SI2DPCA: A Low-Computation Face Recognition Approach

Ying-Kuei Yang, Wei-Li Fang and Jung-Kuei Pan
Dept. of Electrical Engineering, National Taiwan Uni. of Sci. & Technology, Taipei, Taiwan
Email: yingkyang@yahoo.com

Abstract - *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 experimental results have demonstrated that SI2DPCA works well on reaching the goals of reducing computation cost and improving recognition simultaneously after comparing its performance against several better-known approaches.*

Keywords: face recognition, feature extraction, principle component analysis, covariance computation, eigen-decomposition.

1 Introduction

Face recognition in image processing has been significantly important because it can be applied in human life efficaciously [1]-[10]. Several algorithms have been proposed in face recognition. The Yang *et al.*proposed the so-called two-dimensional principal component (2DPCA) algorithm aiming for better feature extraction of face images to increase recognition rate and reduce computation cost simultaneously [11]. 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 *et al.* [12] is to process a face image from transverse and longitudinal axis respectively and then perform the recognition by analyzing their shortest dimension. Unfortunately, its improvement on recognition rate is not ubiquitous in relatively large scale of training samples [13]. Sanguansat *et al.* [14] proposed the approach of "Two-dimensional principal component combined two-dimensional Linear discriminant analysis (2DPCA&2DLDA)" [14] 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 *et al.* [15] 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.* [16] 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.* [17] 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 [18].

Aforesaid face recognition algorithms based on 2DPCA have tried to either increasing recognition rate or reducing computation cost, but not both. In this paper, an approach named *sub-image 2-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 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.

2 The sub-image 2-dimensional principal component analysis

2.1 Two-dimensional principal component analysis (2DPCA)

The 2DPCA approach by Yang *et al.* [11] in 2004 is proposed particularly for two dimensional image data. Suppose there is an image data set Z={$\mathbf{A}_1$, $\mathbf{A}_2$,…, $\mathbf{A}_N$ } 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. (1) and the average value of the data set is computed by Eq. (2).

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

$$\overline{\mathbf{A}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}_i \quad (2)$$

where $\mathbf{A}_i$ 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. (3).

$$\mathbf{Y}_i = \mathbf{A}\,\mathbf{X}_i \qquad i=1,2,\ldots,k \tag{3}$$

where $\mathbf{Y}_i$ are projected feature vectors, $\mathbf{X}_i$ means eigenvectors. Suppose 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.The maximum allowed number of pages is seven for Regular Research Papers (RRP) and Regular Research Reports (RRR); four for Short Research Papers (SRP); and two for Posters (PST).

2.2 Sub-image 2-dimensional principal component (SI2DPCA)

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 [19]. Therefore, SI2DPCA proposes to equally divide a face image into smaller sub-images to be processed so that the total computation cost can be reduced.

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. (4).

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & \cdots & a_{1m} \\ a_{21} & a_{22}-\lambda & \cdots & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & \cdots & a_{mn}-\lambda \end{vmatrix} = 0 \tag{4}$$

The next step is to apply the extension method [19] to extend the square matrix in Eq. (4). The initial procedure is to choose the first element of every column from the determinant in Eq. (4) 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. (5) 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.. Eq. (5) is the result of extending Eq. (4).

$$(a_{11}-\lambda)\cdot\underbrace{\begin{vmatrix} a_{22}-\lambda & \cdots & a_{2m} \\ \vdots & \ddots & \vdots \\ a_{m2} & \cdots & a_{mm}-\lambda \end{vmatrix}}_{\text{smaller determinant}} - a_{12}\cdot\begin{vmatrix} a_{21} & \cdots & a_{2m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm}-\lambda \end{vmatrix} + a_{13}\cdot\begin{vmatrix} a_{21} & \cdots & a_{2m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm}-\lambda \end{vmatrix}$$
$$-\cdots\cdots+(-1)^{(1+m)}\cdot a_{1m}\cdot\begin{vmatrix} a_{21} & \cdots & a_{2(m-1)} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{m(m-1)} \end{vmatrix} = 0 \tag{5}$$

Similar extension process needs to be performed against every smaller determinant in Eq. (5). 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. (5), it is obvious to observe that the higher dimension a square matrix has, the higher computation cost is.

Eq. (6) shows the result of dividing the determinant of the square matrix in Eq. (4) into four smaller ones.

$$\left|\begin{array}{cccc|ccc} a_{11}-\lambda & a_{12} & \cdots & a_{1\frac{m}{2}} & a_{1(\frac{m}{2}+1)} & \cdots\cdots & a_{1m} \\ a_{21} & a_{22}-\lambda & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{\frac{m}{2}1} & \cdots & \cdots & a_{\frac{m}{2}\frac{m}{2}}-\lambda & a_{\frac{m}{2}(\frac{m}{2}+1)} & \cdots\cdots & a_{\frac{m}{2}m} \\ \hline a_{(\frac{m}{2}+1)1} & \cdots & \cdots & a_{(\frac{m}{2}+1)\frac{m}{2}} & a_{(\frac{m}{2}+1)(\frac{m}{2}+1)}-\lambda & \cdots\cdots & a_{(\frac{m}{2}+1)m} \\ \vdots & \ddots & & \vdots & \vdots & \ddots & \vdots \\ \vdots & & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & \cdots & a_{m\frac{m}{2}} & a_{m(\frac{m}{2}+1)} & \cdots\cdots & a_{mm}-\lambda \end{array}\right| = 0 \tag{6}$$

Because the time complexity of computing a determinant is $O(n!)$, the total computation cost of computing the determinants for each of smaller square matrices in Eq. (6) is much less than the computation cost for Eq. (5). From the final equation that is set to be zero, several λ values can be obtained. The eigenvectors can consequently be calculated by substituting λ values into its corresponding matrix in Eq. (6). 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.

Because current computers are mostly binary-based systems, ill condition problem that gives incorrect result could be caused when computing eigen-decomposition [19]. To avoid such problem, many studies use singular value decomposition (SVD) to replace the process of eigen-decomposition. For a matrix with dimension $m\times n$, the computation cost of SVD can be described as Eq. (7) [20].

$$4m^2n + 8mn^2 + 9n^3 \tag{7}$$

The big-order of (15) is $O(n^3)$ when $m < n$, meaning the dimension variation causes significant difference in terms of computation cost. 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.

The comparison of computation cost between the conventional 2DPCA and the proposed SI2DPCA is shown in Table 1 that details the formulas of computation cost. In Table 1, the formula (a) that comes from Eq. (2) is to compute the data average by 2DPCA. The N means computation summation of N images. The computation amount of m×n in Formula (a) is for matrix addition. Adding one in formula (a) is for the division operation in Eq. (2). Formula (b) is from Eq. (1) to compute the covariance matrix. The N and the plus-one have same meaning as in formula (a). There are three operations in Eq. (1). The first one is subtracting the data average obtained by formula (a) from the original image data, causing m×n computation. There are two such operations in Eq. (1), so the computation cost is 2×m×n. The second operation is the multiplication of the two matrices shown in Eq. (1). The computation cost of such matrix multiplication is n^3. The third operation is the computation of transposing a matrix, causing m×n computation. Formula (c) comes from Eq. (7) and is the computation cost of eigen-decomposition for 2DPCA.

The rest of formulas in Table 1 are for SI2DPCA. Suppose an original image with dimension m×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 $((m/ k^{1/2}) \times (n/ k^{1/2}))$. The Eq. (2), Eq. (1) and Eq. (7) need to be applied for each of k smaller matrices in that order. As explained previously for formulas (a) to (c), the computation costs indicated in formulas (d) to (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.

Table 1: Analysis of computation cost

Computation type	2DPCA	SI2DPCA
Data average computation	$N\times(m\times n)+1$ (a)	$k\times N\times[(m\times n)/k]+k$ (d)
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)
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)

In previous discussions, an original image is assumed in m×n dimension. In reality, a 2-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 1 can therefore be summarized in Table 2. Although Table 2 shows SI2DPCA has no advantage over 2DPCA in terms of time complexity, its actual computation cost is much less when k is greater than one. The bigger the k is, the greater the decreased amount is for computation cost. In general, the most reasonable k is 4 meaning at least roughly half computation cost is reduced by SI2DPCA.

Table 2: Time complexity analysis in big order for matrices

Computation type	2DPCA	SI2DPCA
Data average computation	m^2	m^2
Covariance computation	m^3	$(m/ k^{1/2})^3$
Eigen-decomposition computation	m^3	$(m/ k^{1/2})^3$

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 can not be easily recognized. Even these two features have been recognized, their weights in its image may not be as great as those extracted from smaller sub-images because of the co-existence of other features in the whole bigger image.

3 Experiments and analysis

3.1 The ORL database

The ORL database [21] 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 degrees [11][21]. In ORL database, all images are grayscale with dimension of 112×92. The pixel value range is 0~255.

3.2 Experiments and analysis of SI2DPCA

According to Table 1, 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 1 are 112 and 92 respectively. And each image is divided into 4 smaller sub-images, meaning the value of k in Table 1 is 4. Suppose 200 images are taken as training data, meaning N in Table 1 is 200, and 8 features are selected and extracted. Putting these values into Table 1, the result is shown in Table 3.

Table 3 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

Table 3: Analysis of computation cost for ORL database

Computation type	2DPCA	SI2DPCA
Data average computation	$200\times(112\times92)+1$ $=2060801$	$4\times200\times(56\times46)$ $+4=2060804$
Covariance matrix computation	$200\times(2\times112\times92$ $+ 92^3+112\times92)+1$ $= 161920001$	$4\times200\times(2\times56\times46$ $+46^3+56\times46)+4$ $=84051204$
Eigen-decomposition computation	$4\times112^2\times92 +$ $8\times112\times92^2 + 9\times92^3$ $=19208128$	$4\times(4\times56^2\times46+8\times56$ $\times46^2+9\times46^3)$ $= 9604064$
Sum of computation cost	183188930	95716072

Besides the computation cost, the proposed SI2DPCA and conventional 2DPCA also need be compared on their recognition performance. The recognition is performed by the nearest neighbor rule (NNR) [11] that is based on Euclidean distance.

In this experiment, the first 5 images of every face are treated as training and the remained 5 images of every face are treated as testing images. That is, there are 200 images for training and 200 images for testing. Eight important features are extracted in the experiment, meaning a 8-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 8-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 recognition rate comparison between 2DPCA and SI2DPCA is shown in Table 4. Earlier discussions argued that important features can be better recognized and extracted in smaller sub-images. This can be observed in Table 4 that shows slight better recognition rate for SI2DPCA over conventional 2DPCA. Both Table 3 and Table 4 together show that the SI2DPCA reduces computation cost without compromising its recognition performance.

Table 4: Recognition comparison between 2DPCA and SI2DPCA

Strategy	Recognition rate
2DPCA	93%
SI2DPCA	93.5%

Various methodologies based on 2DPCA have been proposed. Table 5 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 5 are conducted based on the face images in ORL database. In Table 5, 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 5, 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.

Table 5: Comparison among other methods and SI2DPCA

Method number	Method	Recognition rate	Computation cost
1	$(2D)^2$PCA [12]	90.5%	high
2	2DPCA+Fusion method based on bidirectional [17]	92.5%	high
3	2DPCA+2DLDA [14]	93.5%	high
4	**SI2DPCA (proposed)**	**93.5%**	**low**
5	2DPCA+Kernel [22]	94.58%	very high
6	2DPCA+Feature fusion approach [23]	98.1%	very very high

4 Conclusions

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.

5 References

[1] A. Z. Kouzani, F. He, and K. Sammut, “Towards invariant face recognition,” *Inf. Sci.*, vol. 123, pp. 75–101, 2000.

[2] M. J. Er, S. Wu, J. Lu, and H. L. Toh, “Face recognition with radial basis function RBF neural networks,” *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 697–710, May 2002.

[3] 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.*, vol. 14, no. 5, pp. 1162–1175, Sep. 2003.

[4] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Face recognition using kernel direct discriminant analysis algorithms,” *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 117–126, Jan. 2003.

[5] B. K. Gunturk, A. U. Batur, and Y. Altunbasak, “Eigenface-domain super-resolution for face recognition,” *IEEE Trans. Image Process.*, vol. 12, no. 5, pp. 597–606, May 2003.

[6] B. L. Zhang, H. Zhang, and S. S. Ge, “Face recognition by applying wavelet subband representation and kernel associative memory,” *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 166–177, Jan. 2004.

[7] Q. Liu, X. Tang, H. Lu, and S. Ma, “Face recognition using kernel scatter-difference-based discriminant analysis,” *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 1081–1085, Jul. 2006.

[8] W. Zheng, X. Zhou, C. Zou, and L. Zhao, “Facial expression recognition using kernel canonical correlation analysis (KCCA),” *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 233–238, Jan. 2006.

[9] 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.*, vol. 16, no. 4, pp. 875–886, Jul. 2005.

[10] G. Huang, “Fusion (2D)2PCALDA: A new method for face recognition,” *Applied Mathematics and Computation*, 216, pp. 3195–3199, 2010.

[11] 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 Analysis and Machine Intelligence.*, vol. 26, no. 1, pp. 131-137, Jan. 2004.

[12] D. Zhang and Z. H. Zhoua, “(2D)^2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition,” *Neurocomputing* 69, pp. 224–231, Jun. 2005.

[13] W. H. Yang and D. Q. Dai, “Two-Dimensional Maximum Margin Feature Extraction for Face Recognition,” *IEEE Transactions on Systems, Man and Cybernetics.*, vol. 39, no. 4, pp. 1002-1012, Aug, 2009.

[14] P. Sanguansat, W. Asdornwised, S. Jitapunkul and S. Marukatat, “Tow-dimensional linear discriminant analysis of principle component vectors for face recognition,” *IEEE.*, pp. 345-348, 2006.

[15] J. Meng and W. Zhang, “Volume measure in 2DPCA-based face recognition,” *Pattern Recognition Letters* 28, pp. 1203–1208, Jan. 2007.

[16] 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 & Applic* 17, pp. 541–547, 2008.

[17] 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,” *Applied Mathematics and Computation.*, vol. 205, pp. 601–607, 2008.

[18] Y. Qi and J. Zhang, “$(2D)^2$PCALDA: An efficient approach for face recognition,” *Applied Mathematics and Computation.*, vol. 213(1), pp. 1-7, Jul. 2009.

[19] S. J. Leon, *Linear algebra with applications*, seventh ed. Prentice Hall, 2006.

[20] 3D Computer Vision Winter Term 2005/06, Nassir Navab

[21]“The ORL face database”, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.htm

[22] 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 & Applic.*, 17, pp. 59-64, 2008.

[23] 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, pp. 1857-1865, Feb, 2008.