BILATERAL TWO-DIMENSIONAL LOCALITY PRESERVING PROJECTIONS

Si-Bao Chen[1], *Bin Luo*[2], *Guo-Ping Hu*[1], *Ren-Hua Wang*[1]

[1]Anhui USTC Iflytek Lab, Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China
[2]School of Computer Science and Technology, Anhui University, Hefei 230039, China

ABSTRACT

In this paper, we investigate Locality Preserving Projections (LPP) in two-dimensional sense. Recently, LPP was proposed for dimensionality reduction, which can detect the intrinsic manifold structure of data and preserve the local information. When image data are concerned, they are often vectorized for LPP. However, the dimension of image data is usually very high, LPP can't be implemented due to singularity of matrix. We propose two methods for image dimensionality reduction: two-dimensional LPP (2DLPP) and bilateral two-dimensional LPP (B2DLPP), which are based directly on 2D image matrices rather than 1D vectors as LPP does. Experiments are conducted on the ORL face database, which shows higher recognition performance of the proposed methods.

***Index Terms*—** Pattern recognition, image analysis, locality preserving projection, dimensionality reduction, two-dimensional method

1. INTRODUCTION

In pattern recognition, when data have high dimension, such as image data, recognition becomes very hard. Due to the wide application of image recognition, many methods have been developed for it over the past few decades. Appearance-based methods are among those well-investigated. However, they are often confronted with dimensionality reduction problems because the dimension of vector representation of an $(s \times t)$ image is too high to allow fast and good recognition. Two of the most classical dimensionality reduction methods are Principal Component Analysis (PCA) [1] and Linear Discriminant Analysis (LDA) [2]. And now two new techniques, namely 2DPCA [3] and Laplacianfaces [4], have appeared in recent literature.

PCA [1] aims to find a linear mapping which preserves variance. However, PCA focuses on low-dimensional representation of data and does not take use of label information in training set for recognition. LDA [2] pursues a linear mapping which preserves discriminant information. Therefore, LDA contains the most discriminant information in the training set (rather than the test set). Given a sufficient training sample, LDA is superior to PCA. While for a small sample size problem, PCA can outperform LDA because LDA is sensitive to the training data set.

Compared with traditional PCA, 2DPCA [3] extracts image features directly from 2D image matrices rather than 1D vectors. Therefore, image matrices do not need to be transformed into vectors. An image covariance matrix is constructed from original image matrices for feature extraction. The optimal projection axes are its orthogonal eigenvectors corresponding to its largest eigenvalues. Due to smaller size of image variance matrix than original variance matrix, 2DPCA requires less time to extract image features and achieves higher recognition performance. Recently, Sanguansat et al.[5] performed 2DLDA in 2DPCA feature space to reduce number of coefficients needed and obtained higher recognition rate.

Laplacianfaces is based on a technique called Locality Preserving Projections (LPP) [4], which finds an embedding that preserves local information, and obtains a face subspace that best detects the essential face manifold structure. He et al. [4] constructed a similarity matrix of data points, then minimized the sum of square difference of features weighted by similarity. The optimal projection axes best preserve the local structure of the underlying distribution in some sense. From analysis they found that LPP is connected with PCA and LDA. LPP can be seen as a generalization of LDA.

2DPCA and 2DLDA are simple in computational complexity, which can only see the Euclidean structure of image space. LPP can find an embedding that preserves local information. But if the training samples are insufficient and data dimension is high especially for image data, LPP can't be used directly due to singularity of matrices. In this paper, we investigate LPP in two-dimensional sense and in bilateral two-dimensional sense based on image matrices directly. they are referred as Two-Dimensional Locality Preserving Projections (2DLPP) and Bilateral Two-Dimensional Locality Preserving Projections (B2DLPP) respectively hereafter. Their recognition performances are evaluated.

The rest of this paper is arranged as the following: LPP method is recalled in section 2; In section 3, 2DLPP is demonstrated in detail; B2DLPP is deduced and described in section 4; Section 5 contains experiments on ORL face database to

Thanks to the China National 863 Program (No. 2004AA114030) and National Natural Science Foundation of China (No.60375010) for funding.

test the performance of the proposed methods comparing with other methods; Conclusion is drawn in Section 6.

2. LOCALITY PRESERVING PROJECTIONS

He et al.[4] considered a low-dimensional manifold $\mathcal{M}$ which is embedded in p-dimensional Euclidean space $\mathbf{R}^p$. A set of data $\{\mathbf{x}_i,\ i=1,\ldots,n\}$ is extracted from the manifold. Now a projection axis $\mathbf{w}$ is expected such that after projection $y_i = \mathbf{w}^\top\mathbf{x}_i$, $\{y_i\}$ represent $\{\mathbf{x}_i\}$ as much as possible. A proper criterion of selecting this projection axis $\mathbf{w}$ is to minimize the following objective function under some constraint:

$$\arg\min_{\mathbf{w}} \sum_{ij} (y_i - y_j)^2 s_{ij}, \tag{1}$$

where matrix $\mathbf{S}$ is a similarity matrix with element s_{ij} being similarity between the ith and the jth training data. Through some mathematical deduction, we can see

$$\sum_{i<j} s_{ij}(y_i - y_j)^2 = \mathbf{w}^\top\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{w}, \tag{2}$$

where $\top$ is transpose of vector (matrix), $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$ is the training data matrix, $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is Laplacian matrix and $\mathbf{D}$ is a diagonal matrix whose entries d_{ii} are column (or row) sums of $\mathbf{S}$, $d_{ii} = \sum_j s_{ij}$. d_{ii} corresponds to the ith training data point. The larger d_{ii} is, the more "important" the ith training data point is, which determines that the more "important" y_i is after the projection. Therefore, to eliminate arbitrary scalability and translation of the minimization problem, it is proper to impose a constraint:

$$\sum_i d_{ii} y_i^2 = 1 \ \ \Rightarrow\ \ \mathbf{w}^\top\mathbf{X}\mathbf{D}\mathbf{X}^\top\mathbf{w} = 1. \tag{3}$$

Now the minimization problem turns into:

$$\arg\min_{\mathbf{w}^\top\mathbf{X}\mathbf{D}\mathbf{X}^\top\mathbf{w}=1} \mathbf{w}^\top\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{w}. \tag{4}$$

The optimal projection axis $\mathbf{w}$ is given by the minimal eigenvalue solution to the generalized eigenvalue problem:

$$\mathbf{X}\mathbf{L}\mathbf{X}^\top\mathbf{w} = \lambda\mathbf{X}\mathbf{D}\mathbf{X}^\top\mathbf{w}. \tag{5}$$

Similarly, to pursue a projection matrix $\mathbf{W}$, projecting $\mathbf{x}_i$ into subspace $\mathbf{R}^q$ by $\mathbf{y}_i = \mathbf{W}^\top\mathbf{x}_i$, we can get $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q]$ through similar procedure described above, which exactly consists of the q eigenvectors corresponding to the q smallest generalized eigenvalues.

3. TWO-DIMENSIONAL LOCALITY PRESERVING PROJECTIONS

Now let us consider a set of n sample images $\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_n$ taken from an $(s\times t)$-dimensional image space. Due to the high dimensionality, it is difficult to apply conventional algorithms directly for recognition. Therefore, dimensionality reduction is especially of importance. We design a linear transformation which maps the original $(s\times t)$-dimensional image space into an s-dimensional feature space. Let $\mathbf{w}$ be a t-dimensional unitary column vector. The method proposed here is projecting each image $\mathbf{A}_i$, an $(s\times t)$ matrix, onto $\mathbf{w}$ by the following transformation:

$$\mathbf{x}_i = \mathbf{A}_i\mathbf{w},\ \ i = 1, 2, ..., n. \tag{6}$$

Then we get an s-dimensional projected feature $\mathbf{x}_i$ for each image $\mathbf{A}_i$. To achieve the highest recognition rate, it is important to select a good projection vector $\mathbf{w}$.

A reasonable criterion for choosing this mapping is to minimize the following objective function:

$$\min \sum_{i<j} s_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|^2. \tag{7}$$

The weight s_{ij} incurs a heavy penalty when neighboring points $\mathbf{A}_i$ and $\mathbf{A}_j$ are mapped far apart. Therefore, minimizing the objective function is an attempt to ensure that, if $\mathbf{A}_i$ and $\mathbf{A}_j$ are "close", then $\mathbf{x}_i$ and $\mathbf{x}_j$ are close as well. By simple algebra operation, we see that

$$\sum_{i<j} s_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|^2 \ = \ \mathbf{w}^\top\mathbf{P}^\top(\mathbf{L}\otimes\mathbf{I}_s)\mathbf{P}\mathbf{w}, \tag{8}$$

where $\mathbf{P} = [\mathbf{A}_1^\top, \mathbf{A}_2^\top, ..., \mathbf{A}_n^\top]^\top$ is an $(sn\times t)$ matrix generated by arranging all the image matrices in column. Operator $\otimes$ is the Kronecker product of matrices. $\mathbf{I}_s$ is the identity matrix of order s. Furthermore, to remove an arbitrary scaling factor and translation in the embedding, we impose a constraint as the following:

$$\sum_i d_{ii}\mathbf{x}_i^\top\mathbf{x}_i = 1 \Rightarrow \mathbf{w}^\top\mathbf{P}^\top(\mathbf{D}\otimes\mathbf{I}_s)\mathbf{P}\mathbf{w} = 1. \tag{9}$$

Now the minimization problem is reduced to be:

$$\arg\min_{\mathbf{w}^\top\mathbf{P}^\top(\mathbf{D}\otimes\mathbf{I}_s)\mathbf{P}\mathbf{w}=1} \mathbf{w}^\top\mathbf{P}^\top(\mathbf{L}\otimes\mathbf{I}_s)\mathbf{P}\mathbf{w}. \tag{10}$$

The transformation vector $\mathbf{w}$ that minimizes the objective function is given by the minimum eigenvector solution to the generalized eigenvalue problem:

$$\mathbf{P}^\top(\mathbf{L}\otimes\mathbf{I}_s)\mathbf{P}\mathbf{w} = \lambda\mathbf{P}^\top(\mathbf{D}\otimes\mathbf{I}_s)\mathbf{P}\mathbf{w}. \tag{11}$$

Note that the matrices $\mathbf{P}^\top(\mathbf{L}\otimes\mathbf{I}_s)\mathbf{P}$ and $\mathbf{P}^\top(\mathbf{D}\otimes\mathbf{I}_s)\mathbf{P}$ are both symmetric and positive semidefinite. And the vectors $\mathbf{w}_i$ that minimize the objective function are the minimum eigenvector solutions to the generalized eigenvalue problem.

4. BILATERAL TWO-DIMENSIONAL LOCALITY PRESERVING PROJECTIONS

In image recognition, LPP needs vectorizing all the images, which results in singularity of matrix. 2DLPP operates directly on image matrices. In fact, it performs compression only in row direction. Similarly, we can operate alternative 2DLPP which operates in column direction of image matrices. In this section, we introduce bilateral two-dimensional locality preserving projections (B2DLPP) which performs dimensionality reduction both in row and in column direction of image matrices.

Let there be n training images $\mathbf{A}_i$ $(i = 1, ..., n)$ all sharing $(s\times t)$ size. We wish to perform the following projection:

$$\mathbf{X}_i = \mathbf{U}^\top\mathbf{A}_i\mathbf{V},\ \ i = 1, 2, ..., n. \tag{12}$$

where $\mathbf{U}\in\mathbf{R}^{s\times l}\,(l < s)$ and $\mathbf{V}\in\mathbf{R}^{t\times r}\,(r < t)$ are left- and right-projection matrix. We get $(l\times r)$ feature matrices $\mathbf{X}_i$ $(i = 1, ..., n)$. Let $\mathbf{S}$ be the $(n\times n)$ neighborhood similarity matrix of original images $\mathbf{A}_i$ $(i = 1, ..., n)$. We wish that the projection preserves neighborhood relationship, i.e., each two neighboring points in original image space should be in neighborhood after the projection. A proper optimal criterion can be set as minimizing the following objective function under some constraint:

$$\min_{\mathbf{U},\mathbf{V}} F(\mathbf{U},\mathbf{V}) = \min_{\mathbf{U},\mathbf{V}} \sum_{i<j} \|\mathbf{X}_i - \mathbf{X}_j\|_F^2 s_{ij}. \quad (13)$$

where $\|\star\|_F^2$ is square of Frobenius norm, i.e., sum of square of all elements of a matrix. The weight s_{ij} incurs a heavy penalty when neighboring points $\mathbf{A}_i$ and $\mathbf{A}_j$ are mapped far apart. Therefore, minimizing the objective function is an attempt to ensure that, if $\mathbf{A}_i$ and $\mathbf{A}_j$ are "close", then $\mathbf{X}_i$ and $\mathbf{X}_j$ are close as well. By simple algebra operation, we can see

$$\begin{aligned} F(\mathbf{U},\mathbf{V}) &= tr[\mathbf{U}^\top \mathbf{Q}(\mathbf{L} \otimes \mathbf{V}\mathbf{V}^\top)\mathbf{Q}^\top \mathbf{U}] \\ &= tr[\mathbf{V}^\top \mathbf{P}^\top(\mathbf{L} \otimes \mathbf{U}\mathbf{U}^\top)\mathbf{P}\mathbf{V}]. \quad (14) \end{aligned}$$

where d_{ii}, the diagonal element of $\mathbf{D}$, satisfy $d_{ii} = \sum_j s_{ij}$. $\mathbf{L} = \mathbf{D} - \mathbf{S}$. $\mathbf{Q} = [\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_n]$ is an $(s \times tn)$ matrix generated by concatenating all the image matrices in row direction, and $\mathbf{P} = [\mathbf{A}_1^\top, \mathbf{A}_2^\top, ..., \mathbf{A}_n^\top]^\top$ is an $(sn \times t)$ matrix generated by concatenating all the image matrices in column direction. Operator $\otimes$ is the Kronecker product of matrices. $tr[\star]$ is trace of square matrix.

Matrix $\mathbf{D}$ provides a natural measure for original data points. The larger d_{ii} (corresponding to the ith image $\mathbf{A}_i$) is, the more important $\mathbf{X}_i$ is after the projection. Therefore, to eliminate the arbitrary scalability and translation, it's proper to impose a constraint on the optimization problem:

$$G(\mathbf{U},\mathbf{V}) = \sum_i d_{ii}\|\mathbf{X}_i\|_F^2 = 1 \quad (15)$$

$$\Rightarrow \quad tr[\mathbf{U}^\top \mathbf{Q}(\mathbf{D} \otimes \mathbf{V}\mathbf{V}^\top)\mathbf{Q}^\top \mathbf{U}] = 1 \quad (16)$$

$$\Rightarrow \quad tr[\mathbf{V}^\top \mathbf{P}^\top(\mathbf{D} \otimes \mathbf{U}\mathbf{U}^\top)\mathbf{P}\mathbf{V}] = 1. \quad (17)$$

Now the optimization criterion turns into solving the following minimization problem:

$$\arg\min_{\mathbf{U},\mathbf{V}} \frac{tr[\mathbf{V}^\top \mathbf{P}^\top(\mathbf{L} \otimes \mathbf{U}\mathbf{U}^\top)\mathbf{P}\mathbf{V}]}{tr[\mathbf{V}^\top \mathbf{P}^\top(\mathbf{D} \otimes \mathbf{U}\mathbf{U}^\top)\mathbf{P}\mathbf{V}]} \quad (18)$$

or

$$\arg\min_{\mathbf{U},\mathbf{V}} \frac{tr[\mathbf{U}^\top \mathbf{Q}(\mathbf{L} \otimes \mathbf{V}\mathbf{V}^\top)\mathbf{Q}^\top \mathbf{U}]}{tr[\mathbf{U}^\top \mathbf{Q}(\mathbf{D} \otimes \mathbf{V}\mathbf{V}^\top)\mathbf{Q}^\top \mathbf{U}]}. \quad (19)$$

Then each column of projection matrices $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_r]$ and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_l]$ are the eigenvectors corresponding to the r and l smallest eigenvalues of the following generalized eigenvalue problems:

$$\begin{aligned} \mathbf{P}^\top(\mathbf{L} \otimes \mathbf{U}\mathbf{U}^\top)\mathbf{P}\mathbf{v} &= \lambda \mathbf{P}^\top(\mathbf{D} \otimes \mathbf{U}\mathbf{U}^\top)\mathbf{P}\mathbf{v} \quad (20) \\ \mathbf{Q}(\mathbf{L} \otimes \mathbf{V}\mathbf{V}^\top)\mathbf{Q}^\top \mathbf{u} &= \gamma \mathbf{Q}(\mathbf{D} \otimes \mathbf{V}\mathbf{V}^\top)\mathbf{Q}^\top \mathbf{u}. \quad (21) \end{aligned}$$

The above two equations depend on each other. $\mathbf{U}$ and $\mathbf{V}$ can't be solved separately. We adopt an iterative procedure to solve them. First let $\mathbf{U}_0 = \mathbf{I}_s$, which is identity matrix, substitute it in equation (20), solve the eigenvectors corresponding to the r smallest eigenvalues and construct $\mathbf{V}_1$, then substitute $\mathbf{V}_1$ in equation (21), solve the eigenvectors corresponding to the l smallest eigenvalues and construct $\mathbf{U}_1$. Perform the two steps iteratively till convergence. The proof of the convergence property can be easily done and is omitted here due to the space limited. Experiments show this procedure converges in three iterations.

Now the entire procedure of B2DLPP algorithm is summarized as follows:

1. **Constructing nearest-neighbor graph:** Let $\mathcal{G}$ denote a graph with n nodes, ith node corresponding to data point $\mathbf{A}_i$. We put an edge between nodes i and j if $\mathbf{A}_i$ and $\mathbf{A}_j$ are "close". There are several methods to measure "close". Here are two:
   (a) k-nearest neighbors. Nodes i and j are connected by an edge if i is among k nearest neighbors of j or j is among k nearest neighbors of i.
   (b) ε-neighborhoods. Nodes i and j are connected if $\|\mathbf{A}_i - \mathbf{A}_j\|_F < \varepsilon$.

   Note: Here we can add label information of training samples (if there is) to improve discriminant performance, which can be done by restricting k-nearest neighbors of each data point to be from the same class.
2. **Choosing the weights:** If there is an edge between nodes i and j, put a similarity weight s_{ij} on it, otherwise zero, and we get a sparse symmetric $(n \times n)$ similarity matrix $\mathbf{S}$. The similarity weight s_{ij} can be:
   (a) Simple-minded. $s_{ij} = 1$ if and only if nodes i and j are linked by an edge.
   (b) Heat kernel. If nodes i and j are linked, put $s_{ij} = \exp\{-\frac{\|\mathbf{A}_i - \mathbf{A}_j\|_F^2}{c}\}$, where c is a suitable constant.
3. **Computing left- and right-projection matrices iteratively:** Let $\mathbf{U}_0 = \mathbf{I}_s$, solve equations (20) and (21) iteratively till convergence, the eigenvectors corresponding to the r and l smallest eigenvalues construct left- and right-projection matrices $\mathbf{U}$ and $\mathbf{V}$.

Once left- and right-projection matrices $\mathbf{U}$ and $\mathbf{V}$ are obtained, we can get the feature matrices of all training images in low dimensional space. When there is a new image $\mathbf{A}$ for recognition, firstly get its feature matrix: $\mathbf{X} = \mathbf{U}^\top \mathbf{A}\mathbf{V}$, then a nearest neighbor classifier can be adopted for recognition using Frobenius norm as distance between feature matrices.

5. EXPERIMENTAL RESULTS

The proposed B2DLPP and other methods were tested and compared for recognition on the ORL (AT&T) face database (http://www.uk.research.att.com/facedatabase.html). Details about the database can be found on the website. To simplify the computation of the experiments and to improve the recognition performance, we cropped each image manually to let the eyes be at the same positions for each image and resized to (64×64) pixels.

Firstly, we examined the variation of F value along iterations. It changed very little after two iterations. In fact, more

Table 1. Top recognition accuracy (%) comparisons between B2DLPP and other methods on ORL

# training images/person	2	3	4	5	6
Eigenfaces	73.1 (72)	76.4 (80)	85.8 (110)	88.0 (70)	88.8 (100)
Laplacianfaces	58.8 (70)	57.5 (90)	69.2 (120)	70.5 (100)	75.6 (110)
2DLPP + PCA	80.3 (50)	81.1 (60)	90.0 (80)	92.0 (60)	93.8 (100)
2DPCA	75.0 (64x10)	79.0 (64x10)	88.0 (64x8)	90.0 (64x4)	91.0 (64x14)
2DLDA	76.3 (64x7)	85.4 (64x5)	91.3 (64x5)	93.5 (64x5)	94.4 (64x5)
2DLPP	81.9 (64x4)	85.0 (64x5)	92.5 (64x5)	93.0 (64x3)	93.8 (64x4)
B2DLPP	72.2 (10x10)	84.3 (9x9)	93.8 (9x9)	95.5 (11x11)	95.6 (12x12)

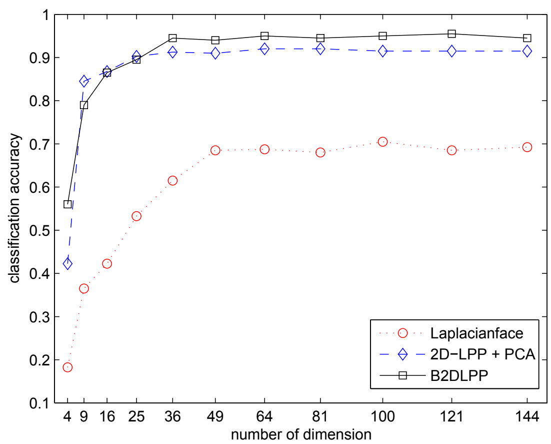

Fig. 1. Recognition comparison of B2DLPP on ORL

analysis in the iteration, we can see that $\mathbf{V}_1$ is exactly the projection matrix of 2DLPP. The following iteration let F smaller than that of 2DLPP, i.e., B2DLPP can obtain smaller objective function value than 2DLPP with fewer coefficients. In the following experiments, we just let the iteration run twice. And for simplicity, we chose the same number of eigenvectors in the two eigenvalue problems in B2DLPP, i.e., $r = l$.

Now we take five images per person for training. We test the recognition performance of B2DLPP with Laplacianfaces (PCA+LPP) and 2DLPP+PCA, which is doing a further PCA step on concatenated features obtained by 2DLPP. The rest of images are used for testing. In this part of experiment, we choose $l = 2, 3, ..., 12$. Then the dimensions of features are square of them. Fig.1 shows our results. B2DLPP and 2DLPP+PCA are generally the same although B2DLPP show a little superiority. Both of them are better than Laplacianfaces. Due to lack of training data, the performance of Laplacianfaces is bad (even worse than PCA, to see the following experiment).

Then we compare B2DLPP with more algorithms related. We take a look at the coefficients needed by each algorithm when obtaining top recognition accuracy. We choose k ($k = 2, ..., 6$) images per person for training. The rest images construct test set. All algorithms are trained, and nearest neighbor classifier is adopted for recognition. Compute recognition accuracy under different feature dimension, and write down the top recognition accuracy and the corresponding dimension. Table 1 lists the top recognition accuracy and the corresponding dimension of B2DLPP and other methods using different number of training images per person. From the table, we can see two-dimensional methods (2DPCA, 2DLDA and 2DLPP) outperform one-dimensional methods (Eigenfaces and Laplacianfaces), but two-dimensional methods need too more coefficients for feature representation. 2DLPP+PCA is worse than 2DLPP though it needs fewer coefficients. B2DLPP obtains higher recognition accuracy than two-dimensional methods while needs only the number of coefficients as that of one-dimensional methods. B2DLPP outperforms 2DLPP+PCA when number of training images per person is enough (only more than two). It should be noticed that 2DLDA+2DPCA only obtained 93.5% accuracy when five images per person were used in [5].

6. CONCLUSION

Based directly on image matrices, we proposed two new methods for dimensionality reduction: two-dimensional locality preserving projections (2DLPP) and bilateral two-dimensional locality preserving projections (B2DLPP). 2DLPP performs matrix compression in row direction while B2DLPP performs compression both in row and in column direction. Experiments on the ORL face database showed that B2DLPP outperforms other methods and requires fewer coefficients.

7. REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Anal. & Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[3] 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.*, vol. 26, no. 1, pp. 131–137, 2004.

[4] X. F. He, S. Yan, Y. Hu, P. Niyogi, and H. J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. & Mach. Intell.*, vol. 27, no. 3, pp. 328–340, 2005.

[5] P. Sanguansat, W. Asdornwised, S. Jitapunkul, and S. Marukatat, "Two-dimensional linear discriminant analysis of principle component vectors for face recognition," in *ICASSP*. IEEE, 2006, vol. 2, pp. 345–348.