

Characters Recognition Method Based on Vector Field and Simple Linear Regression Model

Tetsuya Izumi, Tetsuo Hattori,
Hiroyuki Kitajima, and Toshinori Yamasaki, Non-members

ABSTRACT

In order to obtain a low computational cost method (or rough classification) for automatic handwritten characters recognition, this paper proposes a combined system of two feature representation methods based on a vector field: one is autocorrelation matrix, and another is a low frequency Fourier expansion. In each method, the similarity is defined as a weighted sum of the squared values of the inner product between input pattern feature vector and the reference pattern ones that are normalized eigenvectors of KL (Karhunen-Loeve) expansion. This paper also describes a way of deciding the weight coefficients using a simple linear regression model, and shows the effectiveness of the proposed method by illustrating some experimentation results for 3036 categories of handwritten Japanese characters.

Keywords: Handwritten Characters Recognition, Feature Extraction, Vector Field, Fourier Transform

1. INTRODUCTION

Since there are very many kinds of categories (or pattern classes) in Japanese characters (Hiragana and Chinese characters) and so there are many similar patterns in those characters, it needs much computational cost, i.e., computing time and memory storage, to automatically recognize those handwritten character patterns at high correct recognition rate.

For this problem, many researches have been done in recent years [1]-[14]. Some of those researches are focusing how to define the similarity function (or distance function) that effectively discriminates the similar patterns, using a modification of Mahalanobis distance or so [2]-[5]. Some are devising ways how to absorb the error like deformation variations that appear with common tendency in handwritten character patterns [6]-[9]. And some are applying neural network technology to this kind of characters recognition as a large-scale perceptual problem, such as exclusive

learning network (ELNET) [12], multilayer perceptron [13], and large circuit network (LSNN) [14].

However, we consider that they still require considerably high computation cost for the automated recognition of all Japanese handwritten characters. So, in order to obtain a low cost recognition system with high accuracy, we think we still have to pursue simple and efficient rough classification based on more effective feature extraction and similarity measure.

In this paper, we propose a recognition method using a vector field as an rough classification one, aiming to effectively obtain the feature information on directions of character lines and their juxtaposition situation and so on [10], [11], [17]. The field is a two dimensional gradient vector field like a static electric field, which is constructed from a binarized input pattern by distance transform [15], [16].

Based on feature points in the vector field, we present two rough classification methods that depend on different representations for the distribution of feature points vectors: one is an autocorrelation matrix and another Fourier expansion on low frequency domain that can be interpreted as a complex-valued function. In each of the methods, the representation is expressed as high dimensional vector, and the similarity is defined as a weighted sum of the squared values of the inner product between input pattern and the reference patterns that are eigenvectors of KL (Karhunen-Loeve) expansion.

This paper also describes a way of deciding the weight coefficients based on simple linear regression. And, it shows the effectiveness of the proposed combined method by giving the experimental results that the correct recognition rate is 99.81% for learned samples (50 patterns/category) and 92.10% for unknown samples (150 patterns/category) in 3036 categories of handwritten Japanese characters (including Hiragana) using ETL9B (Electro Technical Laboratory in Japan) data.

Although we use the simple linear regression model for decision of weight coefficients in similarity, we find that it causes almost the same improvement as in the case of using neural network [12].

Manuscript received on January 3, 2005 ; revised on October 24, 2005.

The authors are with Graduate School of Engineering, Kagawa University, 2217-20 Hayashi-Cho, Takamatsu City, 761-0396 Japan. e-mail: s04d451@stmail.eng.kagawa-u.ac.jp, {hattori, kitaji, yamasaki}@eng.kagawa-u.ac.jp

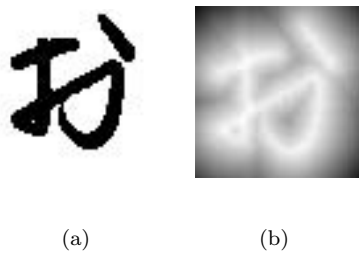


Fig.1: (a) Binarized Input Pattern. (b) Distance Transformation (64×64 pixels).

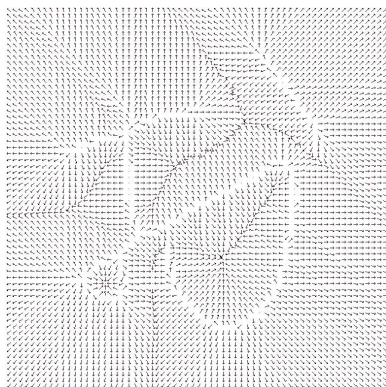


Fig.2: Vector Field.

2. FEATURE EXTRACTION

2.1 Vector Field

After a distance transformation is done for the binarized input pattern (Fig.1), and two-dimensional vector field is constructed by (1), where each vector corresponds to the gradient of the distance distribution at each point P , as shown in Fig. 2.

Let $T(P)$ and $V(P)$ be the value of distance transformation and two-dimensional vector at the point P , respectively. The $V(P)$ is defined as follows.

$$V(p) = \sum_{i=1}^8 \{T(P) - T(Q_i)\} \cdot e_i \quad (1)$$

where, Q_i ($1 \leq i \leq 8$) shows each point of the eight neighborhood of point P , and e_i shows a unit length vector in the direction from the point P to Q_i .

2.2 Normalization and Divergence

The length of each vector on the field is normalized to be one or zero by a threshold. By divergence operation on the field, source points and sink points can be extracted as feature points. Those are called “flow-out point” and “flow-in point”, respectively.

Then at the same time, feature point vectors are obtained (Fig. 3), which are vectors on the source points and sink points, what we call “flow-out point

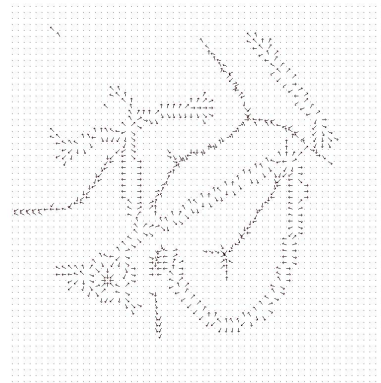


Fig.3: Feature Points Vector Field.

vectors” and “flow-in point vectors”, respectively in the same manner to the above naming.

As a characteristic property of the feature point’s vector field, flow-out and flow-in point vectors are located on the character lines (or strokes) and the background, respectively. They show not only the directional information on the strokes but also the juxtaposition situation of those strokes.

3. REPRESENTATION AND SIMILARITY

After the construction of the above feature points vector field, a combined method of two rough classifications is performed. The two classifications are based on different expressions of feature point’s vector field: one is Autocorrelation Matrix Representation Method (AMRM) and another Fourier expansion in low frequency domain method (shortly we call Low Frequency Domain Method (LFDM)).

3.1 Autocorrelation Matrix Representation Method (AMRM)

The neighborhood vector pattern X of a feature point vector, i.e., 2-dimensional vectors on 3×3 points centering the feature point, can be represented as a 9-dimensional complex vector in which each complex-valued component means 2-dimensional vector. So, the X can also be regarded as an 18-dimensional real vector. In order to express the neighborhood pattern X effectively, we use an orthonormal (orthogonal and normalized) system that can be made from a set of nine typical neighborhood patterns by the well-known KL (Karhunen-Loeve) expansion. Actually, a set of five orthonormal bases (or patterns) $\{\mu_i\}$ ($i=1, \dots, 5$) is obtained by the KL expansion.

Then, we can represent the neighborhood vector pattern X of a feature point P as the following 9-dimensional real vector $\chi(P)$, using the coordinate (i, j) of the point P and a set of the real-valued inner products between the neighborhood pattern and each basis of the above orthonormal system, i.e., $\{(X|\mu_i)\}$.

$$\chi(p) = \left(i, j, \frac{1}{n} \sum_{k=1}^n g(k, j), \frac{1}{n} \sum_{k=1}^n g(i, k), \langle X|\mu_1 \rangle, \dots, \langle X|\mu_5 \rangle \right)^T \quad (2)$$

$$g(i, j) = \begin{cases} 1 & ; \text{if coordinate } (i, j) \text{ is a feature point.} \\ 0 & ; \text{otherwise} \end{cases}$$

where $\frac{1}{n} \sum_{k=1}^n g(k, j)$, $\frac{1}{n} \sum_{k=1}^n g(i, k)$ means the frequency of feature points in each i direction (i.e., horizontal row) and j direction (i.e., vertical column), respectively, and n shows a size of the both side of image.

A set of $\{\chi(P)\}$ is extracted from the feature point vector field. Then, we express the distribution of the set of $\{\chi(P)\}$ in the 9-dimensional real vector space by an autocorrelation matrix in 9×9 size. Because the matrix is symmetric, it can be corresponded to a 45-dimensional real vector.

3.2 Low Frequency Domain Method (LFDM)

As another representation, Fourier expansion on the low frequency domain is used after the Fourier transform over the feature point vector field. The Fourier transform is described as follows. Let x and ω be two 2-dimensional real positional vectors on a real plane and the frequency domain, respectively. The Fourier transform $F(\omega)$ of the input pattern or complex-valued function $f(x)$ is defined in the following (3).

$$F(\omega) = \int_{R^2} f(x) e^{-j\langle \omega|x \rangle} d\mu(x) \quad (3)$$

where j and $d\mu(x)$ mean an imaginary number unit and an area element, respectively.

For example of the Fourier transform, a character pattern and its amplitude spectrum image on the frequency domain are shown in Fig. 4. In this figure, we can see that much information of the input pattern is in the low frequency domain (near the center of the image).

Actually, as a feature representation of input pattern, we use the information on 10×10 points around the original point in the frequency domain of the Fourier transform from the feature point vector field. Therefore, an input pattern is corresponded to a 100-dimensional complex vector.

3.3 Reference Pattern and Similarity

As aforementioned, an input pattern is represented as a correspondent feature vector in each of the two classification methods. Then, some vectors, which are some orthonormal bases made from eigenvectors



Fig.4: Input Pattern (Left, the Same Pattern in Fig. 1(a)) and Its Amplitude Fourier Spectrum Image (Right).

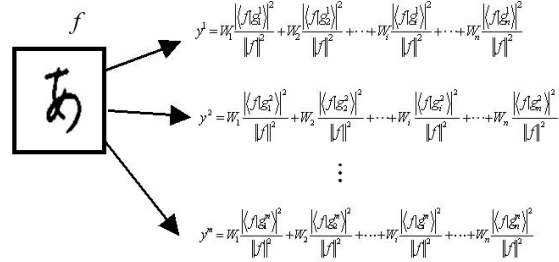


Fig.5: Similarity Computation between Input Pattern f and Reference Patterns of Each Category g^k ($k=1, \dots, m$), using the Same Weight Coefficient. (In the figure, y^k ($k=1, \dots, m$) means the similarity value.)

of KL (Karhunen-Loeve) expansion for learning patterns of each category (or pattern class) can be used as reference patterns for the category.

The similarity between input pattern and each category is defined as a weighted sum of the squared values of inner product between the feature vector and the reference patterns belonging to the category, as in the following (4) (as also shown in Fig. 5).

$$sim(f, g^k) = \sum_{i=1}^n \frac{W_i \times |\langle f|g_i^k \rangle|^2}{\|f\|^2} \quad (4)$$

where $\|f\| = \sqrt{\langle f|f \rangle}$, W_i ($W_i > 0$, $i = 1, \dots, n$) shows one of weight coefficients.

Let f and g be an input pattern (or feature vector) and a category, respectively. Let g_i^k ($i=1, \dots, n$) ($k=1, \dots, m$) be a set of reference patterns of the category g^k , and let $sim(f, g^k)$ be the similarity between f and g^k , the definition is given by (4).

4. DECISION OF WEIGHT COEFFICIENTS AND COMBINED METHOD

4.1 Weight Coefficients

After the similarity computation, input pattern is classified into a category that gives the highest similarity in the above computation. Therefore, the weight coefficient is very influential in the similarity evaluation. In many cases, a set of the coefficients is defined by the eigenvalues of the KL expansion as the

following (5), what we simply call *Eigenvalue Similarity*.

$$W_i = \frac{\lambda_i}{\lambda_1} \quad (5)$$

where λ_i ($\lambda_i > 0, i = 1, \dots, n$) shows the i -th largest eigenvalue in the KL expansion.

However, from our experiences in this kind of character recognition, the largest eigenvalue is often much greater than the other eigenvalues, and so the similarity is decided by the first term of the inner product between the input and the first reference pattern. As a result, the recognition rate is sometimes worse than the case when $W_i = 1$ for all i .

In order to decide the suitable weight coefficients for good recognition rate, we propose an iteration method based on a linear regression model, starting the initial condition that $W_i = 1$ for all i . We explain the concept of the iteration method in the following.

Let y^s be the similarity value between the input pattern f^s and the category g^s that is the same as input's one, as defined in (6). Let y^d ($s \neq d$) be the maximum value of the similarities between the input pattern f^s and the category g^d that is different from the input's one, as defined in (7).

$$y^s = \sum_{i=1}^n W_i \frac{|\langle f^s | g_i^s \rangle|^2}{\|f^s\|^2} \quad (6)$$

$$y^d = \max_d \left\{ \sum_{i=1}^n W_i \frac{|\langle f^s | g_i^d \rangle|^2}{\|f^s\|^2} \right\} \quad (7)$$

For the input pattern f^s , if $f^s < f^d$ ($s \neq d$), then it results in error recognition. In this case, we need to change the known coefficients $\{W_i\}$ so that the inequality situation does not occur in the similarity. Then, we update the coefficients by calculating a set of new coefficients $\{W_i^*\}$ using a linear regression model based on the following (8), (9). Those new coefficients are computed by putting the unchanged value y^s to the left hand side (LHS) of (8) for the same category g^s as input f^s , and also the same value to the LHS of (9) for the different category g^d .

$$y^s = \sum_{i=1}^n W_i^* W_i \frac{|\langle f^s | g_i^s \rangle|^2}{\|f^s\|^2} \quad (8)$$

$$y^d - |y^s - y^d| = \sum_{i=1}^n W_i^* W_i \frac{|\langle f^s | g_i^d \rangle|^2}{\|f^s\|^2} \quad (9)$$

Then, substituting the product of old and new coefficient into W_i , (i.e., $W_i^* W_i \rightarrow W_i$), the updated coefficients are obtained. Thus, we can iteratively search the suitable coefficients. The iteration terminates when no improvement of the recognition rate can be seen.



Fig. 6: Example of Some Japanese Handwritten Characters in 3036 Categories.

4.2 Synthesized Similarity

The aforementioned two classification methods are combined by using a synthesized similarity as defined in (10). Let x and y be the similarity value between the input pattern and each category in the AMRM and LFDm, respectively. The following sum of squared similarity (like Euclid norm) is used.

$$S_Similarity = x^2 + y^2 \quad (10)$$

5. EXPERIMENTATION

We have experimented the above three kinds of methods for 3036 categories of Japanese handwritten characters (total number of character patterns: 3036 200 patterns per category = 607,200) in ETL9B (Electro Technical Laboratory in Japan) database. The data used for experimentation includes not only Chinese characters but also Japanese Hiraganas as shown in Fig. 6.

In the experimentation, 50 samples (or character pattern) per category were used for learning, i.e., decision of reference patterns and the weight coefficients. Therefore, they are what we call *learning patterns*. Actually, we have decided that the number of the reference patterns (or eigenvectors of KL expansion) per category is eight, because we have considered that the number is appropriate for rough classification as a first step. The rest of 150 patterns are experimented as *unknown pattern*.

The specification of the computer, OS, Programming Language, etc. that we used in our experimentation is as follows.

- OS: Microsoft Windows XP Professional.
- CPU: Intel Pentium4 (2.4GHz).
- Main Memory: 1024Mbytes.
- Programming Language: Borland C++5.02J.

5.1 Preliminary Examination

Preliminary to the experimentation for all patterns, we have examined the effect applying the three kinds of methods, i.e., AMRM, LFDM and Combined Method (CM) that uses the aforementioned synthesized similarity, for a somewhat small set of character patterns as follows.

Data: 3036 categories \times 40 patterns.

Learning patterns: 20 samples/category.

Unknown patterns: 20 samples/category.

And, we have compared the results according to each case of the weight coefficients, i.e., No Weight (all values: 1.0), Eigenvalue, and the Linear Regression Model (LRM) as described before.

As for the number of eigenvectors (or standard patterns), we have examined the difference of correct recognition rate for the learning patterns and unknown patterns in AMRM. The results are shown in Fig. 7. Similarly, the difference of the recognition rate for the learning pattern and unknown patterns in LFDM is shown in Fig. 8. Also, the results by the three kinds of methods using weight coefficients based on the LRM are shown in Fig. 9.

From the results as shown in Fig. 7 and Fig. 8, we can see that the recognition rate does not grow so much even if the number of eigenvectors increases, in the case where Eigenvalue is used for the weight coefficient.

On the other hand, we can notice that the weight coefficient based on the LRM works effectively and that it tends to improve the recognition rate for unknown pattern. In addition, Fig. 9 shows that the CM exceeds both AMRM and LFDM in recognition rate. It also illustrates that the recognition rate by the CM becomes almost saturated at eight eigenvectors. Therefore, we estimate that eight standard patterns per category are enough for recognition in the CM.

5.2 Results for All Patterns

As aforementioned, in our experimentation for all patterns of ETL9B (3036 categories \times 200 samples), 50 samples/category are learning patterns and the rest 150 samples/category are unknown ones. As for the standard patterns, we have used eight patterns/category, because we think it appropriate for first step rough classification.

The experimental results are shown in Table 1 through 3. In order to compare the effects in the case of three kinds of weight coefficient, i.e., No Weight ($W_i=1$ for all i), Eigenvalue, and the coefficient by LRM, the results of the three cases are shown in the tables at the same time.

In those tables, the execution time means an average time needed to recognize one character pattern. And, the required memory storage means the neces-

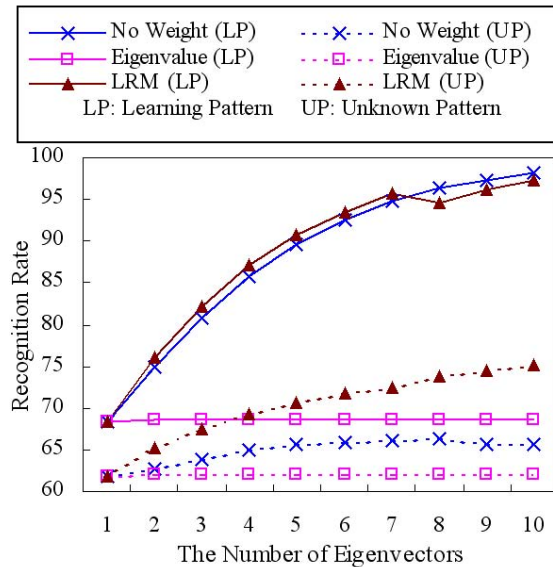


Fig. 7: Difference of Correct Recognition Rate in AMRM.

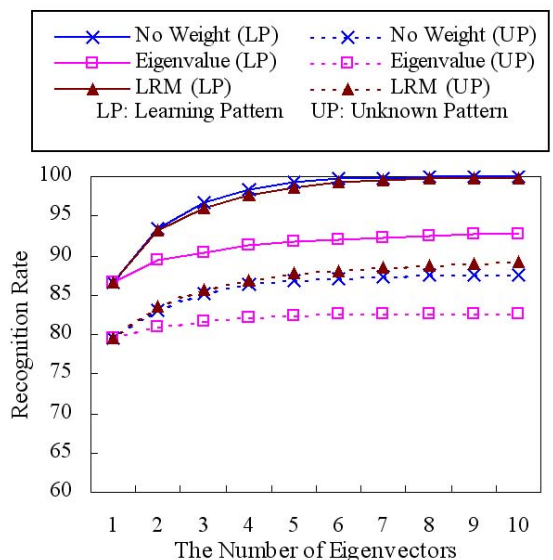


Fig. 8: Difference of Correct Recognition Rate in LFDM.

sary memory size for referencing the standard 3036×8 patterns in the computer system.

Table 1 and Table 2 indicate that LRM raises the recognition rate for unknown pattern up to 7.58% and 1.32% in AMRM and LFDM, respectively, comparing with No Weight.

And Table 3 shows that the recognition rate by CM using LRM is 99.81% for learning pattern and 92.10% for unknown pattern.

We consider that the resultant recognition rate is not extremely high, but very effective in the computational cost, comparing with the other methods as discussed later.

Table 1: Correct Recognition Rate by AMRM.

Input Pattern \ Weight Coefficient	No Weighth	Eigenvalue	Linear Regression Model (LRM)
<i>LearningPattern</i>	96.12%	66.02%	94.10%
<i>UnknowPattern</i>	63.97%	59.13%	71.55%

Execution time in LRM: 24 msec/pattern.

Required memory storage: 8.3 MBytes.

Table 2: Correct Recognition Rate by LFDM.

Input Pattern \ Weight Coefficient	No Weighth	Eigenvalue	Linear Regression Model (LRM)
<i>LearningPattern</i>	99.94%	91.67%	99.65%
<i>UnknowPattern</i>	86.14%	81.30%	87.46%

Execution time in LRM: 50 msec/pattern.

Required memory storage: 18.5 MBytes.

Table 3: Correct Recognition Rate by CM.

Input Pattern \ Weight Coefficient	No Weighth	Eigenvalue	Linear Regression Model (LRM)
<i>LearningPattern</i>	99.96%	95.30%	99.81%
<i>UnknowPattern</i>	90.17%	88.64%	92.10%

Execution time in LRM: 75 msec/pattern.

Required memory storage: 27 MBytes.

5.3 Discussion

In order to know the relation between the two similarities by AMRM and LFDM, we plot a dotted graph for some unknown patterns that belong to the same category.

For an input character pattern P , a point with coordinates (X, Y) is generated based on the definition as follows.

$$(X, Y) = (Simil(P, C) \text{ by AMRM}, Simil(P, C) \text{ by LFDM}) \quad (11)$$

where $Simil(P, C)$ means the similarity between the pattern P and a category C .

Because there are 3036 categories, the same number of points are generated for one input pattern P . When the category C is the same one that the pattern P belongs to, we call the similarity $Simil(P, C)$ “the Same Category Similarity”, otherwise “Different Category Similarity”.

Fig. 10 shows a dotted graph for ten input patterns (unknown patterns) that belong to the same category. That is, 10×3036 points are plotted in the graph. In the graph, points in the cluster (a) are generated by the Same Category Similarity, the other points in another cluster (b) by Different Category Similarity. This graph presents an example of the distribution by the Same Category Similarity and Different Category

Similarity. However, the distribution situations are similar for unknown patterns, as a whole.

From Fig.10, we can see that single similarity by AMRM or LFDM can hardly separate the two clusters, i.e., the same category and different ones. In fact, both the recognition rates are not so good for unknown patterns, as we have seen in Table 1 and Table 2.

But, also from the Fig.10, the synthesized similarity seems to work better to distinguish the two clusters. It has been verified by the experimental results, as shown in Table 3.

After all, we can consider that the two kinds of feature representations by AMRM and LFDM should be in a complementary relation.

More specifically speaking, the experimental result reveals that AMRM comparatively well discriminates curve patterns such as Japanese Hiraganas. So we can conjecture that AMRM stably represents the features of hole or loop and the juxtaposition situation of the curve strokes.

On the contrary, LFDM does not distinguish such curve patterns so much as AMRM. However, it works well for Chinese characters that contain many linear strokes. In this sense, we consider that AMRM and LFDM can function complementarily.

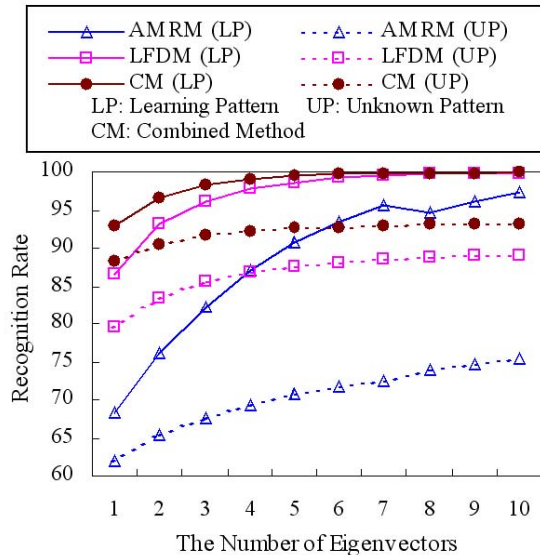


Fig. 9: Difference of Correct Recognition Rate by the three kinds of methods using the weight coefficients based on LRM in the preliminary examination.

5.4 Comparison of Computational Cost

In order to show that the CM is effective in low cost recognition, we try to compare the computational cost with other methods in the reference papers. The execution time of the CM based on LRM is 75 msec/pattern, and the required memory storage is 27 Mbytes, as shown in Table 3.

As for the methods in reference [1] and [2], the memory storage approximately requires 200 Mbyte and 406.3 Mbyte, respectively. The size of 406.3 Mbyte can be estimated by the following manner, according to the description in paper [2].

$406.3 \text{ Mbyte} = 196 \text{ (feature dimension)} \times 4 \text{ (floating point data size)} \times 4179 \text{ (the number of eigenvectors)} \times 3036 \text{ (categories)}$

The execution time reported in paper [1] and [2] is 2.04 sec/pattern (Model:HPC160) and 1.68 sec/pattern (Model:EWS4800/360), respectively.

In the other papers, the computational costs are not described concretely. So we only have very limited information to make comparison with the other methods.

However, as for pattern or feature matching based methods, there is a proportional relation between the memory storage and the execution time, in general. Then, we consider that the CM gives low cost handwritten characters recognition, because it is one of matching based methods, and because it requires comparatively small memory storage.

6. CONCLUSION

In this paper, we have presented two classification methods and a combined one for low cost handwritten characters recognition or rough classification, us-

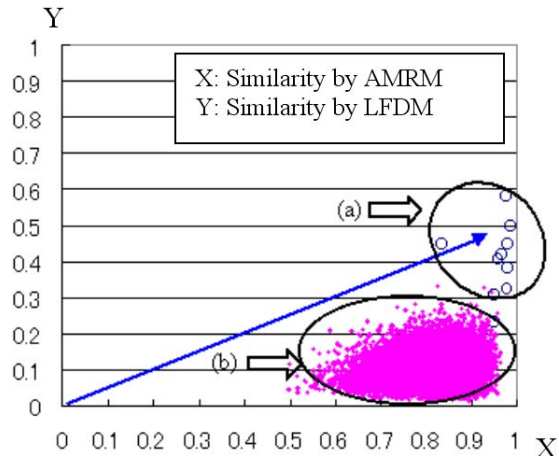


Fig. 10: Relational Graph of Two Similarities by AMRM and LFD. (a) Points by the Same Category Similarity. (b) Points by Different Category Similarity. Arrow corresponds to the distance between the origin and a point in (a), which means the synthesized similarity.)

ing features of the vector field. We have also proposed a decision method of the suitable weight coefficients in the similarity, using simple linear regression model (LRM). Moreover, we have revealed the experimental results. From the results, we can see that it is very effective to use the feature of the vector field and the decision of weight coefficients based on LRM. We have found that the linear regression method has shown almost the same improvement effect as in the case of using neural network [12]. Therefore, we expect that our feature points vector field method is promising and worth refining as a very effective and low computational cost method for hand written characters recognition.

References

- [1] Fang S. et al., "Fast and Precise Discriminant Function Considering Correlations of Elements of Feature Vectors and Its Application to Character Recognition," *IEICE Transactions*, Vol. J81-D2, No. 9, pp. 2027-2034, 1998 (in Japanese).
- [2] Masato S. et al., "A Discrimination Method of Similar Characters Using Compound Mahalanobis Function," *IEICE Transactions*, Vol. J80-D2, No. 10, pp. 2752-2760, 1997 (in Japanese).
- [3] Masato S. et al., "A Discriminant Method of Similar Characters with Quadratic Compound Mahalanobis Function," *IEICE Transactions*, Vol. J84-D2, No. 4, pp. 659-667, 2001 (in Japanese).
- [4] Masato S. et al., "A Discriminant Method of Similar Characters with Quadratic Compound Function," *IEICE Transactions*, Vol. J84-D2, No. 8, pp. 1557-1565, 2001 (in Japanese).
- [5] Takashi N. et al., "Accuracy Improvement by

Compound Discriminant Functions for Resembling Character Recognition," *IEICE Transactions*, Vol. J83-D2, No. 2, pp. 623-633, 2000 (in Japanese).

[6] Kazuhiro S. et al., "Accuracy Improvement by Gradient Feature and Variance Absorbing Covariance Matrix in Handwritten Chinese Character Recognition," *IEICE Transactions*, Vol. J84-D2, No. 11, pp. 2387-2397, 2001 (in Japanese).

[7] Tsuyoshi K. et al., "Handprinted Character Recognition Using Elastic Models," *IEICE Transactions*, Vol. J83-D2, No. 12, pp. 2578-2586, 2000 (in Japanese).

[8] Keitaro H. et al., "A Study of Feature Extraction by Information on Outline of Handwritten Chinese Characters -Peripheral Local Outline Vector and Peripheral Local Moment-," *IEICE Transactions*, Vol. J82-D2, No. 2, pp. 188-195, 1999 (in Japanese).

[9] Minoru M. and Toru W., "Handwritten Kanji Character Recognition Using Relative Direction Contributivity," *IEICE Transactions*, Vol. J84-D2, No. 7, pp. 1360-1368, 2001 (in Japanese).

[10] Tetsuo H. et al., "Handwritten Characters Recognition Using Vector Field Matching Based on Chessboard Distance Distribution," *IEICE Transactions*, Vol. J64-D, No. 12, pp. 1097-1104, 1981 (in Japanese).

[11] Tetsuo H. et al., "Absorption of Local Variations in Handwritten Character by an Elastic Transformation Using Vector Field," *IEICE Transactions*, Vol. J66-D, No. 6, pp. 645-652, 1983 (in Japanese).

[12] Kazuki S. et al., "A Fine Classification Method of Handwritten Character Recognition Using Exclusive Learning Neural Network (ELNET)," *IEICE Transactions*, Vol. J79-D2, No. 5, pp. 851-859, 1996 (in Japanese).

[13] Yuji W. et al., "High Accuracy Recognition for Handwritten Characters Using Multi Layered Perceptron with Squared Connections," *IEICE Transactions*, Vol. J83-D2, No. 10, pp. 1969-1976, 2000 (in Japanese).

[14] Yoshihiro H. and Hidefumi K., "A Neural Network with Multiple Large-Scale Sub-Networks and Its Application to Recognition of Handwritten Characters," *IEICE Transactions*, Vol. J82-D2, No. 11, pp. 1940-1948, 1999 (in Japanese).

[15] Rafael C. Gonzalez and Paul Wintz, *Digital Image Processing*, Addison-Wesley Publishing Company, 1977.

[16] D.H.Ballard and C.M.Brown, *Computer Vision*, Prentice Hall, 1982.

[17] T. Hattori, T. Yamasaki, Y. Watanabe, H. Sanada, and Y. Tezuka, "Distance based vector field method for feature extraction of characters and figures," *Proceedings of IEEE Systems, Man, and Cybernetics*, pp. 207-212, 1991.

[18] Tetsuya Izumi, Tetsuo Hattori, Hiroyuki Kitajima, and Toshinori Yamasaki, "Characters Recognition Method Based on Vector Field and Simple Linear Regression Model," *Proceedings of IEEE International Symposium on Communications and Information Technologies 2004 (ISCIT 2004)*, pp. 498-503, Sapporo, Japan, 2004.



Tetsuya Izumi received the B.E. and M.E. degree in Information Systems Engineering from Kagawa University, Kagawa Japan, in 2002 and 2004, respectively. He is currently a doctoral student in the same university. His research interests include Image Processing, Pattern Recognition and Human Interface Engineering.



Tetsuo Hattori received the B.E. and M.E. degree in Electrical Engineering from Doshisha University, Kyoto Japan, and Ph.D. in Electronics and Information Engineering from Osaka University, Osaka Japan. During 1983-1988, he worked for Toshiba Group Company, developing some image processors and automated visual inspection systems. Since after 1988, he has been with Kagawa University, Kagawa Japan. He has also been a visiting researcher at University of Connecticut, Connecticut USA, in 2001. Currently, he is a professor at the Department of Reliability-based Information Systems Engineering in Kagawa University. His research interests include signal and image processing, pattern recognition, numerical computation, and hardware implementation for fast computing.



Hiroyuki Kitajima received the B.Eng., M.Eng., and Dr.Eng. degrees in Electrical and Electronic Engineering from Tokushima University, in 1993, 1995 and 1998 respectively. He is now an Associate Professor of Reliability-based Information Systems Engineering, Kagawa University, Japan. He is interested in bifurcation problems.



Toshinori Yamasaki received his B.S., M.S. and Ph.D. degrees from the Faculty of Engineering of Osaka University in 1966, 1968 and 1984, respectively. He then joined the Faculty of Engineering of Kansai University. He became an associate professor on the Faculty of Education of Kagawa University in 1970. He was a visiting research member of the Information Science Laboratory, Berne University, Switzerland, in 1993. Subsequently, he served as a principal of Takamatsu Lower Secondary School. He has been a professor on the Faculty of Engineering of Kagawa University since 1998. His research areas include educational engineering, pattern measurement and recognition. He is a member of the Institute of Electronics, Information and Communication Engineers, the Information Processing Society, the Society for Information and Systems in Education and the Society for Systems, Control and Information.