clusters	500														
No. of candidate clusters	1	2	5	10	20	35	36	37	38	39	40	60	80	150	
pre-classification time (sec.)	35.6	35.7	37	39.8	44.2	50.3	51	52.3	54.4	55.1	56	64	68	91	
pre-classification rate (%)	46.5	68.8	81.1	89.4	95.1	97.5	97.5	97.6	97.6	97.7	97.8	98.6	99.1	99.6	
recognition time (sec.)	46.5	55.4	69.8	102	107	111	112	112	114	115	116	124	129	154	
recognition rate (%)	46.4	63	79.9	88.1	93.8	96.4	96.2	96.3	96.4	96.5	96.6	97.5	97.8	98.3	

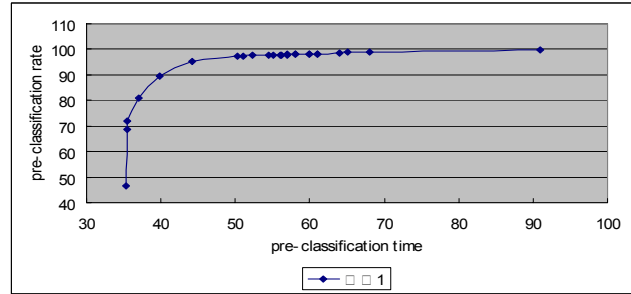


Fig. 2. Pre-classification performances for 500 clusters.

500, and the pre-classification time has been reduced to 91 sec. i.e., 61% from the original 150 sec. without sacrificing the rate at all. When we look at the effect to the total recognizer, 220 sec. is reduced to 154 (70%).

If we consider the optimal point as the point to produce the high rate while suppressing the time, Fig. 2 shows that it would be obtained when the time is around 53 sec. and the rate is about 97.5%, that is when the number of candidate clusters is 37.

Table 2 and Fig. 3 are made when template vectors were divided into 125 clusters. It shows almost the same relation and improvement as was discovered when the number of clusters was 500.

Table 1. Performances when template vectors are clustered into 125.

No. of clusters	125									
No. of candidate clusters	1	2	4	8	15	30	60	65	70	80
pre-classification time (sec.)	25.1	27.2	30.8	37.1	46.3	64.4	95.1	100.2	104.5	113.5
pre-classification rate (%)	49.7	67.6	81.5	91.6	96.8	99	99.5	99.6	99.6	99.6
recognition time (sec.)	41	57.3	93.1	102	115.9	130.7	161.1	166.5	171.8	180.8
recognition rate (%)	49.1	66.5	80.4	90.2	96.4	97.6	98.2	98.3	98.3	98.3

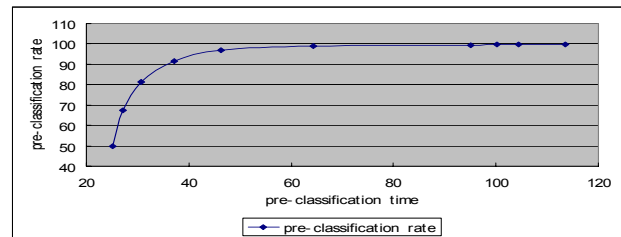


Fig. 3. Pre-classification performances for 125 clusters.

6.1.2 Optimal clustering size for Method-I

Table 3 shows the number of candidate clusters when the pre-classification rate around 97.6% is achieved for different clustering numbers for the template vectors.

Fig.4 shows that the pre-classification time increases if the cluster number is smaller than 125 or bigger than 810. In the range 125-810, however, the pre-classification time is stable and reduced up to 52.8 sec. i.e., 35% from the

original 150 sec. while the number of candidate clusters need to be linear to that of the clusters but it is only from 5 to 15% of the total number of clusters. When we look at the effect to the total recognizer, 220 sec. is reduced to 113 sec. (51%) while achieving 96.3% recognition rate.

Table 4. Performances when pre-classification rate is around 97.6%.

No. of clusters	2	60	125	190	250	310	500	625	750	810	1000	1375	1500	2000
pre-classification time (sec.)	96.2	65.3	56.4	54.1	53.2	54.4	52.8	56.1	66.5	56.3	65.1	73	76.8	86.8
pre-classification rate (%)	90.5	97.4	97.5	97.6	97.5	97.6	97.6	97.7	97.5	97.6	97.5	97.6	97.6	97.6
recognition time (sec.)	156	125	116	114	113	114	114	117	117	118	124	133	136	148
recognition rate (%)	86.6	96.3	96.4	96.4	96.3	96.4	96.4	96.4	96.4	96.4	96.4	96.4	96.4	96.4
No. of candidate clusters	1	12	15	24	27	32	38	41	38	39	46	46	48	44

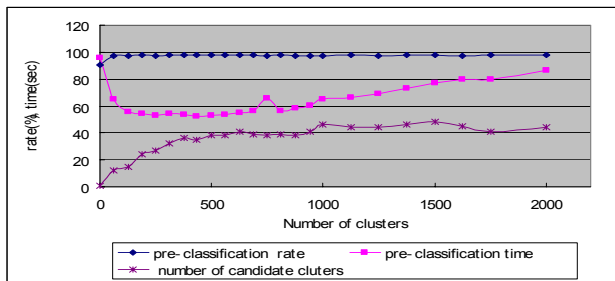


Fig. 4. Pre-classification settings satisfying 97.6% rate for Method-I.

6.2 Unfixed number of candidate clusters

This section considers the performances when the number of candidate clusters is not fixed and determined according to Method-II described in 4.2.

6.2.1 Optimal multiplying coefficient

Table 4 and Fig. 5 show the performances of the pre-classifier and the total recognizer with respect to the range of the multiplying coefficient m when the template vectors are divided into 500 clusters. When the multiplying coefficient m is 1.8, the same pre-classification rate as the original was achieved in 93.8 sec. that was 63% of the original 150 sec. and the same recognition rate was realized in 155.6 sec. (71%) from 220 sec.

6.2.2 Optimal clustering size for Method-II

Table 5 shows the pre-classification time and the total recognition time when the pre-classification rate around 97.7% is achieved for different clustering numbers (10 to 2,000) for the template vectors with the multiplying coefficient adjusted. Fig. 6 summarizes them in a figure.

Although the recognition time was linear to the pre-classification time in the meaningful range of the

Table 3. The recognition performances for multiplying coefficient m .

No. of clusters	500								
multiplying coefficient	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
pre-classification time (sec.)	35.2	36.7	38.5	43.1	50.4	59.2	70.9	83.5	93.8
pre-classification rate (%)	49.5	72.7	87.6	94.7	97.5	98.8	99.3	99.5	99.6
recognition time (sec.)	57.6	71.8	93.5	103.4	112.5	123.3	134.7	145.3	155.6
recognition rate (%)	48.8	79.7	86.5	93.5	96.2	96.5	98	98.2	98.3

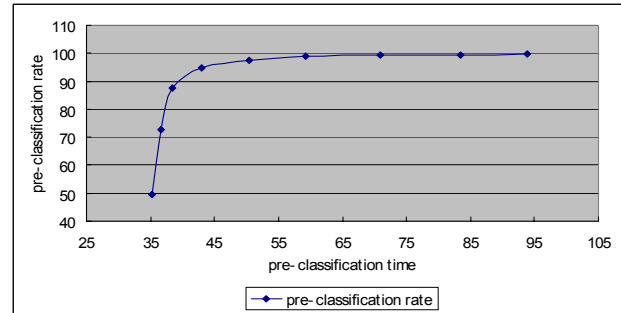


Fig.5; Pre-classification performances for multiplying coefficient m .

number of clusters in Method I, here we can see that the total recognition time is almost flat regardless of the pre-classification time.

It is easy to find that the least pre-classification time can be obtained while the number of clusters is around 250, and the pre-classification time increases fast as clusters are less than 250 and increases gradually as they are bigger than 250. For the recognition time, however, we find the least time when the number of clusters is 500 so that the best point for the total recognition time shifted from that of the pre-classification.

The reason of these can be explained as follows. When Method II is used to select candidate clusters, there might be a few clusters near the input pattern for some test patterns with each cluster having only a small number of templates (as the number of clusters increases, the number of templates in each cluster decreases), thus the total number of candidate templates could less than 40, which leads to the final recognition time reduced too. While the number of clusters is 250, although the pre-classification time is shortest, there are almost 40 candidate templates need to be transferred to the final classification for every input pattern. While the number of clusters is 500 or more, the pre-classification time is slightly longer, but the number of candidate templates is likely less than 40 with the result that the recognition time is less than that when the number of clusters is 250. As the number of clusters increases from 250, the pre-classification time increase gradually, but the final classification time decreases due to the above reason, so that the increase of the total recognition time is very small.

Table 5. Performances when pre-classification rate is around 97.7%.

No. of clusters	10	60	250	500	750	875	1000	1375	1500	1750	1875	2000
pre-classification time (sec.)	79.6	60.4	51.4	53.2	57	60.5	65	77	81	89	92	97
pre-classification rate (%)	97.7	97.6	97.7	97.7	97.7	97.7	97.7	97.6	97.7	97.7	97.7	97.7
recognition time (sec.)	150.7	131	123.3	113.3	115.8	116.3	118	119.1	123.6	123.7	124.3	132.5
recognition rate (%)	96.4	96.4	96.3	96.3	96.2	96.3	96.3	96.2	96.3	96.4	96.4	96.4
multiplying coefficient	1.23	1.34	1.35	1.355	1.375	1.375	1.385	1.4	1.41	1.415	1.42	1.42
New recognition time/original recognition time(220)	60	60	56	51	53	53	54	54	56	56	57	60

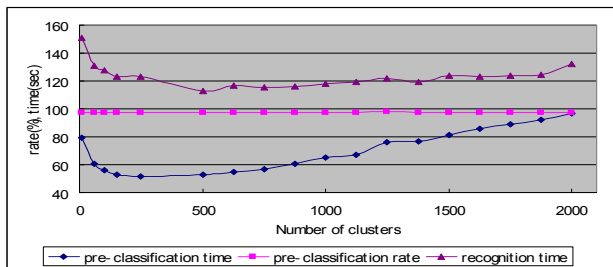


Fig.6. Pre-classification settings satisfying 97.7% rate for Method-II.

Although the characteristic of Method II is a bit different from Method I, we can also obtain the similar advantages as Method I.

When we 250 clusters, the pre-classification time is reduced to 51.4 sec. i.e., 34% from 150 sec. and the total recognition time reduced from 220 sec. to 123.3 sec. (56%) while achieving 96.3% recognition rate.

When we 500 clusters, the pre-classification time is reduced to 53.2 sec. i.e., 35% from 150 sec. and the total recognition time reduced from 220 sec. to 113.3 sec. (51%) while achieving 96.3% recognition rate. Since the total performance is more important, this condition should be selected.

The multiplying coefficient is stable to produce the best performances so that it should be set in range 1.3-1.45.

7. Conclusion

This paper presented our approach to accelerating large character set recognition by employing pivots into the search space. We divided the feature space of character categories into smaller clusters and derive the centroid of each cluster as a pivot. An input pattern is compared with all the pivots and only a limited number of clusters whose pivots have higher similarities (or smaller distances) to the

input pattern are searched for. We have designed two methods to select candidate clusters and applied them to pre-classification of a practical off-line Japanese character recognizer with significant effects to accelerate the pre-classification and the total recognition. As far as we have experimented, the two methods are competing and show little difference. We have also investigated relation among a number of clusters and that of candidate clusters to the pre-classification and the total recognition performances. There remains work to compare the two methods in more various conditions. We expect that if the number of candidates to the final classifier is changed, their competence will change as well.

6. References

- [1] S. Mori, K. Yamamoto, M. Yasuda: Research on machine recognition of handprinted characters, *IEEE PAMI*, Vol.6, No.4, 386-405, 1984.
- [2] T. Kumamoto, et al: On speeding candidate selection in handprinted Chinese character recognition, *Pattern Recognition*, Vol.24, No. XXX, 793-799, 1991.
- [3] T. H. Hildebrandt, W. Liu: Optical recognition of handwritten Chinese characters: advances since 1980, *Pattern Recognition*, Vol.26, No.2, 205-225, 1993.
- [4] C.-H. Tung, H.-J. Lee, J.-Y. Tsai: Multi-stage pre-candidate selection in handwritten Chinese character recognition systems, *Pattern Recognition*, Vol.27, No.8, 1093-1102, 1994.
- [5] Y.-H. Tseng, C.-C. Kuo, H.-J. Lee: Speeding up Chinese character recognition in an automatic document reading system, *Pattern Recognition*, Vol.31, No.11, 1601-1612, 1998.
- [6] C.-L. Liu and M. Nakagawa: Precise candidate selection for large character set recognition by confidence evaluation, *IEEE PAMI*, Vol. 22, No. 6, 636-642, 2000.
- [7] J. Tsukumo, H. Tanaka: Classification of handprinted Chinese characters using non-linear normalization and correlation methods, *Proc. 9th ICPR*, Roma, Italy, 168-171, 1988.
- [8] F. Kimura: Modified quadratic discriminant function and the application to Chinese characters, *IEEE PAMI* Vol.9, No.1, 149-153, 1987.
- [9] M. Nakagawa, et al.: On-line character pattern database sampled in a sequence of sentences without any writing instructions, *Proc. 4th ICDAR*, 376-380, 1997.
- [10] K. Matsumoto, T. Fukushima and M. Nakagawa: Collection and analysis of on-line handwritten Japanese character patterns, *Proc. 6th ICDAR*, Seattle, 496-500, 2001.
- [11] O. Velek, M. Nakagawa, C.-L. Liu: Vector-to-image transformation of character patterns for on-line and off-line recognition, *International Journal of Computer Processing of Oriental Languages*, Vol.15, No2, 187-209, 2002.

[12] Y. Linde, A. Buzo, and R. M. Gray: An algorithm for vector quantization design, *IEEE Trans. on Communications*, Vol. COM-28, 84-95, 1980.