Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Robust Feature Extractions from Geometric Data using Geometric Algebra

Minh Tuan Pham*, Tomohiro Yoshikawa*, Takeshi Furuhashi* and Kanta Tachibana*
*School of Engineering, Nagoya University
Furou-cho Chikusa Nagoya Aichi, 464-8603 JAPAN
Email: minhtuan@cmplx.cse.nagoya-u.ac.jp

***Abstract*—Most conventional methods of feature extraction for pattern recognition do not pay sufficient attention to inherent geometric properties of data, even in the case where the data have spatial features. This paper introduces geometric algebra to extract invariant geometric features from spatial data given in a vector space. Geometric algebra is a multidimensional generalization of complex numbers and of quaternions, and it ables to accurately describe oriented spatial objects and relations between them. This paper proposes to combine several geometric features using Gaussian mixture models. It applies the proposed method to the classification of hand-written digits.**

***Index Terms*—Geometric Algebra, Feature extraction, Pattern recognition, Gaussian mixture model.**

I. Introduction

Because nowadays, enormous amounts of data are available in various fields, data mining which analyzes data to reveal interesting information becomes more and more necessary. However most conventional methods of feature extraction have not focused on the geometric properties of data so far even in the case where the data have spatial features. For example, when m vectors are measured from an object in three-dimensional space, conventional methods represent the object by $x \in \mathbb{R}^{3m}$ which is the vector made by arranging m groups of 3 coordinates of each vector in a row. However, because these coordinate values depend on the definition of the coordinate system, inference or classification becomes remarkably poor, when objects are measured in a coordinate system different from the one used for learning. Furthermore, spatial information of the object in three-dimensional space is lost when each element $x \in \mathbb{R}^{3m}$ is, for example, normalized. Some conventional methods may extract coordinate-free features, but it depends on the experience of the model builder whether such features arise and are adopted.

This study employs geometric algebra (GA) [1],[2],[3] which can describe spatial vectors and higher order subspace relations between them, to systematically undertake various kinds of feature extractions, and to improve precision and robustness in classification problems. There are already many successful examples of its application such as image processing, signal processing, colored image processing, or multi-dimensional time-series signal processing with complex numbers or quaternions, which are low dimensional GAs [4],[5],[6],[7],[8],[9],[10]. In addition, GA-valued neural network learning methods for learning input-output relationships [11] are well studied.

In the proposed method, geometric features extracted with GA can also be used for learning a data distribution. This paper employs geometric features to learn a Gaussian mixture model (GMM) with the expectation maximization (EM) algorithm [12], and applies them to a classification problem. Because each feature extraction derived by the proposed method has its own advantages and disadvantages, this paper applies a plural mixture of GMMs for the classification problem. As an example of multi-class classification of geometric data, it applies the proposed method to a hand-written digit dataset of the UCI Machine Learning Repository [13]. When classifying new hand-written digits in practice with the learning model, it is expected that the coordinate system in a new environment differs from the one used for obtaining the learning dataset. Therefore, this paper evaluates the classification performance for randomly rotated test data.

II. Proposed Method

A. Feature extraction with GA

GA is also called Clifford algebra. An orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ can be chosen for a real vector space $\mathbb{R}^n$. The GA of $\mathbb{R}^n$, denoted by $\mathcal{G}_n$, is constructed by an associative and bilinear product of vectors, the geometric product, which is defined by

$$\mathbf{e}_i\mathbf{e}_j = \begin{cases} 1 & (i = j), \\ -\mathbf{e}_j\mathbf{e}_i & (i \neq j). \end{cases} \tag{1}$$

GAs are also defined for negative squares $\mathbf{e}_i^2 = -1$ of some or all basis vectors. Such GAs have many applications in computer graphics, robotics, virtual reality, etc [3]. But for the purposes of this study, definition (1) will be sufficient.

Now it considers two vectors $\{\mathbf{a}_l = \sum_i a_{li}\mathbf{e}_i, l = 1, 2\}$ in $\mathbb{R}^n$. Their geometric product is

$$\mathbf{a}_1\mathbf{a}_2 = \sum_{i=1}^{n} a_{1i}a_{2i} + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(a_{1i}a_{2j} - a_{1j}a_{2i}\right)\mathbf{e}_{ij}, \tag{2}$$

where the abbreviation $\mathbf{e}_{ij} = \mathbf{e}_i\mathbf{e}_j$ is adopted in the following. The first term is the commutative inner product $\mathbf{a}_1 \cdot \mathbf{a}_2 = \mathbf{a}_2 \cdot \mathbf{a}_1$. The second term is the anti-commutative outer product $\mathbf{a}_1 \wedge \mathbf{a}_2 = -\mathbf{a}_2 \wedge \mathbf{a}_1$. The second term is called a 2-blade. Linear combinations of 2-blades are called bivectors, expressed by $\sum_{I \in \mathcal{I}_2} w_I \mathbf{e}_I, w_I \in \mathbb{R}$, where $\mathcal{I}_2 = \{i_1 i_2 \mid 1 \leq i_1 < i_2 \leq n\}$ is the ordered combination set of two different elements from

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

$\{1, \ldots, n\}$. For parallel vectors $\mathbf{a}_1 = \kappa \mathbf{a}_2, \kappa \in \mathbb{R}$, and the second term becomes 0.

Next, it considers the geometric product of $\mathbf{a}_1 \mathbf{a}_2 = \mathbf{a}_2 \cdot \mathbf{a}_1 + \mathbf{a}_2 \wedge \mathbf{a}_1$ with a third vector $\mathbf{a}_3$. First, because $\mathbf{a}_1 \cdot \mathbf{a}_2 \in \mathbb{R}$, $(\mathbf{a}_1 \cdot \mathbf{a}_2) \mathbf{a}_3$ is a vector. Next, $(\mathbf{a}_1 \wedge \mathbf{a}_2) \mathbf{a}_3 = \left(\sum_{I \in \mathcal{I}_2} w_I \mathbf{e}_I\right) \sum_i a_{3i} \mathbf{e}_i$. For a certain $I = i_1 i_2$,

$$\mathbf{e}_I \mathbf{e}_i = \mathbf{e}_{i_1} \mathbf{e}_{i_2} \mathbf{e}_i = \begin{cases} \mathbf{e}_{i_1} & (i = i_2), \\ -\mathbf{e}_{i_2} & (i = i_1), \\ \mathbf{e}_{i_1 i_2 i} & (i_1 \neq i \neq i_2). \end{cases} \tag{3}$$

Therefore $(\mathbf{a}_1 \wedge \mathbf{a}_2) \mathbf{a}_3$ can be separated into a vector, i.e. the sum of terms corresponding to the first two lines of eq. (3), and a 3-blade $\mathbf{a}_1 \wedge \mathbf{a}_2 \wedge \mathbf{a}_3$, i.e. the sum of terms corresponding to the bottom line of (3). A linear combination of 3-blades is called a trivector which can be represented by $\sum_{I \in \mathcal{I}_3} w_I \mathbf{e}_I$, where $\mathcal{I}_3 = \{i_1 i_2 i_3 \mid 1 \leq i_1 < i_2 < i_3 \leq n\}$ is the combinatorial set of three different elements from $\{1, \ldots, n\}$.

In the same way the geometric product $\mathbf{a}_1 \ldots \mathbf{a}_k, (k \leq n)$ of linear independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_k$ has as its maximum grade term the k-blade $\mathbf{a}_1 \wedge \ldots \wedge \mathbf{a}_k$. Linear combinations of k-blades are called k-vectors, represented by $\sum_{I \in \mathcal{I}_k} w_I \mathbf{e}_I$, where $\mathcal{I}_k = \{i_1 \ldots i_k \mid 1 \leq i_1 < \ldots < i_k \leq n\}$ is the combination set of k elements from $\{1, \ldots, n\}$. 1-blades are vectors with $\mathcal{I}_1 = \{i_1 \mid 1 \leq i_1 \leq n\}$. 0-blades are scalars with $\mathcal{I}_0 = \{\emptyset\}$. For the unit real scalar, many authors simply write $\mathbf{e}_\emptyset = 1$. For $\mathcal{G}_n$, $\bigwedge^k \mathbb{R}^n$ denotes the set of all k-blades and $\mathcal{G}_n^k$ denotes set of k-vectors. The relationship between k-blades and k-vectors is $\bigwedge^k \mathbb{R}^n = \{A \in \mathcal{G}_n^k \mid \exists \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}, A = \mathbf{b}_1 \wedge \ldots \wedge \mathbf{b}_k\}$. The most general element of $\mathcal{G}_n$ is $A = \sum_{I \in \mathcal{I}} w_I \mathbf{e}_I$, where $\mathcal{I} = \bigcup_{k=0}^n \mathcal{I}_k = \mathcal{P}(\{1, \ldots, n\})$, and $\mathcal{P}(\cdot)$ denotes the power set. Concrete examples are the general elements of $\mathcal{G}_2$ which are linear combinations of $\{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_{12}\}$; and general elements of $\mathcal{G}_3$ which are linear combinations of $\{1, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_{12}, \mathbf{e}_{13}, \mathbf{e}_{23}, \mathbf{e}_{123}\}$. A general element of $\mathcal{G}_n$ can always be represented by $A = \sum_{k=0}^n \langle A \rangle_k$ with $\langle A \rangle_k = \sum_{I \in \mathcal{I}_k} w_I \mathbf{e}_I$, where $\langle \cdot \rangle_k$ inclicates an operator which extracts the k-vector part. The operator that selects the scalar part is abbreviated as $\langle \cdot \rangle = \langle \cdot \rangle_0$.

The geometric product of $k \, (1 \leq k \leq n)$ vectors yields

$$\begin{aligned} \mathbf{a}_1 &\in \mathcal{G}_n^1 = \mathbb{R}^n = \bigwedge^1 \mathbb{R}^n, \\ \mathbf{a}_1 \mathbf{a}_2 &\in \mathcal{G}_n^0 \oplus \bigwedge^2 \mathbb{R}^n, \\ \mathbf{a}_1 \mathbf{a}_2 \mathbf{a}_3 &\in \mathcal{G}_n^1 \oplus \bigwedge^3 \mathbb{R}^n, \\ &\vdots \\ \mathbf{a}_1 \ldots \mathbf{a}_k &\in \begin{cases} \mathcal{G}_n^1 \oplus \ldots \oplus \mathcal{G}_n^{k-2} \oplus \bigwedge^k \mathbb{R}^n & (\text{odd } k), \\ \mathcal{G}_n^0 \oplus \ldots \oplus \mathcal{G}_n^{k-2} \oplus \bigwedge^k \mathbb{R}^n & (\text{even } k). \end{cases} \end{aligned} \tag{4}$$

Now this paper proposes a systematic derivation of feature extractions from a set or a series of spatial vectors $\xi = \{\mathbf{p}_l \in \mathbb{R}^n, l = 1, \ldots, m\}$. This method is to extract k-vectors of different grades k; which encode the variations of the features. Scalars appeared in eq. (4) in the case of $k = 2$ can also be extracted.

First, assuming ξ is a series of n-dimensional vectors, $n'+1$ feature extractions are derived where $n' = \min\{n, m\}$. For $k = 1, \ldots, n'$, it writes $k' = k - 1$,

$$\begin{aligned} f_k(\xi) = \{\langle \mathbf{p}_l \ldots \mathbf{p}_{l+k'} \mathbf{e}_I^{-1} \rangle, I \in \mathcal{I}_k, l = 1, \ldots, n' - k'\} \\ \in \mathbb{R}^{(m-k')|\mathcal{I}_k|}, \end{aligned} \tag{5}$$

where, $|\mathcal{I}_k|$ is the number of combinations of k elements from n elements, and $\mathbf{e}_I^{-1}$ is the inverse of $\mathbf{e}_I$. For $I = i_1 \ldots i_k$, $\mathbf{e}_I^{-1} = \mathbf{e}_{i_k} \ldots \mathbf{e}_{i_2} \mathbf{e}_{i_1}$. It further defines

$$f_0(\xi) = \{\langle \mathbf{p}_l \mathbf{p}_{l+1} \rangle, l = 1, \ldots, n' - 1\} \in \mathbb{R}^{m-1}. \tag{6}$$

Next, assuming that ξ is a set of vectors, $n' + 1$ feature extractions can also be derived in the same way

$$f_k(\xi) = \{\langle \mathbf{p}_{l_1} \cdots \mathbf{p}_{l_k} \mathbf{e}_I^{-1} \rangle, I \in \mathcal{I}_k\} \in \mathbb{R}^{({}_m C_k)|\mathcal{I}_k|}, \tag{7}$$

$$f_0(\xi) = \{\langle \mathbf{p}_{l_1} \mathbf{p}_{l_2} \rangle\} \in \mathbb{R}^{{}_m C_2}, \tag{8}$$

where ${}_m C_k$ is the number of combinations when we choose k elements are chosen from m elements. The dimension of the feature space becomes different from the case where ξ is a series.

It denotes by f_k a feature vector extracted by a feature extraction f_k. f_0 is the scalar part in the geometric product of 2 vectors which is chosen from n vectors. f_k consists of coefficient of k-blade in the geometric product of k vectors which is chosen from n vectors. f_0 and f_n do not depend on the measurement coordinate system.

Below it shows several feature extractions for the case of hand-written digit data of the UCI Repository [13]. Each of the digit data is given by 8 points $\xi = \{\mathbf{p}_1, \ldots, \mathbf{p}_8\}$, measured by dividing the hand-written curves in equally long curve segments. A 2-dimensional point is given by $\mathbf{p}_l = x_l \mathbf{e}_1 + y_l \mathbf{e}_2$ with $\sum_{l=1}^8 x_l = \sum_{l=1}^8 y_l = 0$. Using GA, various kinds of feature extraction can be undertaken systematically. Fig. 1 shows some examples of the handwritten digit '7'. They are shown with straight line segments, different from the real curved trajectories of a pen.

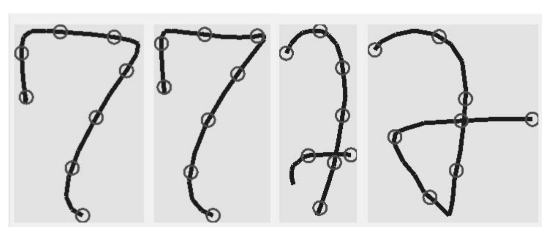

Fig. 1. Examples of hand-written digit '7'.

The simplest feature extraction f_1, which is also used in conventional methods [14],[15],[16], is

$$\begin{aligned} f_1(\xi) &= \left[\langle \mathbf{p}_1 \mathbf{e}_1^{-1} \rangle, \langle \mathbf{p}_1 \mathbf{e}_2^{-1} \rangle, \ldots, \langle \mathbf{p}_8 \mathbf{e}_1^{-1} \rangle, \langle \mathbf{p}_8 \mathbf{e}_2^{-1} \rangle\right] \\ &= [\mathbf{p}_1 \cdot \mathbf{e}_1, \mathbf{p}_1 \cdot \mathbf{e}_2, \ldots, \mathbf{p}_8 \cdot \mathbf{e}_1, \mathbf{p}_8 \cdot \mathbf{e}_2] \\ &= [x_1, y_1, \ldots, x_8, y_8] \in \mathbb{R}^{16}. \end{aligned} \tag{9}$$

Second feature extraction f_2 uses the directed magnitudes of outer products of consecutive points,

$$\begin{aligned} f_2(\xi) &= \left[\langle \mathbf{p}_1\mathbf{p}_2\mathbf{e}_{12}^{-1}\rangle, \ldots, \langle \mathbf{p}_7\mathbf{p}_8\mathbf{e}_{12}^{-1}\rangle\right] \\ &= [x_1y_2 - x_2y_1, \ldots, x_7y_8 - x_8y_7] \in \mathbb{R}^7. \end{aligned} \tag{10}$$

Third feature extraction f_0 uses the inner product of consecutive points

$$\begin{aligned} f_0(\xi) &= [\langle \mathbf{p}_1\mathbf{p}_2\rangle, \ldots, \langle \mathbf{p}_7\mathbf{p}_8\rangle] \\ &= [x_1x_2 + y_1y_2, \ldots, x_7x_8 + y_7y_8] \in \mathbb{R}^7. \end{aligned} \tag{11}$$

B. Data Distribution Learning for Multi-class Classification

A GMM is useful to approximate a data distribution in a data space. A GMM is charecterized by parameters $\Theta = \{\beta_j, \mu_j, \Sigma_j\}$, where β_j, μ_j and Σ_j are the mixture ratio, the mean vector, and the variance covariance matrix of the j-th Gaussian, respectively. The output is

$$p(\xi \mid \Theta) = \sum_{j=1}^{M} \beta_j \mathcal{N}_d\left(f(\xi) - \mu_j; \Sigma_j\right), \tag{12}$$

where $\mathcal{N}_d(\cdot;\cdot)$ is the d-dimensional Gaussian distribution function whose center is fixed at the origin.

To train M Gaussians with given incomplete data $X = \{x_i = f(\xi_i) \mid 1 \le i \le N\}$, the EM algorithm [12] is often utilized. The algorithm identifies both parameters Θ and latent variables $Z = \{z_{ij} \in \{0,1\} \mid 1 \le j \le M\}$. The z_{ij} are random variables, which for $z_{ij} = 1$ indicate that the individual datum x_i belongs the j-th of M Gaussian distributions. Thus $\sum_{j=1}^{M} P(z_{ij}) = 1$. The EM algorithm repeats the E-step and the M-step until $P(Z)$ and Θ converge. The E-step updates the probabilities of Z according to Bayes' theorem $P(Z \mid X, \Theta) \propto p(X \mid Z, \Theta)$. The M-step updates the parameters Θ of the Gaussians to maximize the likelyhood $l(\Theta, X, Z) = p(X \mid Z, \Theta)$.

A major drawback of the GMM is its large number of free parameters whose order is $O\left(Md^2\right)$. Moreover when the correlation between any pair of features is close to 1, the calculation of the inverse matrix is numerically unstable. So, the GMM becomes unable to compute the correct probability distribution. To remedy this, Tipping and Bishop [17] have proposed to use only the eigencomponents with the largest q eigenvalues, where q is a preset value of a Gaussian distribution. But the number of free parameters is not necessarily the same for different Gaussian distributions. With fixed q therefore the GMM produces a bad approximation of a probability distribution. To improve this, Meinicke and Ritter [18] have proposed to use only the eigencomponents of Gaussian distributions which are larger than a certain cutoff. The cutoff eigenvalue is set at $\lambda_- = \alpha\lambda_{\max}$, where $\alpha \in (0,1]$ is a hyper parameter and $\lambda_{\max}$ is the largest eigenvalue of the variance covariance matrix of incomplete data X. It means that

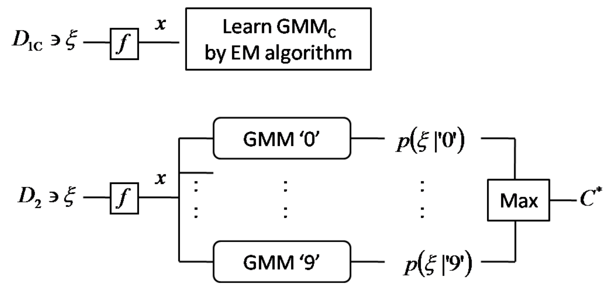

Fig. 2. Flow of multi-class classification. The top diagram shows the training of the GMM for class $C \in \{‘0’, \ldots, ‘9’\}$. The D_{1C} denotes a subset of training samples whose label is C. The $f : \xi \mapsto x$ shows feature extraction. Either of $\{f_1, f_2, f_0\}$ is chosen as f. The bottom diagram shows estimation by the learned GMMs. The same f chosen for training is used here. The GMM$_C$ outputs $p(\xi \mid C)$. The final estimation is $C^* = \arg\max_C \; p(\xi \mid C)\, P(C)$, where $P(C) = 0.1$ is the prior distribution. The set D_2 consists of independent test data.

we develop eq. (12)

$$\begin{aligned} p(x \mid \Theta) &= \sum_{j=1}^{M} \beta_j \prod_{k=1}^{d} \mathcal{N}_1\left((x - \mu_j) \cdot v_k; \lambda_k\right) \\ &\approx \sum_{j=1}^{M} \beta_j \left\{ \prod_{k=1}^{q_j} \mathcal{N}_1\left((x - \mu_j) \cdot v_k; \lambda_k\right) \right\} \\ &\quad \times \left\{ \mathcal{N}_1\left((x - \mu_j)_- \,; \lambda_- \right) \right\}^{d - q_j}, \end{aligned} \tag{13}$$

where, λ_k is the k-th largest eigenvalue of the j-th Gausian distribution. The v_k is the corresponding eigenvector. Because $k \le q_j \Leftrightarrow \lambda_k > \lambda_-$, the bracket $(x - \mu_j)_-$ is the length of the component, which is perpendicular to the q_j eigenvectors with the largest eigenvalues.

The flow of training and estimation of hand-written digit classification, as an example of multi-class classsification, is shown in Fig. 2.

C. Mixture of Experts

Each feature extraction derived with GA has advantages and disadvantages. A big merit of adopting learning of distributions rather than learning of input-output relations is that the learned distributions allow us to obtain reliable inference by mixing plural weak learners. This study therefore uses a mixture of GMMs. Inferences are mixed as

$$p(\xi \mid C) = \prod_{k=0}^{2} p\left(f_k(\xi) \mid C\right). \tag{14}$$

Fig. 3 shows the mixture of different GMMs. Inferences via different feature extractions are mixed to produce the output $p(\xi \mid C)$.

III. Experimental Results and Discussion

A. Classification of Hand-written Digits

This paper employs the Pen-Based Recognition of Hand-written Digits dataset (Pendigits dataset in the following) of

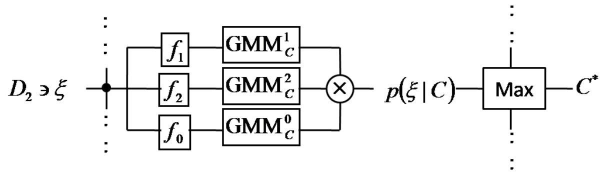

Fig. 3. Mixture of GMMs. Three GMMs via different feature extractions are mixed to yield output $p(\xi \mid C)$.

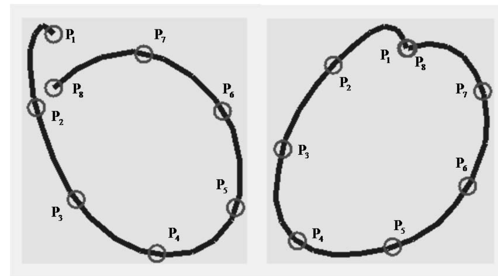

Fig. 4. Misclassified example by f_1.

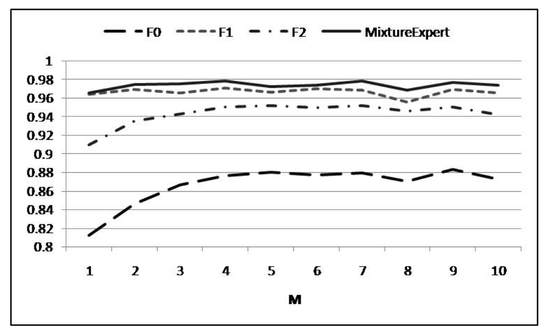

Fig. 5. Correct classification rate with f_0, f_1, f_2 and mixture expert.

the UCI Repository [13] as an example application for multi-class classification because digits have two-dimensional spatial features. The Pendigits dataset consists of 10992 samples written by 44 people. Among these samples, 7494 samples written by 30 people were used as learning data D_1. The 3498 remaining samples written by 14 other people were used as test data D_2. In the sample collection, the pen point coordinates at 100 msec intervals were measured on a tablet with a resolution of 500×500 pixels. Eight points $\{\mathbf{r}_l\}$ dividing the orbit of the pen point into 7 equally long segments were chosen, and scaled vertical ranges became $[0, 100]$. The aspect ratio changed when scaling, and thus some original geometric information was lost. But the influence of changing these aspect ratios is not investigated in this paper. It carried out the feature extraction with GA, after computing $\mathbf{p}_l = \mathbf{r}_l - \bar{\mathbf{r}}$, i.e. setting the origin $\bar{\mathbf{r}}$ at the center of the digit.

B. Classification Result of Each Feature Extraction and Mixture of Experts

For each feature extraction $f \in \{f_1, f_2, f_0\}$, the GMM learned from D_1 with 10 values of $M = \{1, \ldots, 10\}$ and then it evaluated the correct classification rate for D_2. Fig. 5 shows the results of correct classification rate based on 3 kinds of feature extractions and a mixture expert.

Because the feature extraction f_1 kept the coordinate information of each point, it shows higher correct classifications rate than others. However it also has other disadvantages. For example, because the first point $\mathbf{p}_1$ of most learning data in '0' was in the top left area, some test data whose $\mathbf{p}_1$ was in the top right area were misclassified (Fig. 4). The feature extractions f_2 and f_0, on the other hand, lost the coordinate information of each point. They extracted partial shape features which were not affected by the starting position of the part of digits. Therefore f_2 and f_0 correctly classified the '0' data with curves beginning from the top right area. In Fig. 5, the mixture of different f_1, f_2, f_0 expert feature extractions worked best, i.e. the correct classification rate became highest at $M = 4$.

C. The Robustness Against Measurement Environment Fluctuation

This paper carried out the cases that different measurement environments from the one in which D_1 was measured. It generated a dataset $D_2' = \{R(\xi, \varphi) \mid \xi \in D_2, \varphi \sim \mathcal{U}(\varepsilon)\}$ that randomly rotated each digit of the test data D_2. $\mathcal{U}(\varepsilon)$ is a uniform distribution of $[-\varepsilon, \varepsilon]$ and φ is a random variable. 20 different sets D_2' were generated from the test dataset D_2 for each $\varepsilon \in \{\frac{\pi}{5}, \frac{2\pi}{5}, \ldots, \pi\}$ and classified them. $R(\xi, \theta)$ rotated all the points of ξ by θ. Fig. 6 shows the average and the standard deviation of the correct classification rate at $M = 4$ using the feature extraction f_1, f_2, f_0 and a mixture expert. The classification precision using only feature extraction f_1 decreased remarkably with increasing ε, while the mixture expert did not decrease that much. The rotations had no influence in the cases of f_2 and f_0.

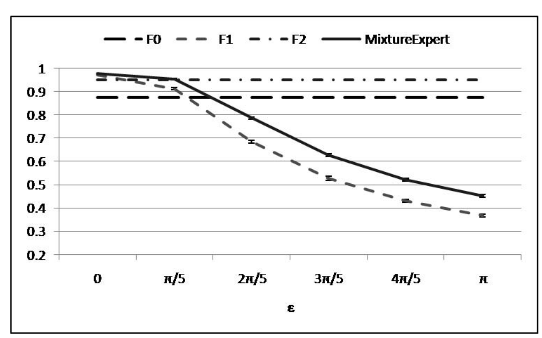

Fig. 6. Correct classification rate with $f_0, f_1, f2$ and mixture expert at $M = 4$ when D_2 was rotated.

The classification rate became remarkably poor, when objects were measured in a coordinate system different from the one used for learning. To remedy this, Simard [19], [20] has proposed to encourage invariance of a model using the training set with rotated D_1. This study also generated a dataset $D_1' = \{R(\xi, \varphi) \mid \xi \in D_1, \varphi \sim \mathcal{U}(\pi)\}$ that randomly rotated each digit of the learning data D_1. Then, it used D_1' to learn the GMM at $M = 4$ and classified D_2'. Fig. 7 shows this result. By the learning to give feature extraction f_1 the invariance for rotation, classification precision did not decrease with increasing ε. The correct classification rate using a mixture of experts was better than that using only f_1, f_2 or f_0.

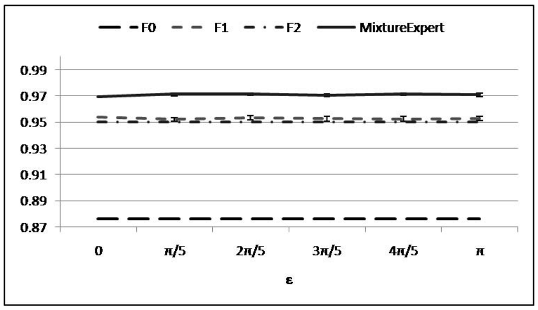

Fig. 7. Correct classification rate with $f_0, f_1, f2$ and mixture expert at $M = 4$ when both D_1 and D_2 were rotated.

IV. Conclusion

This paper proposed a systematic feature extraction method using GA. It applied the proposed method to the extraction of features in the classification of hand-written digits using GMMs. It applied to two-dimensional objects of hand-written digits deriving 3 ways of feature extraction, via coordinates, outer products, and inner products. The classification success rate using coordinate value feature extraction was higher than only using outer product feature extraction or inner product feature extraction. This paper showed that of the classification success rate by pure coordinate value feature extraction dropped substantially for rotated test data, i.e. different measurement environments from learning environments. It also showed that the strategy to mix different GA feature extractions was superior in both classification precision and robustness comparing with pure coordinate value features which were often used in conventional methods.

References

[1] C. Doran and A. Lasenby, Geometric algebra for physicists, Cambridge University Press, 2003.

[2] D. Hestenes, New foundations for classical mechanics, Dordrecht, 1986.

[3] L. Dorst, D. Fontijne, and S. Mann, Geometric Algebra for Computer Science: An Object-oriented Approach to Geometry (Morgan Kaufmann Series in Computer Graphics), 2007.

[4] I. Sekita, T. Kurita, and N. Otsu, Complex Autoregressive Model for Shape Recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 4, 1992.

[5] A. Hirose, Complex-Valued Neural Networks: Theories and Applications, Series on Innovative Intelligence, Vol. 5, 2006.

[6] N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura, Quaternion neural network with geometrical operators, Journal of Intelligent and Fuzzy Systems, Volume 15, Numbers 3–4, pp. 149–164, 2004.

[7] S. Buchholz and N. Le Bihan, Optimal separation of polarized signals by quaternionic neural networks, 14th European Signal Processing Conference, EUSIPCO 2006, September 4–8, Florence, Italy, 2006.

[8] T. Nitta, An Extension of the Back-Propagation Algorithm to Complex Numbers, Neural Networks, Volume 10, Number 8, pp. 1391–1415(25), November 1997.

[9] D. Hildenbrand and E. Hitzer, Analysis of point clouds using conformal geometric algebra, 3rd International conference on computer graphics theory and applications, Funchal, Madeira, Portugal, 2008.

[10] E. Hitzer, Quaternion Fourier Transform on Quaternion Fields and Generalizations, Advances in Applied Clifford Algebras, 17(3), pp. 497–517 (2007).

[11] G. Sommer, Geometric Computing with Clifford Algebras, Springer, 2001.

[12] A. Dempster, N. Laird, and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B, Vol. 39, No. 1, pp. 1–38, 1977.

[13] A. Asuncion, and D. J. Newman, UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science, 2007.

[14] C. Bahlmann and H. Burkhardt, The Writer Independent On-line Handwriting Recognition System frog on hand and Cluster Generative Statistical Dynamic Time Warping, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26(3), pp. 299-310, (2004).

[15] I. Guyon, P. Albrecht, Y. LeCun, J. Denker, and Hubbard, Design of a neural network character recognizer for a touch terminal, Vol. 24, pp. 105-119, 1991.

[16] J. Hu, G. Lim and , M.K. Brown, Writer independent on-line handwriting recognition using an HMM approach, Pattern Recognition, Vol. 33, pp. 133-147, 1999.

[17] M. E. Tipping and C. M. Bishop, Mixtures of probabilistic principal component analysers, Neural Computation, Vol. 11, pp. 443–482, 1999.

[18] P. Meinicke and H. Ritter, Resolution-Based Complexity Control for Gaussian Mixture Models, Neural computation Vol. 13, Issue 2, pp. 453–475, February 2001.

[19] Patrice Simard, Dave Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In Proceedings of ICDAR 2003, pages 958–962, 2003.

[20] C. M. Bishop, Pattern recognition and machine learning. New York: Springer, 2006.