



Evolutionary algorithm-based face verification

Jun-Su Jang ^{a,*}, Kuk-Hyun Han ^b, Jong-Hwan Kim ^a

^a Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), 373-1, Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

^b Digital Media R&D Center, Samsung Electronics Co., Ltd., 416, Maetan-3dong, Youngtong-gu, Suwon, Gyeonggi 442-742, Republic of Korea

Received 21 May 2004

Available online 8 October 2004

Abstract

This paper proposes a novel face verification method using principal components analysis (PCA) and evolutionary algorithm (EA). Although PCA related algorithms have shown outstanding performance, the problem lies in making decision rules or distance measures. To solve this problem, quantum-inspired evolutionary algorithm (QEA) is employed to find out the optimal weight factors in the distance measure for a predetermined threshold value which distinguishes between face images and non-face images. Experimental results show the effectiveness of the proposed method through the improved verification rate and false alarm rate.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Face verification; Principal components analysis; Evolutionary algorithm; Quantum-inspired evolutionary algorithm

1. Introduction

Face detection is one of the visual tasks which human can do easily. However, computer is not good at this task. In computer vision terms, this task can be defined as follows: given a still or video image, detect and localize an unknown number of faces. These subtasks include segmentation,

extraction and verification of faces. Face verification is the task which discriminate between face and non-face, while face detection is locating the face in some images. In general, face verification algorithm is employed to face detection task by window scanning technique. In this paper, our focus is the verification of frontal face images.

Most approaches to face verification fall into one of two categories. They are either based on local features or on holistic templates. In the former category, facial features such as eyes, mouth and some other constraints are used to verify face patterns. In the latter category, 2-D images

* Corresponding author. Tel.: +82 42 869 8048; fax: +82 42 869 8877.

E-mail address: jsjang@rit.kaist.ac.kr (J.-S. Jang).

are directly classified into face groups using pattern recognition algorithms.

We focus on the face verification under the holistic approach. The basic approach in verifying face patterns is a training procedure which classifies examples into face and non-face prototype categories. The simplest holistic approaches rely on template matching, but these approaches have poor performance.

The first neural network approach to face verification was based on multi-layer perceptrons (Propp and Samal, 1992), and advanced algorithms were studied by Rowley et al. (1998). The neural network was designed to look through a window of 20×20 pixels and was trained by face and non-face data. Based on window scanning technique, the face detection task was performed. It means that face verification network was applied to input image for possible face locations at all scales.

One of the most famous methods among holistic approaches is principal components analysis (PCA), which is well known as eigenfaces (Turk and Pentland, 1991). Given an ensemble of different face images, the technique first finds the principal components of the face training data set, expressed in terms of eigenvectors of the covariance matrix of the face vector distribution. Each individual face in the face set can then be approximated by a linear combination of the eigenvectors. Since the face reconstruction by its principal components is an approximation, a residual reconstruction error is defined in the algorithm as a measure of faceness. The residual reconstruction error which they termed as “distance-from-face space”(DFFS) gives a good indication of the existence of a face (Pentland et al., 1994). Moghaddam and Pentland (1997) further developed this technique within a probabilistic framework.

PCA is an appropriate way of constructing a subspace for representing an object class in many cases, but it is not necessarily optimal for distinguishing between the face class and the non-face class. Face space might be better represented by dividing it into subclasses. Several methods have been proposed for doing this. Sung and Poggio (1998) proposed the mixture of multidimensional Gaussian model. They used an adaptively chang-

ing normalized Mahalanobis distance metric. Afterward, many face space analysis algorithms have been investigated and some of them have outstanding performance. But the problem of the PCA-related approaches lies in making decision rules or distance measures. They may not be the optimal ones for distinguishing between face images and non-face images. Moreover, a threshold value should be selected properly considering the verification rate and false alarm rate. To solve such a complex problem, evolutionary algorithm (EA) is an efficient tool. EA has powerful search and optimization performance in a complex problem.

EA is a probabilistic algorithm which maintains a population of individuals. In any evolutionary algorithm, each individual represents a potential solution to the problem at hand and is implemented as some data structure (e.g. chromosome). Each solution is evaluated to give some measure of fitness. Then, a new population is formed by selecting the fitter individuals. Some variation operators change the population to form new solutions. After repeating these procedures the best individual converges to a sub-optimal solution.

Prakash and Murty (1995) studied optimal subset of principal component selection and applied to vowel recognition. They used genetic algorithm to select the principal components which maximize the recognition rate. Liu and Wechsler (2000) proposed a different approach to maximize the recognition rate using evolutionary algorithm. They attempted to find optimal basis for the dual purpose of data compression and pattern classification.

In this paper, we propose a novel evolutionary approach to improve the PCA-based classifier for face verification. We apply evolutionary algorithm such as quantum-inspired evolutionary algorithm (QEA) (Han and Kim, 2004) to search for the optimal weight factors in the distance measure, while previous studies focussed on the selection or the modification of the basis vectors. Eigenfaces are constructed based on PCA and a set of weight factors for a given threshold value is selected by using QEA. QEA is based on the concept and principles of quantum computing such as a quantum bit and superposition of states. Instead of binary, numeric

or symbolic representation, it uses a Q-bit as a probabilistic representation. Its performance was tested on the knapsack problem, which produced outstanding results in computation time and success rate (Han and Kim, 2002). The proposed verification system is compared with the previous works to show the improvement in performance.

This paper is organized as follows. Section 2 presents PCA and density estimation. Section 3 describes QEA briefly. Section 4 presents how the QEA is applied to optimize the distance measure between face images and non-face images. Section 5 presents the experimental results and discussions. Finally, conclusion follows in Section 6.

2. PCA and density estimation

In this section, we present PCA concept and density estimation using Gaussian densities as a basic technique of this study.

2.1. PCA concept

A technique commonly used for dimensionality reduction is PCA. Sirovich and Kirby (1987) efficiently represented human faces using PCA which is currently a popular technique.

Given a set of $m \times n$ pixels images $\{I_1, I_2, \dots, I_K\}$, we can form a set of 1-D vectors $X = \{x_1, x_2, \dots, x_K\}$, where $x_i \in \mathfrak{R}^{N=mn}$, $i = 1, 2, \dots, K$. The basis functions for the Karhunen–Loeve transform are obtained by solving the following eigenvalue problem:

$$A = \Phi^T \Sigma \Phi \quad (1)$$

where Σ is the covariance matrix of X , Φ is the eigenvector matrix of Σ , and A is the corresponding diagonal matrix of eigenvalues. We can obtain M largest eigenvalues of the covariance matrix and their corresponding eigenvectors. Then feature vector is given as follows:

$$y = \Phi_M^T \tilde{x} \quad (2)$$

where $\tilde{x} = x - \bar{x}$ is the difference between the image vector and the mean image vector, and Φ_M is a submatrix of Φ containing the M largest eigenvectors. These principal components preserve the

major linear correlations in the given set of image vectors. By projecting to Φ_M^T , original image vector x is transformed to feature vector y . It is a linear transformation which reduces N dimensions to M dimensions as follows:

$$y = T(x) : \mathfrak{R}^N \rightarrow \mathfrak{R}^M. \quad (3)$$

By selecting M largest eigenvectors, we can obtain two subspaces. One is the principal subspace (or feature space) F containing the principal components, and the other is the orthogonal space \bar{F} . These two spaces are described in Fig. 1, where DFFS stands for “distance-from-feature-space” and DIFS “distance-in-feature-space”.

In a partial Karhunen–Loeve expansion, the residual reconstruction error is defined as

$$\epsilon^2(x) = \sum_{i=M+1}^N y_i^2 = \|\tilde{x}\|^2 - \sum_{i=1}^M y_i^2 \quad (4)$$

and this is the DFFS as stated before, which is basically the Euclidean distance. The component of x which lies in the feature space F is referred to as the DIFS.

2.2. Density estimation

In the previous subsection, we obtained DFFS and DIFS. DFFS is an Euclidean distance, but DIFS is generally not a distance norm. However, it can be interpreted in terms of the probability distribution of y in F . Moghaddam and Pentland (1997) estimated DIFS as the high-dimensional

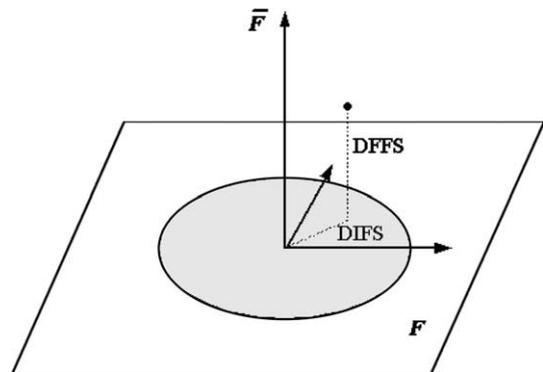


Fig. 1. DFFS and DIFS.

Gaussian densities. This is the likelihood of an input image vector \mathbf{x} formulated as follows:

$$P(\mathbf{x}|\Omega) = \frac{\exp[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \Sigma^{-1}(\mathbf{x} - \bar{\mathbf{x}})]}{(2\pi)^{N/2} |\Sigma|^{1/2}} \quad (5)$$

where Ω is a class of the image vector \mathbf{x} . This likelihood is characterized by the following Mahalanobis distance

$$d(\mathbf{x}) = (\mathbf{x} - \bar{\mathbf{x}})^T \Sigma^{-1}(\mathbf{x} - \bar{\mathbf{x}}) \quad (6)$$

and it can be also calculated efficiently as follows:

$$\begin{aligned} d(\mathbf{x}) &= \tilde{\mathbf{x}}^T \Sigma^{-1} \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^T [\Phi \Lambda^{-1} \Phi^T] \tilde{\mathbf{x}} = \mathbf{y}^T \Lambda^{-1} \mathbf{y} \\ &= \sum_{i=1}^N \frac{y_i^2}{\lambda_i} \end{aligned} \quad (7)$$

where λ is the eigenvalue of the covariance matrix. Now, we can divide this distance into two subspaces. It is determined as

$$d(\mathbf{x}) = \sum_{i=1}^M \frac{y_i^2}{\lambda_i} + \sum_{i=M+1}^N \frac{y_i^2}{\lambda_i}. \quad (8)$$

It should be noted that the first term can be computed by projecting \mathbf{x} onto the M -dimensional principal subspace F . However, the second term cannot be computed explicitly in practice because of the high-dimensionality. So, we use the residual reconstruction error to estimate the distance as follows:

$$\hat{d}(\mathbf{x}) = \sum_{i=1}^M \frac{y_i^2}{\lambda_i} + \frac{1}{\rho} \sum_{i=M+1}^N y_i^2 = \sum_{i=1}^M \frac{y_i^2}{\lambda_i} + \frac{\epsilon^2(\mathbf{x})}{\rho}. \quad (9)$$

The optimal value of ρ can be determined by minimizing a cost function, but $\rho = \frac{1}{2} \lambda_{M+1}$ was used as a thumb rule (Cootes et al., 1994).

Finally, we can extract the estimated probability distribution using (Eqs. (5) and (9)). The estimated form is determined by

$$\begin{aligned} \hat{P}(\mathbf{x}|\Omega) &= \frac{\exp\left(-\frac{1}{2} \sum_{i=1}^M \frac{y_i^2}{\lambda_i}\right) \exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi)^{M/2} \prod_{i=1}^M \lambda_i^{1/2} (2\pi\rho)^{(N-M)/2}} \\ &= P_F(\mathbf{x}|\Omega) \cdot \hat{P}_{\bar{F}}(\mathbf{x}|\Omega). \end{aligned} \quad (10)$$

Using Eq. (10), we can distinguish the face class from the non-face class by setting a threshold value for $\hat{P}(\mathbf{x}|\Omega)$, which is the Maximum Likelihood (ML) estimation method. In this case, the thresh-

old value becomes the decision factor between the verification rate and false alarm rate. If the threshold value is too low, the verification rate would be quite good but the false alarm rate would also increase. For this reason, the threshold value has to be carefully selected. In the next section, we introduce QEA as one of the evolutionary algorithms to optimize the distance measure in Eq. (10) for a given threshold value.

3. Quantum-inspired evolutionary algorithm (QEA)

QEA is designed with a novel Q-bit representation, a Q-gate as a variation operator, an observation process, a global migration process, and a local migration process. It uses a new representation, called Q-bit, for the probabilistic representation that is based on the concept of qubits, and a Q-bit individual as a string of Q-bits. A Q-bit is defined as the smallest unit of information, which is defined with a pair of numbers, (α, β) , where $|\alpha|^2 + |\beta|^2 = 1$. $|\alpha|^2$ gives the probability that the Q-bit will be found in the '0' state and $|\beta|^2$ gives the probability that the Q-bit will be found in the '1' state. A Q-bit may be in the '1' state, in the '0' state, or in a linear superposition of the two. A Q-bit individual is defined as a string of m Q-bits. QEA maintains a population of Q-bit individuals, $Q(t) = \{\mathbf{q}_1^t, \mathbf{q}_2^t, \dots, \mathbf{q}_n^t\}$ at generation t , where n is the size of population, and $\mathbf{q}_j^t, j = 1, 2, \dots, n$, is a Q-bit individual.

Fig. 2 shows the standard procedure of QEA. The procedure of QEA is explained as follows:

(i) In the step of 'initialize $Q(t)$ ', α_i^0 and β_i^0 , $i = 1, 2, \dots, m$, of all \mathbf{q}_j^0 , are initialized to $\frac{1}{\sqrt{2}}$. It means that one Q-bit individual, \mathbf{q}_j^0 represents the linear superposition of all possible states with the same probability.

(ii) This step generates binary solutions in $P(0)$ by observing the states of $Q(0)$, where $P(0) = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_n^0\}$ at generation $t = 0$. One binary solution, \mathbf{x}_j^0 , is a binary string of length m , which is formed by selecting either 0 or 1 for each bit by using the probability, either $|\alpha_i^0|^2$ or $|\beta_i^0|^2$ of \mathbf{q}_j^0 , respectively.

(iii) Each binary solution \mathbf{x}_j^0 is evaluated to give a level of its fitness.

```

Procedure QEA
begin
     $t \leftarrow 0$ 
    i) initialize  $Q(t)$ 
    ii) make  $P(t)$  by observing the states of  $Q(t)$ 
    iii) evaluate  $P(t)$ 
    iv) store the best solutions among  $P(t)$  into  $B(t)$ 
        while (not termination condition) do
            begin
                 $t \leftarrow t + 1$ 
                v) make  $P(t)$  by observing the states of  $Q(t - 1)$ 
                vi) evaluate  $P(t)$ 
                vii) update  $Q(t)$  using Q-gates
                viii) store the best solutions among  $B(t - 1)$  and  $P(t)$  into  $B(t)$ 
                ix) store the best solution  $\mathbf{b}$  among  $B(t)$ 
                x) if (global migration condition)
                    then migrate  $\mathbf{b}$  to  $B(t)$  globally
                xi) else if (local migration condition)
                    then migrate  $\mathbf{b}'_j$  in  $B(t)$  to  $B(t)$  locally
            end
        end

```

Fig. 2. Procedure of QEA.

(iv) The initial best solutions are then selected among the binary solutions, $P(0)$, and stored into $B(0)$, where $B(0) = \{\mathbf{b}_1^0, \mathbf{b}_2^0, \dots, \mathbf{b}_n^0\}$, and \mathbf{b}_j^0 is the same as \mathbf{x}_j^0 at the initial generation.

(v), (vi) In the **while** loop, binary solutions in $P(t)$ are formed by observing the states of $Q(t - 1)$ as in step (ii), and each binary solution is evaluated for the fitness value. It should be noted that \mathbf{x}'_j in $P(t)$ can be formed by multiple observations of \mathbf{q}_j^{t-1} in $Q(t - 1)$.

(vii) In this step, Q-bit individuals in $Q(t)$ are updated by applying Q-gates defined as a variation operator. The following rotation gate is used as a basic Q-gate:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}, \quad (11)$$

where $\Delta\theta_i, i = 1, 2, \dots, m$, is a rotation angle of each Q-bit. $\Delta\theta_i$ should be designed in compliance with the application problem.

(viii), (ix) The best solutions among $B(t - 1)$ and $P(t)$ are selected and stored into $B(t)$, and if the best solution stored in $B(t)$ is a better solution fitting than the stored best solution \mathbf{b} , the stored solution \mathbf{b} is replaced by the new one.

(x), (xi) If a global migration condition is satisfied, the best solution \mathbf{b} is migrated to $B(t)$ globally. If a local migration condition is satisfied, the best one among some of the solutions in $B(t)$ is migrated to them. The migration condition is a design parameter, and the migration process can induce a variation of the probabilities of a Q-bit individual. A local-group is defined to be the sub-population affected mutually by a local migration, and a local-group size is the number of the individuals in a local-group. Until the termination condition is satisfied, QEA is running in the **while** loop.

4. Optimization of distance measure

To improve the performance of the verification rate and to reduce the false alarm rate, in this section we present a novel evolutionary scheme to search for the optimal distance measure for a given threshold value.

The Mahalanobis distance-based probability guarantees quite good performance, but it is not optimal for discriminating face images from the non-face images including the similar face images (position-shifted or different-scale face images).

In Eq. (10), an eigenvalue can be considered as the weight factor of the corresponding feature value. These weight factors can be optimized on a training data set. To perform the optimization, we construct the training data set. It consists of two classes: face class (positive training data) and non-face class (negative training data). Fig. 3(a) shows an example of a face training data set. It was produced from the face region of the [Martinez and Benavente \(1998\)](#). A non-face training data set consists of arbitrarily chosen images and not exact face images which are different in scale, translation and rotation. Fig. 3(b) shows an example of a non-face training data set.

To search for the optimal weight factors for a given threshold value, QEA is used. The number of weight factors to be optimized is M , which is the same as the number of principal components. Using the weight factors obtained by QEA, we can compute the probability distribution as follows:

$$P_{\text{opt}}(\mathbf{x}|\Omega) = \frac{\exp\left(-\frac{1}{2}\sum_{i=1}^M \frac{y_i^2}{\omega_i}\right)}{(2\pi)^{M/2} \prod_{i=1}^M \lambda_i^{1/2}} \cdot \frac{\exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(N-M)/2}}. \quad (12)$$

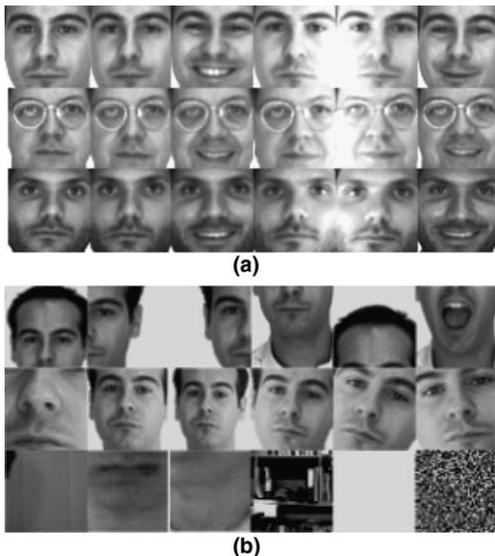


Fig. 3. Examples of training data set: (a) face training data, (b) non-face training data.

It is the same as Eq. (10) except for the weight factors ω_i , $i = 1, 2, \dots, M$. If we apply Eq. (12) to face verification with some weight factors, a threshold value should be assigned. However, the threshold value needs not to be tuned for better performance since QEA yields optimized weight factors to the predetermined threshold value.

To evaluate the fitness value, we calculate the score. The score is added by +1 for every correct verification. The score is used as a fitness measure considering both the verification rate, P score for the face class (positive training data) and the false alarm rate, N score for the non-face class (negative training data). Then the fitness is evaluated as

$$\text{Fitness} = P \text{ score} + N \text{ score}. \quad (13)$$

Using this fitness function, we can find the optimal weight factors for training data set for the predetermined threshold value.

Our methodology for finding the weight factors is shown in Fig. 4. It shows each generation of evolution which maximizes the fitness value. After reaching termination condition, we can obtain the final weight factors ω_i , which maximize the fitness value on the training data. By using the obtained weight factors, we calculate Eq. (12) for classification.

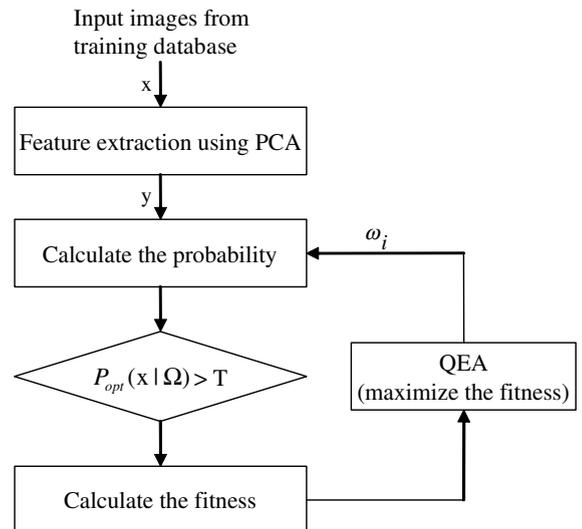


Fig. 4. Flowchart of the proposed method (T : Threshold value).

5. Experimental results and discussions

We constructed three types of database for the experiment. First, 70 face images were used for extracting principal components. Second, 1000 images (300 images for face and 700 images for non-face) were used for training weight factors. Third, 2168 images (1084 images for face and 1084 images for non-face) were used for the generalization test. For this test, two kinds of database were used. One is AR face database and the other is RIT face database. RIT face database consists of the images acquired in our laboratory (RIT lab, KAIST) by capturing a video from USB camera, which have relatively poor quality than AR database. We intended to test the video camera images which differ from still camera images. An example of the RIT database is shown in Fig. 5.

All images are 40×40 pixels with 256 gray levels. We chose 40 principal components from the 70 face images. For pre-processing, histogram equalization was performed to normalize the lighting condition.

Positive training data were produced from the face region of the AR face database except occlusion such as sunglasses. These are normalized by specific position, scale and rotation. An example of a face training data set is shown in Fig. 3(a). Variations of the facial expression and illumination changes were allowed. Negative training data consisted of both randomly generated images and natural images excluding the face images. Position-shifted face images, different-scale or different-rotation face images were also included as negative training data. An example of the negative training data set is shown in Fig. 3(b).



Fig. 5. RIT face database.

As shown in Fig. 4, QEA was employed to train the weight factors ω_i . The following boundary of each weight factor was considered as a domain constraint:

$$0.1\lambda_i < \omega_i < 10\lambda_i, \quad (1 \leq i \leq 40). \tag{14}$$

If $\omega_i = \lambda_i$ for all i , our method is equivalent to ML classifier. A Q-bit individual is illustrated in Fig. 6. Each weight factor was represented by 10 Q-bits.

We performed QEA to 1000 training images using the parameters in Table 1. In Eq. (11), rotation angles should be selected properly. For each Q-bit, $\theta_1 = 0, \theta_2 = 0, \theta_3 = 0.01\pi, \theta_4 = 0, \theta_5 = -0.01\pi, \theta_6 = 0, \theta_7 = 0, \theta_8 = 0$ were used.

The termination condition was given by the maximum generation. The perfect score was 1000 points, which is equivalent to the number of training images. If the score did not reach 1000 points before the maximum generation, the evolution process stopped at the maximum generation. After the searching procedure, we obtained a set of weight factors which maximized the fitness function. By using it, generalization test was performed. We compared our algorithm with DFPS classifier and ML classifier. All three classifiers applied PCA process for obtaining the feature vector y . At next step, DFPS classifier was tested using Eq. (4), and ML classifier was tested using Eq. (10). Our algorithm was tested using Eq. (12) with

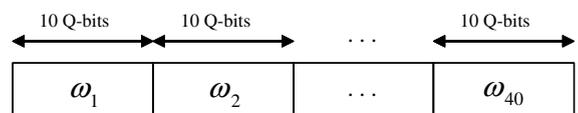


Fig. 6. Structure of a Q-bit individual.

Table 1
Parameters for QEA

Parameters	No.
Population size	15
No. of weight factors	40
No. of Q-bits per weight factor	10
No. of observations	2
Global migration period	100
Local migration period	1
No. of individuals per group	3
Max. generation	2000

the weight factors obtained from QEA. For the DFSS and the ML classifier, we selected the threshold value that provoked the best fitness score. For QEA-based classifier, we used the same threshold value set for the ML classifier. It should be noted that there was no need to choose a threshold value for better performance in our classifier because the weight factors had been already optimized for the predetermined threshold value.

Table 2 shows the results for the generalization test. It shows that the proposed method performs better than the DFSS or the ML classifier.

The results suggest that the QEA-based classifier works well not only in terms of the verification rate (P score), but also in terms of the false alarm rate (N score). The verification rate of the QEA-based classifier was higher than that of the ML classifier. The false alarm rate was lower than that of the ML classifier.

In summary, the advantage of our system can be described in three aspects. First, it provides the optimal distance measure to discriminate face images from non-face images to given training data set. It also has good generalization performance. Second, our system does not need an effort to select the exact threshold value. We only need to assign a threshold value, since QEA finds the weight factors in the distance measure based on the given threshold value. Third, our system can be adapted to various negative data like other training algorithms such as neural networks and support vector machine. However, a fixed structured classifier such as the ML classifier can not change its characteristic in frequent failure situations. Our system can be adapted by reconstruct-

ing the training data and following the optimization procedure. Furthermore, by setting the negative training data with position-shifted or different-scale face images, our system can discriminate exact frontal faces from non-exact-format faces. Therefore, our method is very helpful to detect a normalized face. It is very important to detect a normalized face which is generally used for face recognition, since recognition results depend on detection accuracy.

6. Conclusion

In this paper, we have proposed a novel evolutionary scheme for optimizing the distance measure for face verification. The approach is basically related to the eigenspace density estimation technique. To improve the previous Mahalanobis distance-based classifier, we have used a novel distance measure which consists of the weight factors optimized by the training set. The performance of the proposed face verification system has been demonstrated through the improved face verification rate and false alarm rate.

Acknowledgment

This work was supported by the ITRC-IRRC (Intelligent Robot Research Center) of the Korea Ministry of Information and Communication in 2003.

References

- Cootes, T., Hill, A., Taylor, C., Haslam, J., 1994. Use of active shape models for locating structures in medical images. *Image Vision Comput.* 12, 355–365.
- Han, K.-H., Kim, J.-H., 2002. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans. Evolut. Comput.* 6, 580–593.
- Han, K.-H., Kim, J.-H., 2004. Quantum-inspired evolutionary algorithms with a new termination criterion, He gate, and two phase scheme. *IEEE Trans. Evolut. Comput.* 8, 156–169.
- Liu, C., Wechsler, H., 2000. Evolutionary pursuit and its application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 570–582.

Table 2
Results for generalization test

Face database	Classifier	Verification rate (%)	False alarm rate (%)
AR	DFFS	94.39	7.40
	ML	95.15	6.51
	QEA-based	98.09	3.32
RIT	DFFS	91.00	9.67
	ML	93.00	8.67
	QEA-based	96.33	6.33

- Martinez, A.M., Benavente, R., 1998. The AR face database. Technical Report 24, CVC. Available at <http://rv11.ecn.purdue.edu/~aleix/aleixfaceDB.html>.
- Moghaddam, B., Pentland, A., 1997. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 696–710.
- Pentland, A., Moghaddam, B., Strarner, T., 1994. View-based and modular eigenspaces for face recognition. *IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pp. 84–91.
- Prakash, M., Murty, N., 1995. A genetic approach for selection of (near-) optimal subsets of principal components for discrimination. *Pattern Recogn. Lett.* 16, 781–787.
- Propp, M., Samal, A., 1992. Artificial neural network architecture for human face detection. *Intell. Eng. Syst. Artificial Neural Networks* 1, 535–540.
- Rowley, H.A., Baluja, S., Kanade, T., 1998. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 23–38.
- Sirovich, L., Kirby, M., 1987. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am.* 4, 519–524.
- Sung, K.-K., Poggio, T., 1998. Example-based learning for view-based human face detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 39–51.
- Turk, M., Pentland, A., 1991. Eigenfaces for recognition. *J. Cognitive Neurosci.* 3, 71–86.