

Article

A Kernel Gabor-Based Weighted Region Covariance Matrix for Face Recognition

Huafeng Qin ^{1,*}, Lan Qin ¹, Lian Xue ¹ and Yantao Li ²

¹ Key Laboratory of Optoelectronic Technology and Systems of Ministry of Education, College of Opto-Electronic Engineering, Chongqing University, Chongqing 400030, China; E-Mails: qinlan@cqu.edu.cn (L.Q.); xuelian@cqu.edu.cn (L.X.)

² College of Computer Science, Chongqing University, Chongqing 400030, China; E-Mail: yantaoli@foxmail.com

* Author to whom correspondence should be addressed; E-Mail: N1107450k@ntu.edu.sg; Tel.: +86-65-8455-5456.

Received: 6 May 2012; in revised form: 16 May 2012 / Accepted: 17 May 2012 /

Published: 31 May 2012

Abstract: This paper proposes a novel image region descriptor for face recognition, named kernel Gabor-based weighted region covariance matrix (KGWRCM). As different parts are different effectual in characterizing and recognizing faces, we construct a weighting matrix by computing the similarity of each pixel within a face sample to emphasize features. We then incorporate the weighting matrices into a region covariance matrix, named weighted region covariance matrix (WRCM), to obtain the discriminative features of faces for recognition. Finally, to further preserve discriminative features in higher dimensional space, we develop the kernel Gabor-based weighted region covariance matrix (KGWRCM). Experimental results show that the KGWRCM outperforms other algorithms including the kernel Gabor-based region covariance matrix (KGCRM).

Keywords: face recognition; Gabor features; weighted region covariance matrix; kernalization

1. Introduction

Feature extraction from images or image regions is a key step for image recognition and video analysis problems. Recently, matrix-based feature representations [1–6] have been developed and

employed for feature extraction. Tuzel *et al.* [3] introduced the region covariance matrix (RCM) as a new image region descriptor and have applied it to object detection and texture classification. RCM is a covariance matrix of basic features extracted from a region. The diagonal entries of the covariance matrix represent the variance of each feature, while the nondiagonal entries represent the respective correlations. Using RCM as region descriptor has several advantages. Firstly, RCM provides a natural fusion method because it can fuse multiple basic features without any normalization or weight operations. Secondly, RCM can be invariant to rotations. Thirdly, its computational cost does not depend on the size of the region. Due to these advantages, RCM has been employed to detect and track objects [3,5], and has achieved promising results. The RCM in [3] and [5] were constructed using the basic features including the pixel locations, color values and the norm of the first and second order derivatives. However, directly employing RCM for human face recognition cannot achieve higher recognition rates. In order to improve face recognition rates, Pang *et al.* [4] proposed the Gabor-based RCM (GRCM) method using pixel locations and Gabor features to construct region covariance. As Gabor features can carry more discriminating information, GRCM displayed better performance. Subsequently, they also proposed a kernel Gabor RCM (KGRCM) method [7] to capture the higher order statistics in the original low-dimensional space. Their experimental results have demonstrated that the KGRCM can improve the classification performance. Recently, KGRCM has been applied to object detection and tracking [6]. The nonlinear descriptor can capture nonlinear relationships within image regions due to the usage of nonlinear region covariance matrix.

However, the previous methods based on RCM consider each pixel in the training image to be contributing equally when reconstructing the RCM, *i.e.*, the contribution of each pixel is usually set to be $1/N^2$, where N is the number of pixels in a local region. However, this assumption of equal contribution does not hold in real-world applications because it is possible that different pixels in different image parts may have different discriminative powers. For example, pixels at important facial features such as eyes, mouth, and nose should be emphasized and others such as cheek and forehead should be deemphasized.

Motivated by the above-mentioned reasons, we hence propose in this paper a weighted region covariance matrix (WRCM) to explicitly exploit the different importance of each pixel of a sample. WRCM can only extract linear face features. However, by using nonlinear features it can achieve higher performance for face recognition tasks [7–9]. To further preserve nonlinear features, we develop the kernel Gabor-based weighted region covariance matrix (KGWRCM). Experimental results on the ORL Face database [10], the Yale Face database [11] and the AR database [12] show that the KGWRCM algorithm outperforms the RCM, the WRCM, the RCM with Gabor features (GRCM) [4], the KRCM with Gabor features (KGRCM) [7], and the conventional KPCA [9], Gabor + PCA [13], and Gabor +LDA [13] algorithms in terms of the recognition rate.

2. Region Covariance Matrix (RCM)

The RCM [3] is a matrix of covariance of features computed inside a region of an image. Let F be a two-dimensional image size of $h \times w$, where w and h are the height and width of the image region. The number of pixels in image region is $N = h \times w$. Define a mapping ϕ that maps a pixel (k, l) of F onto the d dimensional feature vector x_i :

$$\phi(F, k, l) = x_i \in \Phi^d \quad (1)$$

As a result there are N d -dimensional feature vectors $(x_i)_{i=1,\dots,N}$. For the intensity image, the feature mapping function ϕ is defined by pixel locations, gray values and the norm of the first and second order derivatives of the intensities with respect to k and l :

$$\phi(F, k, l) = [k \quad l \quad F(k, l) \quad |F_k| \quad |F_l| \quad |F_{kk}| \quad |F_{ll}|] \quad (2)$$

The image region can then be represented by a $d \times d$ covariance matrix of the basic feature vectors x_i :

$$C = \frac{1}{N} \sum_{i=1}^N (x_i - u)(x_i - u)^T \quad (3)$$

where μ is the mean of the feature vectors x_i :

$$u = \frac{1}{N} \sum_{i=1}^N x_i \quad (4)$$

Equation (3) can also be expressed by following equation:

$$C = \frac{1}{2} \frac{1}{N \times N} \sum_{i=1}^N \sum_{j=1}^N (x_i - x_j)(x_i - x_j)^T \quad (5)$$

The computation process is given in Appendix A.

3. Weighted Region Covariance Matrix (WRCM)

Based on the feature vectors x_i , the $d \times d$ weighted region covariance matrix of the image region is defined as follows:

$$\begin{aligned} C_W &= \frac{1}{2} \sum_{j=1}^N \sum_{i=1}^N (x_i - x_j)^2 S_{ij} \\ &= \sum_{j=1}^N \sum_{i=1}^N x_i S_{ij} (x_i)^T - \sum_{j=1}^N \sum_{i=1}^N x_i S_{ij} (x_j)^T \\ &= \sum_{i=1}^N x_i D_{ii} (x_i)^T - X S X^T \\ &= X D X^T - X S X^T \\ &= X (D - S) X^T \\ &= X L X^T \end{aligned} \quad (6)$$

where the matrix S is a similarity matrix [14], which is chosen as:

$$S_{ij} = \exp(-\|x_i - x_j\|^2 / \sigma) \quad (7)$$

with a value ranging from 0 to 1, and σ is a suitable constant. D_{ii} is a diagonal matrixes whose entries are column or row sums of S , $D_{ii} = \sum_{j=1}^N S_{ij}$. $L = D - S$ is a matrix of $N \times N$.

Comparing Equations (5) and (6), we can see that the WRCM is just the RCM if $S_{ij} = 1/N^2$, which implies that RCM is a special case of the WRCM method. However, as all the weights in RCM are $1/N^2$, RCM cannot exploit the different importance of each pixel of a sample. On the other hand, the

WRCM can assign different weights for each pixel of a sample, so it can preserve more discrimination information than RCM.

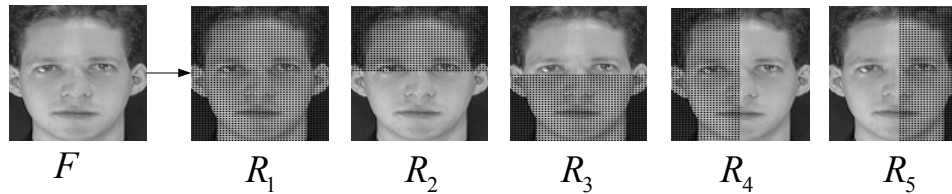
As C_W in Equation (6) is a matrix-form feature, the commonly used distances are not used. The generalized eigenvalue based distance proposed by Forstner [15] is hence used to measure the distance/dissimilarity between the WRCMs C_W^p and C_W^g :

$$D(C_W^p, C_W^g) = \sqrt{\sum_{i=1}^c \ln^2 \lambda_i(C_W^p, C_W^g)} \quad (8)$$

where $\lambda_1, \dots, \lambda_c$ are the generalized eigenvalues of covariance C_W^p and C_W^g computed from:

$$\lambda_i C_W^p v_i = C_W^g v_i \quad i = 1, \dots, c \quad (9)$$

Figure 1. Five regions of a face image. Five WRCMs are constructed from the corresponding regions.



To preserve the local and global patterns, similar to [3,4], we represent a face image with five WRCMs from five different regions (R_1, R_2, R_3, R_4 , and R_5) (Figure 1). The five WRCMs ($C_{W_1}, C_{W_2}, C_{W_3}, C_{W_4}$, and C_{W_5}) are constructed from five different regions. As C_{W_1} is the weighted region covariance matrix of the entire image region R_1 , it is a global representation of the face. The $C_{W_2}, C_{W_3}, C_{W_4}$, and C_{W_5} are extracted from four local image regions (R_2, R_3, R_4 , and R_5), so they are part-based representations of the face.

After obtaining WRCMs of each region, it is necessary to measure the distance between the gallery and probe sets. Let C_W^p and C_W^g be WRCMs from the gallery and probe sets. The distance between a gallery WRCM and a probe one is computed as follows:

$$\begin{aligned} d(C_W^p, C_W^g) &= \min_j \left[\sum_{i=1}^5 D(C_{W_i}^p, C_{W_i}^g) - D(C_{W_j}^p, C_{W_j}^g) \right] \\ &= \sum_{i=1}^5 D(C_{W_i}^p, C_{W_i}^g) - \max_j [D(C_{W_j}^p, C_{W_j}^g)] \end{aligned} \quad (10)$$

4. Kernel Weighted Region Covariance Matrix (KWRCM)

To generalize WRCM to the nonlinear case, we use a nonlinear kernel mapping $x \in \Phi^d \rightarrow \phi(x) \in \Omega$ to map the feature data $x \in \Phi^d$ into a higher dimensional subspace. Then a linear WRCM is performed to preserve intrinsic geometric structures in subspace Ω . Suppose that R_1 and R_2 are two rectangular regions in the gallery and probe set images, respectively. Let m and n be number of pixels located in regions R_1 and R_2 , respectively. $\phi(x)$ and $\phi(y)$ are the higher dimensional features extracted from regions R_1 and R_2 , where, $\phi(X) = [\phi(x_1), \phi(x_2), \dots, \phi(x_m)]$ and $\phi(Y) = [\phi(y_1), \phi(y_2), \dots, \phi(y_n)]$.

Let ϕC_W^p and ϕC_W^g be the kernel weighted region covariance matrices of regions R_1 and R_2 , respectively. ϕC_W^p and ϕC_W^g are computed as follows:

$$\begin{aligned}
 \phi C_W^p &= \sum_{j=1}^m \sum_{i=1}^m \phi(x_i) S_{ij}^* \phi^T(x_i) - \sum_{j=1}^m \sum_{i=1}^m \phi(x_i) S_{ij}^* \phi^T(x_j) \\
 &= \sum_{i=1}^m \phi(x_i) D_{ii}^* \phi^T(x_i) - \phi(X) S^* \phi^T(X) \\
 &= \phi(X) D^* \phi^T(X) - \phi(X) S^* \phi^T(X) \\
 &= \phi(X) (D^* - S^*) \phi^T(X) \\
 &= \phi(X) L^* \phi^T(X)
 \end{aligned} \tag{11}$$

where $S_{ij}^* = \exp(-\|x_i - x_j\|^2 / t)$, $D_{ii}^* = \sum_{j=1}^n S_{ij}^*$, and $L^* = D^* - S^*$:

$$\begin{aligned}
 \phi C_W^g &= \sum_{j=1}^n \sum_{i=1}^n \phi(y_i) S_{ij}^\# \phi^T(y_i) - \sum_{j=1}^n \sum_{i=1}^n \phi(y_i) S_{ij}^\# \phi^T(y_j) \\
 &= \sum_{i=1}^n \phi(y_i) D_{ii}^\# \phi^T(y_i) - \phi(Y) S^\# \phi^T(Y) \\
 &= \phi(Y) D^\# \phi^T(Y) - \phi(Y) S^\# \phi^T(Y) \\
 &= \phi(Y) (D^\# - S^\#) \phi^T(Y) \\
 &= \phi(Y) L^\# \phi^T(Y)
 \end{aligned} \tag{12}$$

where $S_{ij}^\# = \exp(-\|y_i - y_j\|^2 / t)$, $D_{ii}^\# = \sum_{j=1}^n S_{ij}^\#$, and $L^\# = D^\# - S^\#$.

Hence Equation (9) can be written as follows:

$$\phi(X) L^* \phi^T(X) V = \lambda \phi(Y) L^\# \phi^T(Y) V \tag{13}$$

As any eigenvector can be expressed by a linear combination of the elements, there exist coefficients α_i ($i = 1, 2, \dots, m$) and β_j ($j = 1, 2, \dots, n$) such that:

$$V = \sum_{i=1}^m \alpha_i \phi(x_i) + \sum_{j=1}^n \beta_j \phi(y_j) = \phi(X) \alpha + \phi(Y) \beta \tag{14}$$

where $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_m]^T$ and $\beta = [\beta_1, \beta_2, \dots, \beta_n]^T$.

Combining Equations (13) and (14), the generalized eigenvalue task in Equation (13) can be expressed in the form of block matrices:

$$\begin{aligned}
 &\begin{bmatrix} K(X, X) L^* K(X, X) & K(X, X) L^* K(X, Y) \\ K(Y, X) L^* K(X, X) & K(Y, X) L^* K(X, Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
 &= \lambda \begin{bmatrix} K(X, Y) L^\# K(Y, X) & K(X, Y) L^\# K(Y, Y) \\ K(Y, Y) L^\# K(Y, X) & K(Y, Y) L^\# K(Y, Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}
 \end{aligned} \tag{15}$$

The detailed derivation of Equation (15) is given in Appendix B.

We defined matrices U , A , and B as

$$U = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{16}$$

$$A = \begin{bmatrix} K(X, X)L^*K(X, X) & K(X, X)L^*K(X, Y) \\ K(Y, X)L^*K(X, X) & K(Y, X)L^*K(X, Y) \end{bmatrix} \quad (17)$$

$$B = \begin{bmatrix} K(Y, Y)L^\#K(Y, Y) & K(Y, Y)L^\#K(Y, X) \\ K(X, Y)L^\#K(Y, Y) & K(X, Y)L^\#K(Y, X) \end{bmatrix} \quad (18)$$

Equation (15) can be rewritten as:

$$AU = \lambda BU \quad (19)$$

When A is positive definite, the generalized eigenvalues are obtained through solving the following eigenvalue problem:

$$\lambda BA^{-1}U = U \quad (20)$$

However, in many cases, A is a singular matrix, we hence incorporate a regularization parameter $u > 0$ on both sides, respectively:

$$(A + uI)U = \lambda(B + uI)U \quad (21)$$

where I is an identity matrix. When u is large enough, $(B + uI)$ is positive definite. Equation (21) becomes a standard eigenvalue problem:

$$(A + uI)(B + uI)^{-1}U = \lambda U \quad (22)$$

Based on eigenvalues obtained by Equation (9) or Equation (22), we compute the distance between the two image regions R_1 and R_2 using Equation (8).

5. Kernel Gabor-Based Weighted Region Covariance Matrix (KGWRCM)

In Equation (2), these features such as pixel locations (k, l) , intensity values and the norm of the first and second order derivatives of the intensities with respect to k and l are effective for tracking and detecting objects. However, their discriminating ability is not strong enough for face recognition [4]. To further improve the performance, Gabor features are added to the feature space. A 2-D Gabor wavelet kernel is the product of an elliptical Gaussian envelope and a complex plane wave, defined as:

$$\psi_{u,v}(z) = \frac{\|k_{u,v}\|^2}{\sigma^2} e^{-\|k_{u,v}\|^2 \|z\|^2 / 2\sigma^2} (e^{ik_{u,v}z} - e^{-\sigma^2/2}) \quad (23)$$

where u and v define the orientation and scale of the Gabor kernels, $z = (x, y)$, $\|\bullet\|$ denotes the norm operator, and the wave vector $k_{u,v}$ is defined as follows:

$$k_{u,v} = k_v e^{i\phi_u} \quad (24)$$

where $k_v = k_{max}/f^v$ and $\phi_u = \pi u/8$. k_{max} is the maximum frequency, and f is the spacing factor between kernels in the frequency domain. We use Gabor functions with eight orientations $u = \{0, \dots, 7\}$ and five scales $v = \{0, \dots, 4\}$, making a total of 40 Gabor functions. The value of other parameters follows the setting in [16]: $\sigma = 2\pi$, $k_{max} = \frac{\pi}{2}$, $f = \sqrt{2}$. The Gabor-based features are obtained by the convolution of image F with the 40 Gabor wavelets, using the following formula:

$$g_{uv}(z) = |F(z) * \psi_{u,v}(z)| \quad (25)$$

where, $*$ denotes the convolution operator, and $|\bullet|$ is a magnitude operator.

Therefore, a feature mapping function based on Gabor features is obtain by:

$$\phi(F, k, l) = [k \quad l \quad F(k, l) \quad g_{00}(k, l), g_{01}(k, l), \dots, g_{74}(k, l)] \quad (26)$$

As the Gabor wavelet representation can capture salient visual properties such as spatial localization, orientation selectivity, and spatial frequency characteristic, Gabor-based features can carry more important information. The proposed KGWRCM method can be briefly summarized as follows:

- (1) partition a face image into five regions (R_2 , R_3 , R_4 , and R_5), and extract basic features of five regions using Equation (26).
- (2) compute two weight matrices L^* and $L^\#$ using Equations (11) and (12), and obtain four kernel matrices $K(X, X)$, $K(X, Y)$, $K(Y, X)$, and $K(Y, Y)$, using Equations (30)–(33). Based on these matrices, the matrices A and B are computed utilizing Equations (17) and (18), respectively.
- (3) with A and B , the eigenvalues are obtained by Equation (20) or Equation (22) and submitted into Equation (8) to calculate the distance.
- (4) based on the distance defined in Equation (10), the nearest neighborhood classifier is employed to performance classification.

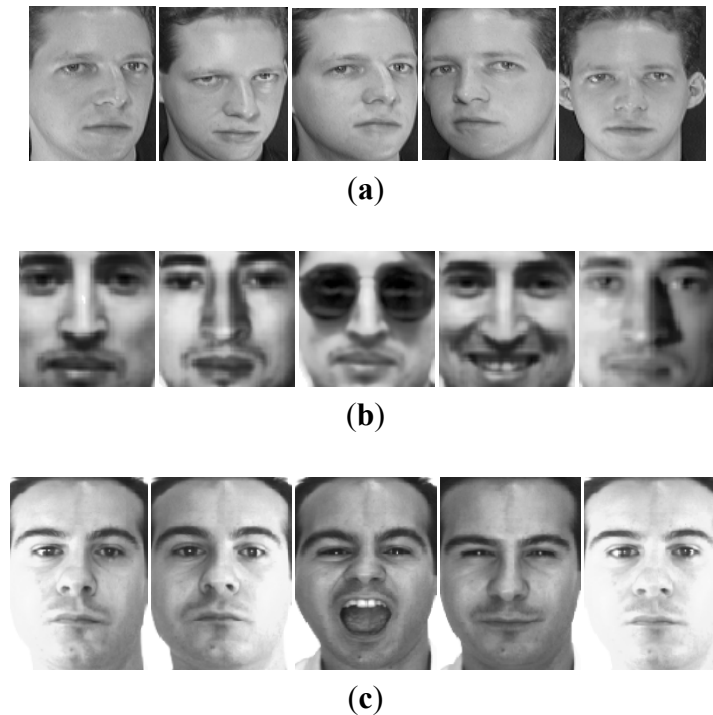
6. Experimental Results

We tested the GKWRCM algorithm on the ORL Face database [10], the Yale Face database [11] and AR Face database [12]. The ORL Face database comprises of 400 different images of 40 distinct subjects. Each subject provides 10 images that include variations in pose and scale. To reduce computational cost, each original image is resized to 56×46 by the nearest-neighbor interpolation function. A random subset with five images per individual is taken with labels to comprise the training set, and the remaining constructs the testing set. There are totally 252 different ways of selecting five images for training and five for testing. We select 20 random subsets with five images for training and five for testing.

The Yale face database contains 165 grayscale images with 11 images for each of 15 individuals. These images are subject to expression and lighting variations. In this recognition experiment, all face images with size of 80×80 were resized to 40×40 . Five images of each subject were randomly chosen for training and the remaining six images were used for testing. There are hence 462 different selection ways. We select 20 random subsets with five images for training and six for testing.

The AR database consists of over 4,000 images corresponding to 126 people's faces (70 men and 56 women). These images include more facial variations, including illumination change, and facial occlusions (sun glasses and scarf). For each individual, 26 pictures were taken in two separate sessions and each section contains 13 images. In the experiment, we chose a subset of the data set consisting of 50 male subjects and 50 female subjects with seven images for each subject. The size of images are 165×120 . We select two images for training and five for testing from the seven images. There are 21 different selection ways. Figure 2 shows some examples of the first object in each database used here.

Figure 2. Five examples of the first subject in (a) the ORL face database, (b) the Yale face database and (c) the AR face database.



In our experiment, all images are normalized to zero mean and unit variance. We compare the developed WRCM and KGWRCM with the RCM, the RCM with Gabor features GRCM [4], the KGRCM [7] and the conventional methods including KPCA [11], Gabor + PCA [12], and Gabor +LDA [12]. For GPCA and the PCA stage of KPCA, we keep 99% information to determine the number of the principal components. For the PCA stage of Gabor+LDA, we selected the number of principal components as $M - c$, where M is the number of training samples and c is the number of classes ($M = 200$ and $c = 40$ for ORL database, and $M = 55$ and $c = 11$ for Yale database, and $M = 200$ and $c = 100$ for Yale database). For KPCA, KGRCM, and the proposed KGWRCM, a Gaussian kernel function is used as kernel. We select their parameters for following algorithms with cross-validation method. Their parameters are summarized as follows: (1) parameter σ for the WRCM and KGWRCM methods. (2) the kernel parameters for the KPCA, KGRCM, and KGWRCM methods. In all these experiments, the classifiers of nearest neighborhood are employed. The performance of all methods is evaluated by the mean and standard deviation (std) of the recognition accuracies on 20 data sets for ORL and Yale databases, and the 21 data sets for AR database.

The average recognition accuracies on the ORL, Yale and AR databases are shown in Tables 1, 2 and 3, respectively. For the ORL face database and the Yale face database, the proposed KGWRCM method achieves 99.21% and 79.20% mean recognition accuracy which is higher than that of other methods. For the AR database, the proposed KGWRCM method achieves 95.95% mean recognition accuracy, which is 4.15% higher than that of KGRCM and much higher than that of other methods.

Table 1. The performance of different approaches on the ORL face database.

Method Mean	Recognition rates (%)	Standard deviations (%)
KGWRCM	99.21	1.12
KGRCM	98.41	1.24
GRCM	97.06	1.28
WRCM	93.83	2.11
RCM	91.88	2.57
GPCA	89.78	2.43
GLDA	97.5	1.37
KPCA	94.43	1.55

Table 2. The performance of different approaches on the Yale face database.

Method Mean	Recognition rates (%)	Standard deviations (%)
KGWRCM	79.20	8.72
KGRCM	76.23	9.04
GRCM	72.00	10.58
RWCM	61.67	8.76
RCM	51.94	7.22
GPCA	67.94	9.36
GLDA	73.47	7.06
KPCA	73.28	8.11

Table 3. The performance of different approaches on the AR face database.

Method Mean	Recognition rates (%)	Standard deviations (%)
KGWRCM	95.95	1.46
KGRCM	91.80	2.58
GRCM	81.46	11.73
WRCM	48.56	11.08
RCM	41.31	12.54
GPCA	78.64	5.35
GLDA	88.99	4.18
KPCA	66.89	7.68

These results clearly show that the proposed KGWRCM method can capture more discriminative information than other methods for face recognition. Particularly KGWRCM and WRCM outperform KGRCM and RCM, which implies that the weighted approaches can better emphasize more important parts in faces and deemphasize the less important parts, and also preserve discriminated information for face recognition.

7. Conclusions

In this paper, an efficient image representation method for face recognition called KGWRCM is proposed. Considering that some pixels in face image are more effectual in representing and recognizing faces, we have constructed KGWRCM based on weighted score of each pixel within a sample to duly emphasize the face features. As the weighted matrix can carry more important

information, the proposed method has shown good performance. Experimental results also show that the proposed KGWRCM method outperforms other approaches in terms of recognition accuracy. However, similar to KGRCM, the computational cost of KGWRCM is high due to the computation of the high dimensional matrix. In future work, an effective KGWRCM method with low computational complexity will be developed for face recognition.

Acknowledgments

This work is supported by the Research Fund for the Doctoral Program of Higher Education of China (NO. 20100191110011), and the Fundamental Research Funds for the Central Universities (NO. CDJXS11122220, CDJXS11121145).

References

1. Jensen, A.C.; Loog, M.; Solberg, A.H.S. Using Multiscale Spectra in Regularizing Covariance Matrices for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1851–1859.
2. Porikliand, F.; Kocak, T. Robust License Plate Detection Using Covariance Descriptor in a Neural Network Framework. In *Proceedings of 2006 IEEE International Conference on Advanced Video and Signal*, Sydney, Australia, 22–24 November 2006; pp. 107–113.
3. Tuzel, O.; Porikli, F.; Meer, P. Region Covariance: A Fast Description for Detection and Classification. In *Proceedings of 9th European Conference on Computer Vision*, Graz, Austria, 7–13 May 2006; pp. 589–600.
4. Pang, Y.W.; Yuan, Y.; Li, X.D. Gabor-Based Region Covariance Matrices for Face Recognition. *IEEE Trans. Circ. Syst. Video Technol.* **2008**, *18*, 989–993.
5. Tuzel, O.; Porikli, F.; Meer, P. Pedestrian Detection via Classification on Riemannian Manifolds. *IEEE Trans. Patt. Anal. Mach. Intell.* **2008**, *30*, 1713–1727.
6. Arif, O.; Vela, P.A. Kernel Covariance Image Region Description for Object Tracking. In *Proceedings of 16th IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, 7–10 November 2009; pp. 865–868.
7. Pang, Y.W.; Yuan, Y.; Li, X.D. Effective Feature Extraction in High-Dimensional Space. *IEEE Trans. Syst. Man Cyber. B Cybern.* **2008**, *38*, 1652–656.
8. Zhao, X.; Zhang, S. Facial Expression Recognition Based on Local Binary Patterns and Kernel Discriminant Isomap. *Sensors* **2011**, *11*, 9573–9588.
9. Scholkopf, B.; Smola, A.; Muller, K.R. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neur. Comput.* **1998**, *10*, 1299–1319.
10. The ORL Face Database. Available online: http://www.cam_orl.co.uk/ (accessed on 10 May 2012).
11. Yale University. Face Database. Available online: <http://cvc.yale.edu/projects/yalefaces/yalefaces.html> (accessed on 10 May 2012).
12. Martinez, A.M.; Kak, A.C. PCA versus LDA. *IEEE Trans. Patt. Anal. Mach. Intell.* **2001**, *23*, 228–233.

13. Liu, C.J.; Wechsler, H. Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition. *IEEE Trans. Image Process.* **2002**, *11*, 467–476.
14. He, X.F.; Yan, S.C.; Hu, Y.X.; Partha, N.Y.; Zhang, H.J. Face Recognition Using Laplacianfaces. *IEEE Trans. Patt. Anal. Mach. Intell.* **2005**, *27*, 328–340.
15. Forstner, W.; Moonen, B. *A Metric for Covariance Matrices*; Technical Report; Stuttgart University: Stuttgart, Germany, 1999.
16. Yang, X.; Zhou, Y.; Zhang, T.; Zheng, E.; Yang, J. Gabor Phase Based Gait Recognition. *Electr. Lett.* **2008**, *44*, 620–621.

Appendix A

Equation (3) can be formulated

$$\begin{aligned}
 C &= \frac{1}{N} \sum_{i=1}^N (x_i - u)(x_i - u)^T \\
 &= \frac{1}{N} \sum_{i=1}^N x_i x_i^T - \frac{1}{N^2} (Nu)(Nu)^T \\
 &= \frac{1}{N} \sum_{i=1}^N x_i x_i^T - \frac{1}{N^2} (Xe)(Xe)^T \\
 &= \frac{1}{N} \sum_{i=1}^N x_i x_i^T - X \frac{ee^T}{N^2} X^T \\
 &= X \frac{I}{N} X^T - X \frac{ee^T}{N^2} X^T \\
 &= X \left(\frac{I}{N} - \frac{ee^T}{N^2} \right) X^T
 \end{aligned} \tag{27}$$

where e is a column vector taking one at each entry and I is the identity matrix.

By some simple algebraic, Equation (5) is expressed by

$$\begin{aligned}
 C' &= \frac{1}{2} \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N (x_i - x_j)^2 \\
 &= \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N x_i x_i^T - \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N x_i x_j^T \\
 &= \frac{1}{N} \sum_{i=1}^N x_i x_i^T - X \frac{ee^T}{N^2} X^T \\
 &= X \frac{I}{N} X^T - X \frac{ee^T}{N^2} X^T \\
 &= X \left(\frac{I}{N} - \frac{ee^T}{N^2} \right) X^T
 \end{aligned} \tag{28}$$

Based on Equations (27) and (28), the following equation is obtained by

$$C = C' = \frac{1}{2} \frac{1}{N^2} \sum_{j=1}^N \sum_{i=1}^N (x_i - x_j)^2 \tag{29}$$

Appendix B

Let $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ be the kernel function, the following four kernel matrices $K(X, X)$, $K(X, Y)$, $K(Y, X)$, and $K(Y, Y)$ with sizes of $m \times m$, $m \times n$, $n \times m$, and $n \times n$ can be obtained by

$$K(X, X) = \phi(X)^T \phi(X) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_m) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_m) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_m, x_1) & k(x_m, x_2) & \cdots & k(x_m, x_m) \end{bmatrix} \quad (30)$$

$$K(X, Y) = \phi(X)^T \phi(Y) = \begin{bmatrix} k(x_1, y_1) & k(x_1, y_2) & \cdots & k(x_1, y_n) \\ k(x_2, y_1) & k(x_2, y_2) & \cdots & k(x_2, y_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_m, y_1) & k(x_m, y_2) & \cdots & k(x_m, y_n) \end{bmatrix} \quad (31)$$

$$K(Y, X) = \phi(Y)^T \phi(X) = \begin{bmatrix} k(y_1, x_1) & k(y_1, x_2) & \cdots & k(y_1, x_m) \\ k(y_2, x_1) & k(y_2, x_2) & \cdots & k(y_2, x_m) \\ \vdots & \vdots & \ddots & \vdots \\ k(y_n, x_1) & k(y_n, x_2) & \cdots & k(y_n, x_m) \end{bmatrix} \quad (32)$$

$$K(Y, Y) = \phi(Y)^T \phi(Y) = \begin{bmatrix} k(y_1, y_1) & k(y_1, y_2) & \cdots & k(y_1, y_n) \\ k(y_2, y_1) & k(y_2, y_2) & \cdots & k(y_2, y_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(y_n, y_1) & k(y_n, y_2) & \cdots & k(y_n, y_n) \end{bmatrix} \quad (33)$$

Substituting Equation (11) into Equation (10), we can obtain

$$\phi(X)L^* \phi^T(X)(\phi(X)\alpha + \phi(Y)\beta) = \lambda \phi(Y)L^\# \phi^T(Y)(\phi(X)\alpha + \phi(Y)\beta) \quad (34)$$

Based on Equations (21)–(24), Equation (25) can be expressed as

$$\phi(X)L^* K(X, X)\alpha + \phi(X)L^* K(X, Y)\beta = \lambda \phi(Y)L^\# K(Y, X)\alpha + \lambda \phi(Y)L^\# K(Y, Y)\beta \quad (35)$$

To express Equation (26) in the form of a kernel function, we multiply $\phi^T(X)$ on both sides, respectively. The following equation is obtained

$$\begin{aligned} & K(X, X)L^* K(X, X)\alpha + K(X, X)L^* K(X, Y)\beta \\ &= \lambda K(X, Y)L^\# K(Y, X)\alpha + \lambda K(X, Y)L^\# K(Y, Y)\beta \end{aligned} \quad (36)$$

Similarly, we multiply $\phi^T(Y)$ on both sides of Equation (26), and have

$$\begin{aligned} & K(Y, X)L^* K(X, X)\alpha + K(Y, X)L^* K(X, Y)\beta \\ &= \lambda K(Y, Y)L^\# K(Y, X)\alpha + \lambda K(Y, Y)L^\# K(Y, Y)\beta \end{aligned} \quad (37)$$

Equations (27) and (28) can be expressed using matrices and vectors

$$\begin{aligned}
& \begin{bmatrix} K(X,X)L^*K(X,X) & K(X,X)L^*K(X,Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
&= \lambda \begin{bmatrix} K(X,Y)L^\#K(Y,X) & K(X,Y)L^\#K(Y,Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}
\end{aligned} \tag{38}$$

$$\begin{aligned}
& \begin{bmatrix} K(Y,X)L^*K(X,X) & K(Y,X)L^*K(X,Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
&= \lambda \begin{bmatrix} K(Y,Y)L^\#K(Y,X) & K(Y,Y)L^\#K(Y,Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}
\end{aligned} \tag{39}$$

Combining Equation (29) and Equation (30) obtains

$$\begin{aligned}
& \begin{bmatrix} K(X,X)L^*K(X,X) & K(X,X)L^*K(X,Y) \\ K(Y,X)L^*K(X,X) & K(Y,X)L^*K(X,Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\
&= \lambda \begin{bmatrix} K(X,Y)L^\#K(Y,X) & K(X,Y)L^\#K(Y,Y) \\ K(Y,Y)L^\#K(Y,X) & K(Y,Y)L^\#K(Y,Y) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}
\end{aligned} \tag{40}$$