3-1

MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

A Novel Approach to Eigenpalm Features using Feature-Partitioning Framework

Kadappagari Vijaya Kumar
Department of Computer Applications
Vasavi College of Engineering Hyderabad India
e-mail:kvkumar28@yahoo.com

Atul Negi
Department of CIS, University of Hyderabad
Hyderabad India
e-mail address:atulcs@uohyd.ernet.in

Abstract

Eigenpalms, a well-known approach, extracts features from palmprint images using conventional PCA technique. However eigenpalms does not exploit neighbourhood (local) information due to its vector representation of palmprint images. In our work here, we propose a feature-partitioning framework that uses a more efficient and appropriate matrix representation of images. Our novel feature partitioning approach shows a considerably better and consistent recognition performance than eigenpalms approach (PCA).

1. Introduction

Palmprint is one of the relatively new biometric technologies, which has unique and stable characteristics. Palmprint recognition is a powerful biometric for person identification. Palmprint biometric recognizes a person based on principal lines, wrinkles and ridges, which are stable throughout life of a person. Moreover, no two persons have the same palmprints. Unlike hand geometry, which measures hand-size and finger-length, palmprint focuses on the inner surface of a hand, its pattern of lines and the shape of its surface. Palmprint provides a much larger recognizable surface than a fingertip [7][10]. Therefore, palmprint verification is one of promising technologies for security access control system. An important issue in palmprint recognition is to extract salient palmprint features, which distinguishes one person from other. One of the popular approaches for feature extraction and dimensionality reduction is Principal Component Analysis (PCA) and is widely used for palmprint recognition.

Principal component analysis is one of the widely used techniques for feature extraction and recognition. PCA is also known as K-L expansion and a detailed discussion may be found in [1][2]. The usefulness and hence popularity of PCA comes from its properties -- it is an optimal linear scheme (in terms of mean squarred error) for reducing data to a lower dimensionality and uses only matrix multiplication operations for reduction and reconstruction. Palmprint recognition methods [6][7][8][9] rely upon classical PCA to be applied on palmprint images. Such approaches involve treating an $m \times n$ sized gray image input as an mn sized feature vector. Clearly the complexity for PCA computation is enormous and also the feature redundancy is high. A considerable improvement is obtained by structuring the input image to a more appropriate matrix representation. The improvements are well documented and reported in [4][5]. More recently, an interesting approach to extract local features in multidimensional data using feature partitioning approach was presented [3]. Here we exploit the advantages of feature partitioning concept and matrix representation, to propose a hybrid approach, FP-IPCA (Feature Partitioning based PCA for Image data).

The organization of the paper is as follows. We present a brief review of a more efficient implementation of classical PCA in the section 2. In section 3, we present our proposed feature partitioning approach, FP-IPCA in detail. Experimental results on palmprint recognition are discussed in section 4. We present concluding remarks in the section 5.

2. Review of Time Efficient Principal Component Analysis

Here we review the time efficient approach to traditional PCA [6] in brief. Consider a set of N palmprint images, $\mathbf{A}=\{\mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_N}\}$. Each image is represented by an $m \times n$ matrix of image features (pixels) and is finally represented by a vector of mn features. Now form an $N \times (mn)$ matrix, $\mathbf{X}$, where rows represent corresponding mn sized images. We know that classical PCA requires computation of an $mn \times mn$ covariance matrix, **CM**, which is quite expensive. They [6] proposed a more efficient method to compute eigenvectors and eigenvalues without computing **CM** as follows. Since N is usually smaller than mn, compute $N \times N$ matrix **EC**, ($=(1/N)\mathbf{XX^T}$), then find eigenvectors ($\mathbf{v_i}$) and eigenvalues (λ_i) of **EC**. Next compute eigenvectors of original covariance matrix (**CM**), $\mathbf{u_i}$, using $\mathbf{u_i}=\mathbf{X^T v_i}$ and λ_i is its corresponding eigenvalue. Using this approach we get eigenvectors of **CM** upto N. Finally project the image data onto selected eigenvectors, $\{\mathbf{u_i}\}$, to get the reduced image.

We explain this method in order to set the background for the experimental comparison of section 4.

3. Feature Partitioning Framework for PCA of Image Data (FP-IPCA) - A Novel Approach in Feature Extraction

In this section, we discuss our feature partitioning approach in detail. The proposed approach is based on feature partitioning concept and exploitation of matrix structure of images. The approach can be better understood using figure 2.

As a first step, we *partition* the i^{th} image $(\mathbf{A_i})_{m\times n}$, into k, ($k\geq 2$), sub-images, $\{\mathbf{A_i^j}\}$; $j=1...k$, each of size $m\times d$, where $d=n/k$. Form $\mathbf{S_j}$, the set of j^{th} sub-images of images, $\{\mathbf{A_i}\}$, $i=1...N$, given by $\mathbf{S_j}=\{\mathbf{A_1^j}, \mathbf{A_2^j}, ..., \mathbf{A_N^j}\}$. For each sub-image set, $\mathbf{S_j}$, $j=1...k$: Compute sub-image covariance matrix as given by,

$$\mathbf{M^j}_{d\times d}=(1/N)\sum_{i=1}^{N}(\mathbf{A_i^j}-\boldsymbol{\mu}^\mathbf{j})^T(\mathbf{A_i^j}-\boldsymbol{\mu}^\mathbf{j}),$$

Where $\boldsymbol{\mu}^\mathbf{j}$ is the mean matrix of all training sub-image matrices of $\mathbf{S_j}$. Next, we perform *local feature extraction* in sub-images by computing q orthonormal eigenvectors ($\mathbf{e_i}$) corresponding to first q largest eigenvalues (λ_i) satisfying $\mathbf{M^j e_i} = \lambda_i \mathbf{e_i}$ and project j^{th} sub-image data set, $\mathbf{S_j}$, onto these q eigenvectors which gives locally reduced sub-image set, $\mathbf{I_j}$. $\mathbf{I_j}$ is given by $\mathbf{I_j} = \{\mathbf{B_i^j} / \mathbf{B_i^j} = (\mathbf{A_i^j})_{m\times d} (\mathbf{E^j})_{d\times q}\}$, where $\mathbf{E^j}$ represents q d-dimensional eigenvectors of $\mathbf{M^j}$. Next we *collate* all locally reduced k sub-images, $\{\mathbf{B_i^1}, \mathbf{B_i^2}, ..., \mathbf{B_i^k}\}$(one from each $\mathbf{I_j}$, $j=1...k$), corresponding to the image, $\mathbf{A_i}$, to form locally reduced image, $(\mathbf{B_i})_{m\times kq}$, $i=1...N$. (See steps (a) – (g) in figure 2)

As a second step, we use inter-subimage correlations for *global feature extraction* of locally reduced images $\{\mathbf{B_i}\}$. We compute image covariance matrix, $(\mathbf{M^F})_{kq\times kq}$ of $\{\mathbf{B_i}\}$. Find r orthonormal eigenvectors corresponding to first r largest eigenvalues of $\mathbf{M^F}$ and project $\{\mathbf{B_i}\}$ onto each of them to give globally reduced data $\{\mathbf{C_i}\}$, $i=1...N$, which is corresponding to original image set, $\{\mathbf{A_i}\}$, $i=1...N$.. An i^{th} reduced image, $\mathbf{C_i}$ is given by $(\mathbf{C_i})_{m\times r} = (\mathbf{B_i})_{m\times kq} (\mathbf{E^F})_{kq\times r}$, where $\mathbf{E^F}$ is set of r kq-dimensional eigenvectors (See steps (h) – (i) in figure 2).

As a last step, we treat each reduced $m\times r$ image, $\mathbf{C_i}$, as a (mr)-dimensional *feature vector* for recognition tasks.

In a nutshell, we first divide each image into sub-images and extracts local features from each of the sub-images, then further extract features among locally reduced images using missing correlations among them.

4. Experimental Results

In this section, we report our experimental results based upon the benchmarking approach as suggested in [8]. We explain the experiments conducted using our implementation and compare the results of PCA and FP-IPCA. We considered PolyU palmprint data [10] for our experiments and we summarize the results in the following.

4.1. PolyU palmprint data set

The PolyU palmprint database [10] contains *7752* grayscale images corresponding to *386* different persons in BMP image format. Around twenty samples from each of these palms were collected in two sessions, where around *10* samples were captured in the first session and the second session, respectively. The average interval between the first and the second collection was two months. The palmprint images in the database belong to subjects ranging from *1* to *386*. Out of *386* subjects, we have chosen all the images from first *336* subjects, a total of 6746 palmprints. We converted palmprint images to PGM format, where size of each image is 284×384 and are used for our experiments.

4.2. Experimental setup

We have chosen training, clients (for enrollment and testing) and impostor data sets from different subjects without overlapping as described in [8]. First, we find eigenvectors and eigenvalues using training data, and then client and impostor data sets are projected on selected eigenvectors to get the data in reduced form. For each reduced client testing and impostor testing data follow the steps: (i) Find the Euclidean distance of test image to every client template (i.e. enrolled client), (ii) Next, find minimum Euclidean distance for the test image among distances found in step (i). The steps (i)-(ii) are repeated for every test image. Next, normalize all the minimum Euclidean distances of test images found in the previous steps with respect to maximum of minimum distances. Find the similarity values of test images by subtracting normalized minimum distances from *one*. For each test image: *accept* it, if its similarity value is greater than some threshold, $\delta \in (0,1)$; otherwise *reject* it. We considered different threshold values.

False Rejection Ratio (FRR), False Acceptance Ratio (FAR) and Recognition Rates are calculated using the formulas: FAR = Number of Impostor data accepted/ Number of impostor testing data or attempts; FRR = Number of client data rejected/Number of client testing data or attempts; Recognition Rate = (Number of client test data accepted + Number of impostor test data rejected)/Total number of client and impostor test data.

We conducted experiments by varying number of projection vectors (eigenvectors). We have divided each image into *8* sub-images (i.e. $k=8$) in FP-IMPCA and number of projection vectors per sub-image set is taken as *1*. For PCA, k is taken as *1*. We considered *120* palmprints from first 6 subjects for training, which are used to find eigenvectors (principal components). Next *12* subjects are used as clients (legal users). Further, palmprints of each client subject are divided as follows: first *12* palmprints are used as templates (enrollment data); remaining 8 are used for client testing. Thus a total of *144* palmprints are enrolled as templates and a total of *96* palmprints are used for client testing. The last *318* subjects (a total of *6386* palmprints) are used as impostors (illegal users). FAR, FRR and Recognition rates using FP-IPCA and PCA are shown in tables 1-2. The recognition rates of FP-IPCA and PCA are plotted in figure 1.

Classical PCA and FP-IPCA are implemented in C language using the procedures described in sections 2 and 3 respectively. We used some routines - *tredt, tqli, eigsrt* from [11] to find eigenvectors and eigenvalues. We used a Pentium 4 based system with a CPU clock speed of 2.4 GHz, 256MB RAM and Fedora Core 5 Linux running on it.

4.3. Discussion of results

The experimental results shown in tables 1-2 reveal that partitioning based approach, FP-IPCA, outperforms an efficient implementation of PCA [6] in terms of recognition rates. We believe that excellent recognition rates are attributed due to partitioning concept and image structure consideration. For PolyU palmprint data, proposed partitioning approach shows predominant recognition rates (93.9%) than PCA (73.5%) and shows lower error rates (FAR: 6.0%, FRR: 12.5%) in comparison to PCA (FAR: 26.4%, FRR: 37.5%), for 8 projection vectors. By observing tables 1-2 and figure 1, we conclude the superiority of FP-IPCA over PCA, for other projection vectors as well. How can the effectiveness of a feature be evaluated? Here we know from text book (section 3.7.1 on accuracy, dimension and training sample size) [2] that an efficient feature reduces the error. This can be seen from the fact that while a single Projection Vector (PV) was sufficient to produce an FAR of *19.4%,* this same (or lower) FAR value could not be achieved even with *8* PVs from PCA. We observed from figure 1 that FP-IPCA shows steady growth and PCA shows oscillations of recognition rate

Table 1. Recognition and error rates of FP-IPCA

PVs	FAR	FRR	Recognition Rate (%)
1	19.4	19.8	80.6
2	10.9	18.8	89.0
3	8.3	14.6	91.6
4	10.7	12.5	89.3
5	8.1	12.5	91.9
6	7.9	11.5	92.1
7	6.0	12.5	93.9
8	6.0	12.5	93.9

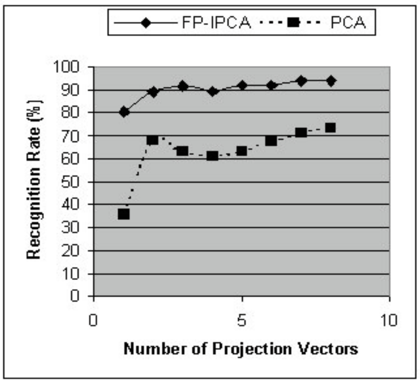
Table 2. Recognition rates using PCA approach

PVs	FAR	FRR	Recognition Rate (%)
1	65.0	19.8	35.7
2	31.7	45.8	68.1
3	36.8	40.6	63.2
4	39.2	27.1	61.0
5	36.8	27.1	63.3
6	32.7	33.3	67.3
7	29.0	34.4	71.0
8	26.4	37.5	73.5

5. Conclusions

In this paper, we have proposed a novel feature extraction approach, FP-IPCA, based on feature partitioning framework and matrix structure of image, and its application to biometrics (palmprint recognition). Feature partitioning framework provides an excellent framework for feature extraction and recognition of palmprints. Experimental results reveal that FP-IPCA outperforms more efficient implementation of PCA (i.e. eigenpalms) in terms of recognition. FP-IPCA extracts local features and then combines locally extracted features globally. We demonstrated the applicability of FP-IPCA technique to 6746 PolyU palmprint images. FP-IPCA may be extensively used in designing other biometric systems such as face recognition, fingerprint recognition, etc.

Figure 1. Recognition performance of FP-IPCA vs. PCA



References

[1] I. T. Jolliffe: “Principal component analysis,” *Springer-Verelag, New York*, 2002.

[2] Richard O, Duda, Peter E. Hart and David G. Stork: “Pattern Classification, Second Edition,” *John Wiley and sons (Asia) Pte Ltd, Singapore*, 2001.

[3] Atul Negi, Vijaya Kumar Kadappa: “An experimental study of sub-pattern based principal component analysis and Cross-subpattern correlation based principal component analysis,” *Proceedings of Image and Vision Computing 2005, New Zealand*, pp.20-25, 2005.

[4] Jian Yang, Jing-Yu Yang: “From image vector to matrix: a straightforward image projection technique – IMPCA vs. PCA,” *Pattern Recognition*, vol.35, pp.1997-1999, 2002.

[5] Benito M, Pena D: “A fast approach for dimensionality reduction with image data,” *Pattern Recognition*, vol.38, pp.2400-2408, 2005.

[6] Guangming Lu, David Zhang, Kuanquan Wang: “Palmprint Recognition using eigenpalms features,” *Pattern Recognition Letters*, vol.25, Issue.9-10, pp.1463-1467, 2003.

[7] Tee Connee, Andrew Teoh, Michael Goh, David Ngo: “Palmprint Recognition using PCA and ICA,” *Proceedings of image and Vision Computing, NZ*, pp.227-232, 2003.

[8] Slobadan Riboric, I.F: “A biometric identification system based on eigenpalm and eigenfinger features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.27, no.11, pp.1698-1709, 2005.

[9] M. Turk and A. Pentland: “Face Recognition using eigenfaces,” *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pp.586-591, 1991.

[10] PolyU Palmprint Database, Second Edition: http:// www.comp.polyu.edu.hk/~biometrics/.

[11] William H. Press, Saul A. Teukolsky and William T. Vetterling: “Numerical recipes in C, Second Edition,” *Cambridge University Press, New Delhi*, 2002.

Palmprint images (gray-scale)

Grayscale matrices (A_i) (m=4, n=6)

Partitioning (k=2) (a)

Sub-images $A_i = \{A_i^1, A_i^2\}$ (b)

Sub-image-sets $\{S_1, S_2\}$ (c)

Extract Local features (q=2) (d)

Reduced Sub-images $\{I_1, I_2\}$ (e)

Collate reduced sub-images (f)

Reduced images $B_i = [B_i^1\ B_i^2]$ (g)

Extract global features among reduced images (r=2) (h)

Finally reduced images (C_i) (i)

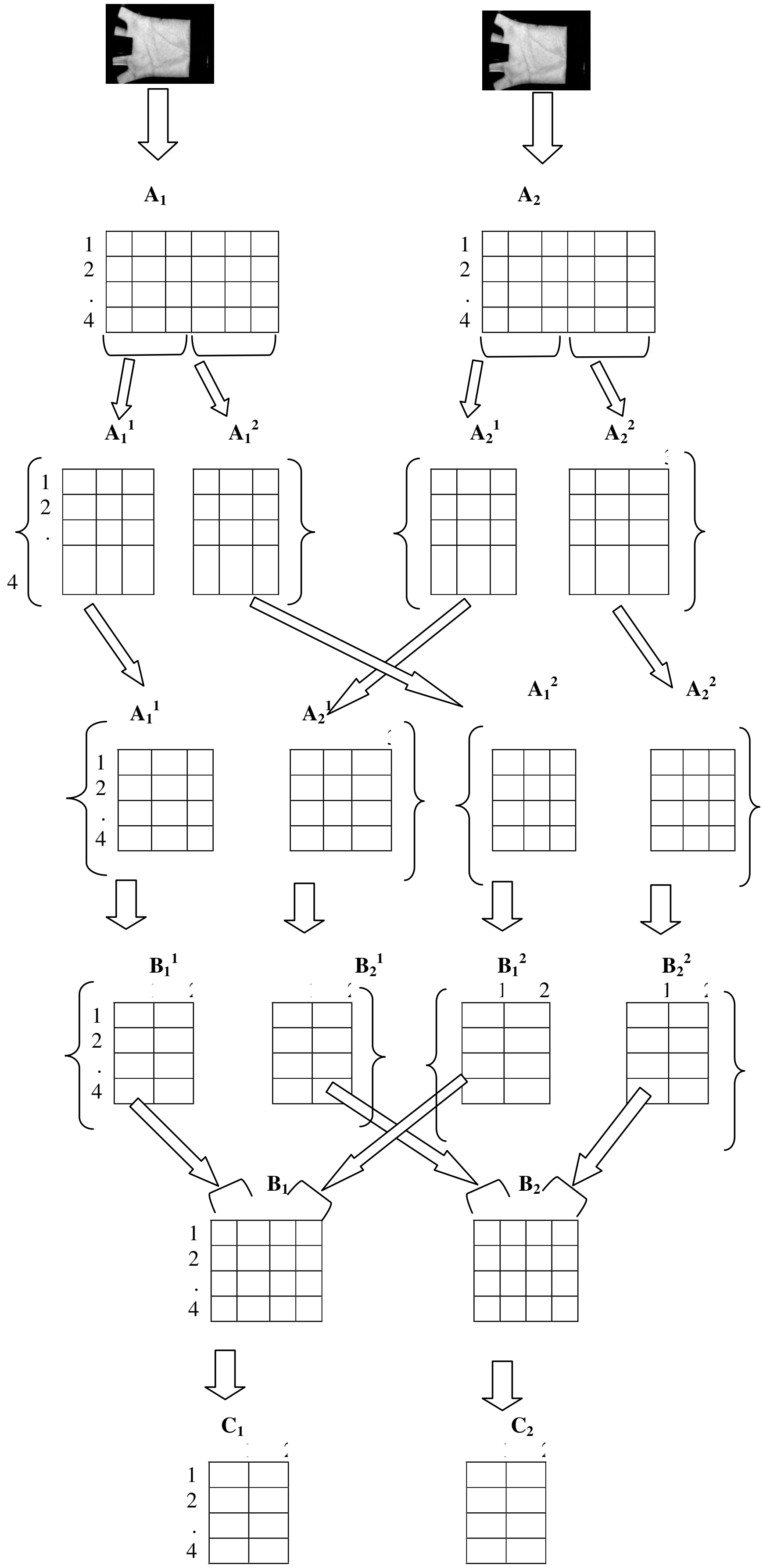


Figure 2. FP-IPCA approach