

Quantum-Inspired Evolutionary Algorithm-Based Face Verification

Jun-Su Jang, Kuk-Hyun Han, and Jong-Hwan Kim

Dept. of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
{jsjang, khhan, johkim}@rit.kaist.ac.kr

Abstract. Face verification is considered to be the main part of the face detection system. To detect human faces in images, face candidates are extracted and face verification is performed. This paper proposes a new face verification algorithm using Quantum-inspired Evolutionary Algorithm (QEA). The proposed verification system is based on Principal Components Analysis (PCA). Although PCA related algorithms have shown outstanding performance, the problem lies in the selection of eigenvectors. They may not be the optimal ones for representing the face features. Moreover, a threshold value should be selected properly considering the verification rate and false alarm rate. To solve these problems, QEA is employed to find out the optimal distance measure under the predetermined threshold value which distinguishes between face images and non-face images. The proposed verification system is tested on the AR face database and the results are compared with the previous works to show the improvement in performance.

1 Introduction

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 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 [1], but these approaches have poor performance compared to more complex techniques like neural networks.

The first neural network approach to face verification was based on multi-layer perceptrons [2], and advanced algorithms were studied by Rowley [3]. 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 [4]. 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 face-ness. The residual reconstruction error which they termed as “distance-from-face space” (DFFS) gives a good indication of the existence of a face [5]. Moghaddam and Pentland have further developed this technique within a probabilistic framework [6].

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 from the non-face class. Face space might be better represented by dividing it into subclasses, and several methods have been proposed for doing this. Sung and Poggio proposed the mixture of multidimensional Gaussian model and they used an adaptively changing normalized Mahalanobis distance metric [7]. Afterward, many face space analysis algorithms have been investigated and some of them have outstanding performance. The problem of the PCA related approaches lies in the selection of eigenvectors. They may not be the optimal ones for representing the face features. Moreover, a threshold value should be selected properly considering the verification rate and false alarm rate. By employing QEA, the performance of the face verification is improved enough to distinguish between face images and non-face images.

In this paper, eigenfaces are constructed based on PCA and a set of weight factors is selected by using Quantum-inspired Evolutionary Algorithm (QEA) [8]. QEA has lately become a subject of special interest in evolutionary computation. It 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 an outstanding result [8].

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

2 Quantum-Inspired Evolutionary Algorithm (QEA)

QEA [8] can treat the balance between exploration and exploitation more easily when compared to conventional GAs (CGAs). Also, QEA can explore the search space with a small number of individuals (even with only one individual for real-time application) and exploit the global solution in the search space within a short span of time. QEA is based on the concept and principles of quantum computing, such as a quantum bit and superposition of states. However, QEA is not a quantum algorithm, but a novel evolutionary algorithm. Like other

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-gatesviii) 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 $B(t)$ globallyxi) **else if** (local migration condition)**then** migrate \mathbf{b}_j^t in $B(t)$ to $B(t)$ locally**end****end****Fig. 1.** Procedure of QEA.

evolutionary algorithms, QEA is also characterized by the representation of the individual, the evaluation function, and the population dynamics.

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. QEA 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 in QEA, 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. 1 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.

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^t 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 of QEA. The following rotation gate is used as a basic Q-gate in QEA:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}, \quad (1)$$

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 in QEA is defined to be the subpopulation 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.

3 PCA and Density Estimation

In this section, we present the PCA concept and density estimation using Gaussian densities. It should be noted that this method is a basic technique in pattern recognition and it lays the background of this study.

3.1 PCA Concept

A technique commonly used for dimensionality reduction is PCA. In the late 1980's, Sirovich and Kirby [9] efficiently represented human faces using PCA and it 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 $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$, where $\mathbf{x}_i \in \mathbb{R}^{N=mn}$, $i = 1, 2, \dots, K$. The basis functions for the Karhunen-Loeve Transform (KLT) [10] are obtained by solving the eigenvalue problem

$$\Lambda = \Phi^T \Sigma \Phi \quad (2)$$

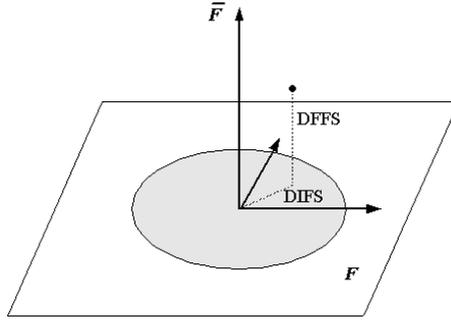


Fig. 2. Two subspaces

where Σ is the covariance matrix of X , Φ is the eigenvector matrix of Σ , and Λ 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:

$$\mathbf{y} = \Phi_M^T \tilde{\mathbf{x}} \tag{3}$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$ 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 \mathbf{x} is transformed to feature vector \mathbf{y} . It is a linear transformation which reduces N dimensions to M dimensions as follows:

$$\mathbf{y} = T(\mathbf{x}) : \mathfrak{R}^N \longrightarrow \mathfrak{R}^M. \tag{4}$$

By selecting M largest eigenvectors, we can obtain two subspaces. One is the principal subspace (or feature space) F containing the principal components, and another is the orthogonal space \bar{F} . These two spaces are described in Fig. 2, where DFFS stands for “distance-from-feature-space” and DIFS “distance-in-feature-space”.

In a partial KL expansion, the residual reconstruction error is defined as

$$\epsilon^2(\mathbf{x}) = \sum_{i=M+1}^N y_i^2 = \|\tilde{\mathbf{x}}\|^2 - \sum_{i=1}^M y_i^2 \tag{5}$$

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

3.2 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 \mathbf{y} in F . Moghaddam estimated

DIFS as the high-dimensional Gaussian densities [6]. This is the likelihood of an input image vector \mathbf{x} formulated as follows:

$$P(\mathbf{x}|\Omega) = \frac{\exp[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \Sigma^{-1}(\mathbf{x} - \bar{\mathbf{x}})]}{(2\pi)^{N/2} |\Sigma|^{1/2}} \quad (6)$$

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

$$d(\mathbf{x}) = (\mathbf{x} - \bar{\mathbf{x}})^T \Sigma^{-1}(\mathbf{x} - \bar{\mathbf{x}}) \quad (7)$$

and it can be also calculated efficiently as follows:

$$\begin{aligned} d(\mathbf{x}) &= \tilde{\mathbf{x}}^T \Sigma^{-1} \tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T [\Phi \Lambda^{-1} \Phi^T] \tilde{\mathbf{x}} \\ &= \mathbf{y}^T \Lambda^{-1} \mathbf{y} \\ &= \sum_{i=1}^N \frac{y_i^2}{\lambda_i} \end{aligned} \quad (8)$$

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

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

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:

$$\begin{aligned} \hat{d}(\mathbf{x}) &= \sum_{i=1}^M \frac{y_i^2}{\lambda_i} + \frac{1}{\rho} \sum_{i=M+1}^N y_i^2 \\ &= \sum_{i=1}^M \frac{y_i^2}{\lambda_i} + \frac{\epsilon^2(\mathbf{x})}{\rho}. \end{aligned} \quad (10)$$

The optimal value of ρ can be determined by minimizing a cost function, but $\rho = \frac{1}{2} \lambda_{M+1}$ may be used as a thumb rule [11].

Finally, we can extract the estimated probability distribution using (6) and (10). The estimated form is determined by

$$\begin{aligned} \hat{P}(\mathbf{x}|\Omega) &= \frac{\exp\left(-\frac{1}{2} \sum_{i=1}^M \frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2} \prod_{i=1}^M \lambda_i^{1/2}} \cdot \frac{\exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(N-M)/2}} \\ &= P_F(\mathbf{x}|\Omega) \cdot \hat{P}_{\bar{F}}(\mathbf{x}|\Omega). \end{aligned} \quad (11)$$

Using (11), 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 threshold value becomes the deciding 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 following section, we propose an optimization procedure for selecting a set of eigenvalues to determine the decision boundary between the face and non-face classes. By optimizing the Mahalanobis distance term in (11) under the given threshold value, we can find a better distance measure and need not refer to the threshold value.

4 Optimization of Decision Boundary

In this section, we describe a framework for decision boundary optimization using QEA. In the case of ML estimation, the decision boundary is determined by a probability equation and an appropriate threshold value. To improve the performance of the verification rate and reduce the false alarm rate, we attempt to find a better decision boundary.

The Mahalanobis distance-based probability guarantees quite good performance, but it is not optimal. In (11), an eigenvalue is used 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 data) and non-face class (negative data). Fig. 3 shows an example of a face training data set. A non-face training data set consists of arbitrarily chosen images, including randomly generated images.

To search for the weight factors, 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:



Fig. 3. Example of the face training data set

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

It is the same as (11) except for the weight factors $\omega_i, i = 1, 2, \dots, M$. To apply (12) to face verification, the threshold value should be assigned. But, since QEA yields optimized weight factors to the predetermined threshold value, we need not assign the 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) and the false alarm rate (N_score) because the training data set consists of both face and non-face data. Then the fitness is evaluated as

$$Fitness = P_score + N_score \quad (13)$$

where P_score is for the face class (positive data) and N_score is for the non-face class (negative data). Using this fitness function, we can find the optimal weight factors for training data set under the predetermined threshold value.

5 Experimental Results and Discussions

We constructed 3 types of database for the experiment. First, 70 face images were used for extracting principal components. Second, 560 images (280 images for face and 280 images for non-face) were used for training weight factors. Third, 1568 images (784 images for face and 784 images for non-face) were used for the generalization test.

All images are 50×50 pixels with 256 gray levels. We chose 50 principal components from the 70 face images. For pre-processing, histogram equalization was performed to normalize the lighting condition.

Positive data were produced from the face region of the AR face database [12]. An example of a face training data set is shown in Fig. 3. Variations of the facial expression and illumination were allowed. Negative data consisted of both randomly generated images and natural images excluding the face images. Position-shifted face images and different-scale face images were also included as negative data.

The following boundary of each weight factor was considered as a domain constraint:

$$0.1\lambda_i < \omega_i < 10\lambda_i, \quad (1 \leq i \leq 50). \quad (14)$$

By setting the constraint of the boundary using the eigenvalue, it becomes a constraint optimization problem. We performed QEA for 560 training images using the parameters in Table 1. In (1), 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 560 points. If the score does not reach 560 points before maximum

Table 1. Parameters for QEA

Parameters	No.
Population size	15
No. of variables	50
No. of Q-bits per variable	10
No. of observations	2
Global migration period	100
Local migration period	1
No. of individuals per group	3
Max. generation	2000

Table 2. Results for generalization test

	P_score (784)	Verification rate(%)	N_score (784)	False Alarm rate(%)	$Fitness$ (1568)
DFFS classifier	726	92.60	716	0.087	1442
ML classifier	728	92.86	726	0.074	1454
QEA-based classifier	741	94.52	740	0.056	1481

generation, the evolution process stops at maximum generation. After the searching procedure, we performed a generalization test to 1568 images using the weight factors obtained by QEA. We also compared the results with the DFFS and the ML classifier. For the DFFS and the ML classifier, we selected the threshold value that provoked the best score. For QEA-based classifier, we used the same threshold value set for the ML classifier. It should be noted that there is no need to choose a threshold value for better performance in our classifier because the weight factors have been already optimized under the predetermined threshold value.

Table 2 shows the results for the generalization test. The results show that the proposed method performs better than the DFFS or the ML classifier. The results described above 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 1.66% higher than that of the ML classifier. The false alarm rate was 0.018% lower than that of the ML classifier. The advantage of the proposed classifier is that more training data can improve its performance.

6 Conclusion and Further Works

In this paper, we have proposed a decision boundary optimization method for face verification using QEA. The approach is basically related to eigenspace density estimation technique. To improve the previous Mahalanobis distance-based probability, we have used a new distance which consists of the weight factors optimized at the training set. The proposed face verification system has

been tested by face and non-face images extracted from AR database, and very good results have been achieved both in terms of the face verification rate and false alarm rate.

The advantage of our system can be summarized in two aspects. First, our system does not need an exact threshold value to perform optimally. We only need to choose an appropriate threshold value and QEA will find the optimal decision boundary based on the threshold value. Second, our system can be adapted to various negative data. A fixed structured classifier such as the ML classifier can not change its character in frequently failure situation. But our system can be adapted to this case by reconstructing the training data and following the optimization procedure.

As a future research, we will construct face detection system using this verification method. In face detection, it is clear that verification performance is very important. Most of the image-based face detection approaches apply a window scanning technique. It is an exhaustive search of the input image for possible face locations at all scales. In this case, overlapping detection arises easily. A powerful verification method is therefore needed to find exact face locations. We expect that our verification method will also work well for the face detection task.

References

1. Holst, G.: Face detection by facets : Combined bottom-up and top-down search using compound templates. Proc. of the International Conference on Image Processing (2000) TA07-08
2. Propp, M., Samal, A.: Artificial neural network architecture for human face detection. *Intell. Eng. Systems Artificial Neural Networks* **1** (1992) 535-540
3. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **20** (1998) 23-38
4. Turk, M., Pentland, A.: Eigenfaces for recognition. *J. Cog. Neurosci.* **3** (1991) 71-86
5. Pentland, A., Moghaddam, B., Strarner, T.: View-based and modular eigenspaces for face recognition. *IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition* (1994) 84-91
6. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation, *IEEE Trans. Pattern Anal. Mach. Intell.* **19** (1997) 696-710
7. Sung, K.-K., Poggio, T.: Example-based learning for view-based human face detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **20** (1998) 39-51
8. Han, K.-H., Kim, J.-H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans. Evolutionary Computation* **6** (2002) 580-593
9. Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Amer.* **4** (1987) 519-524
10. Loeve, M.: *Probability Theory*. Princeton, N.J., Van Nostrand (1955)
11. Cootes, T., Hill, A. Taylor, C., Haslam, J.: Use of active shape models for locating structures in medical images. *Image and Vision Computing* **12** (1994) 355-365
12. AR face database: http://rv11.ecn.purdue.edu/~aleix/aleix_face_DB.html