Symmetrical Two-Dimensional PCA with Image Measures in Face Recognition

Regular Paper

Jicheng Meng1,∗, and Yuming Yang2

1 School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, P.R. China
2 School of Applied Mathematics, University of Electronic Science and Technology of China, Chengdu, P.R. China

* Corresponding author E-mail: jcmeng@uestc.edu.cn

Received 14 May 2012; Accepted 2 Oct 2012

© 2012 Meng and Yang; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract In this paper, we extensively investigate symmetrical two-dimensional principal component analysis (S2DPCA) and introduce two image measures for S2DPCA-based face recognition, volume measure (VM) and subspace distance measure (SM). Although symmetrical features are an obviously but not absolutely facial characteristic, they have been successfully applied to PCA and 2DPCA. The paper gives detailed evidence that even and odd subspaces in S2DPCA are mutually orthogonal, and particularly that S2DPCA can be constructed using a quarter of the conventional S2DPCA even/odd covariance matrix. Based on these theories, we investigate the time and memory complexities of S2PDCA further, and find that S2DPCA can in fact be computed using a quarter of the time and memory compared to conventional S2DPCA. Finally, VM and SM are introduced to S2DPCA for final classification. Our experiments compare S2DPCA with 2DPCA on YALE, AR and FERET face databases, and the results indicate that S2DPCA+VM generally outperforms other algorithms.

Keywords Symmetrical Two-Dimensional PCA, Image Measures, Volume Measure, Subspace Distance Measure, Face Recognition

1. Introduction

Face recognition is now becoming a very popular subject for research. It can be applied in many fields[1], including mobile robots, information security, and entertainment. Over the past two decades, many face recognition approaches have been proposed. Typical approaches include principal component analysis (PCA), linear discriminant analysis (LDA), independent component analysis (ICA), neural network (NN), support vector machine (SVM), and kernel PCA/LDA [1-3]. On the whole, these approaches are based on appearance image and have some statistical significance. Recently, 3D-based face recognition has also been a hot research area[4].

There are many approaches, including those cited above, to improve recognition performance from a statistical point of view whilst combating the “dimensionality curse” problem [5]. However, these approaches barely consider characters of image itself. Local binary patterns (LBP), elastic graph matching (EGM), and so on [6] consider the topologic or local characters of images, and have also been applied in face recognition.
Appearance-based statistical face recognition algorithms usually use vector-based face samples obtained from two-dimensional (2D) face images by techniques such as concatenation [7]. Vector-based face samples usually have a large number of dimensions. However, the number of the samples is small, so the “dimensionality curse” problem is always bred. 2DPCA, which is founded on a 2D face data format rather than 1D vector-based face images, shows high performance in dealing with the “dimensionality curse” problem. Why does the 2DPCA work? The row or column of the 2D face image has a low dimensionality, so the covariance matrix in 2DPCA is more precise than that in PCA. This is the key to solving the impact of the “dimensionality curse” problem in 2DPCA-based face recognition [4,8]. (The covariance matrix in face recognition based on PCA is not so precise, because the sample is of high dimensionality while the number of samples is small.) Other algorithms based on 2D face data and which use 2DPCA are also proposed, e.g., two-dimensional LDA (2DLDA) [9], two-dimensional Laplacianfaces [10], and diagonal PCA [11]. In [12], Gao et al. proved that 2DPCA is not equivalent to a special case of block-based PCA, where each block is a column (or row) of the image. Moreover, from the view of tensor analysis, 2DPCA or 2DLDA can be considered as the variant of the second rank tensor decomposition [13,14]. The typical classification measure used in 2DPCA is the sum of the Euclidean distance between two feature vectors in a feature matrix [7] and is called distance measure (DM) or Yang distance. Zuo et al. furthermore proposed a generalized Yang distance [15]. Volume measure (VM) [16] based on the volume of matrix has also been proposed for 2DPCA in face recognition. Recently, another rule, subspace distance measure (SM) [17,18], has also been proposed to measure the similarity between two images.

In fact, 2DPCA(2DLDA)-based approaches do not acknowledge the characters of the face image itself, and solve the impact of the “dimensionality curse” problem based on pure statistics. Symmetrical principal component analysis (SPCA) [19], an approach obtained by the even-odd decomposition principle, expands samples from the facial symmetry [19,20]. In essence, this approach improves recognition performance by exploiting prior knowledge, i.e., the symmetrical characters of the face image. Good performance has also been obtained by SKPCA [21]. However, these approaches are all based on 1D vectorized face images. Recently, a new facial symmetry approach called symmetry two-dimensional PCA (S2DPCA), has been proposed [22]. S2DPCA utilizes even and odd decompositions based on the 2D original and mirror face samples. On the one hand, S2DPCA can exploit more prior knowledge than SPCA, producing more new samples. It can also obtain a more precise covariance matrix, just like 2DPCA, using more samples than PCA or SPCA.

Although S2DPCA has been proposed in [22], the theories of 2DPCA, e.g., even/odd subspace orthogonality and computational cost, remain open, and the classifier was unique. Motivated by the conclusions on SPCA [5,19], we give more proofs of the corresponding theories on S2DPCA here. We offer further proof and clarify the point that even and odd subspaces in S2DPCA are mutually orthogonal [23] and in particular give a theoretical foundation to the observation that the even or odd subspace can be founded on a quarter of the conventional S2DPCA’s even/odd covariance matrix. The time and memory space complexities of 2DPCA, conventional S2DPCA, and our S2DPCA are also compared. S2DPCA based on our theories requires the least time and memory space among these algorithms, with a quarter of the time and memory costs compared to conventional S2DPCA.

VM, mentioned above, is directly based on matrix data, i.e., a 2D image format [16]; it can therefore be regarded as a kind of image measure like SM. From the point of view of data format, image measures fit better for 2DPCA or S2DPCA-based face recognition, as compared to DM. We therefore investigate 2DPCA and S2DPCA with DM, VM and SM separately through experiments using a variety of face databases, and find that S2DPCA with VM generally gives better performance than others.

The remainder of this paper is organized in the following way. Ideas on S2DPCA and classification measures are developed in Section 2. Section 3 shows our experimental results. The last section concludes this paper and discusses future work.

2. Symmetrical two-dimensional PCA

Compared with PCA, 2DPCA gives better performance by computing the covariance matrix more accurately. SPCA exploits the original face training samples and the mirror samples, as well as even and odd decompositions, thus making use of more discriminative information [21]. S2DPCA, on the other hand, can use a pure statistical mathematical technique, as well as the characters of face image, to improve the recognition performance.

2.1 Principles

S2DPCA combines the idea of SPCA with that of 2DPCA. Suppose the set \( \{X_{m_1}, X_{m_2}, \ldots, X_{m_N}\} \) is the mirror set of the original training sample set \( \{X_1, X_2, \ldots, X_N\} \). Then, \( X_{m_i} \) \((i = 1, 2, \ldots, N) \) can be obtained by the following equation:

\[
X_{m_i} = X_i M
\]

(1)

Here, \( M \) denotes the mirror transform matrix, which is a unit sub-diagonal matrix; i.e., the sub-diagonal entries of \( M \) are all 1, and the rest are all 0. If set \( X_i \) is a \( h \times w \)
matrix as mentioned above, then \( M \) is a \( w \times w \) matrix.
The even and odd samples can therefore be computed separately by the following equations:

\[
X_{ei} = (X_i + X_{mi}) / 2 = X_i (I + M) / 2 \tag{2}
\]

\[
X_{oi} = (X_i - X_{mi}) / 2 = X_i (I - M) / 2 \tag{3}
\]

\[
C = E[(X - E(X))^T (X - E(X))] = \frac{1}{N} \left\{ \sum_{i=1}^{N} (X_i - \bar{X})^T (X_i - \bar{X}) \right\} \tag{4}
\]

In equations (2) and (3), \( I \) denotes the \( w \times w \) unit matrix.
As the equation (4) defined, \( C \) denotes the covariance matrix of the original training samples. In addition, we set \( C_e \) and \( C_o \) respectively to be the covariance matrix of even, odd and mirror training samples. We apply PCA to \( C_e \) and \( C_o \), respectively, and then choose \( U_e \) and \( U_o \) as the feature matrices. The selection strategy is to order even and odd eigenvectors according to their eigenvalues, and then select eigenvectors corresponding to larger eigenvalues [22].

Then, the low dimensional even and odd features (coefficients) can be extracted respectively by projecting the original sample to even/odd subspaces. This process is called conventional S2DPCA [22]. To further improve conventional S2DPCA, we have proved several theorems, outlined below.

2.2 Theorems on S2DPCA

In fact, if we set each even and odd training sample as \( X_e := X_o - \bar{X}_e \) and \( X_o := X_o - \bar{X}_o \) (\( \bar{X}_e \) and \( \bar{X}_o \) denote the even and odd centroid, respectively), the mean sample matrix of the new training sample set is 0 mean matrix. So, without loss of generality, we suppose the mean sample matrix of the even and odd training sample matrix is a 0 mean matrix. Then, the mean sample matrix of all training sample matrices is also a 0 mean matrix. This hypothesis can simplify the deduction of theorems.

Theorem 1: Set \( C^o = C_e + C_o \). Then the following equation holds: \( C^o = C + C_m / 2 \)

Proof: Since the mean sample matrices of even, odd and all samples are all 0 mean matrices, from the definition of \( C_e \) and \( C_o \) we can get:

\[
C_e = \frac{1}{N} \sum_{i=1}^{N} X_{ei} X_{ei} = \frac{1}{N} \sum_{i=1}^{N} (I + M)^T X_i^T X_i (I + M) / 4
\]

\[
C_o = \frac{1}{N} \sum_{i=1}^{N} X_{oi} X_{oi} = \frac{1}{N} \sum_{i=1}^{N} (I - M)^T X_i^T X_i (I - M) / 4
\]

\[
\Rightarrow C^o = C_e + C_o
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} ((I + M)^T X_i^T X_i (I + M)) / 4 + \frac{1}{N} \sum_{i=1}^{N} ((I - M)^T X_i^T X_i (I - M)) / 4
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} (X_i^T X_i) / 2 + \frac{1}{N} \sum_{i=1}^{N} (M^T X_i X_i M) / 2
\]

\[
= (C + C_m) / 2
\]

Theorem 2: Let \( U^o \) and \( \Lambda^o \) denote the eigenvector matrix and the non-zero eigenvalue matrix of the covariance matrix \( C^o \), respectively. \( \Lambda_e \) denotes the non-zero eigenvalue matrix corresponding to the eigenvector matrix \( U_e \) of the covariance matrix \( C_e \). Similarly, \( \Lambda_o \) denotes the non-zero eigenvalue matrix corresponding to the eigenvector matrix \( U_o \) of the covariance matrix \( C_o \). The following equations then hold:

1. \( U_e^T U_o = 0 \); 
2. \( U^o = [U_e U_o], \Lambda^o = [\Lambda_e 0] \) .

Proofs: 1. Set \( U_e = [u_{e1} \dots u_{em}] \), \( U_o = [u_{o1} \dots u_{om}] \), then, to prove the equation \( U_e^T U_o = 0 \) holds, \( u_{ei}, u_{oj} \geq 0 \), \( i, j \in [1, 2, \dots, m] \).

Since \( U_e \) and \( U_o \) denote \( C_e \) and \( C_o \) ’s eigenvector matrix, while \( C_e \) and \( C_o \) denote the even samples and odd samples’ covariance matrix separately, and the eigenvector can be linearly approximated by samples [24], we have \( u_{ei} = \sum \alpha_k x_{k} \), \( u_{oj} = \sum \alpha_l x_{l} \). \( x_{k} \) and \( x_{l} \) denote the \( k \)th row of even samples and the \( l \)th row of odd samples; \( \alpha_k \) and \( \alpha_l \) are the corresponding coefficients. \( x_{k} \) is a mirrored symmetric vector and \( x_{l} \) is a mirrored anti-symmetric vector, \( u_{e} \) is a mirrored symmetric vector and \( u_{o} \) is a mirrored anti-symmetric vector. They can be written as \( u_{e} = [u_{e1} \dots u_{em}] \) and \( u_{o} = [-u_{e1} \dots -u_{em}] \) respectively; here, \( u_{e1} \) and \( u_{o1} \) are mirrored vectors of \( u_{e1} \) and \( u_{o1} \), i.e., if \( u_{e1} = [u_{e1}^1, u_{e1}^2, \dots, u_{em}^1] \), \( u_{o1} = [u_{o1}^1, u_{o1}^2, \dots, u_{em}^1] \), then \( u_{e1} = [u_{o1}^1, u_{o1}^2, \dots, u_{em}^1] \), \( u_{o1} = [-u_{em}^1, \dots, -u_{o1}^1] \). So, from the above, we can say \( u_{ei}, u_{oj} \geq 0 \).

Therefore, \( U_e^T U_o = 0 \).

2. From conclusion 1, we have \( U_e^T U_o = U_e^T U_o \Lambda_o = 0 \), so

\[
U_e^T C_o U_e = 0, U_e^T C_o U_o = 0, U_o^T C_e U_o = 0,
\]

\[
U_o^T C_e U_e = 0, U_o^T C_o U_o = 0, U_e^T C_e U_o = 0.
\]
\[
\Rightarrow \Lambda^o = \begin{bmatrix}
U_e^T C^o [U_e & U_o]
\end{bmatrix}
\]
\[
= \begin{bmatrix}
U_e^T (C_e + C_o) [U_e & U_o]
\end{bmatrix}
\]
\[
= \begin{bmatrix}
U_e^T C_e U_e & U_e^T C_o U_o
U_e^T C_o U_e & U_e^T C_o U_o
\end{bmatrix}
= \begin{bmatrix}
\Lambda_e & 0
0 & \Lambda_o
\end{bmatrix}
\]
\[
= \begin{bmatrix}
\Lambda_e & 0
0 & \Lambda_o
\end{bmatrix}
\]
That is to say, \([U_e & U_o]\) is the eigenvector matrix of the covariance matrix \(C^o\). Hence,
\[
U^o = [U_e & U_o].
\]

Theorem 3: The covariance matrices \(C_e\) and \(C_o\) are defined as \(\omega \times \omega\) matrices. Suppose \(\omega\) is even, then \(C_e\) and \(C_o\) can be written as \[
\begin{bmatrix}
B & BM
MB & MBM
\end{bmatrix}
\]
and \[
\begin{bmatrix}
D & DM
MD & MDM
\end{bmatrix}
\]
respectively. \(M\) denotes the mirror transform matrix as defined above. Furthermore, \(U_e\) and \(U_o\) can be obtained just by PCA on the matrix \(B\) and \(D\), i.e., the quarter of the matrix \(C_e\) and \(C_o\), respectively.

Proof: When the \(\omega\) is even, and \(X_{ei} = (X_i + X_{ei}) / 2\), the even training sample \(X_{ei}\) can be written as \[
\begin{bmatrix}
X_{ei}^a & X_{ei}^o
\end{bmatrix}
\]
In addition, we have supposed the mean even and odd sample matrices are 0; the mean sample matrix of all the training samples is also 0. Therefore:
\[
\begin{align*}
C_e &= \frac{1}{N} \sum_{i=1}^{N} (X_{ei}^a X_{ei}^a) \\
&= \frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix}
X_{ei}^a & X_{ei}^o
\end{bmatrix}^T \begin{bmatrix}
X_{ei}^a & X_{ei}^o
\end{bmatrix} \\
&= \frac{1}{N} \sum_{i=1}^{N} (X_{ei}^a M)^T X_{ei}^a \ (X_{ei}^a M)^T X_{ei}^a M \\
&= \frac{1}{N} \sum_{i=1}^{N} (M^T X_{ei}^a M)^T X_{ei}^a M \\
&= \begin{bmatrix}
B & BM
MB & MBM
\end{bmatrix}
\end{align*}
\]
For \(X_{o}\) = \((X_i - X_{o}) / 2\), the odd training sample \(X_{oi}\) can be written as \[
\begin{bmatrix}
X_{oi}^a & X_{oi}^o
\end{bmatrix}
\]. Therefore:
\[
\begin{align*}
C_o &= \sum_{i=1}^{N} (X_{oi}^a X_{oi}^a) \\
&= \frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix}
X_{oi}^a & -X_{oi}^o
\end{bmatrix}^T \begin{bmatrix}
X_{oi}^a & -X_{oi}^o
\end{bmatrix} \\
&= \frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix}
X_{oi}^a -X_{oi}^o
\end{bmatrix}^T \begin{bmatrix}
X_{oi}^a -X_{oi}^o
\end{bmatrix} \\
&= \frac{1}{N} \sum_{i=1}^{N} \begin{bmatrix}
X_{oi}^a -X_{oi}^o
\end{bmatrix}^T \begin{bmatrix}
X_{oi}^a -X_{oi}^o
\end{bmatrix} \\
&= \begin{bmatrix}
D & DM
MD & MDM
\end{bmatrix}
\end{align*}
\]
From the proofs of Theorem 2, we know \(u_{ei}\) can be written as \[
\begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix}
\]. Therefore:
\[
\begin{align*}
C_e u_{ei} &= \lambda_e u_{ei} \\
\Rightarrow [B & BM] \begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix} &= \lambda_e \begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix} \\
\Rightarrow [B & BM] \begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix} &= \lambda_e \begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix} \\
\Rightarrow M(Bu_{ei}^a) &= \lambda_e Mu_{ei}^o
\end{align*}
\]
Note that when \(MM = I\) (I is a unit matrix), then
\[
\begin{bmatrix}
2Bu_{ei}^a \\
2Bu_{ei}^o
\end{bmatrix} &= \lambda_e \begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix} \\
\Rightarrow \begin{bmatrix}
2Bu_{ei}^a \\
2Bu_{ei}^o
\end{bmatrix} &= \lambda_e \begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix} \\
\Rightarrow \begin{bmatrix}
2Bu_{ei}^a \\
2Bu_{ei}^o
\end{bmatrix} &= \lambda_e \begin{bmatrix}
u_{ei}^a \\
u_{ei}^o
\end{bmatrix}
\]
Therefore,
\[
Bu_{ei} = (\lambda_e / 2) u_{ei}^a
\]
So, we can obtain \(u_{ei}\) by solving the eigenvector \(u_{ei}^a\) of the matrix \(B\).

In a similar way, \(u_{oj}\) can be written as \[
\begin{bmatrix}
u_{oj}^a \\
u_{oj}^o
\end{bmatrix}
\], so:
\[
\begin{align*}
C_o u_{oj} &= \lambda_o u_{oj} \\
\Rightarrow \begin{bmatrix}
D & DM
MD & MDM
\end{bmatrix} \begin{bmatrix}
u_{oj}^a \\
u_{oj}^o
\end{bmatrix} &= \lambda_o \begin{bmatrix}
u_{oj}^a \\
u_{oj}^o
\end{bmatrix} \\
\Rightarrow \begin{bmatrix}
D & DM
MD & MDM
\end{bmatrix} \begin{bmatrix}
u_{oj}^a \\
u_{oj}^o
\end{bmatrix} &= \lambda_o \begin{bmatrix}
u_{oj}^a \\
u_{oj}^o
\end{bmatrix} \\
\Rightarrow 2Du_{oj}^a &= \lambda_o u_{oj}^a \\
\Rightarrow -(MDu_{oj}^o) &= \lambda_o u_{oj}^o
\end{align*}
\]
Therefore:
\[ \textbf{D}u_{ij}^o = (\lambda_{ij} / 2)u_{ij}^o \] (12)

We can thus obtain \textbf{u}_{ij} by solving the eigenvector \textbf{u}_{ij}^o of the matrix \textbf{D}. As formulated above, we can come to the conclusion that \textbf{U}_o and \textbf{U}_o are mutually orthogonal, and can be obtained just by running PCA on the matrices \textbf{B} and \textbf{D}. In particular, Theorem 1 shows that S2DPCA is equal to 2DPCA with another training set in substance. Note that the conventional S2DPCA obtain \textbf{U}_o and \textbf{U}_o by running PCA on the matrices \textbf{C}_o and \textbf{C}_o. That is to say, there is no doubt that our S2DPCA are saving time and memory space compared to the conventional S2DPCA. Based on these theories, we will investigate the time and memory space complexities of S2DPCA in the next section.

### 2.3 Time and Memory Space Complexities

The time and memory space complexities of 2DPCA, conventional S2DPCA and our S2DPCA are shown in Table 1. Here, \( w \) is the number of columns of face image, and \( L \) is the number of the projection vectors. Note that we only give the training time and memory space complexities; the testing time(memory space) complexities of these algorithms are the same if we suppose the projection vectors selected. In Table 1, it is very clear that the time and memory space complexities of 2DPCA[11], conventional S2DPCA and our S2DPCA, from the point of view of the computational complexity theory, are the same. The time and memory costs of our S2DPCA, however, can both be reduced to a quarter of that of conventional S2DPCA. Note that the time and memory costs of conventional S2DPCA are double those of 2DPCA, while the costs of our S2DPCA are less than 2DPCA. In fact, the computational cost and memory space cost of our S2DPCA are least among these algorithms. To make this more straightforward, the computing (CPU) time and memory costs of conventional S2DPCA and our S2DPCA on YALE will be given in Section 4.1.

| Algorithm     | Complexity          | Time | Memory |
|---------------|---------------------|------|--------|
| 2DPCA         | \( \text{O}(w^2L) \) | \( \text{O}(w^2) \) | |
| Conventional S2DPCA | \( 2\text{O}(w^2L) \) | \( 2\text{O}(w^2) \) | |
| Our S2DPCA    | \( 2\text{O}(w^2L/4) \) | \( 2\text{O}(w^2/4) \) | |

Table 1. Time and memory space complexities

### 2.4 Classification

A feature matrix \( \textbf{Y}_i = [y_{i1}, y_{i2}, \ldots, y_{id}] \) for each training face sample can be obtained by \( y_{ik} = \textbf{X}_i\textbf{u}_k, k=1,2,\ldots,d \) after the transformation by S2DPCA (2DPCA). Here, \( \textbf{u}_k \) is the eigenvector of subspace. In the same way, we can also get a feature matrix \( \textbf{Y}_j = [y_{j1}, y_{j2}, \ldots, y_{jd}] \) for each testing face sample. Then, a nearest neighbour classifier is used for classification [23].

\[ c = \arg \min_{i \in [1,2,\ldots,N]} d(\textbf{Y}_i, \textbf{Y}_j) \] (13)

The distance between \( \textbf{Y}_i \) and \( \textbf{Y}_j \) is minimal, so \( \textbf{Y}_i \) belongs to the class \( \textbf{Y}_j \) belongs to. The similarity measure (i.e. distance) between any training and testing feature matrices is traditionally defined by

\[ d(\textbf{Y}_i, \textbf{Y}_j) = \| \textbf{Y}_i - \textbf{Y}_j \|_2 \] (14)

For the sake of brevity, these procedures are called S2DPCA+DM and 2DPCA+DM. Two other similarity measures are:

\[ d(\textbf{Y}_i, \textbf{Y}_j) = \sqrt{Y_i^T \cdot Y_j} \] (15)

\[ d(\textbf{Y}_i, \textbf{Y}_j) = \sqrt{\max(m,n) - \sum_{i=1}^{m} \sum_{j=1}^{n} (y_{ij}^2)^2} \] (16)

In equation (15), we give the VM [16],SM is defined in equation (16), \( m \) is the rank of \( \textbf{Y}_i \), and \( n \) is the rank of \( \textbf{Y}_j \). Note that in equation (16), \( \textbf{Y}_i \) and \( \textbf{Y}_j \) need to be orthonormal subspaces. So, the Gram-Schmidt orthogonalization should be applied first. The Gram-Schmidt orthogonalization process is numerically unstable, particularly when applied to large bases [25]. Fortunately, the command “orth” in Matlab gives a numerically stable process to produce an orthonormal subspace (matrix). Therefore, we can use equation (16) expediently. For DM every column of each sample is independent. But VM can build a high dimensional geometrical body, so it can be illustrated by the difference of corresponding elements (points) of two samples. In geometry, any geometrical body has a volume. The size of the volume can be computed to measure the similarity of the two samples. In this measure the points are not independent anymore, and the integral characteristic would be used. SM can measure the similarity between two images by subspace distance.

Similarly, for simplicity, we call these algorithms S2DPCA+VM, 2DPCA+VM, S2DPCA+SM, and 2DPCA+SM. In contrast to traditional DM, VM and SM are regarded as image measures. The steps of S2DPCA-based algorithms in our paper are depicted as follows.

1. Decompose each training face sample into an even symmetrical sample and an odd symmetrical one to get the even/odd symmetrical training sets.
2. Apply 2DPCA to the even/odd symmetrical training sets separately to get even/odd subspaces. Combine features in even and odd subspaces based on eigenvalues.

3. Extract low dimension features by projecting the face sample to the even/odd subspace for final classification.

4. Apply nearest neighbour classifier with DM, VM and SM.

3. Experimental results

Many studies (see [7],[13],[16]), have shown that the performance of 2DPCA is better than that of PCA, so we only compare S2DPCA with 2DPCA using different similarity measures on three well-known face databases, i.e., YALE [26], AR [27], and FERET face databases [28] [29]. On YALE and AR face databases, we investigate the algorithms performance when facial expressions, illumination, etc., are varied. The algorithms performance by cumulative match score (CMS) [28] is also investigated on FERET face database. The experiments are performed using Matlab 7.0 on a Pentium IV 3.0GHz processor and a total RAM memory of 1GB.

3.1 Results on YALE Database

The YALE database includes 11 different images of each of 15 individuals, in which the face images vary in different expressions and light conditions [2] [16]. All grey-scaled images were cropped and normalized to a resolution of 66x56 pixels in the experiments.

Our experiments are performed using two and three random face samples per person for training, and the remaining images for testing. Here, training samples are randomly selected in one person’s samples, but the ordinal of samples for different people is the same. The average recognition rates (ARR), which are calculated over ten runs, are shown in Fig. 1 and Fig. 2.

Figure 1. ARR on YALE database with 2 samples per person.

Figure 2. ARR on YALE database with 3 samples per person

The corresponding standard deviations ($\sigma$) are described in Fig. 3 and Fig. 4. We illustrate the
performance with the first two and three face samples per person for training (others for testing) in Fig. 5 and Fig. 6. Comparing Fig.1 and Fig.2 with Fig. 5 and Fig.6, the ARRs are smooth, while the recognition rates show marked fluctuations. It can be seen that S2DPCA+VM generally outperforms others. Particularly with four to six features selected, the best ARRs can be obtained by S2DPCA+VM. The standard deviation of these algorithms is about 2% when the number of features is bigger than two. S2DPCA-based algorithms have the potential to outperform 2DPCA-based ones. S2DPCA and 2DPCA with SM could not obtain a good performance.

![Figure 5](image5.png)

**Figure 5.** Recognition rate on YALE database with the first 2 samples per person

![Figure 6](image6.png)

**Figure 6.** Recognition rate on YALE database with the first 3 samples per person

We compare the computing (CPU) time and the size of the covariance matrices for our S2DPCA and conventional S2DPCA in Table 2. The classifier based on the nearest neighbour with DM is applied (the first two face samples per person for training, others for testing). The number of features selected is nine.

| Algorithm       | Average time (ms) | Memory space cost |
|-----------------|-------------------|-------------------|
|                 | Training time     | Testing time      | Size of matrix |
| Conventional    | 531.0             | 810.0             | 56*56          |
| S2DPCA          | 203.0             | 808.0             | 28*28          |

**Table 2.** CPU time and memory space cost

In Table 2, our S2DPCA takes only about 203.0ms for training, while conventional S2DPCA takes about 531.0ms. The size of matrix is reduced from 56*56 to 28*28. Here, the time cost of our S2DPCA can be reduced to about 203/531 compared to the conventional S2DPCA. Like the theory analysis mentioned above, it is not distinct (1/4), for the CPU time involves other requested operations, such as reading images, constructing sample matrices, etc. Testing data by Conventional S2DPCA and our S2DPCA takes about the same time for 135 samples, which means less than 6.0 ms per sample.

3.2 Results on AR Database

The AR database consists of over 4,000 colour face images, and the images vary in different lighting conditions, facial expressions and occlusions, etc. [27]. The images of most persons were taken in two sessions, separated by two weeks, each section consisting of 13 colour face images for one person. The face portion of the images is cropped, grey-scaled and normalized to a resolution of 66x56 pixels. The face images of 119 persons are used in the experiments. The histogram equalization is used in preprocessing. The images of one person used in our experiments are shown in Fig. 7.

![Figure 7](image7.png)

**Figure 7.** Sample images for one person on AR database

We select the first five images per person in every two sessions as the training and testing samples. The experiments are performed using five random face samples per person for training, and the remaining images for testing. The average recognition rates and standard deviation, calculated over ten runs, are shown in Fig. 8 and Fig. 9. S2DPCA-based algorithms have the potential to outperform 2DPCA-based ones. The standard deviations of these algorithms are about 1.6%. S2DPCA+VM generally outperform others. S2DPCA and 2DPCA with SM still could not obtain good performance.
3.3 Results on FERET Database

For the purpose of some applications, e.g., verification from biometric signatures stored on smart cards [28], we validate S2DPCA on the FERET face database [29], which has been widely used to evaluate face recognition approaches. Details on this database and the terms used in the experiments can be found in [28]. Two FERET image sets are used in the experiments. FA is a regular frontal face library used as the gallery set, including frontal images of 1196 individuals. FB, including frontal images of 1195 individuals used as the probe set, is an alternative frontal face library, with images taken seconds after the corresponding images in FA [16]. 998 face images are selected as the training set, which includes samples from FA and FB.

Before performing the experiments, all images in the FA and FB sets are rectified using the positions of the eyes, provided by FERET, and are cropped and standardized to the 60x50 face images. To further reduce the effect of illumination, the histograms of these images are equalized for preprocessing. The recognition rate with varying number of features is given in Table 3. CMS is the performance statistics, which are shown in Fig.10. The number of features is that where algorithms can obtain the best recognition rates, that is, the number for 2DPCA+DM is 10, the number for both 2DPCA+VM and S2DPCA+VM is 8, and for S2DPCA+DM it is 12. The performances of 2DPCA+SM and S2DPCA+SM were so poor they have not been presented in Fig.10. As can be seen from Table 3 and Fig. 10, of these algorithms S2DPCA+VM can obtain the best recognition rates with eight features. The CMS of S2DPCA+VM generally outperform those of the other algorithms, though this difference is not marked.

Table 3. Recognition rates (%) with varying number of features

| Algorithm     | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   |
|---------------|------|------|------|------|------|------|------|------|------|------|
| 2DPCA+DM      | 80.3 | 81.2 | 82.3 | 83.6 | 83.6 | 83.9 | 83.8 | 83.3 | 83.8 | 83.6 |
| 2DPCA+VM      | 85.3 | 85.8 | 87.4 | 88.2 | 86.9 | 87.2 | 86.2 | 86.2 | 85.6 | 85.7 |
| 2DPCA+SM      | 63.9 | 64.0 | 64.6 | 63.7 | 61.8 | 62.8 | 60.6 | 59.7 | 58.2 | 56.7 |
| S2DPCA+DM     | 80.5 | 81.3 | 82.3 | 82.7 | 83.0 | 83.7 | 83.8 | 84.3 | 84.0 | 84.0 |
| S2DPCA+VM     | 83.8 | 85.8 | 87.5 | 88.3 | 86.9 | 87.2 | 86.4 | 86.2 | 85.7 | 85.7 |
| S2DPCA+SM     | 56.9 | 64.5 | 64.8 | 63.6 | 61.6 | 62.8 | 60.3 | 59.4 | 59.2 | 56.7 |
4. Conclusions

In this paper, we have investigated the orthogonality of the even and odd subspaces in 2DPCA in detail, and, in particular, have proved that S2DPCA can be constructed on a quarter of conventional S2DPCA seven/odd covariance matrix. Moreover, based on these theories, we find that S2DPCA can be computed using a quarter of the time and memory costs compared to conventional S2DPCA from the point of view of the computational complexity theory. Furthermore, two image measures, volume measure (VM) and subspace distance measure (SM), are introduced to S2DPCA-based face recognition.

S2DPCA can explore more prior-knowledge, i.e., facial symmetry, based on even and odd decompositions, as well as obtain a more precise covariance matrix in the same way as 2DPCA. Therefore, S2DPCA-based algorithms are able to achieve better performance in general. Nevertheless, theoretically, the time and memory space costs of conventional S2DPCA are doubled compared to those of 2DPCA, which is an annoying issue. We solve this problem successfully with the S2DPCA theories proved in this paper. The experimental results on YALE, AR, and FERET databases show that, generally, S2DPCA+VM outperform some other relevant algorithms. Most importantly, the computational cost and memory space needed for our S2DPCA are less compared to 2DPCA and conventional S2DPCA.

Finally, we should point out that there are still two aspects in S2DPCA worthy of further investigation. How many even and odd features should be selected in order to achieve better performance? From the experimental results, the number of features with which good performance can be obtained strongly depends on face database selection. In addition, VM is not satisfied by the triangle inequality, which means VM is not ametric. With constraints, could VM be a metric? S2DPCA+VM would be applied widely when both aspects have been worked out successfully.

5. Acknowledgements

This work is supported in part by the National Natural Science Foundation for Distinguished Young Scholars of China under Grant No.60905011.

6. References

[1] Shan S (2004) Study on some key issues in face recognition. Ph.D. dissertation, Institute of Computing Technology Chinese Academy of Sciences, Beijing, China.

[2] Corcoran P (2011) Reviews, Refinements and New Ideas in Face Recognition. Rijeka: InTech. 328 p.

[3] Li R, Yang Y, Yuan Y (2009) L1-norm-based 2DPCA. IEEE Trans. Syst. Man Cybern. B: Cybern., 40(4): 1170-1175.

[4] Corcoran P (2011). 3D Face Recognition. Rijeka: InTech. 252 p.

[5] Yang Q (2004) Research on appearance-based statistical face recognition. Ph.D. dissertation, Dept. of Electro. Eng. Tsinghua Univ. Beijing, China.

[6] Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. 28(12):2037-2041.

[7] Yang J, Zhang D, Frangi A F, Yang J Y (2004) Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans. Pattern Anal. Mach. Intell. 26(1):131-137.

[8] Kong H, Li X C, Eam T, Wang J G, Ronda V (2005) Generalized 2D principal component analysis. In Proc. IEEE Inter. Joint Conf. Neural Networks, Montreal, pp.108-113.

[9] Li M, Yuan B Z (2005) 2D-LDA: A statistical linear discriminant analysis for image matrix. Pattern Recognit. Lett. 26(5):527-532.

[10] Zhang D, Zhou Z, Chen S (2006) Diagonal principal component analysis for face recognition. Pattern Recognit. 39(1):140-142.

[11] Niu B, Yang Q, Shiui S, Pal S (2008) Two-dimensional Laplacianfaces method for face recognition. Pattern Recognit. 41(10):3237-3243.

[12] Gao Q, Zhang L, Zhang D, Yang J (2007) Comments on “On image matrix based feature extraction algorithms”. IEEE Trans. Syst., Man, Cybern. B: Cybern. 37(5):1373-1374.

[13] Yu H, Zhang J (2010) An Extension of Principal Component Analysis. In: Oravec M, editor. Face Recognition. Rijeka: InTech. pp. 21-34.

[14] Yan S, Xu D, Yang Q, Yang L, Tang X, Zhang H (2007) Multilinear discriminant analysis for facial recognition. IEEE Trans. Image Processing. 16(1):212-220.

[15] Zuo W M, Zhang D, Wang K (2006) Bidirectional PCA with assembled matrix distance metric for image recognition. IEEE Trans. Syst. Man Cybern. B: Cybern. 36(4): 863-871.

[16] Meng J, Zhang W (2007) Volume measure in 2DPCA-based Face Recognition. Pattern Recognit. Lett. 28(10):1203-1208.

[17] Wang L, Wang X, Feng J (2006) Subspace distance analysis with application to adaptive Bayesian algorithm for face recognition. Pattern Recognit. 39(3):456-464.

[18] Wang L, Wang X, Feng J (2007) Further results on the subspace distance. Pattern Recognit. 40(1):328-329.

[19] Yang Q, Ding X (2002) Symmetrical PCA in face recognition. In Proc. IEEE Int’l Conf. on Image Processing, New York, USA, pp. 97-100.
[20] Kirby M, Sirovich L (1990) Application of the Karhunen-Loève procedure for the characterization of human faces. IEEE Trans. Pattern Anal. and Mach. Intell. 12 (1):103-108.

[21] Lu C, Zhang C, Zhang T, Zhang W (2007) Kernel based symmetrical principal component analysis for face classification. Neurocomputing.70(4-6):904-911.

[22] Ding M, Lu C, Lin Y, Tong L (2007) Symmetry based two-dimensional principal component analysis for face recognition. The 4th International Symposium on Neural Networks, NanJing, pp.1048-1055.

[23] Meng J, Zhang W (2008) S2DPCA with DM and FM in Face Recognition. In: IEEE International Conference on Apperceiving Computing and Intelligence Analysis (ICACIA’08), Chengdu, China. pp.365-368.

[24] Scholkopf B, Smola A J, Muller K R (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation.10(5): 1299-1319.

[25] Aster R C, Thurber C H, Borchers B (2005) Parameter Estimation and Inverse Problems, Holland: Elsevier Academic Press.

[26] Yale University (1997) The Yale Face Database. Available: http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

[27] Martinez A M, Benavente R (1998) The AR Face Database. CVC Technical Report, no.24.

[28] Phillips P J, Moon H, Rizvi S A, Rauss P J (2000) The FERET evaluation methodology for face recognition algorithms. IEEE Trans. Pattern Anal. and Mach. Intell.22(10):1090–1104.

[29] Defense Advanced Research Products Agency (DARPA) (2004). The Facial Recognition Technology (FERET) Database. Available: http://www.itl.nist.gov/iad/humaid/feret/feret_master.html.