

# Face Detection using Quantum-inspired Evolutionary Algorithm

Jun-Su Jang  
Dept. of EECS,  
Korea Advanced Institute of  
Science and Technology (KAIST),  
Guseong-dong, Yuseong-gu, Daejeon,  
305-701, Republic of Korea  
Email: jsjang@rit.kaist.ac.kr

Kuk-Hyun Han  
Digital Media R&D Center,  
Samsung Electronics Co., Ltd.,  
416, Maetan-3dong, Youngtong-gu,  
Suwon, Gyeonggi,  
443-742, Republic of Korea  
Email: kukhyun7.han@samsung.com

Jong-Hwan Kim  
Dept. of EECS,  
Korea Advanced Institute of  
Science and Technology (KAIST),  
Guseong-dong, Yuseong-gu, Daejeon,  
305-701, Republic of Korea  
Email: johkim@rit.kaist.ac.kr

**Abstract**—This paper proposes a new face detection system using Quantum-inspired Evolutionary Algorithm (QEA). The proposed detection system is based on elliptical blobs and Principal Components Analysis (PCA). The elliptical blobs in the directional image are used to find the face candidate regions, and then PCA and QEA are employed to verify faces. Although PCA related algorithms have shown outstanding performance, there still exist some problems such as optimal decision boundary or learning capabilities. By PCA, we can obtain the optimal basis but they may not be the optimal ones for discriminating faces from non-faces. Moreover, a threshold value should be selected properly considering the success rate and false alarm rate. To solve these problems, QEA is employed to find out the optimal decision boundary under the predetermined threshold value which distinguishes between face images and non-face images. The proposed system provides learning capability by reconstructing the training database, which means that system performance can be improved as failure trials occur.

## I. 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 involve segmentation, extraction and verification of faces.

Most approaches to face detection 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 detection 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 detection 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  $20 \times 20$  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. For efficient face location strategy, Maio and Maltoni suggested real-time face location algorithm based on the search of elliptical blobs in the directional image by means of the generalized Hough transform [4]. In this paper, we used Maio's approach to find the face candidate regions.

One of the most famous methods among holistic approaches is Principal Components Analysis (PCA), which is well known as eigenfaces [5]. 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 [6]. Moghaddam and Pentland have further developed this technique within a probabilistic framework [7].

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 [8]. 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 success 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. This concept was proved in our previous research work [9].

In this paper, eigenfaces are constructed based on PCA and a set of weight factors is selected by using Quantum-inspired Evolutionary Algorithm (QEA) [10][11]. 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 on outstanding result [10].

This paper is organized as follows. Section II describes face location based on elliptical blobs. Section III presents the PCA and density estimation. Section IV describes a summary of QEA. Section V presents how the QEA is applied to optimize the decision boundary between face images and non-face images. Section VI presents the experimental results and discussions. Finally, conclusion and further works follow in Section VII.

## II. FACE LOCATION

In this section, we present the face location based on the search of elliptical blobs in the directional image. In the face location part, face candidate regions are extracted from input image. Then face verification is performed from the face candidate region. For efficient face detection system, face location strategy 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. To reduce search time, we apply the Maio's approximate location [4].



Fig. 1. Directional image

A directional image is a matrix defined over a discrete grid, superimposed on the gray-scale image, whose elements are in correspondence with the grid nodes. Each element is a vector lying on the  $xy$  plane. The direction of the vector represents the tangent to the image edges in a neighborhood of the node. Directional image is computed by means of the method

proposed by Donahue and Roklin [12]. An example of the directional image is shown in Fig. 1.

After obtaining the directional image, generalized Hough transform is employed to find the face centers (ellipse centers). Fig. 2 shows simple geometric relationship between a directional vector and two ellipse centers. Let  $a$  and  $b$  be the

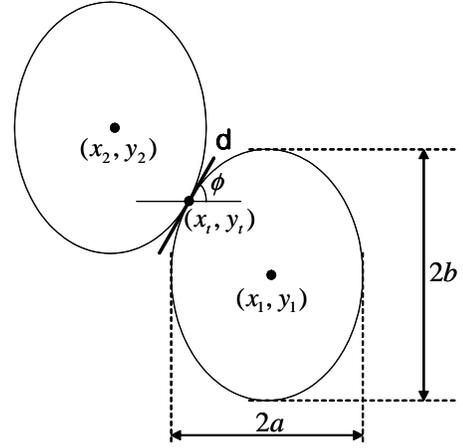


Fig. 2. Location of two ellipse centers from a directional vector

lengths of the semi-axes of an ellipse used as reference face shape,  $\phi$  be the angle of directional vector  $\mathbf{d}$ . There are two ellipses which has a vector  $\mathbf{d}$  as a tangent line at  $(x_t, y_t)$ . Two ellipse center position is calculated as follows:

$$\begin{aligned} x_1 &= x_t + \frac{a^2 m}{a^2 m^2 + b^2}, & y_1 &= y_t - \frac{b^2}{a^2 m^2 + b^2} \\ x_2 &= x_t - \frac{a^2 m}{a^2 m^2 + b^2}, & y_2 &= y_t + \frac{b^2}{a^2 m^2 + b^2} \end{aligned} \quad (1)$$

where  $m = \tan \phi$ . For each directional vector, face center candidates are calculated and accumulated in accumulator cell. By selecting the points which have peak values in the accumulator cell, we can obtain the face center candidates.

## III. 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.

### A. PCA Concept

A technique commonly used for dimensionality reduction is PCA. In the late 1980's, Sirovich and Kirby efficiently represented human faces using PCA [13], 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  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$ , where  $\mathbf{x}_i \in \mathbb{R}^{N=mn}$ ,  $i = 1, 2, \dots, K$ . The basis functions for the Karhunen-Loeve Transform (KLT) [14] are obtained by solving the following eigenvalue problem:

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

where  $\Sigma$  is the covariance matrix of  $X$ ,  $\Phi$  is the eigenvector matrix of  $\Sigma$ , and  $\Lambda$  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:

$$\mathbf{y} = \Phi_M^T \tilde{\mathbf{x}} \quad (3)$$

where  $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$  is the difference between the image vector and the mean image vector, and  $\Phi_M$  is a submatrix of  $\Phi$  containing the  $M$  largest eigenvectors. These principal components preserve the major linear correlations in the given set of image vectors. By projecting to  $\Phi_M^T$ , 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:

$$\mathbf{y} = T(\mathbf{x}) : \Re^N \longrightarrow \Re^M. \quad (4)$$

By selecting  $M$  largest eigenvectors, we can obtain two subspaces. One is the principal subspace (or feature space)  $F$  containing the principal components, and the other is the orthogonal space  $\bar{F}$ . These two spaces are described in Fig. 3, where DFFS stands for “distance-from-feature-space” and DIFS “distance-in-feature-space”.

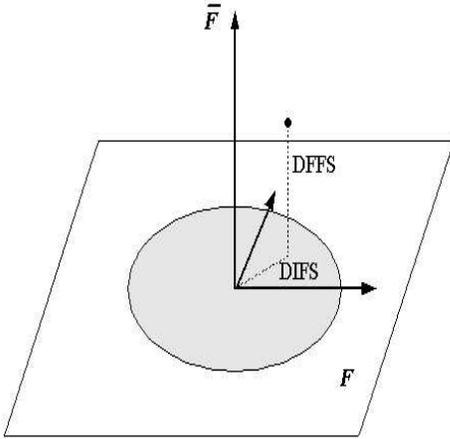


Fig. 3. Two subspaces

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

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

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.

### B. 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 [7]. 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  $\Omega$  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  $\lambda$  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  $\rho$  can be determined by minimizing a cost function, but  $\rho = \frac{1}{2} \lambda_{M+1}$  may be used as a thumb rule [15].

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 success rate and false alarm rate. If the threshold value is too low, the success rate would be quite good but the false alarm rate would also increase. For this reason, the threshold value has to be carefully selected.

#### IV. QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM (QEA)

QEA [10] 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 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,  $(\alpha, \beta)$ , 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. 4 shows the standard procedure of QEA. The procedure of QEA is explained as follows:

i) In the step of 'initialize  $Q(t)$ ,'  $\alpha_i^0$  and  $\beta_i^0, i = 1, 2, \dots, m$ , 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

#### 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  $\mathbf{b}_j^t$  in  $B(t)$  to  $B(t)$  locally

**end**

**end**

Fig. 4. Procedure of QEA.

following rotation gate is used as a basic Q-gate in QEA:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}, \quad (12)$$

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.

#### V. 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 success 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 for discriminating face images from the non-face images including the similar face images (position-shifted or different-scale face images). 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. 5(a) shows an example of a face training data set. It was produced from the face region of the AR face database [16]. 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. 5(b) shows an example of a non-face training data set.



(a) Face training data



(b) Non-face training data

Fig. 5. Examples of training data set

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:

$$P_{opt}(\mathbf{x}|\Omega) = \frac{\exp\left(-\frac{1}{2} \sum_{i=1}^M \frac{y_i^2}{\omega_i}\right)}{(2\pi)^{M/2} \prod_{i=1}^M \lambda_i^{1/2}} \cdot \frac{\exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(N-M)/2}}. \quad (13)$$

It is the same as (11) except for the weight factors  $\omega_i, i =$

$1, 2, \dots, M$ . To apply (13) 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 success 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

$$Fitness = P\_score + N\_score \quad (14)$$

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.

## VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we show two kinds of results. One is the excellence of the verification algorithm, and the other is the performance of the overall face detection system.

We constructed 3 types of database for the verification. 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 the RIT, KAIST laboratory by USB camera. An example of the RIT database is shown in Fig. 6.

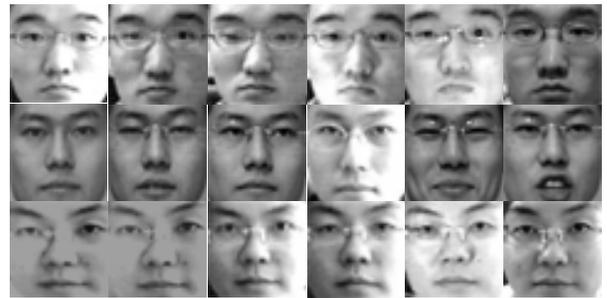


Fig. 6. RIT face database

All images are  $40 \times 40$  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 data were produced from the face region of the AR face database. These are normalized by specific position, scale and rotation. An example of a face training data set is shown in Fig. 5(a). Variations of the facial expression and illumination changes were allowed. Negative 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 data. An example of the negative data set is shown in Fig. 5(b).

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

$$0.1\lambda_i < \omega_i < 10\lambda_i, \quad (1 \leq i \leq 40). \quad (15)$$

Each weight factor was represented by 10 Q-bits. A Q-bit individual is illustrated in Fig. 7. We performed QEA for

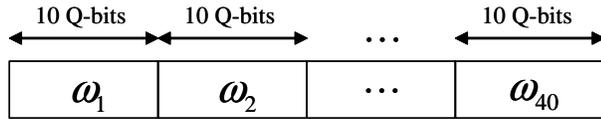


Fig. 7. Structure of a Q-bit individual

1000 training images using the parameters in Table I. In (12), rotation angles should be selected properly. For each Q-bit,  $\theta_1 = 0, \theta_2 = 0, \theta_3 = 0.01\pi, \theta_4 = 0, \theta_5 = -0.01\pi, \theta_6 = 0, \theta_7 = 0$ , and  $\theta_8 = 0$  were used.

TABLE I  
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 termination condition was given by the maximum generation. The perfect score was 1000 points. 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. By using it, we performed a generalization test. 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 fitness 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.

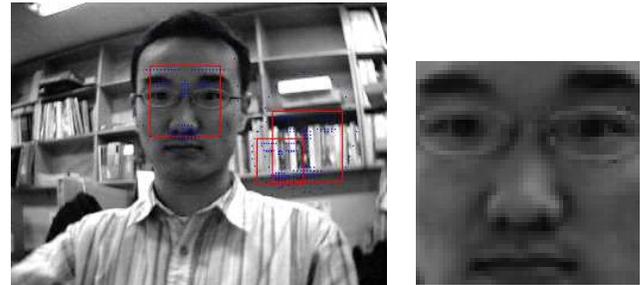
Table II shows the results for the generalization test. It shows that the proposed method performs better than the DFFS or the ML classifier.

The results suggest that the QEA-based classifier works well not only in terms of the success rate ( $P\_score$ ), but also in terms of the false alarm rate ( $N\_score$ ). The success 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.

Fig. 8 shows an example of the face detection results. From the input image, face location is performed as described in Section II. There are three face candidate regions in Fig.

TABLE II  
RESULTS FOR GENERALIZATION TEST

Face database	Classifier	Success 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



(a) Face candidate regions

(b) Detected face

Fig. 8. Example of the face detection

8(a). For each candidate region, face verification is performed by varying translation, scale and rotation. After this process, normalized face region is acquired by selecting the region which has maximum value in (13). Fig. 8(b) shows normalized face region as a detection result. The detected face is well normalized such as the faces in the training data set.

## VII. CONCLUSION

In this paper, we proposed a new face detection system using QEA. Our face detection system consisted of two parts, face location and face verification. In face location part, generalized Hough transform was employed to find the face candidate regions. In face verification part, QEA was used for fine location. To improve the previous Mahalanobis distance-based probability, we used a new distance which consists of the weight factors optimized by the training set. The performance of the proposed face verification algorithm was demonstrated through the improved success rate and false alarm rate. Also overall face detection system was tested by images from USB camera.

## REFERENCES

- [1] G. Holst, "Face detection by facets : Combined bottom-up and top-down search using compound templates," in *Proceedings of the International Conference on Image Processing*, TA07-08, 2000
- [2] M. Propp and A. Samal, "Artificial neural network architecture for human face detection," *Intell. Eng. Systems Artificial Neural Networks*, vol. 2, pp. 535-540, 1992.
- [3] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 23-38, January, 1998
- [4] D. Maio and D. Maltoni, "Real-time face location on gray-scale static images," *Pattern Recognition*, vol. 33, pp. 1525-1539, 2000
- [5] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cog. Neurosci.*, vol. 3, pp. 71-86, 1991

- [6] A. Pentland, B. Moghaddam and T. Strarner, "View-based and modular eigenspaces for face recognition," in *IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 84-91, 1994
- [7] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 696-710, 1997
- [8] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 39-51, 1998
- [9] J.-S. Jang, K.-H. Han and J.-H. Kim, "Quantum-inspired Evolutionary Algorithm-based Face Verification," in *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 2147-2156, 2003
- [10] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evolutionary Computation*, vol. 6, pp. 580-593, 2002
- [11] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, He Gate, and two phase scheme," *accepted for publication in the IEEE Trans. Evolutionary Computation*, vol. 8, No. 2, 2004
- [12] M. J. Donahue, S. I. Rokhlin, "On the use of level curves in image analysis," *Image Understanding*, vol. 57, pp. 185-203, 1993
- [13] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer.*, vol. 4, pp. 519-524, 1987
- [14] M. Loeve, *Probability Theory*, Princeton, N.J., Van Nostrand, 1955
- [15] T. Cootes, A. Hill, C. Taylor and J. Haslam, "Use of active shape models for locating structures in medical images," *Image and Vision Computing*, vol. 12, pp. 355-365, 1994
- [16] [http://rv11.ecn.purdue.edu/~aleix/aleix\\_face\\_DB.html](http://rv11.ecn.purdue.edu/~aleix/aleix_face_DB.html)