OPTO-ELECTRONICS REVIEW **20**(2), 138–147

DOI: 10.2478/s11772-012-0015-0

Feature fusion of palmprint and face via tensor analysis and curvelet transform

X. XU, X. GUAN, D. ZHANG, X. ZHANG*, W. DENG, and Z. WANG

School of Electronics and Information Engineering, Xi'an Jiaotong University, 28 Xian'ning West Road, Xi'an 710049, People' s Republic of China

In order to improve the recognition accuracy of the unimodal biometric system and to address the problem of the small samples recognition, a multimodal biometric recognition approach based on feature fusion level and curve tensor is proposed in this paper. The curve tensor approach is an extension of the tensor analysis method based on curvelet coefficients space. We use two kinds of biometrics: palmprint recognition and face recognition. All image features are extracted by using the curve tensor algorithm and then the normalized features are combined at the feature fusion level by using several fusion strategies. The k-nearest neighbour (KNN) classifier is used to determine the final biometric classification. The experimental results demonstrate that the proposed approach outperforms the unimodal solution and the proposed nearly Gaussian fusion (NGF) strategy has a better performance than other fusion rules.

Keywords: tensor analysis, curvelet, feature fusion, multimodal biometric recognition.

1. Introduction

Biometric techniques are the future of personal identification and have received a lot of attention and interest in recent years [1–4]. Jenkins *et al*. proposed a method that obtained 100% accuracy in automatic face recognition [2]. Unfortunately, 20 training samples of every class were used in their experiments. However, the small samples biometric recognition causes a research difficulty in real-world applications [1,5–11]. Usually, 2–3 training samples of each class are used in the experiments. Little work has been done to address the small samples problem. In real-world applications, it is very difficult to get a satisfactory recognition accuracy with the existing methods that only use the unimodal biometric and small samples [9,11–13]. A multimodal biometric fusion technique is a novel solution to address the problem [1,5–13]. The supplementary information between different biometrics might improve the recognition performance by using small samples.

"There are four levels of fusion: pixel level, feature level, score level, and decision level" [8,10,11]. To data, most research on the multimodal biometric fusion techniques was based on score level and decision level [9–12]. Hong *et al*. achieved improvements by integrating fingerprint and face biometric [5], while Jain *et al*. combined three biometrics: face, fingerprint, and hand geometry [6–8]. Compared to the abundance of research work related to a fusion at the score level, a fusion at the feature level is a relatively understudied problem because of the difficulties in practice.

*e-mail: ccp9999@sina.com

Recently, some work has been carried out at the feature level. There are two approaches to the feature level fusion. In the first one, the features were extracted, then concatenation took place and, finally, dimensionality reduction was used. Ross *et al*. discussed fusion of face and hand modalities at the feature level [10]. Preliminary results are encouraging and help to highlight the pros and cons of the performing fusion at the feature level. Zhang *et al*. proposed a geometry preserving projections (GPP) method at the feature level for face and palmprint fusion, but, unfortunately, 4 samples of each class were used as the training samples in their experiments [11]. Ross and Zhang methods belong to the first approach. In the second one, features were extracted, then dimensionality reduction was done, and finally, feature concatenation was performed. Zhou *et al*. presented a new approach that utilized and integrated information from side face and gait at the feature level by using principal component analysis (PCA) and multiple discriminant analysis (MDA) [12]. The experimental results demonstrated that the synthetic features encoding both face and gait information carry more discriminating power than the individual biometrics features and the feature level fusion strategy outperformed the score level strategy. Yao *et al*. presented a weighting strategy for conducting the feature fusion of palmprint and face based on classical 2D-Gabor transform and PCA method [13]. Zhou and Yao methods belong to the second approach.

Recently, multilinear algebra (algebra of higher-order tensors) has been applied for analysing the multifactor structure of image ensembles [14]. A novel face representation algorithm called "tensorface" has been proposed by Vasilescu and Terzopoulos [15]. Tensorface represents the set of face images by a higher-order tensor and extends singular value decomposition (SVD) to the higher-order tensor data. In this way, the multiple factors related to expression, illumination, and pose can be separated from different dimensions of the tensor. Ye's 2DLDA and Yan's DATER are the tensor extensions of the popular vector-based linear discriminant analysis (LDA) algorithm [14]. Zhang's tensor linear laplacian discrimination (TLLD) algorithm is an extension of LLD [14]. He's tensor subspace analysis method, the combination of tensor and locality preserving projection (LPP), also preserving local neighbour structures of tensor samples [16]. The tensor methods perform well, particularly when the number of samples is relatively small which is the case when the vector-based methods often suffer from a singularity problem [16]. In addition, Tensor PCA, Tensor MFA and Tensor LDA are the extensions of principal component analysis (PCA), marginal fisher analysis (MFA) and linear discriminant analysis (LDA), respectively [17,18].

Unfortunately, the general tensor-based methods are linear. "If the manifold is highly nonlinear, they may fail to discover the intrinsic geometrical structure" [14,16–18]. Thus, it is important to address the nonlinear problem of the tensor-based methods. The nonlinear approximation theory (NAT) has been developed in recent years. Several NAT--based approaches have been proposed, such as ridgelet, curvelet, contourlet, NSCT, etc. [19–22]. These approaches have better nonlinear approximation capabilities than classical wavelet transform [19–22]. Especially, a curvelet transform has better performance in extracting biometric features than a wavelet one [19, 21]. Thus, a curvelet transform provides a better solution for addressing the nonlinear problem of the tensor-based methods.

In this paper, we present a novel curve tensor-based multimodal biometric recognition approach that operates at the feature fusion level. Our method belongs to the second approach. The curve tensor algorithm is used for more accurate recognition results. Here, we use two kinds of biometrics: palmprint feature and face feature.

This paper is organized as follows. Section 1 gives a brief review of the current multimodal recognition techniques and tensor-based methods. Section 2 describes the second-generation curvelet transform. Section 3 gives out the curve tensor algorithm. The experimental results and performance evaluation on several well-known databases are given in Sect. 4. The experimental results are discussed in Sect. 5. The last section summarizes the paper and the conclusions are drawn finally.

2. Second-generation curvelet transform

Curvelet transform is a new multiscale representation, suited for objects which are smooth and away from discontinuities across curves. Curvelet differs from wavelet and it takes the form of basic elements which exhibit a very high directional sensitivity and are highly anisotropic [19,20]. In this section, we briefly review the implementation of the second-generation curvelet which is simpler, faster, and less redundant [19–21,23–25].

Assume that we work throughout in two dimensions, i.e., R^2. Denote x as the spatial variable, denote ω as the frequency domain variable and with r and θ as the polar coordinates in the frequency domain [19–21].We define a pair of windows $W(r)$ and $V(t)$, called the "radial window" and "angular window". The frequency window U_j is defined in the Fourier domain by [19,20]

$$U_j(r,\theta) = 2^{-3j/4} W(2^{-j} r) V\left(\frac{2^{\lfloor j/2 \rfloor}\theta}{2\pi}\right), \tag{1}$$

where $\lfloor j/2 \rfloor$ is the integer part of $j/2$.

We can define the "mother" curvelet as $\varphi_j(x)$, and its Fourier transform $\hat{\varphi}_j(\omega) = U_j(\omega)$.

As the "mother" wavelet, $\varphi_j(x)$ is considered to be a "mother" of curvelet , then all curvelets at scale 2^{-j} are obtained by translations and rotations of $\hat{\varphi}_j$.

Introduce the equispaced sequence of rotation angles $\theta_l = 2\pi \times 2^{\lfloor -j/2 \rfloor}$ (with l = 0, 1, 2, $0 \le \theta < 2\pi$) and the sequence of translation parameters $k = (k_1, k_2) \in Z^2$.

Then, the curvelets at the orientation θ_l, scale 2^{-j} and the position $x_k^{(j,l)} = R_{\theta_l}^{-1}(k_1 \times 2^{-j}, k_2 \times 2^{-j/2})$ can be defined as follows [20]

$$\varphi_{j,l,k}(x) = \varphi_j[R_{\theta_l}(x - x_k^{(j,l)})], \tag{2}$$

where R_θ is the rotation by θ radians and its inverse R_θ^{-1}. $R_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$ and $R_\theta^{-1} = R_\theta^T$.

The curvelet transform and the curvelet transform in the frequency domain are defined as follows

$$c(j,l,k) = \langle f, \varphi_{j,l,k} \rangle = \int_{R^2} f(x)\overline{\varphi_{j,l,k}(x)}dx, \tag{3}$$

$$\begin{aligned} c(j,l,k) &= \frac{1}{(2\pi)^2}\int \hat{f}(\omega)\overline{\hat{\varphi}_{j,l,k}(\omega)}d\omega \\ &= \frac{1}{(2\pi)^2}\int \hat{f}(\omega)U_j(R_{\theta_l}\omega)\exp(i\langle x_k^{(j,l)},\omega\rangle)d\omega \end{aligned} \tag{4}$$

Figure 1 shows curvelet spatial domain and frequency domain.

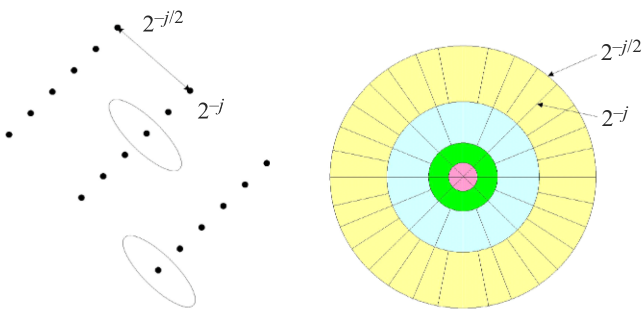

Fig. 1. Curvelet spatial domain and frequency domain.

Introduce the set of equispaced slopes $\tan\theta_l = l \times 2^{-\lfloor j/2 \rfloor}$, $l = -2^{\lfloor j/2 \rfloor},\ldots,2^{\lfloor j/2 \rfloor}-1$, and define

$$\hat{U}_{j,l(\omega)} = W_j(\omega)V_j(S_{\vartheta_l}\omega), \tag{5}$$

where S_θ is the shear matrix

$$S_\theta = \begin{pmatrix} 1 & 0 \\ -\tan\theta & 1 \end{pmatrix},\; W_j(\omega) = W(2^{-j}\omega),$$

$$V_j(\omega) = V(2^{\lfloor j/2 \rfloor}\omega_2/\omega_1),$$

$$\{(\omega_1,\omega_2): 2^j \le \omega_1 \le 2^{j+1}, -2^{-j/2} \le \omega_2/\omega_1 \le 2^{-j/2}\}$$

Figure 2 shows the basic digital tiling. The shaded region represents such a typical wedge [20].

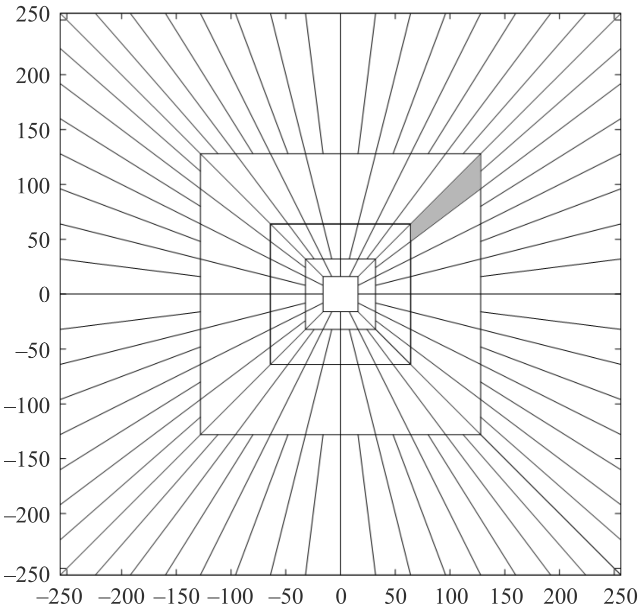

Fig. 2. Sheared wedges obeying parabolic scaling.

Then, the discrete curvelet transform is defined as follows

$$c(j,l,k) = \int \hat{f}(\omega)\hat{U}_j(S_{\theta_l}^{-1}\omega)\exp(i\langle b,\omega\rangle)d\omega, \tag{6}$$

where $b = (k_1 \times 2^{-j},\; k_2 \times 2^{-j/2})$

In this paper, we use the fast discrete curvelet transform (FDCT) *via* wrapping mechanism [19,20,22].

3. Multimodal biometric recognition based on curve tensor

Figure 3 displays the multimodal biometric recognition procedure.

The whole mechanism of the approach is detailed as follows.

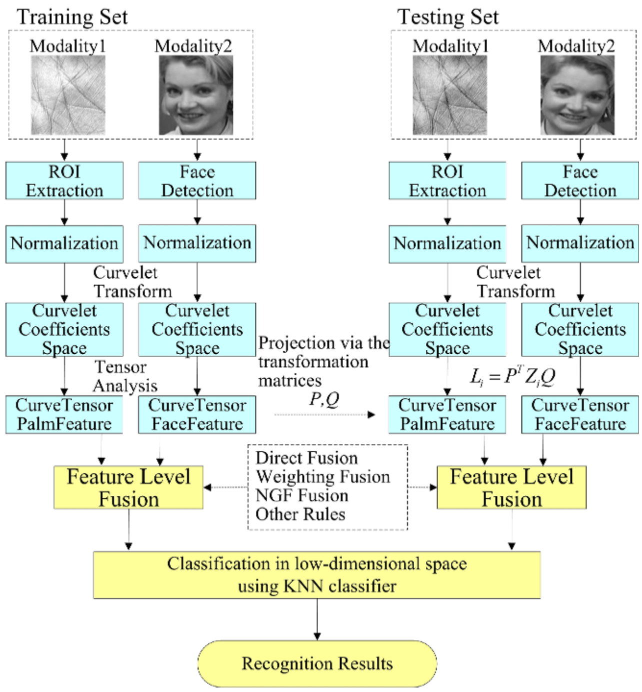

Fig. 3. Schematic of multimodal biometric recognition.

3.1. Pre-processing and image normalization

Let T_{palm} and T_{face} represent the palm and face image sample sets separately. Firstly, we extract a region of interesting (ROI) of the palm images. Please refer to Ref. 25 for details. Then, we use the AdaBoost-based method to realize the face detection [26]. At last, the grey level of all testing and training images should be scaled to 0,1, and the size should be normalized to $d \times d$ pixels. Here $d = 40$.

3.2. Curvelet decomposition and coefficients combination

In this stage, all images (T_{palm} and T_{face}) are decomposed by using a digital curvelet transform. In this paper, we use two scales of decomposition for curvelet transform. Assume that T_{palm} and T_{face} represent a sample of T_{palm} and T_{face}, respectively. T_{palm} and T_{face} are decomposed by curvelet transform into curvelet coefficients, i.e., C_{palm} and C_{face}. Assume that $C_{palm}^{j,l}$ and $C_{face}^{j,l}$ represent the j^{th} scale and l^{th} orientation of the coefficients C_{palm} and C_{face}, where $j = \{1,2\}$. While $l = 1, k_1 = 0$, and $k_2 = 1$ when $j = 1$. While $l = \{1, 2,3,4,5,6,7,8\}$, $0 \le k_1 \le 7$ and $1 \le k_2 \le 8$ when $j = 2$.

In order to reduce the computational cost and redundant information, we resize the coefficients $C_{palm}^{2,l}$ and $C_{face}^{2,l}$ to a matrix with the same size of $C_{palm}^{1,1}$ and $C_{face}^{1,1}$ by a bilinear interpolation [27,28]. Then, we combine $C_{palm}^{1,l}$ and $C_{palm}^{2,l}$, $C_{face}^{1,l}$ and $C_{face}^{2,l}$ as follows

$$Z_{palm} = \begin{bmatrix} C^{2,1}_{palm} & C^{2,1}_{palm} & C^{2,2}_{palm} \\ C^{2,3}_{palm} & C^{2,4}_{palm} & C^{2,5}_{palm} \\ C^{2,6}_{palm} & C^{2,7}_{palm} & C^{2,8}_{palm} \end{bmatrix}, \tag{7}$$

$$Z_{face} = \begin{bmatrix} C^{1,1}_{face} & C^{2,1}_{face} & C^{2,2}_{face} \\ C^{2,3}_{face} & C^{2,4}_{face} & C^{2,5}_{face} \\ C^{2,6}_{face} & C^{2,7}_{face} & C^{2,8}_{face} \end{bmatrix}, \tag{8}$$

where Z_{palm} is the palm feature matrix and Z_{face} is the face feature matrix.

3.3. Tensor analysis based on combined curvelet coefficients space

Tensors are multilinear mappings over a set of vector spaces. A tensor is a higher order generalization of a vector (first order tensor) and a matrix (second order tensor) [14–17]. In this stage, tensor analysis is used to detect the intrinsic geometrical structure of tensor space (data manifold, consist of the curvelet coefficients Z_{palm} and Z_{face}). Curve tensor considers a coefficient matrix as the second order tensor in $R^{n_1} \otimes R^{n_2}$, where R^{n_1} and R^{n_2} are two vector spaces [14–16]. Obviously, the combined curvelet coefficients space is generally a sub-manifold embedded in $R^{n_1} \otimes R^{n_2}$.

Assume that Z_i represents the combined curvelet coefficients of a size of $n_1 \times n_2$. Given a set of data points $Z_1, Z_2 \ldots, Z_m$ in $R^{n_1} \otimes R^{n_2}$, we need to find two transform matrices P of a size of $n_1 \times s_1$, Q of a size of $n_2 \times s_2$ that maps these m set of points $L_1, L_2 \ldots, L_m \in R^{s_1} \otimes R^{s_2}$ ($s_1 < n_1, s_2 < n_2$), where $L_i = P^T Z_i Q$. Here, L_i is the tensor projection that represents Z_i.

Usually, we use the classical PCA, LDA, LPP, LLD, and other methods to address the linear dimensionality reduction problem [16–18,29–31]. Here, we use LPP-based method to reduce dimensionality [29]. After performing the tensor analysis method, L_{palm} and L_{face} (curve tensor features) will be obtained. The tensor dimensionality reduction algorithm is described in detail as follows:

• optimal linear embedding

We can set up the nearest neighbour graph Ω to model the local geometrical structure of the curvelet coefficients submanifold [14,16,17,29]. Let B be the weight matrix of Ω. Assume that the label information is available (supervised learning), a possible definition of B is as follows

$$B_{ij} = \begin{cases} \exp(-\|Z_i - Z_j\|^2 / 2a^2), & \text{if } Z_i \text{ and } Z_j \text{ share the same label} \\ 0, & \text{otherwise.} \end{cases}, \tag{9}$$

where a is the suitable constant. The function $\exp(-\|Z_i - Z_j\|^2 / 2a^2)$ is the heat kernel which is intimately related to the manifold structure. $\|\cdot\|$ is the Frobenius norm of matrix, i.e., $\|D\| = \sqrt{\sum_i \sum_j d_{ij}^2}$.

A reasonable transformation respecting the graph structure can be obtained by solving the following optimization problem [14–17]

$$\arg \min_{P,Q} \sum_{ij} B_{ij} \left\| P^T Z_i Q - P^T Z_j Q \right\|^2, \tag{10}$$

where P and Q are the transformation matrices.

Let $L_i = P^T Z_i Q$. Assume that H represents the diagonal matrix. $H_{ii} = \sum_j B_{ij}$. Since $\|D\|^2 = tr(DD^T)$, we have

$$\begin{aligned} &\min_{P,Q} \sum_{ij} B_{ij} \left\| P^T Z_i Q - P^T Z_j Q \right\|^2 \\ &= \sum_{ij} B_{ij} tr[(L_i - L_j)(L_i - L_j)^T] \\ &= \sum_{ij} B_{ij} tr(L_i L_i^T + L_j L_j^T - L_i L_j^T + L_j L_i^T) \\ &= 2tr\left(\sum_i H_{ii} P^T Z_i Q Q^T Z_i^T P - \sum_{ij} B_{ij} P^T Z_i Q Q^T Z_j^T P \right) \\ &= 2tr\left[P^T \left(\sum_i H_{ii} Z_i Q Q^T Z_i^T - \sum_{ij} B_{ij} Z_i Q Q^T Z_j^T P \right) P \right] \\ &= 2tr(P^T H_Q P) - 2tr(P^T B_Q P), \end{aligned} \tag{11}$$

where

$H_Q = \sum_i H_{ii} Z_i Q Q^T Z_i^T$ and $B_Q = \sum_{ij} B_{ij} Z_i Q Q^T Z_j^T$.

Since $\|D\|^2 = tr(D^T D)$, we also have

$$\begin{aligned} &\min_{P,Q} \sum_{ij} B_{ij} \left\| P^T Z_i Q - P^T Z_j Q \right\|^2 \\ &= \sum_{ij} B_{ij} tr[(L_i - L_j)^T (L_i - L_j)] \\ &= 2tr\left(\sum_i H_{ii} L_i^T L_i - \sum_{ij} B_{ij} L_j^T L_i \right) \\ &= 2tr\left[Q^T \left(\sum_i H_{ii} Z_i^T P P^T Z_i - \sum_{ij} B_{ij} Z_j^T P P^T Z_j \right) Q \right] \\ &= 2tr(Q^T H_P Q) - 2tr(Q^T B_P Q), \end{aligned} \tag{12}$$

where

$H_P = \sum_i H_{ii} Z_i^T P P^T Z_i$ and $B_P = \sum_{ij} B_{ij} Z_i^T P P^T Z_j$.

To preserve the graph structure, we should maximize the global variance on the manifold. The weighted variance of L can be estimated as follows

$$\begin{aligned}\operatorname{var}(L) &= \sum_i \|L_i\|^2 H_{ii} = \sum_i tr(L_i^T L_i) H_{ii} \\ &= tr\left[Q^T\left(\sum_i H_{ii} Z_i^T P P^T Z_i\right)Q\right] = tr(Q^T H_P Q),\end{aligned} \tag{13}$$

$$\begin{aligned}\operatorname{var}(L) &= \sum_i \|L_i\|^2 H_{ii} = \sum_i tr(L_i L_i^T) H_{ii} \\ &= tr\left[P^T\left(\sum_i H_{ii} Z_i Q Q^T Z_i^T\right)P\right] = tr(P^T H_Q P).\end{aligned} \tag{14}$$

At last, we may solve the following optimization problem

$$\arg\min_{P,Q} \frac{tr[P^T(H_Q - B_Q)P]}{tr(P^T H_Q P)}, \tag{15}$$

$$\arg\min_{P,Q} \frac{tr[Q^T(H_P - B_P)Q]}{tr(Q^T H_P Q)}, \tag{16}$$

• iterative algorithm

The optimal P should be the generalized eigenvectors of $(H_Q - B_Q, H_Q)$. The optimal Q should be the generalized eigenvectors of $(H_P - B_P, H_P)$. We can calculate P and Q iteratively [32]. Let $P = I$. I is the identity matrix, then Q and P can be calculated iteratively by solving the following generalized eigenvector problems [16,32]

$$(H_P - B_P)Q = \lambda H_P Q, \tag{17}$$

$$(H_Q - B_Q)P = \lambda H_Q P. \tag{18}$$

The matrices H_P and H_Q are all positive semi-definite and symmetric ones.

Table 1 shows the computation of P and Q by using the iterative algorithm.

Table 1. Procedure of the curve tensor method.

Input: samples data ($Z_1, Z_2 ..., Z_m$), tensor projection dimension u, v
Output: transformation matrices P and Q, curve tensor feature L_i
• Curvelet decomposition and combination;
• Construct the weight matrix B;
• Let $P = I$, N refers to the iterative number ($N \le 5$);
• For $i = 1, \dots N$
Calculate H_P, B_P and ($H_P - B_P$) by using P;
Calculate Q, the generalized eigenvectors of ($H_P - B_P$, H_P);
Calculate H_Q, B_Q and ($H_Q - B_Q$) by using Q;
Calculate P, the generalized eigenvectors of ($H_Q - B_Q$, H_Q);
And
• Let $P = [p_1, p_2 ..., p_u]$, $Q = [q_1, q_2 ..., q_v]$;
• Let $L_i = P^T Z_i Q$;
• Output P, Q, and L.

If the manifold is highly nonlinear, the general tensor-based methods may fail to discover the intrinsic geometrical structure. Curvelet is an approximately true "2-D" sparse representation for 2-D signals like images and can efficiently capture the intrinsic geometrical structures in natural images. Thus, curve tensor method is helpful to address the nonlinear problem.

3.4. Feature level fusion strategies

We reshape L_{palm} and L_{face} into the form of feature vectors and carry on a feature vector normalization. Then, we normalize L_{palm} and L_{face} by using Z-Score formula [8,9]

$$L_{norm\text{-}palm} = \frac{L_{palm} - \mu_{palm}}{\sigma_{palm}}, \tag{19}$$

$$L_{norm\text{-}palm} = \frac{L_{face} - \mu_{face}}{\sigma_{face}}, \tag{20}$$

where μ_{palm} and μ_{face} are the mean value of L_{palm} and L_{face}, σ_{palm}, and σ_{face} are the variance value of L_{palm} and L_{face}.

The previous works mainly deal with the score level and decision level fusion [5–9]. The information obtained by the feature level fusion methods is more abundant than the score level and decision level fusion methods [10,12]. So, we pay more attention to the feature level fusion in this paper.

A palmprint feature vector and its corresponding face feature vector are fused by using the general feature fusion strategies

direct fusion [11,12]:

$$L_{norm\text{-}fuse} = [L_{norm\text{-}palm} \;\; L_{norm\text{-}face}], \tag{21}$$

weighting fusion [13]

$$L_{norm\text{-}fuse} = [\alpha \times L_{norm\text{-}palm} \;\; L_{norm\text{-}face}], \tag{22}$$

where α is the weight and the computation of α has been introduced in Ref. 13.

Here, we propose the feature level fusion strategy and we call it nearly Gaussian fusion (NGF)

nearly Gaussian fusion:

$$L_{norm\text{-}fuse} = \left[\exp\left(\frac{-L_{norm\text{-}palm}}{2\sigma_{rpalm}^2}\right) \exp\left(\frac{-L_{norm\text{-}face}}{2\sigma_{rface}^2}\right)\right], \tag{23}$$

where σ_{rpalm} and σ_{rface} are the width of Gaussian radial basis. Here, we choose the σ_{rpalm} and σ_{rpalm} just like the kernel parameters of the support vector machine (SVM).

At last, the fused sample set $T_{norm\text{-}fuse}$ will be obtained. The KNN classifier is used to determine the final classification. Here, we use Euclidean distance. In order to compare the performance of the multimodal methods and unimodal methods more clearly, we do not use any support vector machine (SVM). KNN classifier and Euclidean distance measurement are used in the experiments.

4. Experimental results

A comparative study between the unimodal methods and the multimodal methods will be presented in this section. The Palm-Tensor LPP, Palm-Wavelet, Face-Tensor LPP, and Face-Wavelet methods are four basic unimodal biometrics recognition methods. We use the Tensor LPP method instead of the classical PCA, LDA, MFA, LLD, and LPP methods in the experiments since the Tensor LPP method has a better performance than the other methods [16]. Daubechies 2 wavelets are used in the Wavelet-based methods. The "direct fusion" [12], "weighting fusion" [13], and "NGF fusion" rules are used as three fusion strategies in this study. The experiments are performed on the several well-known databases: PolyU palmprint database, CMU--PIE, and ORL face databases.

4.1. Experimental results on PolyU and CMU-PIE databases

We use a PolyU palmprint database provided by the Hong Kong Polytechnic University [33,34]. The palmprint subset used in the experiments contains 68 individuals. For each individual, 10 images of a size of 120×120 pixels are used in the experiments. The PIE face database provided by the Carnegie Mellon University (CMU) contains 68 individuals. The face images were captured by 13 synchronized cameras and 21 flashes under varying illumination, different pose and expression [16,18]. Each individual has 10 images (Pose C05, C07, C09, C27, C29) which are used in our experiments. At last, all the images are normalized to 40×40 pixels.

For PolyU and PIE databases, 68 palmprint classes and 68 face classes (10 images/class) are used in the experiments. The image samples are shown in Fig.4.

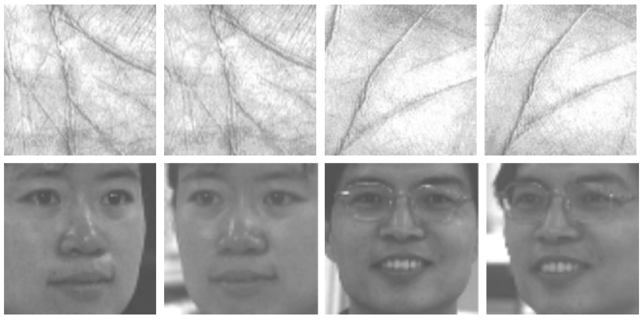

Fig. 4. Image samples from PolyU and CMU-PIE databases.

We randomly select $\lambda(= 2, 3)$ images of each class for the training set (small samples). The rest of the images is considered as the testing set. The training samples are used to learn the tensor subspace. The testing samples are then projected into the low-dimensional representation subspace. For a convenient comparison, we also give out the experimental results by using $\lambda(= 5, 6, 7, 8)$ training samples. In order to reduce the random error, we give more objective evaluation of the algorithm. For each λ, we have 20 times random selection of the training set and take the average accuracy as the final results.

The KNN classifier is used to determine the final classification. Here, we use a Euclidean distance. The experimental results on PolyU and PIE databases are shown in Tables 2 and 3. The highest item among each column is bold printed.

In Table 2, we can see that the performance of the multimodal methods is far better than the one of the unimodal methods. Especially, PIEPalm-CurveTensor-fusion (NGF) always outperforms other 10 methods and with the best recognition accuracy of 90.99% (2Train), 93.07% (3Train) and 99.71% (5Train), respectively. In Table 3, 100% accuracy has been obtained while using more training samples (6,7,8Train).

Table 2. Recognition accuracy on PolyU and PIE databases (2,3,5Train).

Databases/Methods	Recognition accuracy (%)		
	2Train(Dim)	3Train(Dim)	5Train(Dim)
PIE-Wavelet-KNN	43.38(144)	57.56(144)	80.00(49)
Palm-Wavelet-KNN	81.43(144)	85.50(144)	94.50(49)
PIEPalm-Wavelet-fusion(Direct)	84.56(144, 144)	88.03(144, 144)	97.50(49, 49)
PIE-TensorLPP-KNN	58.46(21×21)	75.21(18×18)	80.00(18×18)
Palm-TensorLPP-KNN	82.17(10×10)	88.24(16×16)	94.41(12×12)
PIEPalm-TensorLPP-fusion(Direct)	89.71(17×17, 16×16)	90.76(18×18, 16×16)	97.06(18×18, 12×12)
PIE-CurveTensor-KNN	61.40(28×28)	74.37(27×27)	81.18(26×26)
Palm-CurveTensor-KNN	82.35(26×26)	89.08(16×16)	94.71(18×18)
PIEPalm-CurveTensor-fusion(Direct)	90.63(28×28, 26×26)	91.18(18×18, 12×12)	99.41(18×18,18×18)
PIEPalm-CurveTensor-fusion(Weighting)	90.99(28×28, 26×26)	92.33(18×18, 12×12)	99.41(18×18,18×18)
PIEPalm-CurveTensor-fusion(NGF)	**90.99(28×28, 26×26)**	**93.07(18×18, 12×12)**	**99.71(18×18,18×18)**

Table 3. Recognition accuracy on PolyU and PIE databases (6,7,8Train).

Databases/Methods	Recognition accuracy (%)		
	6Train(Dim)	7Train(Dim)	8Train(Dim)
PIE-Wavelet-KNN	86.03(144)	87.75(144)	88.24(144)
Palm-Wavelet-KNN	95.22(144)	97.06(144)	98.53(144)
PIEPalm-Wavelet-fusion(Direct)	97.43(144, 144)	98.04(144, 144)	99.26(144, 144)
PIE-TensorLPP-KNN	89.34(9×9)	97.06(18×18)	97.06(18×18)
Palm-TensorLPP-KNN	96.32(9×9)	99.02(16×16)	100.00(12×12)
PIEPalm-TensorLPP-fusion(Direct)	98.90(9×9, 9×9)	100.00(18×18, 16×16)	100.00(18×18, 12×12)
PIE-CurveTensor-KNN	94.85(14×14)	97.06(15×15)	97.06(20×20)
Palm-CurveTensor-KNN	96.32(16×16)	99.02(16×16)	100.00(21×21)
PIEPalm-CurveTensor-fusion (Direct)	100.00(14×14, 16×16)	100.00(15×15, 16×16)	100.00(17×17,18×18)
PIEPalm-CurveTensor-fusion(Weighting)	100.00(14×14, 16×16)	100.00(15×15, 16×16)	100.00(17×17,18×18)
PIEPalm-CurveTensor-fusion(NGF)	**100.00(14×14, 16×16)**	**100.00(15×15, 16×16)**	**100.00(17×17,18×18)**

4.2. Experimental results on PolyU and ORL databases

The Cambridge University ORL face database is composed of 400 images of ten different patterns for each of 40 individuals. Some images were captured at different times and have different variations including expression (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). All images are at greyscale and of a size of 112×92 pixels. For PolyU and ORL databases, 40 palmprint classes and 40 face classes (10 images/class) are used in the experiments. The image samples from ORL database are shown in Fig. 5.

Table 4. Recognition accuracy on PolyU and ORL databases(2,3,5Train).

Databases/Methods	Recognition rates (%)		
	2Train(Dim)	3Train(Dim)	5Train(Dim)
ORL-Wavelet-KNN	78.75(49)	81.43(49)	89.00(49)
Palm-Wavelet-KNN	80.94(49)	85.00(49)	94.50(49)
ORLPalm-Wavelet-fusion(Direct)	94.06(49, 49)	95.36(49, 49)	99.50(49, 49)
ORL-TensorLPP-KNN	83.44(12×12)	87.50(22×22)	93.00(22×22)
Palm-TensorLPP-KNN	84.38(17×17)	87.86(24×24)	94.50(12×12)
ORLPalm-TensorLPP-fusion(Direct)	94.37(12×12, 17×17)	96.43(22×22,24×24)	97.50(22×22, 12×12)
ORL-CurveTensor-KNN	87.81(21×21)	89.29(20×20)	94.50(15×15)
Palm-CurveTensor-KNN	84.69(18×18)	88.21(18×18)	95.00(16×16)
ORLPalm-CurveTensor-fusion(Direct)	95.63(21×21, 18×18)	97.14(20×20, 18×18)	100.00(15×15, 16×16)
ORLPalm-CurveTensor-fusion(Weighting)	95.94(21×21,18×18)	97.50(20×20, 18×18)	100.00(15×15, 16×16)
ORLPalm-CurveTensor-fusion(NGF)	**96.56(21×21, 18×18)**	**97.86(20×20, 18×18)**	**100.00(15×15, 16×16)**

Table 5. Recognition accuracy on PolyU and ORL databases (6,7,8Train).

Databases/Methods	Recognition rates (%)		
	6Train(Dim)	7Train(Dim)	8Train(Dim)
ORL-Wavelet-KNN	97.50(49)	98.33(49)	98.75(49)
Palm-Wavelet-KNN	96.25(49)	96.67(49)	97.50(49)
ORLPalm-Wavelet-fusion(Direct)	100.00(49, 49)	100.00(49, 49)	100.00(49, 49)
ORL-TensorLPP-KNN	97.50(15×15)	98.33(21×21)	97.50(20×20)
Palm-TensorLPP-KNN	97.50(19×19)	99.17(20×20)	100.00(14×14)
ORLPalm-TensorLPP-fusion(Direct)	100.00(12×12, 17×17)	100.00(22×22,24×24)	100.00(22×22, 12×12)
ORL-CurveTensor-KNN	97.50(18×18)	98.33(15×15)	98.75(16×16)
Palm-CurveTensor-KNN	98.12(20×20)	99.17(19×19)	100.00(18×18)
ORLPalm-CurveTensor-fusion(Direct)	100.00(18×18, 20×20)	100.00(15×15, 19×19)	100.00(16×16, 18×18)
ORLPalm-CurveTensor-fusion(Weighting)	100.00(18×18,20×20)	100.00(15×15, 19×19)	100.00(16×16, 18×18)
ORLPalm-CurveTensor-fusion(NGF)	**100.00(18×18, 20×20)**	**100.00(15×15, 19×19)**	**100.00(16×16, 18×18)**

Fig. 5. Image samples from ORL database.

The experimental results on PolyU and ORL databases are shown in Tables 4 and 5. The highest item among each column is bold printed.

In Tables 4 and 5, ORLPalm-CurveTensor-fusion (NGF) achieves the best recognition accuracy of 96.56% (2Train), 97.86%(3Train) and 100% (5,6,7,8Train). It is found that curve tensor outperforms the other methods with different numbers of training samples (2,3,5,6,7,8) per individual, and nearly Gaussian fusion (NGF) strategy outperforms the other strategies, i.e., direct fusion and weighting fusion.

4.3. Relationship between σ_{rpalm}, σ_{rface} and recognition accuracy

The relationship between the kernel parameters σ_{rpalm}, σ_{rface} and the recognition accuracy is shown in Fig. 6

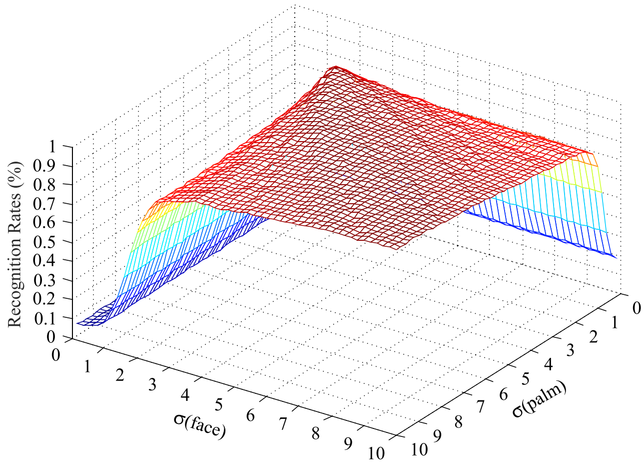

Fig. 6. Relationship between σ_{rpalm}, σ_{rface} and recognition accuracy.

(3Train, PolyU & ORL databases). Here, $\sigma_{rpalm} \in [1,10]$ and $\sigma_{rface} \in [1,10]$. When $\sigma_{rpalm} \leq 3$ and $\sigma_{rpalm} \leq 3$, the recognition accuracy is very low. With the increment of σ_{rpalm} and σ_{rface}, the recognition accuracy reaches its maximum. We find out that there are several points of extrema and that the best recognition accuracy will be obtained according to these points. Then, the recognition accuracy will be descended.

Table 6. Computational cost using PolyU and PIE databases (2Train).

Databases/Methods	Computational time(s)		
	Feature extraction	Feature fusion	Classfication
PIE-Wavelet-KNN	5.8456	/	0.0099
Palm-Wavelet-KNN	5.8941	/	0.0097
PIEPalm-Wavelet-fusion	11.7397	0.0974	0.0124
PIE-TensorLPP-KNN	5.8670	/	0.0787
Palm-TensorLPP-KNN	5.9006	/	0.0809
PIEPalm-TensorLPP-fusion	11.7676	0.4665	0.1472
PIE-CurveTensor-KNN	16.7421	/	0.0792
Palm-CurveTensor-KNN	16.4312	/	0.0780
PIEPalm-CurveTensor-fusion(NGF)	33.2733	0.4816	0.1498

Table 7. Computational cost using PolyU and ORL databases (2Train).

Databases/Methods	Computational time(s)		
	Feature extraction	Feature fusion	Classfication
ORL-Wavelet-KNN	3.2876	/	0.0066
Palm-Wavelet-KNN	3.1983	/	0.0061
ORLPalm-Wavelet-fusion	6.4859	0.0531	0.0095
ORL-TensorLPP-KNN	2.9189	/	0.0331
Palm-TensorLPP-KNN	2.8929	/	0.0323
ORLPalm-TensorLPP-fusion	5.8118	0.1782	0.0543
ORL-CurveTensor-KNN	8.6013	/	0.0317
Palm-CurveTensor-KNN	8.5611	/	0.0309
ORLPalm-CurveTensor-fusion(NGF)	17.1624	0.1795	0.0689

4.4. Computational cost using unimodal and multimodal methods

Tables 6 and 7 show the computational cost using the unimodal and multimodal methods. Since we use the KNN classifier in the experiments, thus, we give the computational cost of the feature extraction, the feature fusion and the classification. In the experiments we use the same data as in Sect. 4.1 and Sect. 4.2. All of the algorithms are implemented in MATLAB 7.12 and executed on the same computer (Intel Core i5-2300 2.8GHz CPU, 2048M RAM). All of the experiments are completed in the same environment.

In Tables 6 and 7, the multimodal methods are slower than the unimodal methods. In addition to that, the feature fusion time is needed in the multimodal methods. It is found that the wavelet-based methods acquire the best performance. Although our method needs more computational time, it obtains better accuracy than the unimodal methods and other multimodal methods.

5. Discussions

- While the previous works mainly deal with subspace just like PCA, LLD, LDA, MFA and their extension of tensor, we now pay more attention to the combination of frequency domain and tensor subspace. Curve tensor is an extension of the tensor analysis method based on curvelet coefficients space. It is helpful to address the nonlinear problem. The experimental results demonstrate that it is a robust and reliable multimodal biometric recognition approach.
- The experimental results also show that the proposed "nearly Gaussian fusion" (NGF) strategy has better performance than other fusion rules. In the future, we plan to design a fusion rule based on a human vision perception and the preliminary results are promising.

6. Conclusions

In order to address the problem of the small samples recognition, in this paper we present the feature level biometric fusion approach based on curve tensor. Experimental results on PolyU, CMU-PIE and ORL databases demonstrate the effectiveness and robustness of our methods.

In conclusion, the curve tensor approach is a novel attempt to apply tensor analysis and curvelet transform to multimodal biometrics and can also be applied to other problems such as target recognition, SAR image processing, medical image processing etc.

Acknowledgements

The work is supported by the grants from the National Natural Science Foundation of China (No. 60972146 and No. 60602025), the grants from the National High Technology Research and Development Program of China (863 Program) (No. 2005AA121130). We would like to thank the Hong Kong Polytechnic University, the Cambridge University and the Carnegie Mellon University for their permission to use the biometrics databases.

References

1. K. Jain, "Biometric recognition", *Nature* **449**, 38–40 (2007).
2. R. Jenkins and A.M. Burton, "100% accuracy in automatic face recognition", *Science* **319**, 435 (2008).
3. C.C. Tsai, J. Taur, and C.W. Tao, "Iris recognition based on relative variation analysis with feature selection", *Opt. Eng.* **47**, 097202 (2008).
4. F.P. Nie, S.M. Xiang, Y.Q. Song, and C.S. Zhang, "Orthogonal locality minimizing globality maximizing projections for feature extraction", *Opt. Eng.* **48**, 017202-017202-5 (2009).
5. K. Chang, K. Bowyer, and V. Barnabas, "Comparison and combination of ear and face images in appearance-based biometrics", *IEEE T. Pattern Anal.* **25**, 1160–1165 (2003).
6. L. Hong and A.K. Jain, "Integrating faces and fingerprints for personal identification", *IEEE T. Pattern Anal.* **20**, 1295–1307 (1998).
7. A.K. Jain and A. Ross, "Learning user-specific parameters in a multibiometric system", *Proc. ICIP*, Vol. 1, 57–60 (2002).
8. A. Ross and A.K. Jain, "Information fusion in biometrics", *Pattern Recogn. Lett.* **24**, 2115–2125 (2003).
9. X.Y. Jing, Y.F. Yao, and D. Zhang, "Face and palmprint pixel level fusion and Kernel DCV-RBF classifier for small sample biometric recognition", *Pattern Recogn.* **40**, 3209–3224 (2007).
10. A. Ross and R. Govindarajan, "Feature level fusion using hand and face biometrics", *Proc. SPIE.* **5779**, 196–204 (2005).
11. T.H. Zhang, X.L. Li, D.C. Tao, and J. Yang, "Multimodal biometrics using geometry preserving projections", *Pattern Recogn.* **41**, 805–812 (2008).
12. X.L. Zhou and B. Bhanu, "Feature fusion of side face and gait for video-based human identification", *Pattern Recogn.* **41**, 778–795 (2008).
13. Y.F. Yao, X.Y. Jing, and H.S. Wong, "Face and palmprint feature level fusion for single sample biometrics recognition", *Neurocomputing* 7**0**, 1582–1588 (2007).
14. W. Zhang, Z.C. Lin, and X.O. Tang, "Tensor linear Laplacian discrimination (TLLD) for feature extraction", *Pattern Recogn.* **42**, 1941–1948 (2009).
15. M.A.O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis for image ensembles", *Proc. CVPR*, 93–99 (2003).
16. X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis", *Proc. NIPS* (2005).
17. D.C. Tao, X.L. Li, and X.D. Wu, "General tensor discriminant analysis and Gabor features for gait recognition", *IEEE T. Pattern Anal.* **29**, 1700–1715 (2007).
18. H.P. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "MPCA: multilinear principal component analysis of tensor objects", *IEEE T. Neural. Networ.* **19**, 18–39 (2008).
19. S.T. Li and B. Yang, "Multifocus image fusion by combining curvelet and wavelet transform", *Pattern Recogn. Lett.* **29**, 1295–1301 (2008).
20. E.J. Candes, L. Demanet, and D.L. Donoho, "Fast discrete curvelet transforms", *Multiscale Mode. Sim.* **5**, 861–899 (2005).

21. T. Mandal, "Curvelet based face recognition via dimension reduction", *Signal Process.* **89**, 2345–2353 (2009).
22. X. Xu, D. Zhang, and X. Zhang, "An efficient method for human face recognition using nonsubsampled contourlet transform and support vector machine", *Opt. Appl.* **39**, 601–615 (2009).
23. M.Y. Fu and C. Zhao, "Fusion of infrared and visible images based on the Second generation curvelet transform", *J. Infrared Milim. W.* **28**, 254–258(2009).
24. Z.H Xie, G.D Liu, and S.Q Wu, "A fast infrared face recognition system using curvelet transformation", *Proc.* 2th *Inter. Symp. Electro. Comm. Secu.* **2**, 145–149 (2009).
25. D. Zhang, W. Kong, and J. You, "Online palmprint identification", *IEEE. T. Pattern. Anal.* **25**, 1041–1050 (2003).
26. Y. Freund, "Boosting a weak learning algorithm by majority", *Inform. Comput.* **121**, 256–285 (1995).
27. W. Hao-jing, W. Jian-li, and W. Ming-hao, "Efficient image inpainting based on bilinear interpolation downscaling", *Optic. Precis. Eng.* **18**, 1234–1241 (2010).
28. S.A. Fahmy, "Generalised parallel bilinear interpolation architecture for vision systems", *Proc. Int. Conf. on Reconfigurable Computing and FPGAs*, Cancun, 331–336 (2008).
29. X. He, S. Yan, and Y. Hu, "Face recognition using Laplacian faces", *IEEE T. Pattern Anal.* **27**, 328–340 (2005).
30. C. Du, J. Yang, and Q. Wu, "Integrating affinity propagation clustering method with linear discriminant analysis for face recognition", *Opt. Eng.* **46**, 110501 (2007).
31. D. Xu, S. Yan, and D. Tao, "Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval", *IEEE T. Image Process.* **216**, 2811–2821 (2007).
32. J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis", http://books.nips.cc/nips17. html, 10–20 (2007).
33. PolyU Palmprint Database.TTTP://www.comp.polyu.edu.hk/~biometrics/
34. W. Jia, D.S. Huang, and D. Zhang, "Palmprint verification based on robust line orientation code", *Pattern Recogn.* **41**, 1504–1513 (2008).