AN INTERPOLATION METHOD FOR THE RECONSTRUCTION AND RECOGNITION OF FACE IMAGES

N. C. Nguyen and J. Peraire
Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 20139, USA
cuongng@mit.edu and peraire@mit.edu

Keywords: Face reconstruction, face recognition, best points interpolation method, principal component analysis.

Abstract: An interpolation method is presented for the reconstruction and recognition of human face images. Basic ingredients include an optimal basis set defining a low-dimensional face space and a set of “best interpolation pixels” capturing the most relevant characteristics of known faces. The best interpolation pixels are chosen as points of the pixel grid so as to best interpolate the set of known face images. These pixels are then used in a least-squares interpolation procedure to determine interpolant components of a face image very inexpensively, thereby providing efficient reconstruction of faces. In addition, the method allows a fully automatic computer system to be developed for the real-time recognition of faces. Two significant advantages of this method are: (1) the computational cost of recognizing a new face is independent of the size of the pixel grid; and (2) it allows for the reconstruction and recognition of incomplete images.

1 INTRODUCTION

Image processing and recognition of human faces constitutes a very active area of research. The field has evolved rapidly and become one of the most successful applications of image analysis and computer vision partly because of availability of many powerful methods and partly because of its significant practical importance in many areas such as authenticity in security and defense systems, banking, human–machine interaction, image and multimedia processing, psychology, and neurology. Principal component analysis (PCA) or the Karhunen-Loève (KL) expansion is a well-established method for the representation (Sirovich and Kirby, 1987; Kirby and Sirovich, 1990) and recognition (Turk and Pentland, 1991) of human faces.

PCA approach (Kirby and Sirovich, 1990) for face representation consists of computing the “eigenfaces” of a set of known face images and approximating any particular face by a linear combination of the leading eigenfaces. For face recognition (Turk and Pentland, 1991), a new face is first projected onto the eigenface space and then classified according to the distances between its PCA coefficient vector and those representing the known faces. There are two drawbacks with this approach. First, PCA may not handle corrupted data well, that is, situations in which only partial information of an input image is available. Secondly, the computational cost per image classification depends on the size of the pixel grid. Despite this, PCA is still one of the most used techniques for face recognition due to its simplicity and efficiency over other methods.

This paper describes an interpolation method that aims to address these deficiencies of PCA. The method was first introduced in (Nguyen et al., 2006) for the approximation of parametrized fields. Here, we investigate the method for face reconstruction and recognition. The basic ingredient is a set of “best interpolation pixels” capturing the most relevant features of known face images. The essential component is a least-squares interpolation procedure for the very rapid computation of the interpolant coefficient vector of any given input face. The interpolant coefficient vector is then used to determine which face in the face set, if any, best matches the input face. A significant advantage of our approach is that the computational cost of recognizing a new face is *independent* of the size of the pixel grid, while achieving a recognition rate comparable to PCA approach. Moreover, the method allows the reconstruction and recognition of corrupted images.

In the past years, there have been a large number of contributions to face recognition and analysis. The reader is referred to a number of papers (Chellappa et al., 1995; Jain and Li, 2005; Delac et al., 2005; Draper et al., 2003; Phillips et al., 1998) for perspectives and recent advances in face recognition. Face analysis and representation have also been extensively studied by many authors (Sirovich and Kirby, 1987; Kirby and Sirovich, 1990; O'Toole et al., 1993; Everson and Sirovich, 1995; Kanade, 2005).

This paper is organized as follows. In Section 2, we present an overview of PCA. In Section 3, we extend the best points interpolation method (BPIM) introduced in (Nguyen et al., 2006) and apply it to develop an automatic *real-time* face recognition system. In section 4, we test and compare our approach with PCA. Finally, in Section 5, we close the paper with some concluding remarks.

2 PRINCIPAL COMPONENT ANALYSIS

2.1 Eigenfaces

An ensemble of face images is denoted by $U_K = \{u_i\}$, $1 \le i \le K$, where u_i represents an i-th mean-subtracted face and K represents the number of faces in the ensemble. It is assumed that after proper normalization and resizing to a fixed pixel grid Ξ of dimension N_1 by N_2, u_i can be considered as a vector in an N-dimensional image space, where $N = N_1 N_2$ is the number of pixels. PCA (Sirovich and Kirby, 1987; Kirby and Sirovich, 1990) derives an optimal representation of the face ensemble in the sense that the average reconstruction error

$$\bar{\varepsilon}^* = \sum_{i=1}^{K} \left\| u_i - \sum_{j=1}^{k} (\phi_j^T u_i)\phi_j \right\|^2 , \tag{1}$$

is minimal for all $k \le K$. In the literature (Turk and Pentland, 1991), the basis vectors ϕ_j are referred as *eigenfaces* and the space spanned by them is known as the *face space*. The construction of the eigenfaces is described as follows.

Let $\mathbf{U}$ be the $N \times K$ matrix whose columns are $[u_1, \ldots, u_K]$. It can be shown that the ϕ_i satisfy

$$\mathbf{A}\phi_i = \lambda_i \phi_i , \tag{2}$$

where the covariance matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \frac{1}{K}\mathbf{U}\mathbf{U}^T . \tag{3}$$

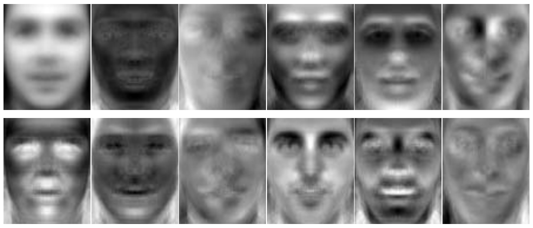

Figure 1: Eigenfaces and the mean face. The mean face is on the top left and followed by 11 top eigenfaces, in order from left to right and top to bottom.

Here the eigenvalues are arranged such that $\lambda_1 \ge \ldots \ge \lambda_K$. Since the matrix $\mathbf{A}$ of size $N \times N$ is large, solving the above eigenvalue problem can be very expensive.

However, if $K < N$, there will be only K meaningful eigenvectors and we may express any ϕ_i as

$$\phi_i = \sum_{j=1}^{K} \varphi_{ij} u_j . \tag{4}$$

Inserting (3) and (4) into (2), we immediately obtain

$$\mathbf{G}\varphi_i = \lambda_i \varphi_i , \tag{5}$$

where $\mathbf{G} = \frac{1}{K}\mathbf{U}^T\mathbf{U}$ is a symmetric positive-definite matrix of size K by K. The eigenvalue problem (5) can be solved for $\varphi_{ij}, 1 \le i, j \le K$, from which the eigenfaces ϕ_i are obtained.

2.2 Face Reconstruction

We briefly describe the reconstruction of face images using PCA and later compare the results with those obtained using our method. To this end, we seek to project an input face u onto the face space $\Phi_k = \text{span}\{\phi_1, \ldots, \phi_k\}$ to obtain

$$u^* = \sum_{i=1}^{k} a_i \phi_i , \tag{6}$$

where for $i = 1, \ldots, k$,

$$a_i = \phi_i^T u . \tag{7}$$

We also define the associated error as

$$\varepsilon^* = \|u - u^*\| . \tag{8}$$

Note that the mean face of the ensemble U_K should be added to u^* to obtain the reconstructed image; and that if k is set equal to K, the reconstruction is exact for all members of the ensemble.

We present in Figure 1 the mean face and a few of the top eigenfaces for a training ensemble of 400 face images extracted from the AT&T database (see Section 4.1 for details). Figure 2 shows $\bar{\varepsilon}^*_{\text{rel}}$ as a function of k. Here $\bar{\varepsilon}^*_{\text{rel}}$ is the average of the relative error

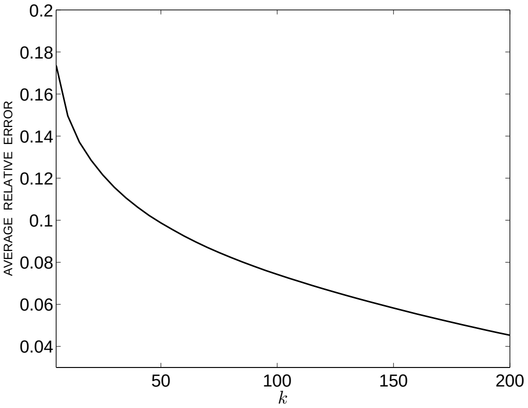

Figure 2: Average relative error $\bar{\varepsilon}^*_{\text{rel}}$ versus k for the training ensemble.

$\varepsilon^*/\|u\|$ over the training ensemble. Note that the slow convergence is mostly due to the wide and complex variation of faces in the ensemble. Much faster convergence and smaller error can be achieved by only considering the frontal view and the oval-shaped portion for the faces (Kirby and Sirovich, 1990).

2.3 Face Recognition

We briefly describe the eigenface recognition procedure of Turk and Pentland (Turk and Pentland, 1991). To classify an input image, one first obtains PCA coefficients $a_i, 1 \le i \le k$, as described above. One then computes the Euclidean distances between its PCA coefficient vector $\mathbf{a} = [a_1, \ldots, a_k]^T$ and those representing each individual in the training ensemble. Depending on the smallest distance and the PCA reconstruction error ε^*, the image is classified as belonging to a familiar individual, as a new face, or a non-face image. Several variants of the above procedure are possible via the use of a different classifier such as the nearest-neighbor classifier and a different norm such as L_1 norm or Mahalanobis norm (Delac et al., 2005).

It is generally observed that the recognition performance is improved when using a larger k. Typically, the number of eigenfaces k required for face recognition varies from $O(10)$ to $O(10^2)$ and is *much* smaller than N. We note that classification of an input image requires the evaluation of PCA coefficients according to (7). The computational cost per image classification is thus at least $O(Nk)$. This cost depends linearly on N and is quite acceptable for a *small* number of input images. However, when classification of a *very large* number of faces is performed at the same time, PCA approach appears increasingly intractable. Real-time recognition is thus excluded for large-scale applications. Other subspace methods such as independent component analysis (ICA) (Draper et al., 2003; Bartlett et al., 2002) and linear discriminant analysis (LDA) (Etemad and Chellappa, 1997; Lu et al., 2003) suffer from similar drawbacks.

3 BEST POINTS INTERPOLATION METHOD

In this section, we extend the best points interpolation method (Nguyen et al., 2006) to deal with face images and apply it for face recognition. We shall use the eigenfaces as basis functions in the interpolation process, as they possess optimal L_2 representation of face images. The key idea, however, is to find a set of interpolation points which provides a good uniform approximation.

3.1 Interpolation Procedure

Let us recall the pixel grid Ξ and the face space $\Phi_k = \text{span}\{\phi_1, \ldots, \phi_k\}$. In this space, we shall seek an approximation of any input image u. However, rather than performing the projection of u onto Φ_k for the best approximation u^*, we choose to interpolate u as follows.

In particular, we aim to find a good approximation $\tilde{u} \in \Phi_k$ of u via $m(\ge k)$ interpolation pixels $\{\mathbf{z}_j \in \Xi\}, 1 \le j \le m$, such that

$$\tilde{u} = \sum_{i=1}^{k} \tilde{a}_i \phi_i \tag{9}$$

where the coefficients $\tilde{a}_i$ are the solution of

$$\sum_{i=1}^{k} \phi_i(\mathbf{z}_j)\tilde{a}_i = u(\mathbf{z}_j), \quad j = 1, \ldots, m \ . \tag{10}$$

We define the associated error as

$$\tilde{\varepsilon} = \|u - \tilde{u}\| \ . \tag{11}$$

In general, the linear system (10) is over-determined because there are more equations than unknowns. However, the interpolant coefficient vector $\tilde{\mathbf{a}} = [\tilde{a}_1, \ldots, \tilde{a}_k]^T$ can be determined from

$$\mathbf{C}^T\mathbf{C}\tilde{\mathbf{a}} = \mathbf{C}^T\mathbf{c} \ , \tag{12}$$

where $\mathbf{C} \in \mathbb{R}^{m\times k}$ with $C_{ji} = \phi_i(\mathbf{z}_j), 1 \le i \le k, 1 \le j \le m$ and $\mathbf{c} = [u(\mathbf{z}_1), \ldots, u(\mathbf{z}_m)]^T$. It thus follows that

$$\tilde{\mathbf{a}} = \mathbf{B}\mathbf{c} \ . \tag{13}$$

Here the matrix $\mathbf{B} = \left(\mathbf{C}^T\mathbf{C}\right)^{-1}\mathbf{C}^T$ is precomputed and stored. Therefore, for any new face u, the cost of evaluating the interpolant coefficient vector $\tilde{\mathbf{a}}$ is only $O(mk)$ and becomes $O(k^2)$ when $m = O(k)$.

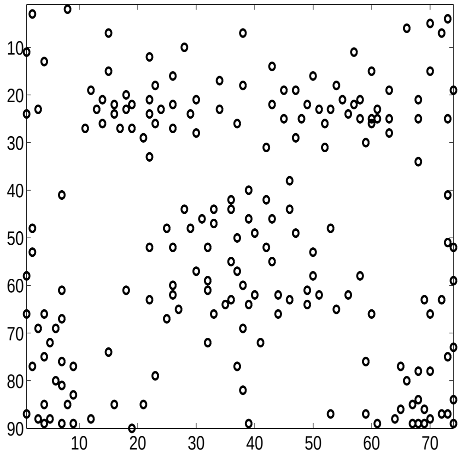

Figure 3: Distribution of the best interpolation pixels on the pixel grid for $k = 100$ and $m = 200$.

k	$\bar{\tilde{\varepsilon}}_{\text{rel}}$	$\overline{\Upsilon}$
20	0.1548	1.2048
40	0.1322	1.2448
60	0.1158	1.2524
80	0.1047	1.2712
100	0.0962	1.2954
120	0.0888	1.3189
140	0.0813	1.3295

Table 1: Average interpolation error $\bar{\tilde{\varepsilon}}_{\text{rel}}$ and average error ratio $\overline{\Upsilon}$ for different values of k for $m = 2k$.

Obviously, the approximation quality depends crucially on the interpolation pixels $\{\mathbf{z}_j\}$. Therefore, it is extremely important to choose $\{\mathbf{z}_j\}$ so as to guarantee accurate and stable interpolation. For instance, Figure 3 shows the interpolation pixels for $k = 100$ and $m = 200$ obtained using our method described below. We see that the pixels are distributed somewhat symmetrically with respect to the symmetry line of the face and largely allocated around main locations of the face such as eyes, nose, mouth, and jaw. Furthermore, we present in Table 1 $\bar{\tilde{\varepsilon}}_{\text{rel}}$ and $\overline{\Upsilon}$ as a function of k for the choice of $m = 2k$. Here $\bar{\tilde{\varepsilon}}_{\text{rel}}$ and $\overline{\Upsilon}$ are the average of the relative error $\tilde{\varepsilon}/\|u\|$ and the average of the error ratio $\tilde{\varepsilon}/\varepsilon^*$ over the training ensemble, respectively. The average error ratio $\overline{\Upsilon}$ is quite close to unity for all k. This means that the approximation $\tilde{u}$ is also close to the "best" approximation u^*. However, unlike u^*, $\tilde{u}$ requires only m intensity values of u.

3.2 Best Interpolation Pixels

We proceed to describe our approach for determining the interpolation pixels. The crucial observation is that much of the surface of a face is smooth with regular texture and that faces are similar in appearance and highly constrained; for example, the frontal view of a face is symmetric. Moreover, the value of a pixel is typically highly correlated with the values of the surrounding pixels. Therefore, a large number of pixels in the image space does not represent physically possible faces. Only a small number of pixels may suffice to represent facial characteristics. The question we aim to address is to find such pixels and proceed with our interpolation.

We choose the interpolation pixels by exploiting the training ensemble U_K. Specifically, we might consider to choose $\{\mathbf{z}_j\}$ by formulating a minimization problem that minimizes the sum of squared errors between $u_\ell, 1 \le \ell \le K$, and their approximations. More precisely, we might wish to find $\{\mathbf{z}_j\}$ as a minimizer of

$$\min_{\mathbf{x}_1\in\Xi,\ldots,\mathbf{x}_m\in\Xi} \sum_{\ell=1}^{K} \left\| u_\ell - \sum_{i=1}^{k} \tilde{a}_{\ell i}(\mathbf{x}_1,\ldots,\mathbf{x}_m)\phi_i \right\|^2 \qquad (14)$$

$$\sum_{i=1}^{k} \phi_i(\mathbf{x}_j)\tilde{a}_{\ell i} = u_\ell(\mathbf{x}_j), \quad 1 \le j \le m, 1 \le \ell \le K.$$

Clearly, the minimizer of the above error minimization problem is optimal for the interpolation of the face images belonging to U_K. However, since the problem is nonconvex with multiple local minima and the Hessian is not easily computed, solving it is particularly difficult. In practice, we find $\{\mathbf{z}_j\}$ by solving a simpler minimization problem introduced below.

To begin, we introduce a set of images, $U_K^* = \{u_\ell^*\}, 1 \le \ell \le K,$, where u_ℓ^* is the best approximation to u_ℓ. It thus follows that

$$u_\ell^* = \sum_{i=1}^{k} a_{\ell i}\phi_i, \qquad (15)$$

where for $1 \le i \le k, 1 \le \ell \le K$,

$$a_{\ell i} = \phi_i^T u_\ell \;. \qquad (16)$$

By replacing u_ℓ in the objective of the problem (14) with u_ℓ^* and expanding the resulting objective, we arrive at the nonlinear least squares minimization

$$\min_{\mathbf{x}_1\in\Xi,\ldots,\mathbf{x}_m\in\Xi} \sum_{\ell=1}^{K} \sum_{i=1}^{k} \left(a_{\ell i} - \tilde{a}_{\ell i}(\mathbf{x}_1,\ldots,\mathbf{x}_m)\right)^2 \qquad (17)$$

$$\sum_{i=1}^{k} \phi_i(\mathbf{x}_j)\tilde{a}_{\ell i} = u_\ell(\mathbf{x}_j), \quad 1 \le j \le m, 1 \le \ell \le K.$$

Let us denote a minimizer of this problem by $\{\mathbf{z}_j\}, 1 \le j \le m$. We shall call the $\mathbf{z}_j$ as *best interpolation pixels*, because these pixels are optimal for the interpolation of the best approximations u_ℓ^*. It remains to describe the solution procedure for (17).

3.3 Solution Procedure

We first write the linear system in (17) for $\tilde{\mathbf{a}}_\ell = [\tilde{a}_{\ell 1}, \ldots, \tilde{a}_{\ell k}]^T$ into the matrix form as

$$\mathbf{D}^T \mathbf{D} \tilde{\mathbf{a}}_\ell = \mathbf{D}^T \mathbf{d}_\ell, \quad 1 \le \ell \le K, \tag{18}$$

where $\mathbf{d}_\ell = [u_\ell(\mathbf{x}_1), \ldots, u_\ell(\mathbf{x}_m)]^T$ and $\mathbf{D} \in \mathbb{R}^{m \times k}$ with $D_{ji} = \phi_i(\mathbf{x}_j)$. Next let $\mathbf{s} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T$, for $1 \le i \le k, 1 \le \ell \le K$, $1 \le q \le Q = kK$, we set

$$f_q(\mathbf{s}) \quad = \quad a_{\ell i} - \tilde{a}_{\ell i}(\mathbf{s}); \tag{19}$$

$$F(\mathbf{s}) \quad = \quad \frac{1}{2} \sum_{q=1}^{Q} f_q^2(\mathbf{s}) \; . \tag{20}$$

The gradient and Hessian of the objective function $F(\mathbf{s})$ can thus be computed as

$$\nabla F(\mathbf{s}) = \sum_{q=1}^{Q} f_q(\mathbf{s}) \nabla f_q(\mathbf{s}) = \mathbf{J}(\mathbf{s})^T \mathbf{f}(\mathbf{s}) \; , \tag{21}$$

$$\nabla^2 F(\mathbf{s}) = \mathbf{J}(\mathbf{s})^T \mathbf{J}(\mathbf{s}) + \sum_{q=1}^{Q} f_q(\mathbf{s}) \nabla^2 f_q(\mathbf{s}) \; , \tag{22}$$

where $\mathbf{J}(\mathbf{s}) \in \mathbb{R}^{Q \times 2m}$, for $1 \le q \le Q$, $1 \le p \le 2m$,

$$J_{qp}(\mathbf{s}) = \frac{\partial f_q(\mathbf{s})}{\partial x_j^d}, \quad 1 \le j \le m, d = 1, 2. \tag{23}$$

Hence, when the residuals $f_q(\mathbf{s})$ are small, we may approximately compute the Hessian in terms of only the Jacobian matrix $\mathbf{J}(\mathbf{s})$ as

$$\nabla^2 F(\mathbf{s}) = \mathbf{J}(\mathbf{s})^T \mathbf{J}(\mathbf{s}) \; . \tag{24}$$

To compute the Jacobian $\mathbf{J}(\mathbf{s})$, we differentiate both sides of (18) with respect to x_j^d to obtain

$$\frac{\partial \tilde{\mathbf{a}}_\ell}{\partial x_j^d} = \mathbf{E}^{-1} \left(\frac{\partial \mathbf{D}^T}{\partial x_j^d} \mathbf{d}_\ell + \mathbf{D}^T \frac{\partial \mathbf{d}_\ell}{\partial x_j^d} - \frac{\partial \mathbf{E}}{\partial x_j^d} \tilde{\mathbf{a}}_\ell \right) \; ,$$

where $\mathbf{E} = \mathbf{D}^T \mathbf{D}$. The partial derivatives $\partial \mathbf{d}_\ell / \partial x_m^d$, $\partial \mathbf{D}^T / \partial x_m^d$, and $\partial \mathbf{E} / \partial x_m^d$ are computed by finite differences. Note also that $\mathbf{x} = (x^1, x^2)$.

Having determined the gradient and the Hessian, we may now use the Levenberg-Marquardt (LM) algorithm (Marquardt, 1963) to solve (17). The LM algorithm is very efficient, but it is sensitive to an initial guess. Hence it is important to start the algorithm with a good initial guess. In our implementation, we use the empirical interpolation method (Barrault et al., 2004; Grepl et al., 2006) to obtain an initial set of interpolation points $\{\mathbf{z}_j^{\text{ig}}\}$ as follows. We first set

$$\mathbf{z}_1^{\text{ig}} = \arg \sup_{\mathbf{x} \in \Xi} |\phi_1(\mathbf{x})| \; . \tag{25}$$

Then for $\ell = 2, \ldots, m$, we solve the linear system

$$\sum_{j=1}^{\ell-1} \phi_j(\mathbf{z}_i^{\text{ig}}) \, \sigma_j = \phi_\ell(\mathbf{z}_i^{\text{ig}}), \quad 1 \le i \le \ell - 1, \tag{26}$$

for $\sigma_j, 1 \le j \le \ell - 1$, and set

$$\mathbf{z}_\ell^{\text{ig}} = \arg \sup_{\mathbf{x} \in \Xi} \left| \phi_\ell(\mathbf{x}) - \sum_{j=1}^{\ell-1} \sigma_j \, \phi_j(\mathbf{x}) \right| \; . \tag{27}$$

This set of points, when used as an initial guess, yields very satisfactory results. For further details of the empirical interpolation method, we refer the reader to (Barrault et al., 2004; Grepl et al., 2006).

3.4 Application to Face Recognition

We now apply the BPIM to develop a fully automatic *real-time* face recognition system involving the generation stage and the recognition stage. The detailed implementation of the system is given below:

1. Determine the dimension of the face space k and choose some m (say $m = 2k$), then calculate $\phi_1, \ldots, \phi_m$. Note m eigenfaces are required to obtain the initial guess $\mathbf{z}_j^{\text{ig}}, 1 \le j \le m$.
2. Compute and store $\{\mathbf{z}_j\}$, $\mathbf{B} = \left(\mathbf{C}^T \mathbf{C}\right)^{-1} \mathbf{C}^T$. Recall that $C_{ji} = \phi_i(\mathbf{z}_j), 1 \le i \le k, 1 \le j \le m$.
3. For a "gallery" of images $V_{K'} = \{v_i\}, 1 \le i \le K'$, compute $\tilde{\mathbf{a}}_i = \mathbf{B}[v_i(\mathbf{z}_1), \ldots, v_i(\mathbf{z}_m)]^T, 1 \le i \le K'$. (Note $V_{K'}$ can be the same or different from U_K).
4. For each new face to be classified u, calculate its interpolant coefficient vector $\tilde{\mathbf{a}}$ from (13) and find
$$i_{\min} = \arg \min_{1 \le i \le K'} \|\tilde{\mathbf{a}} - \tilde{\mathbf{a}}_i\| \; . \tag{28}$$
5. If $\|\tilde{\mathbf{a}} - \tilde{\mathbf{a}}_{i_{\min}}\|$ is less than a chosen threshold, the input image u is identified as the individual associated with the coefficient vector $\tilde{\mathbf{a}}_{i_{\min}}$. Otherwise, the image is classified as a new individual.

The generation stage (steps 1–2) is computationally expensive, but performed only when the training set changes. Furthermore, even if it is necessary to perform the generation stage due to an update of the training set, we may compute only the eigenfaces and *reuse* the best pixels. This can save us some computational time.

However, the recognition stage (steps 4–5) is very inexpensive. The calculation of $\tilde{\mathbf{a}}$ takes $O(mk)$. Note further that the problem (28) is the nearest-neighbor search which can be solved in $O(kK'^{0.25})$ time with the storage of $O(kK' + K' \log K')$ (Andoni and Indyk, 2006). Hence, if K' is in order of $O(k^4)$ or less, the computational cost is only $O(k^2)$. This is usually the case even for large-scale applications; for example, for a training database of 10^4 images, one would need more (or many more) than 10 eigenfaces to achieve acceptable recognition rates.

In summary, the operation count of the recognition stage is about $O(mk)$. The computational complexity of our system is thus *independent* of N. As mentioned earlier, the complexity of PCA-based algorithms is at least $O(Nk)$. Our approach leads to a computational reduction of N/m relative to PCA. Since m is typically much smaller than N, significant savings are expected. The savings per image classification certainly translate to real-time performance especially when many face images need to be classified simultaneously.

However, some applications of face recognition may regard the recognition quality more importantly than the computational performance. Therefore, in order to be useful and gain acceptance, our approach must be tested and compared with existing approaches, particularly here with the PCA.

4 EXPERIMENTS

4.1 Face Database

The AT&T face database (Samaria and Harter, 1994) consists of 400 images of 40 individuals (10 images per individual). The images were taken at different times with variation in lighting, poses, and facial expressions, with and without glasses. The images were cropped and resized by us to a resolution of 74×90. We formed a training ensemble of 400 images by using 200 images of the database, 10 each of 20 different individuals, and including 200 mirror images of these images (Kirby and Sirovich, 1990).

The testing set contains the (200) remaining images of 20 individuals not belonging to the training ensemble. We further divide the testing set into the gallery of 20 individual faces and 180 probe images containing 9 views of every individual in the gallery. The recognition task is to match the probe images to the 20 gallery faces. The fact that the training and testing sets have no common individual serves to assess the performance of a face recognition system more critically — the ability to recognize new faces which are not part of the face space constructed from the training set.

4.2 Results for Face Reconstruction

We first present in Figure 4 the reconstruction results for a face in the training ensemble. The BPIM produces reconstructions almost as well as PCA: most facial features captured by the PCA reconstructed images also appear in the BPIM reconstructed images. We underline the fact that the interpolation method requires less than 5% of the total number of pixels $N = 6660$, but delivers quite satisfactory results.

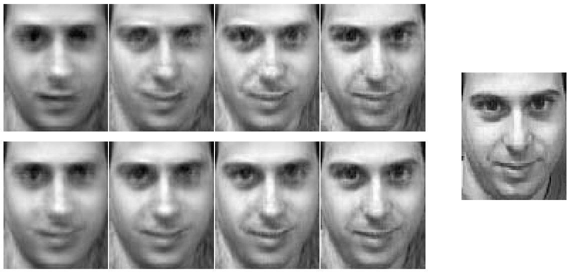

Figure 4: The reconstruction results for a familiar face. The BPIM reconstructed images are placed at the top row for $k = 40, 80, 120, 160$ (from left to right) and $m = 2k$. The PCA reconstructed images are placed at the second row for $k = 40, 80, 120, 160$ (from left to right). The original face is shown on the right.

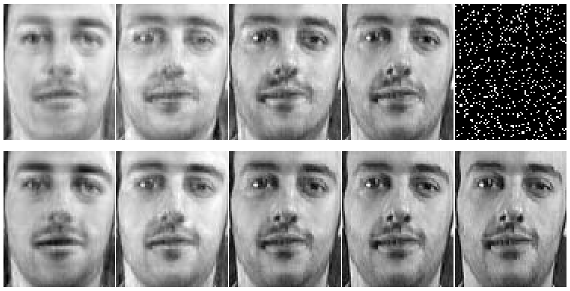

Figure 5: Reconstruction of a familiar face (bottom right) from a 10% mask (top right) with only the white pixels. The reconstructed images are shown at the top row for $k = 40, 80, 120, 160$ (from left to right) and $m = 2k$. The PCA reconstructed images are shown at the second row for $k = 40, 80, 120, 160$ (from left to right) with using all the pixels.

To illustrate the use of the interpolation approach for reconstructing a full image from a partial image, we consider a face (in the training set) shown at the bottom right and a mask shown at the top right in Figure 5. This is a relatively extreme mask that obscures 90% of the pixels in a random manner. Because the masked face may not have intensity values at all the best interpolation pixels, we need to define a new set of interpolation pixels. To this end, we keep the best interpolation pixels which coincide with some of the white pixels of the masked face and replace the remaining best pixels with the “nearest” white pixels. In Figure 5, the reconstructed images using those interpolation pixels are compared with the PCA reconstructed images utilitizing all the pixels. Although the

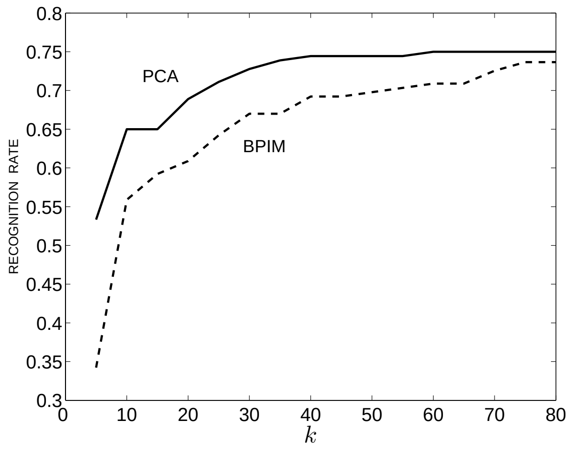

Figure 6: Recognition accuracy of PCA and BPIM with increasing the number of eigenfaces k. Note that the BPIM uses $m = 2k$ best interpolation pixels.

interpolation procedure does not recover the original face exactly, the construction is visually close to the "best" reconstruction.

4.3 Results for Face Recognition

We apply the face recognition system developed in Section 3.4 to classify the probe images. We illustrate in Figure 6 the recognition accuracy as a function of k for the BPIM and PCA. As it may be expected, the BPIM yields smaller recognition rates than PCA. However, as k increases, the BPIM gives recognition rates which are quite comparable to those of PCA for large enough k: PCA achieves a recognition rate of 74.98%, while PBIM results in a recognition rate of 73.66% for $k = 80$. In many applications, the small accuracy loss of only 1.32% is paid off very well by the significant reduction of $6660/160(> 40)$ in complexity. This is confirmed in Table 2 which shows the computational times for the BPIM and PCA. The values are normalized with respect to the time to recognize a face for $k = 10$ and $m = 20$ with the BPIM. Clearly, the BPIM is significantly faster than PCA. This important advantage is very useful to applications that requires a real-time recognition capability.

Finally, in order to appreciate the power of the interpolation approach for classifying corrupted images, we consider a random chosen mask of 10% pixels shown in Figure 7. Next to the mask, we show a few faces which are correctly recognized with using the interpolation procedure when their intensity values are available only at the white pixels of the mask. Note here that the interpolation pixels are chosen in the same way as before.

k	BPIM	PCA
10	1.00	333.30
20	4.20	592.67
30	9.33	873.34
40	15.60	1107.66
50	26.12	1437.35
60	36.47	1708.02
70	47.93	1958.94
80	61.47	2293.73

Table 2: Computational times (normalized with respect to the time to recognize a face for $k = 10$ and $m = 20$ with the BPIM) for the BPIM and PCA at different values of k.

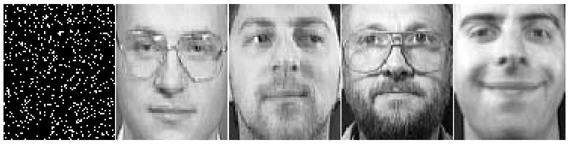

Figure 7: Recognition of corrupted face images. The 10% mask on the left is followed by a few faces which are correctly recognized with using the interpolation procedure.

5 CONCLUSION

We have presented a best points interpolation method for the reconstruction and recognition of face images. The method gives very good reconstruction of face images. Therefore, the method has significant potential to be used as an alternative to PCA for data reconstruction. It is important to note that PCA uses full knowledge of the data in the reconstruction process. In contrast, the BPIM uses only partial knowledge of the data. Therefore, the BPIM is very useful to the restoration of a full image from a partial image.

We have also developed a fully automatic real-time face recognition system based on the BPIM. The system is shown to be able to recognize corrupted images. Moreover, the computational cost of recognizing a new face is only $O(mk)$, translating to a saving of N/m relative to PCA approach. Typically, since N is $O(10^4)$ and m is $O(10^2)$, this implies two orders of magnitude less expensive computationally than PCA. As confirmed in Figure 6 and Table 2, the system is significantly faster, while yielding a comparable recognition rate for large enough k, than a standard PCA-based system. The significant reduction in time should enable us to tackle very large problems. Hence, it is imperative to test our system on a larger database such as the FERET database. We plan to pursue this direction in future research.

The present work may also present an opportunity combining the BPIM with some subspace techniques.

For example, in the context of ICA, instead of using intensity values at all pixels one may use intensity values at *only* the best interpolation pixels to build a ICA-based recognition system, thereby effecting significant computational savings. It should be mentioned that most presented ICA-based algorithms do not perform ICA directly on the training ensemble of face images, but on either eigenfaces or PCA coefficient vectors, to reduce the heavy computational cost. Although we have not put effort to investigate this direction, we believe that using partial image pixels will not reduce the recognition capability of ICA-based algorithms provided that a sufficient (small) number of the interpolation pixels is used. Furthermore, in ICA Architecture II (Draper et al., 2003; Bartlett et al., 2002), one may want to input the interpolant coefficient vectors (instead of PCA coefficient vectors) to ICA. Similarly, one may choose to perform LDA on the interpolant coefficient vectors to reduce the computational burden considerably.

Although we do not claim that our findings necessarily have a wide range of applications, we believe that our work could open a new direction of research in face recognition and image analysis in general.

ACKNOWLEDGEMENTS

We would like to thank Professor A. T. Patera of MIT for his long-standing collaboration and many invaluable contributions to this work. The authors also thank AT&T Laboratories Cambridge for providing the ORL face database. This work was supported by DARPA and AFOSR under Grant FA9550-05-1-0114 and F49620-03-1-0439, and by the Singapore-MIT Alliance.

REFERENCES

Andoni, A. and Indyk, P. (2006). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th IEEE Sym. on Foundations of Computer Science*, Berkeley, CA.

Barrault, M., Maday, Y., Nguyen, N. C., and Patera, A. T. (2004). An “empirical interpolation” method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Série I*, 339:667–672.

Bartlett, M. S., Movellan, J. R., and Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Trans. on Neural Networks*, 13:1450–1464.

Chellappa, R., Wilson, C. L., and Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–740.

Delac, K., Grgic, M., and Grgic, S. (2005). Independent comparative study of pca, ica, and lda on the feret data set. *International Journal of Imaging Systems and Technology*, 15(5):252–260.

Draper, B. A., Baek, K., Bartlett, M. S., and Beveridge, J. R. (2003). Recognizing faces with pca and ica. *Computer Vision and Image Understanding*, 91(1-2):115–137.

Etemad, K. and Chellappa, R. (1997). Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14(8):1724–1733.

Everson, R. and Sirovich, L. (1995). Karhunen-loeve procedure for gappy data. *Opt. Soc. Am. A*, 12(8):1657–1664.

Grepl, M. A., Maday, Y., Nguyen, N. C., and Patera, A. T. (2006). Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN Math. Model. Numer. Anal.* Submitted.

Jain, A. K. and Li, S. Z. (2005). *Handbook of Face Recognition*. Springer-Verlag New York.

Kanade, T. (2005). Facial expression analysis. In *Proceedings of the 2nd International Workshop on Analysis and Modelling of Faces and Gestures*, volume 3723 of *Lecture Notes in Computer Science*, page 1. Springer.

Kirby, M. and Sirovich, L. (1990). Application of the karhunen-loève procedure for the characterization of human face. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:103–108.

Lu, J., Plataniotis, K. N., and Venetsanopoulos, A. N. (2003). Face recognition using lda-based algorithms. *IEEE Trans. on Neural Networks*, 14(1):195–200.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11(1):431–444.

Nguyen, N. C., Patera, A. T., and Peraire, J. (2006). A best points interpolation method for efficient approximation of parametrized functions. *International Journal of Numerical Methods in Engineering*. Submitted.

O’Toole, A., Abdi, H., Deffenbacher, K. A., and Valentin, D. (1993). Low-dimensional representation of faces in higher dimensions of the face space. *J. of the Optical Society of America A*, 10(3):405–411.

Phillips, P. J., Wechsler, H., Huang, J. S., and Rauss, P. J. (1998). The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306.

Samaria, F. S. and Harter, A. C. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of the 2nd IEEE workshop on Applications of Computer Vision*, pages 138–142, Sarasota, Florida.

Sirovich, L. and Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4:519–524.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.