International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 26, No. 6 (2012) 1256015 (23 pages)

DOI: 10.1142/S0218001412560150

A LOW-COMPUTATION APPROACH FOR HUMAN FACE RECOGNITION

WEI-LI FANG, YING-KUEI YANG* and JUNG-KUEI PAN

Department of Electrical Engineering
National Taiwan University of Science and Technology
No. 43, Sec. 4, Keelung Road, Taipei, Taiwan
**ykyang@mouse.ee.ntust.edu.tw*

Received 8 February 2012
Accepted 25 September 2012
Published 7 December 2012

Several 2DPCA-based face recognition algorithms have been proposed hoping to achieve the goal of improving recognition rate while mostly at the expense of computation cost. In this paper, an approach named SI2DPCA is proposed to not only reduce the computation cost but also increase recognition performance at the same time. The approach divides a whole face image into smaller sub-images to increase the weight of features for better feature extraction. Meanwhile, the computation cost that mainly comes from the heavy and complicated operations against matrices is reduced due to the smaller size of sub-images. The reduced amount of computation has been analyzed and the integrity of sub-images has been discussed thoroughly in the paper. The experiments have been conducted to make comparisons among several better-known approaches and SI2DPCA. The experimental results have demonstrated that the proposed approach works well on reaching the goals of reducing computation cost and improving recognition performance simultaneously.

Keywords: Face recognition; feature extraction; principle component analysis; covariance computation; eigen-decomposition; image integrity.

1. Introduction

Face recognition in image processing has been significantly important because it can be applied in human life efficaciously. Research areas include building/store access control, suspect identification, security and surveillance.[1–4,6–8,13,18–20,23,24,26,28,30,33,35,36,38,40,42,43,49,52,53,55,56] Face recognition is not only natural but also intuitively appealing. For example, the most obvious feature of a person is his face observed by our eyes. One typical application to commercial products is technique of automatic face detection in digital camera, mobile phones and notebooks. The theory of face recognition can be combined with many artificial intelligence approaches to achieve better effect, so it has much higher research merit in this field.[6,49]

Several algorithms have been proposed in face recognition. The best ones are those that try not only to reduce computation cost but also to increase recognition rate.[25,50] Based on this viewpoint, principal component analysis (PCA)[37] has become a very popular feature extraction algorithm in recent decades. Sirovich and Kirby[17,37] were the first ones using PCA to recognize human faces successfully. Pujol *et al.*[31] presented topological PCA that combines PCA with the topological relationship existing among the variances of original data for face recognition. But this approach requires high computation cost caused by the diagonalization of covariance matrix. Kim *et al.*[15] proposed kernel PCA that combines with PCA to perform face recognition by projecting low dimensional data to be high dimensional ones. Although this approach has high recognition accuracy rate up to 97.5% when testing in ORL database, its projection process needs much computation cost. Wang *et al.*[46] proposed the approach of combining PCA with linear discriminant analysis (LDA) to do feature extraction in human face. The approach first uses PCA to extract important features, and then LDA is used to make face information more discriminative for easier recognition. However, the approach suffers from the small sample size problem when number of samples is not sufficiently large.[48] Wang *et al.*[45] also proposed a hybrid approach that combines PCA together with symmetrical image correction (SIC) and bit-plane feature fusion (BPFF) algorithms to reduce the inferior influence of recognition when a face image has high contrast of illumination. Hsieh and Tung[9] proposed an integration of PCA with sub-pattern technique and had shown not only good face recognition rate but also the ability of reducing the influence from high contrast of illumination. Unfortunately, its high computation cost is a drawback.

After PCA was proposed, Yang *et al.*[50] proposed the so-called two-dimensional principal component (2DPCA) algorithm aiming for better feature extraction of face images. The 2DPCA has achieved the goal of increasing recognition rate and reducing computation cost simultaneously.[50] Because 2DPCA has such good performance, various face recognition algorithms based on 2DPCA had been proposed and enhanced. For instance, the approach of "two-directional two-dimensional PCA ($(2D)^2$PCA)" proposed by Zhang and Zhoua[54] is to process a face image from transverse and longitudinal axis, respectively, and then perform the recognition by analyzing their shortest dimension. Low computation cost is the advantage of this approach. Unfortunately, its improvement on recognition rate is not ubiquitous in relatively large scale of training samples. Sanguansat *et al.*[34] proposed the approach of "Two-dimensional principal component combined two-dimensional Linear discriminant analysis (2DPCA and 2DLDA)"[34] to face recognition applications. Although this approach solves the small sample size problem, its computation cost is high due to the composition of 2DPCA and 2DLDA. Meng and Zhang[29] proposed the combination of 2DPCA with self-defined volume measure to perform feature extraction by 2DPCA first and then conduct classification by computing the distances of matrix volumes. This approach is more suitable to process applications with high dimensional data. Wang *et al.*[44] proposed "probabilistic two-dimensional principal component analysis" that combines 2DPCA with Gaussian distribution concept to mitigate the noise influence in face image recognition. Kim *et al.*[16] proposed "fusion method based on bidirectional 2DPCA" that reduces dimensions of both row and column vectors before performing face recognition procedure. It does increase recognition rate, but at the expense of high computation cost.[32]

Aforesaid face recognition algorithms have tried to either increasing recognition rate or reducing computation cost, but not both, and all are based on 2DPCA that apparently is well suitable for processing face image data. In this paper, an approach named *sub-image two-dimensional principal component analysis* (SI2DPCA) is proposed hoping to achieve not only reducing the computation but also increasing the recognition rate in face image recognition applications. Unlike conventional 2DPCA, the SI2DPCA divides a whole face image into smaller sub-images to increase the *weight* of features in those sub-images so that features can be better recognized and extracted. At the same time, computation cost can also be reduced due to smaller size of sub-images. For example, eyes, ears, noise, mouth and hair are important features in terms of face recognition and need to be well extracted from a face image data. When a whole image data is processed, its computation is more complex and it is more difficult to well conduct feature extraction because various kinds of features are all included in one image to be extracted. However, if the whole image is divided into smaller sub-images, one sub-image may only include the feature of, say eye, meaning the eye feature is so obvious or outstanding that the feature can be easier to be recognized and therefore can be well recognized and extracted from the sub-image data.

The organization of this paper is as follows: Sec. 2 describes in detail the idea of the proposed SI2DPCA. Its theory is derived and its algorithm is developed in this section. In Sec. 3, various types of experiments are conducted against the ORL image database, and the experimental results and computation costs are analyzed to demonstrate the performance of the SI2DPCA. Finally, conclusions with some discussions are made in Sec. 4.

2. Two-Dimensional Principal Component Analysis

2.1. *Principal component analysis*

For high dimensional data, the task of classification becomes more difficult due to the problems resulting from data noise and high computation cost.[51] To solve this puzzle, it is necessary to reduce data dimension and filter unimportant data. Among many dimension reduction algorithms being proposed so far, the PCA is one of the most popular ones.[17,37,50] The concept of PCA is to get data approximation in lower dimensions from its original data in higher dimensions.[51] Suppose there are n pieces of data in the space, and each data is in v dimensions. The purpose is to find a piece of data $\mathbf{x}_0$ that can mostly approximate (represent) the whole n data. A cost function

J can be set up to indicate the sum of errors:

$$J(x_0) = \sum_{k=1}^{n} \|\mathbf{x}_0 - \mathbf{x}_k\|^2, \tag{1}$$

$$J(x_0) = \sum_{k=1}^{n} \|(\mathbf{x}_0 - \mathbf{m}) - (\mathbf{x}_k - \mathbf{m})\|^2, \tag{2}$$

where

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k. \tag{3}$$

Equation (2) is derived from Eq. (1) and $\mathbf{m}$ is the average value of whole n data. Then Eqs. (4) and (5) can be obtained from Eq. (2) by using the Lagrange multipliers[10,11] that enables the maximum value of an equation to be found when variables are constrained by one or more functions.

$$\mathbf{S}\mathbf{e} = \lambda \mathbf{e}, \tag{4}$$

$$\mathbf{S} = (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t. \tag{5}$$

$\mathbf{S}$ is known as a covariance matrix from Eq. (5). In Eq. (4), $\mathbf{e}$ is a projection vector. The goal is to derive these projection vectors from original data so that the data dimension can be reduced. By applying eigen-decomposition[21] against Eq. (4), the eigenvectors can be obtained and those eigenvectors are the projection vectors from the original data.

In other words, for a set of data, the purpose of PCA is to do feature extraction, reduce data dimension and filter out unimportant data features.[14;27] The first step of PCA is to calculate the covariance matrix of original data and then apply eigen-decomposition against the matrix to get eigenvalues and eigenvectors. Each eigenvalue corresponds to one eigenvector. The eigenvalue with maximum value indicates its corresponding eigenvector has the biggest variance direction. Similarly, the eigenvalue with second-highest value means its corresponding eigenvector has the second biggest variance direction, and so on.

2.2. *Two-dimensional principal component analysis*

The so-called 2DPCA approach by Yang *et al.*[50] in 2004 is proposed particularly for two-dimensional image data. When conventional PCA processes a two-dimensional data, it must transform the data into one-dimensional image data causing high computation cost by performing eigen-decomposition for covariance matrix.[50] Instead of performing dimension reduction, 2DPCA processes against a two-dimensional data directly.

Suppose there is an image data set $Z = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_N\}$ with N images, and the dimension of every image is $n \times n$. The covariance matrix of the image data set is computed by Eq. (6) and the average value of the data set is computed by Eq. (7).

$$\mathbf{R} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{A}_i - \overline{\mathbf{A}})(\mathbf{A}_i - \overline{\mathbf{A}})^T, \tag{6}$$

$$\overline{\mathbf{A}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}_i. \tag{7}$$

In Eq. (6), $\mathbf{A}_i$ is an image in the data set, $\mathbf{R}$ is covariance matrix, and $\overline{\mathbf{A}}$ is data average.

After eigen-decomposition is performed for covariance matrix, k eigenvectors corresponding to the k biggest eigenvalues are selected. These eigenvectors are the projection vectors of the original image data set and the features of the image can therefore be extracted from those projection vectors as shown in Eq. (8).

$$\mathbf{Y}_i = \mathbf{A}\mathbf{X}_i, \quad i = 1, 2, \ldots, k, \tag{8}$$

where $\mathbf{Y}_i$ are projected feature vectors, $\mathbf{X}_i$ means eigenvectors, and there are k biggest eigenvalues being selected. Then a feature vector set $\mathbf{B} = [\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_k]$ in descending order of eigenvalues can be obtained and these projected feature vectors are the resultant principal components of an original image data $\mathbf{A}$ by 2DPCA.

Because 2DPCA processes a two-dimensional face image directly, it can get better result of feature extraction. On the contrary, the conventional PCA needs to transform an image into one-dimensional data and therefore loses some feature information. Consequently, the recognition rate by 2DPCA is better than conventional PCA for two-dimensional face images.

3. Sub-Image Two-Dimensional Principal Component

Compared with PCA, the 2DPCA has better performance on recognition accuracy and computation cost.[50] SI2DPCA is proposed in this paper to further increase the recognition accuracy and decrease the computation cost. As discussed previously, the high computation cost of PCA and 2DPCA comes from computing covariance matrix and eigen-decomposition.[21] Therefore, SI2DPCA proposes to equally divide an image into smaller sub-images to be processed so that the total computation cost can be reduced. The obvious features in a human face usually are eyes, ears, nose, mouth and hairs. If a face image was not divided equally, some of resultant sub-images could be so small that it does not contain any meaningful feature information, such as only skin information. Even if it does, the feature information included in this sub-image could be scanty or incomplete enough to be ignored. For example, a small sub-image may contain only a very small portion of eyes due to the size of the sub-image. Similarly, if a face image is divided into too many sub-images making each of them in very small size, the above-mentioned phenomenon will take place too. In other words, although the smaller size a sub-image is, the lower the total computation cost will be, an original face image can only be divided into appropriate size for better recognition result.

The intuitive way is to equally divide a face image into four smaller sub-images for feature extraction. Suppose there is an $m \times m$ square matrix and the eigen-decomposition is to be performed against it. The eigenvalue λ is obtained by subtracting λ from each of diagonal elements of the square matrix, and then setting the value of the determinant of the square matrix to be zero. The process is described in Eq. (9).

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & \cdots & a_{1m} \\ a_{21} & a_{22}-\lambda & \cdots & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mm}-\lambda \end{vmatrix} = 0. \tag{9}$$

The next step is to apply the extension method[21] to extend the square matrix in Eq. (9). The initial procedure is to choose the first element of every column from the determinant in Eq. (9) and multiply it by the smaller determinant that consists of the elements which belong to neither the column nor the row where the first element is located. The $(-1)^{(i+j)}$ in Eq. (10) is used to get the coefficient sign (+ or −) of every smaller determinant. The symbols of "i" and "j" are the row and column of the first element of a determinant, respectively. For example, "i" is 1 and "j" is 2 for element a_{12}. Equation (10) is the result of extending Eq. (9).

smaller determinant

$$(a_{11}-\lambda)\cdot\begin{vmatrix} a_{22}-\lambda & \cdots & a_{2m} \\ \vdots & \ddots & \vdots \\ a_{m2} & \cdots & a_{mm}-\lambda \end{vmatrix} - a_{12}\cdot\begin{vmatrix} a_{21} & \cdots & a_{2m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm}-\lambda \end{vmatrix} + a_{13} \cdot\begin{vmatrix} a_{21} & \cdots & a_{2m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm}-\lambda \end{vmatrix} - \cdots + (-1)^{(1+m)}\cdot a_{1m}\cdot\begin{vmatrix} a_{21} & \cdots & a_{2(m-1)} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{m(m-1)} \end{vmatrix} = 0. \tag{10}$$

The extension process in Ref. 21 needs to be performed against every smaller determinant in Eq. (10). This procedure of performing extension process continues until no more determinant exists. At this moment, only scalar computation remains in the equation. Based on Eq. (10), it is obvious to observe that the higher the dimension a square matrix has, the higher the computation cost is.

Because the time complexity of computing a determinant is $O(n!)$, the total computation cost of computing the determinants for each of divided smaller square matrices is much less than the computation cost for the originally undivided matrix.

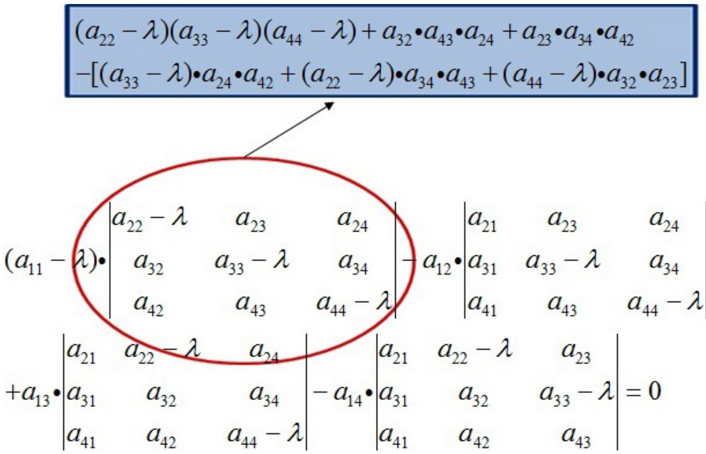

Fig. 1. The process of eigen-decomposition.

For instance, suppose there is a 4×4 matrix to which eigen-decomposition is to be performed. As discussed previously, a value λ is subtracted from diagonal elements of the matrix, and the extension process is repeatedly performed against determinants. The final equation is then set to value zero. The whole process is illustrated in Eq. (11) and Fig. 1.

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22}-\lambda & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33}-\lambda & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44}-\lambda \end{vmatrix} = 0. \tag{11}$$

From the final equation that is set to be zero, four λ values can be obtained. The eigenvectors can consequently be calculated by substituting λ values into Eq. (11). The eigenvector that is based on the largest λ value is the most important feature. The eigenvector based on the second largest λ value is the second important feature, and so on. The illustration can be done by a simple example. Suppose eigen-decomposition is to be performed against below Eq. (12),

$$\begin{vmatrix} a_{11}-\lambda & a_{12} \\ a_{21} & a_{22}-\lambda \end{vmatrix} = 0. \tag{12}$$

Then Eq. (13) is the result of applying extension process to Eq. (12).

$$(a_{11}-\lambda)(a_{22}-\lambda) - a_{12}a_{21} = 0. \tag{13}$$

Two λ values can be obtained from Eq. (13), meaning two eigenvectors can be consequently obtained. For a face image, this means two important features are extracted from the image for recognition purpose. Because current computers are mostly binary-based systems, ill condition problem that gives incorrect result could be caused when computing eigen-decomposition.[21] To avoid such problems, many studies use singular value decomposition (SVD) to replace the process of eigen-decomposition. The computation result of SVD is very close to that of eigen-decomposition, but it can avoid the formation of ill condition problem.[21] For a matrix with dimension $m \times n$, the computation cost of SVD can be described as Eq. (14)[5]

$$4m^2n + 8mn^2 + 9n^3. \tag{14}$$

The big-order of Eq. (14) is $\mathrm{O}(n^3)$ when $m < n$, meaning the dimension variation causes significant difference in terms of computation cost. For example, if a 4×4 matrix is equally divided into four smaller matrices with 2×2 dimension each, then the total computation cost of performing eigen-decomposition against the four smaller matrices is less than that against the original 4×4 matrix. The computation cost is shown in Table 1.

Table 1 indicates that the computation cost can be reduced by half if eigen-decomposition is performed against four 2×2 matrices instead of one 4×4 matrix. This infers that the idea of working on several smaller matrices rather than one original larger matrix by SI2DPCA can lower computation cost when performing eigen-decomposition process. What needs to be explained next for the proposed SI2DPCA is the integrity of getting good feature extraction from separated smaller matrices.

In order to decrease computation cost, the essence of SI2DPCA is to divide an original image into smaller ones to be processed. However, the division process could separate a complete image feature into pieces causing bad performance of feature extraction. Therefore, after eigenvectors are obtained from smaller matrices by SI2DPCA, the smaller matrices need to be integrated together to be processed one more time for correct feature extraction.

The comparison of computation cost between the conventional 2DPCA and the proposed SI2DPCA is shown in Table 2 that details the formulas of computation cost. In Table 2, the formula (a) that comes from Eq. (7) is to compute the data average by 2DPCA. The N means computation summation of N images. The computation amount of $m \times n$ in formula (a) is for matrix addition. Adding one in formula (a) is for the division operation in Eq. (7). Formula (b) is from Eq. (6) to compute the covariance matrix. The N and the plus-one have same meaning as in formula (a). There are three operations in Eq. (6). The first one is subtracting the data average obtained by formula (a) from the original image data, causing $m \times n$ computation. There are two such operations in Eq. (6), so the computation cost is $2 \times m \times n$. The second operation is the multiplication of the two matrices shown in

Table 1. Comparison of computation cost for eigen-decomposition.

	Computation Cost of Eigen-Decomposition: $4m^2n + 8mn^2 + 9n^3$
Four 2×2 matrices	$4 \times (4 \times 2^2 \times 2 + 8 \times 2 \times 2^2 + 9 \times 2^3) = 672$
One 4×4 matrix	$4 \times 4^2 \times 4 + 8 \times 4 \times 4^2 + 9 \times 4^3 = 1344$

Table 2. Analysis of computation cost.

Computation Type	2DPCA	SI2DPCA	
		First Process	Second Process
Data average computation	$N \times (m \times n) + 1$ (a)	$K \times N \times [(m \times n)/k] + k$ (d)	$N \times [m \times s] + 1$ (g)
Covariance computation	$N \times (2 \times m \times n + n^3 + m \times n) + 1$ (b)	$k \times N \times [2 \times (m \times n)/k + (n/k^{1/2})^3 + (m \times n)/k] + k$ (e)	$N \times [2 \times m \times s + s^3 + m \times s] + 1$ (h)
Eigen-decomposition computation	$4 \times m^2 \times n + 8 \times m \times n^2 + 9 \times n^3$ (c)	$k \times (4 \times (m/k^{1/2})^2 \times (n/k^{1/2}) + 8 \times (m/k^{1/2}) \times (n/k^{1/2})^2 + 9 \times (n/k^{1/2})^3)$ (f)	$4 \times m^2 \times s + 8 \times m \times s^2 + 9 \times s^3$ (i)

Eq. (6). The computation cost of such matrix multiplication is n^3. The third operation is the computation of transposing a matrix, causing $m \times n$ computation. Formula (c) comes from Eq. (14) and is the computation cost of eigen-decomposition for 2DPCA.

The rest of formulas in Table 2 are for SI2DPCA. The formulas (d)−(f) are for first process that is taken place to process each of divided smaller matrices. The formulas (g)−(i) are for second process that is for processing the matrix after integration. Suppose an original image with dimension $m \times n$ is equally divided into k smaller images, where k can be square-rooted and m and n are times of $k^{1/2}$. That is, the dimension of each smaller matrix is $((m/k^{1/2}) \times (n/k^{1/2}))$. In first process, Eqs. (7), (6) and (14) need to be applied for each of k smaller matrices in that order. As explained previously for formulas (a)−(c), the computation costs indicated in formulas (d)−(f) are self-explained by reducing dimension to be $((m/k^{1/2}) \times (n/k^{1/2}))$ for each smaller matrix and summing up the whole computation cost of k smaller matrices.

After first process is completed by SI2DPCA, feature extraction result is presented, say h features are extracted meaning only h, instead of original $(n/k^{1/2})$, columns are important and needed for each of k smaller image matrices. Here the value h is usually much smaller than the value of $(n/k^{1/2})$ because only few important features are extracted. As discussed previously, the divided k image matrices need to be combined together in order to obtain complete and correct image features. After putting all these k image matrices together, the dimension of the combined image matrix becomes $m \times s$, where s is h multiplies $k^{1/2}$ representing the number of extracted features of the combined image matrix from smaller image matrices. Similarly, the s value here should be much smaller than n. Then, as discussed before, formulas (g), (h) and (i) can be easily obtained by applying Eqs. (7), (6) and (14) to this resultant matrix with dimension $m \times s$ during the second process of SI2DPCA.

In previous discussions, an original image is assumed in $m \times n$ dimension. In reality, a two-dimensional human face image generally has same dimension on columns and rows, meaning m equals n. Under this assumption, the time complexity in big order for Table 2 can therefore be summarized in Table 3. Although Table 3

Table 3. Time complexity analysis in big order for matrices.

Computation Type	2DPCA	SI2DPCA	
		First Process	Second Process
Data average computation	m^2	m^2	(m/k)
Covariance computation	m^3	$(m/k^{1/2})^3$	s^3
Eigen-decomposition computation	m^3	$(m/k^{1/2})^3$	s^3

Table 4. Time complexity analysis in big order for face images.

Computation Type	2DPCA	SI2DPCA		
		First Process	Second Process	Sum
Data average computation	m^2	m^2	(m/k)	$m^2 + (m/k)$
Covariance computation	m^3	$(m/k^{1/2})^3$	$(m/20)^3$	$(m/k^{1/2})^3 + (m/20)^3$
Eigen-decomposition computation	m^3	$(m/k^{1/2})^3$	$(m/20)^3$	$(m/k^{1/2})^3 + (m/20)^3$

shows SI2DPCA has no advantage over 2DPCA in terms of time complexity, its actual computation cost is less when k is greater than one. The bigger the k is, the greater the decreased amount is for computation cost. In general, the k is at least 4 meaning at least roughly half computation cost is reduced by SI2DPCA.

Generally speaking for face images, since h is the dimension of selected feature vectors in every smaller image, the value of s should be less than the dimension of original image data by about 20 times.[50] Therefore, Table 4 can be obtained after replacing s in Table 3 by the value $m/20$. In Table 4, the column "Sum" is the summation of the column "First Process" and the column "Second Process". From Table 4, it is clear that the computation cost of SI2DPCA is less than that of traditional 2DPCA.

Above discussions prove that SI2DPCA can reduce computation cost. However, the fundamental goal is to perform face image recognition. That is, hoping the proposed SI2DPCA does not improve its computation cost at the expense of recognition performance.

In SI2DPCA, after an original face image is equally divided into several smaller sub-images, each of the sub-images is processed individually for feature extraction. Because the size of a sub-image is smaller, any important features existing in this sub-image can be easier to be found and therefore to be extracted. For example, a sub-image may include only features of eyes and hair, and these features and their detailed textures would then be so obvious to be recognized and extracted in this relatively small image. On the other hand, a whole image includes not only eyes and hair but also many other features. In this situation, the features of eyes and hair may not be so outstanding in such immense image data and therefore cannot be easily recognized. Even these two features have been recognized, their weights may not be

Fig. 2. Original face image (From ORL database).

as great as those extracted from smaller sub-images because of the co-existence of other features in the whole bigger image.

The face image shown in Fig. 2 from ORL database is used to explain above discussions. The dimension of the face image is 112×92. To process the face image by 2DPCA, the covariance matrix is obtained by applying Eq. (6) to the image. The average value $\overline{\mathbf{A}}$ is derived by Eq. (7) and the projected vectors are consequently obtained by Eq. (8). As mentioned before, the dimension of selected feature vectors by 2DPCA is usually much less than the dimension of original data by around 20 times depending on applications. Suppose seven feature vectors $\mathbf{Y}_1 \sim \mathbf{Y}_7$ are selected and are denoted as $\mathbf{B} = [\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3, \mathbf{Y}_4, \mathbf{Y}_5, \mathbf{Y}_6, \mathbf{Y}_7]$ where $\mathbf{B}$ is called feature domain data that can be transformed back to face domain to see how much important information these features have represented for the whole information of the original image. The mechanism of transformation process is shown in Eq. (15).

$$\mathbf{A} = \mathbf{B}\mathbf{X}^{-1}, \tag{15}$$

$$\mathbf{A}\mathbf{X} = \mathbf{B}. \tag{16}$$

Equation (15) is derived from Eq. (16). Equation (16) means that the projected feature vectors $\mathbf{B}$ can be obtained by the original data $\mathbf{A}$ and its projected eigenvectors $\mathbf{X}$. That is, Eq. (15) shows that a face image can be obtained by projected feature vectors $\mathbf{B}$ and the inversed matrix of projected eigenvectors $\mathbf{X}$.

Equation (15) can also be interpreted from the viewpoint of dimension. The image dimension in Fig. 2 is 112×92, and a 92×92 dimensional covariance matrix for the image is computed by Eq. (6). After performing eigen-decomposition against the covariance matrix, seven eigenvectors $\mathbf{X}$ in Eq. (15) corresponding to the seven biggest eigenvalues are selected for the seven most informative features. The dimension of the resultant seven eigenvectors $\mathbf{X}$ is 92×7. From Eq. (8), the dimension of the projected feature vector $\mathbf{Y}$ is 112×1. Because seven features are selected, the

Fig. 3. Face image formed by selected features.

dimension of the seven projected feature vectors is 112×7 forming the matrix $\mathbf{B}$ in Eq. (15). Since matrices $\mathbf{B}$ and $\mathbf{X}$ are known, the face image $\mathbf{A}$ can be obtained by Eq. (15). The resultant image of $\mathbf{A}$ is shown in Fig. 3.

Although Fig. 3 looks more blurred than Fig. 2 because of incomplete information, obvious features such as eyes, ears, mouth and hair are clearly shown. The point is that the dimension of the image in Fig. 3 has been reduced to be 112×7 from 112×92 of the original image in Fig. 2.

Next issue to be discussed would be how small in size a sub-image should be in order to get the best result. As mentioned previously, the horizontal dimension and vertical dimension are usually same for face images. For illustration purpose, a face image in Fig. 2 can be divided into either four or 16 smaller sub-images. Similarly, assume seven features are selected, then after applying Eq. (15), the obtained image formed by the seven selected features are shown in Fig. 4 when dividing into four

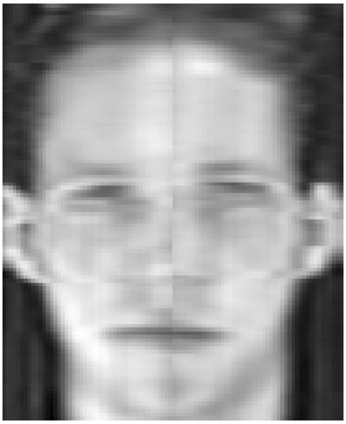

Fig. 4. The image in face domain (division number: 4).

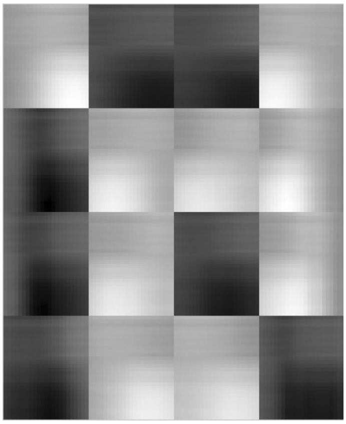

Fig. 5. The image in face domain (division number: 16).

smaller sub-images and in Fig. 5 when dividing into 16 even smaller sub-images. It is obvious that Fig. 5 is a bad one for face recognition because no feature can be clearly seen. The reason is when an original face image is divided into too many small sub-images, the feature information existing in each of them is so little that the feature information cannot be distinguished from other information.

After sub-images are processed and features are extracted, these sub-images have to be combined together to form one image for data integrity. Because only the information, that is actually eigenvectors, related to extracted features are preserved at this moment, some vertical lines can be observed after combining these sub-images together. For example, one vertical line can be seen in the middle of Fig. 4. There is only one vertical line in Fig. 4 because there are only two sub-images being combined along horizontal side. In other words, the line is caused because of discontinuous image data in the seam edge when combining two sub-images. Such phenomenon is called *edge occurrence*. There is no such phenomenon taking place along vertical side because 2DPCA reduces column vectors rather than row vectors. The image data along vertical side is continuous when combining sub-images. The creation of edge occurrence does not affect final recognition performance at all because it is feature domain, not face domain like Fig. 4, to be used for face recognition. The feature domain has preserved important information of extracted features and eliminated noisy characteristics.

Figure 6 is the feature domain formed by the four sub-images divided from the original face image in Fig. 2 that is in 112×92 dimension. That is, each sub-image has dimension of 56×46. Suppose four features are extracted, then the horizontal dimension of the feature domain of every sub-image has been reduced from 46 to 4. After combining the four sub-images together, the integrated feature domain shown in Fig. 6 has dimension of 112×8. In Fig. 6, the lighter area is more informative, while the darker area is less. There are four rectangle bars in Fig. 6 because of four sub-images. In each bar, the color has gradually become darker and darker from left

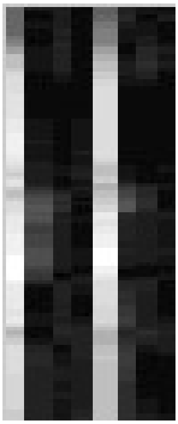

Fig. 6. The feature domain by SI2DPCA.

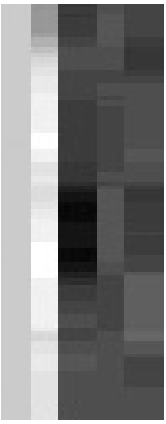

Fig. 7. Feature domain by 2DPCA.

to right. This phenomenon corresponds to the fact that in each image the most important feature is located in the left side while the least important one is located in the right side after feature extraction. Figure 7 shows the feature domain after the original image in Fig. 2 is processed by 2DPCA directly. Same phenomenon can be observed in Fig. 6 as well as in Fig. 7.

Most popular recognition/classification algorithms are based on locations.[12,22,50] For example, the popular nearest neighbor rule (NNR)[50] computes pixel values in same locations of every image. In other words, two images are computed against each other at corresponding locations. That is, operational result is correct as long as location relation of image data is preserved. For example, suppose Fig. 6 is to be compared against other images, the pixel value in location $(1, 1)$ of Fig. 6 is processed against corresponding location $(1, 1)$ of other images. Similarly to the pixel value in location $(1, 2)$, and so on. Because all sub-images are processed by SI2DPCA, the locations of extracted features are arranged by the same way for all sub-images, meaning the location relation of features is preserved among all sub-images. For example, the importance of feature distribution in Fig. 6 is descended from the first column to the 4th column for the first sub-image. Similarly, the importance of feature distribution for the second sub-image is descended from the 5th column to the 8th column. Because such kind of location relation is preserved for every sub-image, the recognition/classification can be performed correctly by SI2DPCA although the sub-images have been combined together for integrity and their row dimension has been reduced.

Above discussions have shown that computation cost can be reduced and at the same time the recognition performance can be improved by SI2DPCA when a face image is properly divided into smaller sub-images to be processed individually. According to the aforesaid analysis, the proper number of dividing a face image into smaller sub-images is four for face recognition. If the division number is higher than four, the recognition performance becomes worse because features have been separated into too small pieces to be recognized in each of sub-images.

4. Experiments and Analysis

4.1. *The ORL database*

The ORL database[41] is a well-known face image database and is used in this paper for experiments. There are 40 individual faces in ORL database. Each individual face has 10 different images making totally 400 face images in the database. The images were taken with a tolerance of some tilting and rotation of the face for up to 20°.[41,50] In ORL database, all images are grayscale with dimension of 112×92. The pixel value range is $0 \sim 255$.

4.2. *Experiments and analysis of SI2DPCA*

In this section, the recognition capabilities of SI2DPCA and 2DPCA are compared. Both SI2DPCA and 2DPCA are feature extraction algorithms, and the classification algorithm is performed by NNR in this experiment.

According to Table 2, the computation cost of SI2DPCA and 2DPCA can be calculated for the images in ORL database. Every image has dimension of 112×92, meaning m and n in Table 2 are 112 and 92, respectively. And each image is divided into four smaller sub-images, meaning the value of k in Table 2 is 4. Suppose 200 images are taken as training data, meaning N in Table 2 is 200, and eight features are selected and extracted. Because the value of s, that is the feature vector dimension in second process, is set to be the number of extracted features multiplies the square root of k, the value of s is therefore $8 \times 2 = 16$. Putting these values into Table 2, the result is shown in Table 5.

Table 5. Analysis of computation cost for ORL database.

Computation Type	2DPCA	SI2DPCA First Process (a)	SI2DPCA Second Process (b)	SI2DPCA (a) + (b)
Data average computation	$200 \times (112 \times 92) + 1 = 2{,}060{,}801$	$4 \times 200 \times (56 \times 46) + 4 = 2{,}060{,}804$	$200 \times (112 \times 16) + 1 = 358{,}401$	2,419,205
Covariance matrix computation	$200 \times (2 \times 112 \times 92 + 92^3 + 112 \times 92) + 1 = 161{,}920{,}001$	$4 \times 200 \times (2 \times 56 \times 46 + 46^3 + 56 \times 46) + 4 = 84{,}051{,}204$	$200 \times (2 \times 112 \times 16 + 16^3 + 112 \times 16) + 1 = 1{,}894{,}401$	85,945,605
Eigen-decomposition computation	$4 \times 112^2 \times 92 + 8 \times 112 \times 92^2 + 9 \times 92^3 = 19{,}208{,}128$	$4 \times (4 \times 56^2 \times 46 + 8 \times 56 \times 46^2 + 9 \times 46^3) = 9{,}604{,}064$	$4 \times 112^2 \times 16 + 8 \times 112 \times 16^2 + 9 \times 16^3 = 1{,}069{,}056$	10,673,120
Sum of computation cost	183,188,930	95,716,072	3,321,858	99,037,930

Table 5 shows that the computation cost for SI2DPCA is only half of 2DPCA. When calculating covariance matrix and eigen-decomposition, there are many quadratic or cubic power computations. Smaller image dimensions operated in SI2DPCA can greatly reduce the computation cost, as discussed previously.

Besides the computation cost, the proposed SI2DPCA and conventional 2DPCA also need to be compared on their recognition performance. Both approaches are feature extraction algorithms. The recognition is performed by the nearest neighbor rule (NNR)[50] that is based on Euclidean distance shown in Eq. (17).

$$d = \|\mathbf{V} - \mathbf{P}\|_2. \tag{17}$$

The symbols $\mathbf{V}$ and $\mathbf{P}$ are two mean vectors, and d is Euclidean distance. Equation (17) computes the norm of **V-P**. Suppose $\mathbf{V} = (v_1, v_2, v_3)$ and $\mathbf{P} = (p_1, p_2, p_3)$, the operation of norm is shown in Eq. (18).

$$\|\mathbf{V} - \mathbf{P}\|_2 = \sqrt{(v_1 - p_1)^2 + (v_2 - p_2)^2 + (v_3 - p_3)^2}. \tag{18}$$

Suppose there are N images, represented as $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_N$, and each image is represented by a projected feature vector, such as $[\mathbf{Y}_1^1, \mathbf{Y}_2^1, \ldots, \mathbf{Y}_d^1]$ for $\mathbf{B}_1$ with $m \times d$ dimension. The classes of these N images are already known. Suppose a class-unknown data $\mathbf{T}_k = [\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_d]$ is to be recognized against these N images. The computation process is shown in Eq. (19).

$$d(\mathbf{B}, \mathbf{T}) = \sum_{k=1}^{d} \|\mathbf{B}_k - \mathbf{T}_k\|_2. \tag{19}$$

The class of $\mathbf{T}_k$ is classified as the class of $\mathbf{B}_k$ if these two have minimum distance d in Eq. (19).

In this experiment, the first five images of every face are treated as training and the remaining five images of every face are treated as testing images. That is, there are 200 images for training and 200 images for testing. Seven important features are extracted in the experiment, meaning a seven-elements feature vector is obtained for each of images. The projected feature vector of each of training and testing images can be calculated by multiplying the seven-elements feature vectors to the data of every training and testing images. The classification for each of testing images can then be performed by NNR against the training images.

The procedure of the experiment is conducted as shown in Fig. 8. In the beginning, the images in ORL database are divided into training and testing images. The important features are then extracted to get feature vector from training images and consequently projected feature vector can be obtained for each of the training and testing images. Finally, the classification result can be performed by NNR based on these projected feature vectors. The procedures of SI2DPCA and 2DPCA are executed respectively for performance comparison.

Figure 9 shows how the classification is performed in this experiment. Since 200 images are used as training and 200 images are used as testing data, there are 200 projected feature vectors for training and 200 projected feature vectors for testing

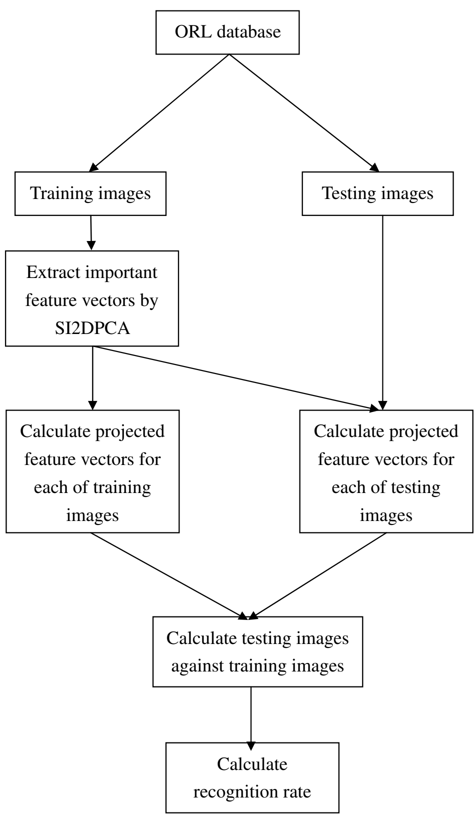

Fig. 8. Flow chart of experimental procedure.

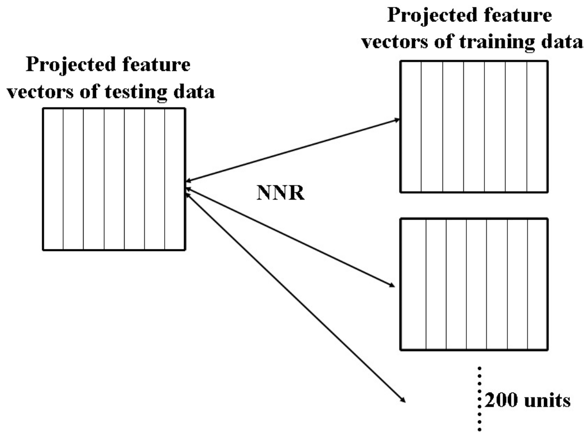

Fig. 9. Operation of classification.

images to be calculated for classification. The first projected feature vector of testing image is classified against the 200 projected feature vectors of training images based on the computational result of NNR, resulting in 200 values. The class of the training image is said to be the class of the testing image that corresponds to the minimum value among the 200 NNR results. At this moment, the classification of the first image can be either correct or wrong. Similar procedure is performed for the second, third, etc. until the 200th testing images, resulting in totally 200 values of either correct or wrong classification. The recognition rate can therefore be calculated by dividing the number of correct classifications by 200.

The recognition rate comparison between 2DPCA and SI2DPCA is shown in Table 6. Earlier discussions argued that important features can be better recognized and extracted in smaller sub-images. This can be observed in Table 6 that shows slight better recognition rate for SI2DPCA over conventional 2DPCA. Both Tables 5 and 6 together show that the SI2DPCA reduces computation cost without compromising its recognition performance.

As mentioned previously, there are two processes being performed by SI2DPCA. The first process is to calculate the projected feature vectors for each of sub-images, and the second process is for image integration. Table 7 compares the recognition performance between the one with only the first process and another one with both first and second processes by SI2DPCA. The result in Table 7 confirms previous

Table 6. Recognition comparison between 2DPCA and SI2DPCA.

Strategy	Recognition Rate (%)
2DPCA	93
SI2DPCA	93.5

Table 7. Comparison between one- and two-process by SI2DPCA.

Number of SI2DPCA Processes	Recognition Rate (%)
One	92
Two	93.5

Table 8. Comparison among other methods and SI2DPCA.

Method Number	Method	Recognition Rate (%)	Computation Cost
1	$(2D)^2$PCA[54]	90.5	high
2	2DPCA + Fusion method based on bidirectional[16]	92.5	high
3	2DPCA + 2DLDA[34]	93.5	high
4	**SI2DPCA (proposed)**	**93.5**	**low (lower than 2DPCA)**
5	2DPCA + Kernel[39]	94.58	very high
6	2DPCA + Feature fusion approach[47]	98.1	very very high

statements that the sub-images are better to be combined together for image integrity and consequently for better recognition after obtaining projected feature vectors.

Various methodologies based on 2DPCA have been proposed. Table 8 shows the performance comparison in terms of recognition rate and computation cost among some of better-known approaches and SI2DPCA. All the experiments for the approaches in Table 8 are conducted based on the face images in ORL database. In Table 8, the computation costs of method 1, method 2 and method 3 are all higher than SI2DPCA while the recognition rates are either lower than or same as SI2DPCA. This is because SI2DPCA operates against matrices in smaller dimensions. For methods 4, 5 and 6 in Table 8, they even put additional processes to 2DPCA. Method 5 combines 2DPCA with Kernel algorithm. This approach projects image data to high dimensional space, causing high computation cost. Although its recognition rate is slightly better than the proposed SI2DPCA, the much higher computation cost makes it difficult for practical applications. Method 6 combines feature fusion with 2DPCA in order to increase recognition rate. The resultant recognition rate is very good at 98.1% that is better than the rate of 93.5% by SI2DPCA in the experiment. Unfortunately, the computation cost of this approach is so high, at least 10 times higher than 2DPCA, that it is impossible to be applied to any practical applications.

5. Conclusion

The feature extraction algorithm 2DPCA is specially developed for face recognition. Its characteristics are low computation cost and good feature extraction, making

2DPCA a popular approach for face recognition. In this paper, an enhanced approach "SI2DPCA" is proposed to operate at even lower computation cost without compromising its good recognition performance. Both of the two goals of reducing computation cost and maintaining good recognition rate have been shown in the results of the conducted experiments in this paper. This paper not only discusses in details on how the proposed SI2DPCA works but also points out that the number of smaller sub-images divided from an original image is critical to the success of face recognition by the approach. For face images in ORL database, the appropriate number is four. Since most face images have same horizontal and vertical dimensions to cover entire face information, dividing original images into four smaller sub-images should be a good way in reality. Nevertheless, even dividing into four sub-images may not reach optimal performance in some other applications, the SI2DPCA should perform as better result as shown in the experiments in this paper.

References

1. H. Chen, Y. Y. Tang, B. Fang and J. Wen, Illumination invariant face recognition using FABEMD decomposition with detail measure weight, *Int. J. Pattern Recogn. Artif. Intell.* **25** (2011) 1261–1273.
2. X. Chen and J. Zhang, Maximum variance difference based embedding approach for facial feature extraction, *Int. J. Pattern Recogn. Artif. Intell.* **24**(7) (2010) 1047–1060.
3. G. Chiachia, A. N. Marana, T. Ruf and A. C. Ernst, Histograms: A simple feature extraction and matching approach for face recognition, *Int. J. Pattern Recogn. Artif. Intell.* **25** (2011) 1337–1348.
4. J. Y. Choi, Y. M. Ro and K. N. Plataniotis, Color local texture features for color face recognition, *IEEE Trans. Image Process.* **21**(3) 1366–1380.
5. M. Groher, Singular Value Decomposition, *3D Computer Vision - WT 2005*, 2005, Nassir Navab, available at http://campar.in.tum.de/twiki/pub/Chair/TeachingWs05ComputerVision/svd.pdf.
6. M. J. Er, S. Wu, J. Lu and H. L. Toh, Face recognition with radial basis function RBF neural networks, *IEEE Trans. Neural Netw.* **13**(3) (2002) 697–710.
7. B. K. Gunturk, A. U. Batur and Y. Altunbasak, Eigenface-domain super-resolution for face recognition, *IEEE Trans. Image Process.* **12**(5) (2003) 597–606.
8. G. He, Y. Tang, B. Fang and P. S. P. Wang, Bionic face recognition using Gabor transformation, *Int. J. Pattern Recogn. Artif. Intell.* **25** (2011) 391–402.
9. P. C. Hsieh and P. C. Tung, A novel hybrid approach based on sub-pattern technique and whitened PCA for face recognition, *Pattern Recogn. Lett.* **42** (2009) 978–984.
10. H. C. Zhang, Lagrange Multiplier Method, available at http://episte.math.ntu.edu.tw/entries/en_lagrange_mul/index.html.
11. Wikipedia, Lagrange multiplier, available at http://en.wikipedia.org/wiki/Lagrange_multiplier.
12. G. Huang, Fusion $(2D)^2$PCALDA: A new method for face recognition, *Appl. Math. Comput.* **216** (2010) 3195–3199.
13. H. Huang, J. Liu and H. Feng, Uncorrelated local Fisher discriminant analysis for face recognition, *Int. J. Pattern Recogn. Artif. Intell.* **25** (2011) 863–887.
14. J. S. R. Jang, Data clustering and pattern recognition, available at http://www.cs.nthu.edu.tw/∼jang (in Chinese).

15. K. I. Kim, K. Jung and H. J. Kim, Face recognition using kernel principal component analysis, *IEEE Signal Process. Lett.* **9**(2) (2002) 40–42.
16. Y. G. Kim, Y. J. Song, U. D. Chang, D. W. Kim, T. S. Yun and J. H. Ahn, Face recognition using a fusion method based on bidirectional 2DPCA, *Appl. Math. Comput.* **205** (2008) 601–607.
17. M. Kirby and L. Sirovich, Application of the KL procedure for the characterization of human faces, *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(1) (1990) 103–108.
18. A. Z. Kouzani, F. He and K. Sammut, Towards invariant face recognition, *Inf. Sci.* **123** (2000) 75–101.
19. K. Lam and H. Yan, An analytic-to-holistic approach for face recognition based on a single frontal view, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(7) (1998) 673–686.
20. S. W. Lee, P. S. P. Wang, S. N. Yanushkevich and S. W. Lee, Noniterative 3D face reconstruction based on photometric stereo, *Int. J. Pattern Recogn. Artif. Intell.* **22**(3) (2008) 389–410.
21. S. J. Leon, *Linear Algebra with Applications*, 7th edn. (Prentice Hall, 2006).
22. M. Li and B. Yuan, 2D-LDA: A statistical linear discriminant analysis for image matrix, *Pattern Recogn. Lett.* **26** (2005) 527–532.
23. Q. Liu, X. Tang, H. Lu and S. Ma, Face recognition using kernel scatter-difference-based discriminant analysis, *IEEE Trans. Neural Netw.* **17**(4) (2006) 1081–1085.
24. J. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms, *IEEE Trans. Neural Netw.* **14**(1) (2003) 117–126.
25. J. Lu, X. Yuan and T. Yahagi, A method of face recognition based on fuzzy c-means clustering and associated sub-NNs, *IEEE Trans. Neural Netw.* **18**(1) (2007) 150–160.
26. Y. Luo, M. L. Gavrilova and P. S. P. Wang, Facial metamorphosis using geometrical methods for biometric applications, *Int. J. Pattern Recogn. Artif. Intell.* **22** (2008) 555–584.
27. A. M. Martinez and A. C. Kak, PCA versus LDA, *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(2) (2001) 228–233.
28. P. Melin, O. Mendoza and O. Castillo, Face recognition with an improved interval type-2 fuzzy logic Sugeno integral and modular neural networks, *IEEE Trans. Syst. Man Cybern. A Syst. Humans* **41**(5) (2011) 1001–1012.
29. J. Meng and W. Zhang, Volume measure in 2DPCA-based face recognition, *Pattern Recogn. Lett.* **28** (2007) 1203–1208.
30. Pattern recognition and artificial intelligence in biometrics-editorial, **22**(3) (2008), available at http://www.worldscinet.com/ijprai/22/2203/S02180014082203.html.
31. A. Pujol, J. Vitria, F. Lumbreras and J. J. Villanueva, Topological principal component analysis for face encoding and recognition, *Pattern Recogn. Lett.* **22** (2001) 769–776.
32. Y. Qi and J. Zhang, $(2D)^2$PCALDA: An efficient approach for face recognition, *Appl. Math. Comput.* **213**(1) (2009) 1–7.
33. K. Ruba Soundar and K. Murugesan, An adaptive face recognition in combined global and local preserving feature space, *Int. J. Pattern Recogn. Artif. Intell.* **25** (2011) 99–115.
34. P. Sanguansat, W. Asdornwised, S. Jitapunkul and S. Marukatat, Two-dimensional linear discriminant analysis of principle component vectors for face recognition, *ICASSP 2006*, May 2006, pp. 345–348.
35. F. Y. Shih, S. Cheng, C. F. Chuang and P. S. P. Wang, Extracting faces and facial features from color images, *Int. J. Pattern Recogn. Artif. Intell.* **22**(3) (2008) 515–534.
36. F. Y. Shih, C. F. Chuang and P. S. P. Wang, Performance comparisons of facial expression recognition in Jaffe database, *Int. J. Pattern Recogn. Artif. Intell.* **22**(3) (2008) 445–459.
37. L. Sirovich and M. Kirby, Low-dimensional procedure for characterization of human faces, *J. Optical Soc. Am.* **4** (1987) 519–524.

38. N. Sudha, A. R. Mohan and P. K. Meher, A self-configurable systolic architecture for face recognition system based on principal component neural network, *IEEE Trans. Circuits Syst. Video Technol.* **21**(8) (2011) 1071–1084.
39. N. Sun, H. X. Wang, Z. H. Ji, C. R. Zou and L. Zhao, An efficient algorithm for kernel two-dimensional principal component analysis, *Neural Comput. Appl.* **17** (2008) 59–64.
40. X. Tan, S. Chen, Z. H. Zhou and F. Zhang, Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble, *IEEE Trans. Neural Netw.* **16**(4) (2005) 875–886.
41. The ORL face database, available at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
42. M. A. Turk and A. P. Pentland, Eigenfaces for recognition, *J. Cognit. Neurosci.* **3** (1991) 71–86.
43. N. S. Vu and A. Caplier, Enhanced patterns of oriented edge magnitudes for face recognition and image matching, *IEEE Trans. Image Process.* **21**(3) (2012) 1352–1365.
44. H. Wang, S. Chen, Z. Hu and B. Luo, Probabilistic two-dimensional principal component analysis and its mixture model for face recognition, *Springer Neural Comput. Appl.* **17** (2008) 541–547.
45. H. Wang, Y. Leng, Z. Wang and X. Wu, Application of image correction and bit-plane fusion in generalized PCA based face recognition, *Pattern Recogn. Lett.* **28** (2007) 2352–2358.
46. H. Wang, Z. Wang, Y. Leng, X. Wu and Q. Li, PCA plus F-LDA: A new approach to face recognition, *Int. J. Pattern Recogn. Artif. Intell.* **21**(6) (2007) 1059–1068.
47. Y. Xu, D. Zhang, J. Yang and J. Y. Yang, An approach for directly extracting features from matrix data and its application in face recognition, *Neurocomputing* **71** (2008) 1857–1865.
48. W. H. Yang and D. Q. Dai, Two-dimensional maximum margin feature extraction for face recognition, *IEEE Trans. Syst. Man Cybern. B, Cybern.* **39**(4) (2009) 1002–1012.
49. F. Yang and M. Paindavoine, Implementation of an RBF neural network on embedded systems: Real-time face tracking and identity verification, *IEEE Trans. Neural Netw.* **14**(5) (2003) 1162–1175.
50. J. Yang, D. Zhang, A. F. Frangi and J. Y. Yang, Two-dimensional PCA: A new approach to appearance-based face representation and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(1) (2004) 131–137.
51. K. L. Yeh, Principal component analysis with missing data, June 2006, pp. 1–12, available at http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/PCAMissingData.pdf (accessed 15 September 2011).
52. H. Yu, J. J. Zhang and X. Yang, Tensor-based feature representation with application to multimodal face recognition, *Int. J. Pattern Recogn. Artif. Intell.* **25** (2011) 1197–1217.
53. B. L. Zhang, H. Zhang and S. S. Ge, Face recognition by applying wavelet subband representation and kernel associative memory, *IEEE Trans. Neural Netw.* **15**(1) (2004) 166–177.
54. D. Zhang and Z. H. Zhoua, $(2D)^2$PCA: Two-directional two-dimensional PCA for efficient face representation and recognition, *Neurocomputing* **69** (2005) 224–231.
55. W. Zheng, X. Zhou, C. Zou and L. Zhao, Facial expression recognition using kernel canonical correlation analysis (KCCA), *IEEE Trans. Neural Netw.* **17**(1) (2006) 233–238.
56. W. W. W. Zou and P. C. Yuen, Very low resolution face recognition problem, *IEEE Trans. Image Process.* **21**(1) (2012) 327–340.

Wei-Li Fang was born in Taiwan, in 1981. He received his B.S. degree from National Changhua University of Education, Changhua, Taiwan, in 2004 and his M.S. degree from the National Taiwan Normal University, Taipei, Taiwan, in 2007. He is currently a Ph.D. candidate in the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include pattern recognition, image processing and artificial intelligence.

Ying-Kuei Yang received his B.S. degree from National Chao-Tung University in Taiwan in 1977, his M.S. degree from North Dakota State University in 1980 and his M.S. and Ph.D. degrees in Electrical Engineering from Purdue University in 1984 and 1986, respectively.

He worked at CyCare System Inc. in Dubuque, Iowa from 1980 to 1982. He had been a Principal Engineer at Harris Corp. in Melbourne, Florida from 1986 to 1990. Currently, he is an Associate Professor at the Electrical Engineering Department of the National Taiwan University of Science and Technology. His research areas include artificial intelligence, fuzzy theory and its applications, pattern recognition, and database systems.

Jung-Kuei Pan received his M.S. degree from the Electrical Engineering Department of National Taiwan University of Science and Technology in 2007. He is currently a Ph.D. candidate at the National Taiwan University of Science and Technology. His research interests include image and video processing, computer vision, speech recognition, signal and embedded systems.