'90 KACC 1990. 10. 26~27

AN INTELLIGENT SYSTEM FOR ISOMORPHIC TRANSFORMATION PATTERN RECOGNITION

†*Qiusheng XIE* †*Akira KOBAYASHI*

†Faculty of Engineering, Tokyo Institute of Technology
12-1, O-okayama 2-Chome Meguro-ku,Tokyo 152, Japan

ABSTRACT

To recognize isomorphic transformation patterns, such as scale-change, translation and rotation transformed patterns, is an old difficult but interesting problem. Many resaerches have been done with a dominant approach of normalization by many eminent pioneers. However, there seems no a perfect system which can even recognize 90°-multiple rotation isomorphic transformation patterns for real needs. Here, as a new challenge, we propose a method of how to recognize 90°-multiple rotation isomorphic and symmetry isomorphic transformation patterns.

1. INTRODUCTION

Usually, a pattern recognition system is composed of two parts, feature extraction part(FEP) and recognition part(RP)[1]. Because a pattern may have many manifestations due to its geometrical isomorphic transformations and stochastic variations, a feature extraction mechanism is often demanded to have some performances of normalization. The geometric aspect can be realized by designing pattern-to-feature mapping process which can absorb the transformation. And on the other hand, the stochastic aspect can be accomplished through performing KL-transformation or PCA(Principal Component Analysis). When all these are achieved, pattern recognition can be simplified to a minimum distance matching with learned standard patterns in some kind of feature space[2]~[11].

In the newly developed intelligent system, we utilize Nth-order Autocorrelation functions of patterns(AFP) to constitute the FEP, where the domain of AFP is confined to a 7×7 lattice and its order is selected to be $N = 2$. By virtue of the characteristics of autocorrelation, it is found that the AFP can be obtained with a set of mask patterns by matching them with object patterns respectively and counting up the number of their matchings instead of using conventional methods. By taking this way, the FEP is realized with a special circuit using parallel processing algorithm. Since the 2nd-order autocorrelation function of a pattern within a 3×3 lattice domain can only has 25 independent values and 5×5 as well as 7×7 is only the expansion of 3×3, patterns are mapped into a 75-D feature space in real time in the system. More than that, based on the property of autocorrelation, it is clear that every translation isomorphic transformation pattern set is normalized into one pattern. So the system is invariant to translation transformation of patterns. However, as our objective is to realize 90°-multiple rotation isomorphic and symmetry isomorphic patterns, we take use of the isomorphic relation between object pattern's rotation-symmetry and mask pattern's rotation-symmetry to perform another mapping. this is realized simply by summing up the feature values correspoding to the mask patterns which form a 90°-multiple rotation-symmetry isomorphic set. As the last result of the geometric normalization, patterns are mapped into a 22-D feature space.

To perform normalization to stochastic variation of patterns, we perform PCA on the 22-D feature space. By supervised learning, standard patterns of every pattern class is learned concentratively. Then, PCA is performed on the covariance matrices of total data (Ω_t) and every class adat (Ω_i), so that linear transform matrices ($\overline{A_t,A_i}$) corresponding to them are calculated using their eigenvectors and eigenvalues in order to map patterns into Mahalanobis spaces(S_t,S_i) and get their centroids (CE_t,CE_i). After that, covariance matrices of every calss are summed up to get within-class convariance (Ω_w). By simultaneous diagnolizing the Ω_w and $\Omega_b = \Omega_t - \Omega_w$, linear transform matrix (A_{bw}) is acquired, by which patterns are mapped into a space(S_{bw}) with a meaning of largest between-class-covariance and whitened within-class-covariance. In S_{bw}, centeroid of each learned pattern class($CE_{bw,i}$) is calculated as the standard pattern. Becase PCA is also a dimension compression method, the dimensions of S_t,S_i,S_{bw} are deminished to lower than 22. These complete the stochastic normalization process and the parameters thus obtained become the knowledge-base.

In pattern recognition phase, unkown pattern's 22-D feature vector is firstly mapped into S_t, where its distance to CE_t is calculated to determine if it is in the range of learned patterns with a criterion of χ^2 distribution of some degrees of freedom. If not in, it will be rejected. If in, it is then mapped into *Sdubi* to verify if it belongs to any class. Here if it doesn't belong at least one class, it will also be rejected. After these two checks, if pattern is receipted, it then is mapped into S_{bw} to determine which class it belongs by minimum distance(to $CE_{sd,i}$) matching method. In the last decision making, if the minimum distance is large than a χ^2 ditribution criterion, the pattern will also be rejected.

The above matters are verified with experiment, where 3 classes of chinese characters are used as main object patterns. The result shows that this system is usful for application such as visual inspections.

2. PRINCIPLE OF FEATURE EXTRACTION

Usually, a real value pattern can be expressed with a two-variable real function $f(x,y)$, where (x,y) is a point in Cartesian coordinates. From this, the Nth order autocorrelation function of any pattern can be defined as follows.

$$\Psi_f(\tau_1,\tau_{y1};\dots;\tau_{xN},\tau_{yN})$$
$$=\iint f(x,y)f(x+\tau_{x1},y+\tau_{y1})\dots f(x+\tau_{xN},y+\tau_{yN})\,dxdy \tag{1}$$

where $(\tau_{xi},\tau_{yi});i=1,2,\dots,N$ is the ith translation vector.

It is obvious that function Ψ_f is tramslation invariant in the sense that $f(x,y)$ and $\overline{f}(x,y)=f(x+\tau_x,y+\tau_y)$ have the same Nth-order autocorrelation function. This kind of functions of patterns was first proposed for pattern recognition by Horwitz and Shelton using optical system[12]. And its detail properties were thoroughly discussed by McLaughlin and Raviv[13]. It is indicated that other than the first order autocorrelation function, for almost all the patterns, the second order (and higher even order) autocorrelation functions are unique. that is, an autocorrelation function is equal to another if and only if their corresponding patterns have a tanslation relation. Such that, the second order (and higher even order) autocorrelation function can be used as a method of feature extraction for its characteristics of uniqueness.

However, as feature extraction in some meanings is also a process of information compression from higher dimension to lower dimension, the domain of autocorrelation function must be confined to a small range. Otsu and others suggested to use local domain near origin such as a 3×3 lattice[14]. In the case of a $m\times m$ lattice domain, a Nth order autocorrelation function can have $(m\times m)^N$ values. So, even for the 3×3 domain, the dimension of feature space is too high to operate. From this point of view, we selected the order to be N=2, so that the feature space may have a dimension of 81 for 3×3 lattice domain. But, because of order degeneration and translation isomerism, the dimension can be degenerated to 25. Table 1 shows the coordinates of the confined domain and their isomeric ones in the case of taking the coordinates as Fig.1. In order to make the feature space have a higher recognizability, we simply extend the 25 coordinates mentioned above to a 5×5 and a 7×7 lattice frame-and-center domain. Fig.2 shows the total domain illustrated with a 7×7 lattice. Such that we can totally get 75 kinds of independent translation coordinates, that is to say, the dimension of feature space will be 75. This can be considered as neither too high nor too low a feature space for microcomputer to process.

Table 1 The relation between mask patterns and translation vectors of autocorrelation

Mask	0th-	1st-	2nd-order	
	(τ_{x1},τ_{y1})	(τ_{x1},τ_{y1})	$(\tau_{x1},\tau_{y1})(\tau_{x2},\tau_{y2})$	
M_1	(-,-)	(0,0)	(0,0)(0,0)	
M_2		(1,0)	(1,0)(1,0)	(-1,0)(-1,0);(0,0)(1,0);(0,0)(-1,0); (1,0)(0,0);(-1,0)(0,0)
M_3		(1,1)	(1,1)(1,1)	(-1,-1)(-1,-1);(0,0)(1,1);(0,0)(-1,-1); (1,1)(0,0);(-1,-1)(0,0)
M_4		(0,1)	(0,1)(0,1)	(0,-1)(0,-1);(0,0)(0,1);(0,0)(0,-1); (0,1)(0,0);(0,-1)(0,0)
M_5		(-1,1)	(-1,1)(-1,1)	(1,-1)(1,-1);(0,0)(-1,1);(0,0)(1,-1); (-1,1)(0,0);(1,-1)(0,0)
M_6			(1,0)(-1,0)	(-1,0)(1,0)
M_7			(1,1)(-1,-1)	(-1,-1)(1,1)
M_8			(0,1)(0,-1)	(0,-1)(0,1)
M_9			(-1,1)(1,-1)	(1,-1)(-1,1)
M_{10}			(1,0)(-1,-1)	(-1,-1)(1,0)
M_{11}			(1,1)(0,-1)	(0,-1)(1,1)
M_{12}			(0,1)(1,-1)	(1,-1)(0,1)
M_{13}			(-1,1)(1,0)	(1,0)(-1,1)
M_{14}			(1,1)(-1,0)	(-1,0)(1,1)
M_{15}			(0,1)(-1,-1)	(-1,-1)(0,1)
M_{16}			(-1,1)(0,-1)	(0,-1)(-1,1)
M_{17}			(-1,0)(1,-1)	(1,-1)(-1,0)
M_{18}			(1,0)(0,-1)	(1,1)(0,1);(0,1)(1,1);(-1,0)(-1,-1); (-1,-1)(-1,0);(0,-1)(1,0)
M_{19}			(1,1)(1,-1)	(1,-1)(1,1)
M_{20}			(0,1)(1,0)	(1,0)(0,1);(-1,1)(-1,0);(-1,0)(-1,1); (0,-1)(1,-1);(1,-1)(0,-1)
M_{21}			(-1,1)(1,1)	(1,1)(-1,1)
M_{22}			(0,1)(-1,0)	(-1,0)(0,1);(1,0)(1,1);(1,1)(0,1); (-1,-1)(0,-1);(0,-1)(-1,-1)
M_{23}			(-1,1)(-1,-1)	(-1,-1)(-1,1)
M_{24}			(-1,0)(0,-1)	(0,-1)(-1,0);(1,0)(1,-1);(0,1)(-1,1); (-1,1)(0,1);(1,-1)(1,0)
M_{25}			(-1,-1)(1,-1)	(1,-1)(-1,-1)

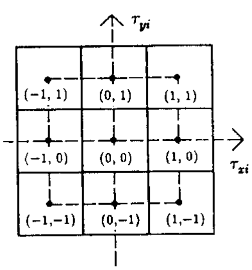

Fig.1 3×3 lattice domain

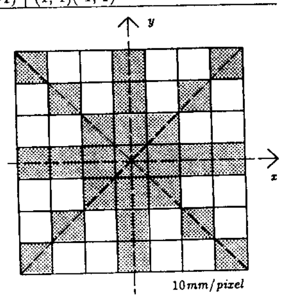

Fig.2 7×7 lattice domain

As a common knowledge, to calculate Nth order autocorrelation function is a very time exhausting thing. So, a high spead method based on hardware parallelism is desired. Because the calculation of autocorrelation function of any pattern $f(x,y)$ with respect to a set of N translation vectors is simply a process of shifting the pattern corresponding to those translation vectors to make out a product pattern of N+1 overlap patterns and then integrating it over the whole domain, the process can be replaced with a mask pattern scanning, where the N translation vectors are distributed on a small lattice. The scanning is taken pixel by pixel over the whole scene, and at each pixel the product of pattern value corresponding to a translation vector on the mask pattern is calculated. The sum of those product becomes the autocorrelation function value vs. the very set of translation vectors. Especially in the case of binary pattern, this process is only a process of calculating mask pattern histogram if mask pattern is designed to be a black and wight one where black pixels present translation vectors and white pixels means "don't care". Since binarization is included in the preprocessing, pattern's autocorrelation is actually its bianry pattern's. It is clear that mask pattern histogram is easy to be realized with hardware parallelism. In deed, this kind of device is developed to extract pattern's autocorrelation feature in real time. Fig.3 shows the mask patterns used to extract feture vectors of 75-D. And, we describe the 75-D feature vector as follows.

$$\Xi = (\xi_1,\xi_2,\ . . .\ ,\xi_{75}) \tag{2}$$

Here what we must pay attention is that, ξ_1, ξ_{26} and ξ_{51} are the same value.

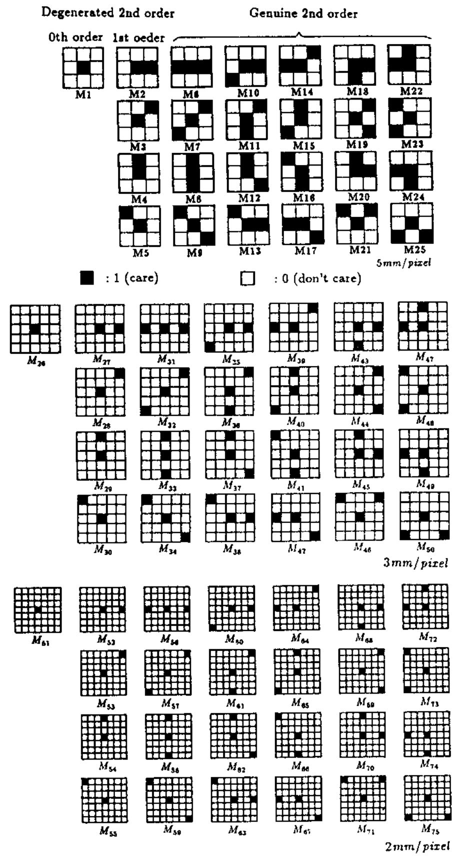

Fig.3 Mask patterns used for feature extraction

3. ABOUT ISOMORPHIC TRANSFORMATION PATTERN

3.1. WHAT IS ISOMORPHIC TRANSFORMATION

Let $f(x,y)$ presents a 2-D real pattern, the transformation between any two patterns expressed by the following formula is called pattern's Isomorphic Transformation(IT).

$$T_i(\alpha,A,\beta) \circ f(x,y) = \alpha f(A\begin{bmatrix} x \\ y \end{bmatrix} + \beta) \tag{3}$$

where, $T_i(\alpha,A,\beta)$ means ith kind of transformation. And α means luminance factor, A means linear transformation matrix, β means translation transformation matrix, respectively.

It is clear that every interesting geormetric characteristic of patterns gives a kind of isomorphic transformation. For example, pattern's translation forms translation isomorphic transformation and pattrn's rotaion forms rotation isomorphic transfomation. If we define the patterns which are generated from a standard pattern according to some kind of isomorphic transformation as a pattern calss named isomorphic transformation patterns(ITP), we can line out an interesting pattern recognition problem which has paramount significance of application. As a matter of fact, every kind of ITP is caused by an operation of transformation group if we deal the problem with group theory.

3.2. HOW TO RECOGNIZE ITP

As we said above that thinking every ITP as a pattern class is an interesting apllication problem for visual inpection, here we describe how to recognize it.

To recognize ITP, a simple method is to extract invariant features of ITP, so that every pattern belonging to the same class can have the same feature vector and different classes have different feature vectors. By this way, we can easily constitute a pattern recognition system for classifying ITP. However, it's difficult to extract features invariant to some kind of pattern transformation from pattern directly. Actually, our mask pattern histogram method is only invariant to pattern translation explicitly. So, to find out some kind of implicit invariance of features to some kind of pattern transformation looms important.

Because a class of some kind of ITP can be considered as generated from a standard pattern by some kind of transformation group $G = \{g_i \mid i=0,1,\ldots,n\}$, where n is finite or infinite, if the G can be quasiisomorphicly mapped in to another kind of transformation group $\hat{G}$ which operates on feature vector sets, an algorithm of feature vector transformation or amalgamation can be worked out for constructing a kind of ITP recognition system. The quasiisomorphic mapping from G to $\hat{G}$ can be illustrated with Fig.4.

$$\begin{tikzcd} f \arrow[r, "\Phi:\Psi(f)"] \arrow[d, "G"'] & \Xi \arrow[d, "\hat{G}"] \\ G\mathrm{O}f \arrow[r, "\Phi:\Psi(G\mathrm{O}f)"'] & \hat{G}\mathrm{O}\Xi \end{tikzcd}$$

Fig.4 Quasiisomorphic mapping from G to $\hat{G}$

Therefore, if we can find a feature measure space which satisfies the if-only-if condition of Fig.4 with respect to some kind of ITP, we can simply work out a recognition system for this kind of ITP.

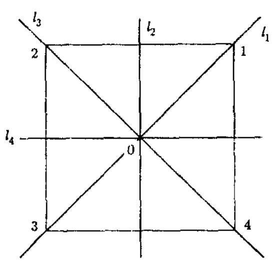

Fig.5 Symmetry of square

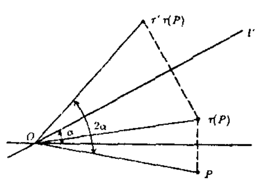

Fig.6 Relation between symmetry & rotation

3.3. 90°-MULTIPLE ROTATION AND SYMMETRY ITP

Fig.5 shows a square with a rotation center "O" and four symmetry axes. we note the transformation group which transforms the square to itself after a rotation or symmetry transformation as G_4. Obviously, this group contains eight elements, that is, the $e,\sigma,\sigma^2,\sigma^3(\sigma^4=e)$ and the $\tau_1,\tau_2,\tau_3,\tau_4$. Here, the former four correspond to the rotation of the square with a degree of 0°,90°,180° and 270° about the center "O", and the latter four correspond to the symmetries with respect to l_1,l_2,l_3,l_4 respectively. If we take the advantage of the relation between symmetry and rotation as shown in Fig.6, we can obtain the following results. That is, if $\tau=\tau_1$, then $\tau_2=\sigma\tau,\tau_3=\sigma^2\tau,\tau_4=\sigma^3\tau$. Such that, we can write G_4 as follows.

$$G_4 = \{e,\sigma,\sigma^2,\sigma^3,\tau,\sigma\tau,\sigma^2\tau,\sigma^3\tau\} \tag{4}$$

In this paper we deal with the problem of how to recognize ITP generated by the very G_4 defined above. Indeed, every calss of this kind of ITP contains eight patterns as illustrated in Fig.7.

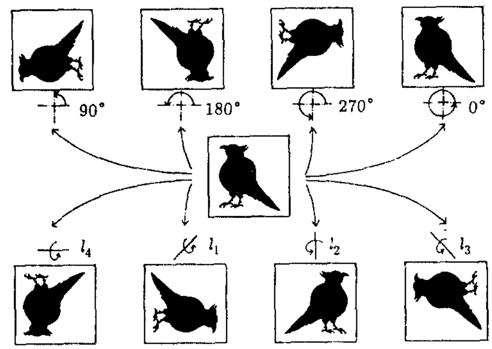

Fig.7 The eight Isomorphic transformation patterns generated by G_4

3.4. ABOUT THE MASK PATTERNS

Because mask pattern histogram method is used in our system to extract pattern's features and it isn't invariant to the G_4-ITP, the mask patterns have to have some kind of property which satisfies the condition shown in Fig.4. Since a feature corresponds to a mask pattern in mask pattern histogram method, mask pattern sets included in Fig.3 must be under an operation of transformation group $\overline{G}$ which is quasi-isomorphic with G_4. As a matter of fact, within the 75 mask patterns in Fig.3 there are 22 mask pattern sets which are generated by G_4. Therefore, the features obtained by mask pattern histogram must include 22 feature sets which are generated by a permutation group $\hat{G}$ of order 8. So our musk patterns shown in Fig.3 satisfy the condition shown in Fig.4.

3.5. DEDUCTING INVARIANT FEATURE VECTOR

Table 2 shows the 22 sets of isomorphic transformation mask patterns under G_4. The mask patterns in the first set are actually the same one. And the sets which contain only two or four mask

Table 2 Isomorphic Transformation Mask Pattern Sets

NO.	Mask Pattern Sets
1	$\{M_1(M_{26},M_{51})\}$
2	$\{M_2,M_4\}$
3	$\{M_3,M_5\}$
4	$\{M_6,M_8\}$
5	$\{M_7,M_9\}$
6	$\{M_{10},M_{11},M_{12},M_{13},M_{14},M_{15},M_{16},M_{17}\}$
7	$\{M_{18},M_{20},M_{22},M_{24}\}$
8	$\{M_{19},M_{21},M_{23},M_{25}\}$
9	$\{M_{27},M_{29}\}$
10	$\{M_{28},M_{30}\}$
11	$\{M_{31},M_{33}\}$
12	$\{M_{32},M_{34}\}$
13	$\{M_{35},M_{36},M_{37},M_{38},M_{39},M_{40},M_{41},M_{42}\}$
14	$\{M_{43},M_{45},M_{47},M_{49}\}$
15	$\{M_{44},M_{46},M_{48},M_{50}\}$
16	$\{M_{52},M_{54}\}$
17	$\{M_{53},M_{55}\}$
18	$\{M_{56},M_{58}\}$
19	$\{M_{57},M_{59}\}$
20	$\{M_{60},M_{61},M_{62},M_{63},M_{64},M_{65},M_{66},M_{67}\}$
21	$\{M_{68},M_{70},M_{72},M_{74}\}$
22	$\{M_{69},M_{71},M_{73},M_{75}\}$

patterns are degenerated ones caused by some kinds of symmetry. It is obvious that if we sum up the features correspond to each set of isomorphic transformation mask patterns, the compressed feature vector of 22-D will be invariant to G_4-ITP. So that, every pattern class as shown in Fig.6 could be mapped into a distinct point in the 22-D feature space if stochastic variations aren't taken into account. Therefore, through configurating a stochatical algorithm of pattern recognition, we can build a system for recognizing G_4-ITP simply.

4. STOCHASTIC PATTERN RECOGNITION ALGORITHM

Although in principle, every class of G_4-ITP could be mapped into a distinct point in the 22-D feature space through the process mentioned above, it is difficult to be realized with hardware dure to some kinds of stochastic noises during the process of standardization, digitization and feature extraction. So, it is inevitable to take the stochastic variation of feature vectors into account. Furthermore, as a pattern class usually contains not only a kind of pattern but also some similar patterns of it, we have to compose and select out the powerful feature in order to have a robust and fast recognition system.

Usually, in order to enable a pattern recognition system to recognize object patterns, one must at first teach some samples of the object patterns to be recognized to the system. And after the learn process, the system must be able to recognize each learned pattern and its similar ones based on the learned knowledge. Furthermore, it could also reject the patterns not learned.

From the above points of view, we use PCA to compose the features and at the same time compress the dimension of feature space. In order to have a powerful rejecting mechanism, multi-layer of rejection decision making is used. The detail is as follows.

Let's condider a U-class pattern recognition problem. In the teaching process, we teach each standard pattern of every class several ten times to the maching. The teaching contents are 22-D Feature vectors and class numbers. Such that, A data matrix of total patterns D_t and data matrices of the patterns of each class D_i ;$i=1,...,U$ are acquired. Then each D's covariance matrix, that is, Ω_t and Ω_i are calculated as R-problem. After that, we conduct PCA to Ω_i ;$i=t,1,...,U$ to get linear transformation matrices A_i ;$i=t,1,...,U$ for composing features and compressing feature space dimension. In fact, the PCA operation is a problem of eigenvalue and eigenvector of Ω_i, that is,

$$\lambda_k\Omega = \lambda_k\zeta_k \tag{5}$$

where λ and ζ present eigenvalue and eigenvector respectively. Since λ_k is indeed the variance of transformed data along the axis of ζ_k, usually the eigenvectors corresponding to the largest several eigenvalues are used in order to achieve the feature composition and dimension compression. In our system, we use only the largest two eigenvalues' eigenvectors to constitute the A_i, so every A_i is a 22×2 matrix. However, as this kind of A_i is only a transformation for diagonalizing covariance matrix, to build rejection decision making rule it is necessary to have the new spcae whitened. This can be done by multiplying the A_i with the $\Lambda_i^{\frac{1}{2}} = diag(\lambda_{i1},\lambda_{i2})^{\frac{1}{2}}$ matrix, that is,

$$\overline{A_i} = A_i\times\Lambda_i^{\frac{1}{2}} \tag{6}$$

If we transform each data matrix D_i through $\overline{A_i}$, then the distance between any two point in the new feature space is nothing but Mahalanobis distance which is invariant to scale change[15]. Furthermore, if we hypothesize that the patterns in the new feature space are ditributed as normal distribution, it is clear that the data after centroidization must follow χ^2 distribution with a freedom of degree two. Since the freedom degree two χ^2 distribution's probability density function is as follows,

$$p(\chi^2) = \frac{1}{2}e^{-\chi^2/2} \quad ;(0\leq\chi^2<\infty) \tag{7}$$

we can build an absolute criterion for rejecting abnormal patterns by intergrating $p(\chi^2)$ from 0 to some upper limit value to which the decision making confidence is nearly 1. It is easy to calculate that if we select the confidence to be 0.999 the criterion, that is, the upper limit must be 13.815.

From the above items, two layers of rejection mechanism can be obtained. First, we use $\overline{A_t}$ and D_t's transformed centroid $CE_t = (CE_{t1},CE_{t2})$ to build the first rejection layer. Based on this, when an unknown object pattern is given to the machine, after being transformed by $\overline{A_t}$ to the 2-D whitened feature space, its square distance to the learned center CE_t is calculated. Because the distance is nothing but χ^2 value, if it is above 13.815, the pattern will be rejected as unlearned one. Otherwise, it will be received for next discrimination.

Second, we use $\overline{A_i}$ and D_i's transformed centroid $CE_i = (CE_{i1},CE_{i2})$ to build the second rejection layer. The decision making is as the first one, but because there are U 2-D whitened feature spaces, only if the unknown pattern is rejected by all the U spaces it will be recjected. Otherwise, it will be received as learned pattern or the similar ones.

Obviously, in the above two layers of rejection mechanism, $(\overline{A_i},CE_i)$;$i=t,1,...,U$ are the learned knowledge from teached sample patterns.

Hereafter, we must build the final rejection layer. This layer must also do recognition and inference. To do this, we first build the within-class and between-class covariance matrices Ω_w and Ω_b. These can be calculated with Equation (8) and Equation (9).

$$\Omega_w = \sum_{i=1}^{U}\frac{n_i}{n_t}\Omega_i \tag{8}$$

$$\Omega_b = \Omega_t - \Omega_w \tag{9}$$

where, n_i and n_t are the sample numbers of each class and total, respectively.

Then, we perform simultaneous diagonalization on Ω_w and Ω_b, that is

$$\Omega_b A_{bw} = \Omega_w A_{bw}\Lambda, \quad A_{bw}^{Tr}\Omega_w A_{bw} = I \tag{10}$$

This is an eigen equation problem. Where, Λ is the diagonal matrix of eigenvalues and I denotes the unit matrix. At fact, this is a process of getting an optimized new feature space in which each class's variance is whitened and the between class variance gets maximum. Here, we also use two eigenvectors corresponding to the largest eigenvalues to compose a dimension compression matrix A_{bw}.

Because after transformation by A_{bw}, the within class distribution of data is whitened, χ^2 distribution hypothesis can also be used to reject unknown patterns. As A_{bw} is a 22×2 matrix, the criterion of rejection is also 13.815 for a 0.999 confidence.

However, this layer must also recognize and inference the class number if an unknown pattern is recieved, so inference rule must also be build up. Here we use the method of calculating the minimum distance to a class center. if the minimum distance(square) to a center of some calss, is below 13.815, then the pattern accepted by the previous two rejection layers will be infered as belonging to that class. otherwise, it will be at last rejected as unlearned. It is clear that this is a mix layer of both rejection and class inference.

5. EXPERIMENTS

In order to confirm the performances of the isomorphic transformation pattern recognition system mentioned above, we used 3 sets of patterns in doing experiments as shown in Fig.8. The first set contains three classes of patterns, where each class contains 6 kinds of patterns. This set is used for teaching the machine. Here, chinese charactors are used for that it is easy to get. The second set consists of the first set's similar patterns used for checking the recognizability of the system. And the third set is composed of abnormal patterns used for testing the rejection mechanism of each layer in the system.

Fig.9 shows the learned space of the first and the second rejection layer. And Fig.10 illustrates the third layer's learned space. In the former, the center of each circle is the CE_i and the inside circle shows the criterion of 13.815. In the latter, the cluster of each class is shown.

The experiment results show that for the learned patterns in

1st class 2nd class 3rd class

(a) 1st set used for teaching

(b) 2nd set used for checking recognizability

(c) 3rd set used for checking rejection.

Fig.8 Patterns used in experiments

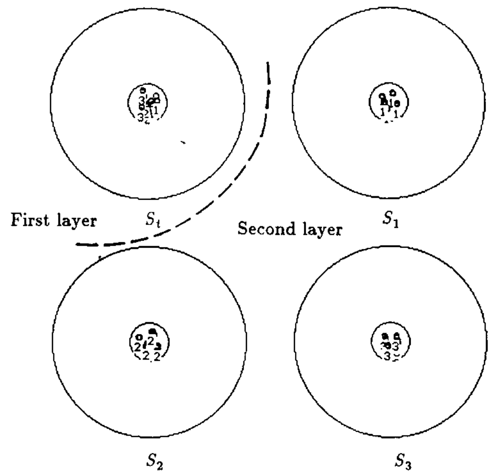

Fig.9 The learned spaces of 1st & 2nd rejection layers

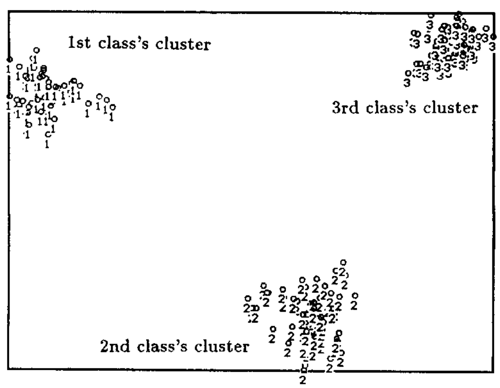

Fig.10 The 3rd layer for rejection and inference

set 1 all can be recognized perfectly by the system. To set 2, the system have a correct recognition rate of 0.67, where 33 percent of the patterns was rejected rather than wrongly infered. And the third set is all rejected, some by the first layer, some by the second layer and others by the third layer. What is interesting is that some patterns which are rejected by the previous two layers may be accepted by the third layer if there were no previous rejections. This shows that the previous two layers of rejection are meaningful.

6. CONCLUSION

In this paper, we described how to recognize isomorphic transformation patterns based on mask pattern histogram method of feature extraction. Being confined by the mask pattern realization with hardware, our system can only recognize 90°-multiple rotation isomorphic transformation and symmetry isomorphic transformation patterns. However, our approach can be considered as having some generalities. Indeed, any kind of isomorphic transformation patterns, to which we can have a measuration space to extract pattern features satisfying the condition as shown in Fig.3, can be recognized by the similar system.

Although our system now can only recognize eight isomorphic transformation patterns of a standard pattern, however it is powerful if used for identifying machine parts. This is because that, almost all the parts have the same shape for top and bottom and they often subject to turn over and rotation change, so if we can adjust their four usual directions we can recognize them by this method. And, it is obvious that to adjust a part to any one of the four directions is not a hard thing to realize. This tells that our system is applicable for real use.

In the three layers of rejection and class inference, every kind of new feature space is confined to 2 dimensions. This is only for fast caculation and visual display. In theory, the dimension of the new space must be determined by accumulation contribution rate of eigenvalues sorted from large to small. Of course, criterion of χ^2 must change with the new feature space's dimension.

In the experiments, the low recognition rate of the second set, that is, the similar patterns of the learned patterns of the first set is possibly because the teached patterns, that is , the ones in the first set are not good representatives of the three classes. So, it may be improved by teaching a great amount of samples of patterns of each class.

Lastly, because the χ^2 distribution criterion is an a priori knowledge, this stochastic pattern recognition method can be considered as having some generalities.

REFERENCES

[1]T.IIJIMA, "Theory of Pattern Recognition", Morikita Syuppan Kabushiki Kaisya, Japan, 1989.

[2]Pattern Recognition," IEEE Trans. on ASSP, VOL. 36, NO. 7, 1109-1118, JULY 1988.

[3]H.HAMA, K.YAMASHITA, "Walsh-Hadamard Power Spectra Invariant to Certain Transform Groups," IEEE Trans. on SMC, VOL. SMC-9, NO. 4, 227-237, APRIL 1979.

[4]K.S.FU, S.Y.LU, "Size Normalization and Pattern Orientation Problems in Syntactic Clustering," IEEE trans on SMC, VOL. SMC-9, NO. 1, 55-58, JANUARY 1979.

[5]M.K.HU, "Visual Pattern Recognition by Moment Invariants," IRE trans. on INFORMATION THEORY, IT-8-2, 179-187, 1962.

[6]H.Wechsler, G.L.Zimmerman, "2-D Invariant Object Recognition Using Distributed Associative Memory, IEEE trans. on PAMI, VOL. 10, NO. 6, 811/821, 1988.

[7]R.A.Messner, H.H.Szu, "An Image Processing Architecture for Real Time Generation of Scale and Rotation", Computer Vision, Graphics, and Image Processing, VOL.31, 50/66, 1985.

[8]S.Amari, "Theory of Normalization of Pattern Signals in Feature Spaces", Magazine of Electricity and Communication Association, VOL. 49, NO. 7, 1342-1350, in Japanese, 1966.

[9]K.UDAKAWA, J.TORIWAKI, K.SUGINO, "Normalization and Recognition of Two-Dimensional Patterns with Linear Distortion by Moments", Magazine of Electricity and Communication Association, VOL. 47, NO. 6, 892-901, in Japanese, 1964.

[10]T.W.ANDERSON,"An Introduction to Multivariate Statistical Analysis", John Wiley & Sons, New York, 1958.

[11]K.FUKUNAGA,"Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.

[12]L.P.HORWITZ, G.L.SHELTON, JR.,"Pattern Recognition Using Autocorrelation", Proceeding of the IRE, VOL. 49, 175-185, 1961.

[13]J.A.McLAUGHLIN, J.RAVIV, "Nth-Order Autocorrelations in Pattern Recognition", Infomation and Control, VOL. 12, 121-142, 1968.

[14]N.OTSU, "Mathematical Studies on Feature Extraction in Pattern Recognition", Researches of the Electrotechnical Laboratory, NO.818, July, 1981.

[15]H.SPATH, "Cluster Analysis Algorithms", ELLIS HORWOOD LIMITED, 1980.