Chapter 7
Recent Developments in Video-Based Face Recognition

Jingxiao Zheng, Vishal M. Patel and Rama Chellappa

Abstract Face recognition with its wide range of commercial and law enforcement applications has been one of the most active areas of research in the field of computer vision and pattern recognition. Personal identification systems based on faces have the advantage that facial images can be obtained from a distance without requiring cooperation of the subject, as compared to other biometrics such as fingerprint, iris, etc. Face recognition is concerned with identifying or verifying one or more persons from still images or video sequences using a stored database of faces. Depending on the particular application, there can be different scenarios, ranging from controlled still images to uncontrolled videos. Since face recognition is essentially the problem of recognizing a 3D object from its 2D image or a video sequence, it has to deal with significant appearance changes due to illumination and pose variations. Current algorithms perform well in controlled scenarios, but their performance is far from satisfactory in uncontrolled scenarios. Most of the current research in this area is focused toward recognizing faces in uncontrolled scenarios. This chapter presents an overview of recent video-based face recognition methods. In particular, recent sparse coding-based, manifold-based, probabilistic, geometric model-based, and dynamic model-based methods are reviewed.

J. Zheng · R. Chellappa (✉)
Center for Automation Research, University of Maryland,
College Park, MD 20742, USA
e-mail: rama@umiacs.umd.edu

J. Zheng
e-mail: jxzheng@umiacs.umd.edu

V.M. Patel
Rutgers, The State University of New Jersey,
94 Brett Road, Piscataway, NJ 08854, USA
e-mail: vishal.m.patel@rutgers.edu

M. Tistarelli and C. Champod (eds.), *Handbook of Biometrics for Forensic Science*, Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-3-319-50673-9_7

7.1 Introduction

Video-based face recognition has received a significant amount of attention in recent years. This is mainly due to the fact that large amounts of video data are becoming available everyday. Millions of cameras have been installed in buildings, streets, and airports around the world, and people are using billions of handheld devices that are capable of capturing videos. As a result, 350 million photos are uploaded to Facebook every day and 100 h of video are uploaded to YouTube each minute.

For video-based face recognition problem, the identification and verification tasks are all based on videos rather than still images compared to the classical image-based face recognition problem. Approaches for video-based face recognition need to identify a person in a video, given some possible candidates, or to decide whether the two people in two different videos are the same person.

In most of the video-based face recognition methods, given video data, tracking algorithms like [38] are first used to detect faces in the video frames. Then fiducial extraction methods like [47] are applied to align the detected faces. After the alignment, traditional feature extraction techniques such as SIFT [30], HoG [14], LBP [31] or the very popular DCNN features [26, 35, 36] are used to extract features for matching.

In video-based face recognition, a key challenge is in exploiting the extra information available in a video, e.g., face, body, and motion identity cues. In addition, different video sequences of the same subject may contain variations in resolution, illumination, pose, and facial expressions. These variations contribute to the challenges in designing an effective video-based face recognition algorithm. Whether the temporal information is considered or not, most video-based face recognition can be divided into sequence-based methods or set-based methods. Sequence-based face recognition methods consider the video as a sequence of images and make use of the temporal information for recognition. On the other hand, set-based face recognition methods only consider the video as a set of images and ignore their order.

Besides using temporal information, video-based face recognition can also be sorted by the techniques used to model the video. These include sparse coding-based methods, manifold-based methods, probabilistic methods, geometrical model-based methods, and dynamical model-based methods. In this chapter, we give an overview of some of these modeling approaches.

7.2 Sparse Coding-Based Methods

For sparse coding-based methods, faces (or features extracted from faces) in videos are modeled as dictionaries, which are overcomplete atoms learned from the training data with sparsity constraints.

Given L video frames with faces of dimension M concatenated in a matrix $\mathbf{Y} = \left[\mathbf{y}_1, \ldots, \mathbf{y}_L\right] \in \mathbb{R}^{M \times L}$, the problem of learning a dictionary, which minimizes

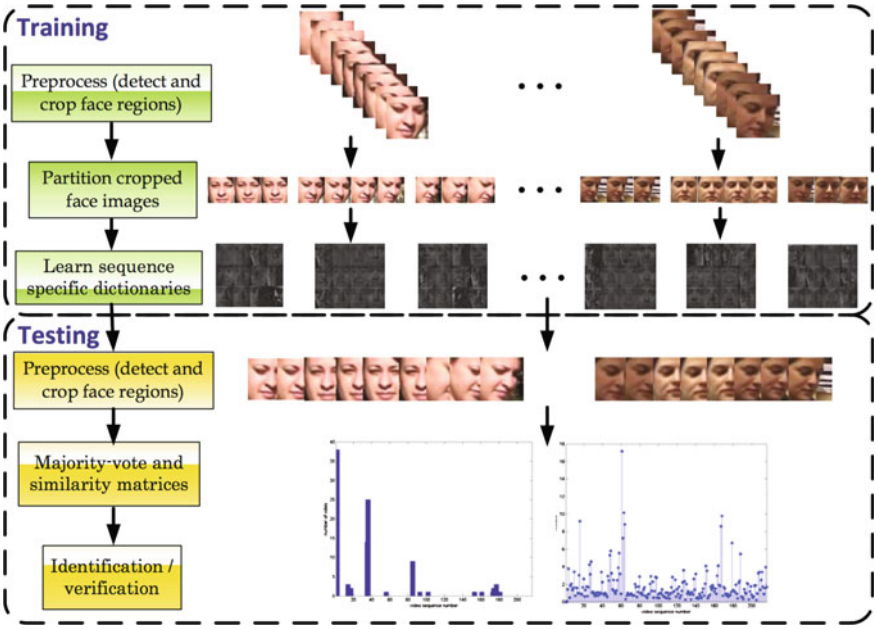

Fig. 7.1 Dictionary-based face recognition from video [12]

the representation error with a sparseness constraint is equivalent to solving the following optimization problem

$$\min_{\mathbf{D},\mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \;\text{ s.t }\; \|\mathbf{x_i}\|_0 \leq T, \;\; \mathbf{d}_i^T \mathbf{d}_i = 1 \quad \forall i, \tag{7.1}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\mathbf{x}\|_0$ is the ℓ_0 norm of $\mathbf{x}$ which counts the number of nonzero elements in $\mathbf{x}$, $\mathbf{D} = \left[\mathbf{d}_1, \ldots, \mathbf{d}_S\right] \in \mathbb{R}^{M \times S}$ is the dictionary, $\mathbf{X} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_N\right] \in \mathbb{R}^{S \times N}$ is the corresponding collection of sparse coefficients, S is the number of atoms in the dictionary, and T is a sparsity parameter. Because of the sparsity constraint, the learned dictionaries are robust to different kinds of variations in video sequences.

[12] proposed a generative dictionary learning method for video-based face recognition. The main idea of the method is to partition the video frames into clusters with different poses and illuminations and learn a set of sub-dictionaries for each cluster. Then the concatenation of the sub-dictionaries removes the temporal redundancy in the videos and can handle large variations on poses and illumination variations. An overview of this method is shown in Fig. 7.1.

For each frame in a video sequence, the face regions are first detected and cropped. Then all the cropped face images are partitioned into K different partitions by a K-means clustering type of algorithm. For each partition, a dictionary is learned with the minimum representation error under a sparseness constraint using (7.1).

Thus, there will be K sub-dictionaries built to represent a video sequence. Then the video sequence-specific dictionary is constructed by concatenating these partition-level sub-dictionaries as $\mathbf{D}_p = \left[\mathbf{D}_p^1, \mathbf{D}_p^2, \ldots, \mathbf{D}_p^K\right]$. Due to changes in pose and lighting in a video sequence, the number of face images in a partition will vary. Those partitions with very few images will be augmented by synthesized face images. This is done by creating horizontally, vertically, or diagonally position shifted face images, or by in-plane rotated face images.

For identification task, testing videos are partitioned into K partitions as well. Given a testing frame $\mathbf{q}_{l,k}$ from the kth partition, the frame-level decision $\hat{p}_{l,k}$ is the sequence p with the minimum residual error from its projection onto the subspace spaned by $\mathbf{D}_p$ as

$$\hat{p}_{l,k} = \underset{p}{\operatorname{argmin}} \|\mathbf{q}_{l,k} - \mathbf{D}_p \mathbf{D}_p^{\dagger} \mathbf{q}_{l,k}\|_2 \tag{7.2}$$

The sequence-level decision $\hat{p}$ is then the weighted sum of votes from K partitions as

$$\hat{p} = \underset{i}{\operatorname{argmax}} \sum_{k=1}^{K} w_k \sum_{l} \mathbf{1}\{\hat{p}_{l,k} = i\} \tag{7.3}$$

For verification task, given a query video sequence m and gallery video sequence p (with learned dictionary $\mathbf{D}_p$), the similarity score is

$$\mathbf{R}^{m,p} = \min_{k} \min_{l} \|\mathbf{q}_{l,k}^m - \mathbf{D}_p \mathbf{D}_p^{\dagger} \mathbf{q}_{l,k}^m\|_2. \tag{7.4}$$

which is the minimum residual among all l and all k, between the frames from query video sequence m and gallery dictionary $\mathbf{D}_p$.

[11] further introduced the joint sparsity constraints into their dictionary learning algorithm. Given video frames sets $\{\mathbf{Y}^k\}$, instead of learning dictionaries from each frame partition independently as

$$\min_{\mathbf{D}^k, \mathbf{X}^k} \|\mathbf{Y}^k - \mathbf{D}^k \mathbf{X}^k\|_F^2 \;\; \text{s.t} \;\; \|\mathbf{x}_\mathbf{i}^\mathbf{k}\|_0 \leq T, \;\; \mathbf{d}_i^{kT} \mathbf{d}_i^k = 1 \quad \forall i \tag{7.5}$$

based on the joint sparse constraints, the dictionaries are learned jointly as

$$\min_{\{\mathbf{D}^k\}, \mathbf{X}} \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{Y}^k - \mathbf{D}^k \mathbf{X}^k\|_F^2 + \lambda \|\mathbf{X}\|_{1,2} \;\; \text{s.t} \;\; \mathbf{d}_i^{kT} \mathbf{d}_i^k = 1 \quad \forall i \tag{7.6}$$

where $\|\mathbf{X}\|_{1,2} = \sum_{i=1}^{d} \|\mathbf{x}_i\|_2$ is the sparse constraint on $\mathbf{X} = \left[\mathbf{X}^1, \ldots, \mathbf{X}^K\right]$. It enforces the sparse pattern for each column of $\mathbf{X}$ to be similar, which makes the learned dictionaries more robust to noise and occlusion. [11] also introduced a kernel version of their algorithm to deal with those non-linearly separable cases and improve the performance.

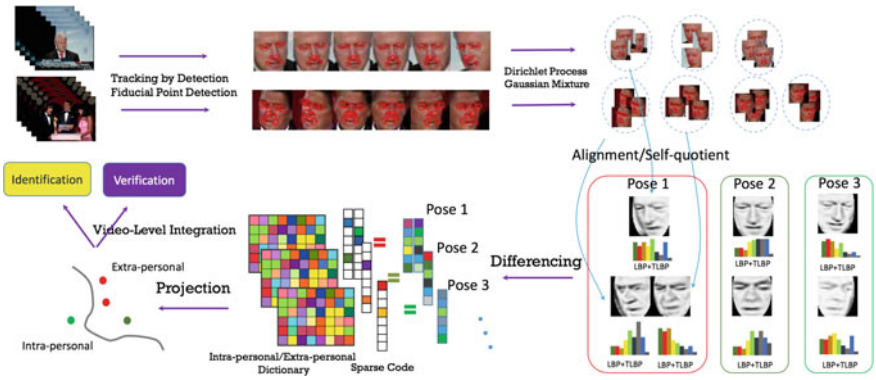

Fig. 7.2 Video-based face recognition using the intra/extrapersonal difference dictionary [16]

Du and Chellappa [16] proposed a video-based face recognition method based on intra/extrapersonal difference dictionary. Since pose variations often cause within-class variance to exceed between-class variance in face recognition, instead of learning dictionaries from the face features directly, pose-specific dictionaries are learned from those intra/extrapersonal difference features. Also, instead of learning generative dictionaries by merely minimizing the reconstruction error, it jointly learns dictionaries and discriminative projection matrices, which improves performance. The overall algorithm is shown in Fig. 7.2.

In their algorithm, given a video $\mathbf{V}$, faces are first detected and cropped from the videos by using a tracking algorithm. Fiducial points are then detected by a structural SVM approach. These cropped faces are aligned and clustered by the K-means algorithm into K clusters according to their poses. Then a given video can be characterized by its K cluster centers $\{\mathbf{v}_k, k = 1, 2, \ldots, K\}$ considered as representative images.

For the training videos, the intrapersonal difference features $\{\mathbf{x}_{In} = \mathbf{v}_i^m - \mathbf{v}_j^n, ID(\mathbf{V}_i) = ID(\mathbf{V}_j)\}$ and the extrapersonal ones $\{\mathbf{x}_{Ex} = \mathbf{v}_i^m - \mathbf{v}_j^n, ID(\mathbf{V}_i) \neq ID(\mathbf{V}_j)\}$ are employed to learn the dictionary $\mathbf{D}$ and the projection matrix $\mathbf{W}$ simultaneously for each pair of poses by solving the following Label-Consistent K-SVD problem (LC-K-SVD):

$$\min_{\mathbf{D},\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_2^2 + \mu\|\mathbf{Q} - \mathbf{B}\mathbf{A}\|_2^2 + \sigma\|\mathbf{F} - \mathbf{W}\mathbf{A}\|_2^2 + \lambda \sum_i \|\boldsymbol{\alpha}_i\|_1, \tag{7.7}$$

where $\mathbf{X} = \left[\mathbf{X}_{In}\ \mathbf{X}_{Ex}\right]$ is the concatenation of intrapersonal and extrapersonal features. The columns of $\mathbf{F} \in \mathbb{R}^{2\times N}$ are the corresponding labels (same or different), represented using the 1-of-K coding scheme. It enforces $\mathbf{W}$ to encode discriminative information from the sparse codes. $\mathbf{B} \in \mathbb{R}^{K\times d}$ is a linear transformation that encourages the samples from the same class to be reconstructed using the entries in the sub-dictionary of that class. $\mathbf{Q} \in \mathbb{R}^{K\times N}$ has a block diagonal form: The c-th block contains entry $\mathbf{Q}_{ij}$, $i \in v_c$, $j \in h_c$, where v_c are the indices of atoms from class c

(i.e., intrapersonal or extrapersonal) and h_c are the indices of training instances from class c. All the nonzero entries in $\mathbf{Q}$ are assigned with unit value. This problem can be converted to a typical K-SVD [3] objective function and solved using the same procedure.

At the testing stage, for every probe-gallery video pair $\{\mathbf{V}_p, \mathbf{V}g\}$, feature difference vectors $\{\mathbf{x}_{p,g}^{m,n} = \mathbf{v}_p^m - \mathbf{v}_g^n\}$ from each pair of poses are calculated. The sparse representation of $\mathbf{x}_{p,g}^{m,n}$ is obtained by solving $\boldsymbol{\alpha}_{p,g}^{m,n} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{1}{2}\|\mathbf{x}_{p,g}^{m,n} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$ using the learned dictionary $\mathbf{D}$ in the training stage. The similarity score for this video pair is then calculated as

$$s(p,g) = \sum_{m=1}^{M}\sum_{n=1}^{N} \mathbf{1}(\mathbf{t}_1\mathbf{W}\boldsymbol{\alpha}_{p,g}^{m,n} > \mathbf{t}_0\mathbf{W}\boldsymbol{\alpha}_{p,g}^{m,n})/MN \tag{7.8}$$

where $\mathbf{t}_0 = [0,1]^T$ and $\mathbf{t}_1 = [1,0]^T$ are the 1-of-K coding label for intrapersonal and extrapersonal class, respectively. For video-based recognition, the decision is made by $ID(\mathbf{V}_p) = \underset{g}{\operatorname{argmax}}\, s(p,g)$.

Some of the other sparse dictionary learning-based methods for video-based face recognition include [18, 32].

7.3 Manifold-Based Methods

In manifold-based methods, videos are usually modeled as image sets. These image sets are considered as the approximation of manifolds and the problem actually turns into looking for a discriminant distance metric between manifolds. The basic idea is shown in Fig. 7.3.

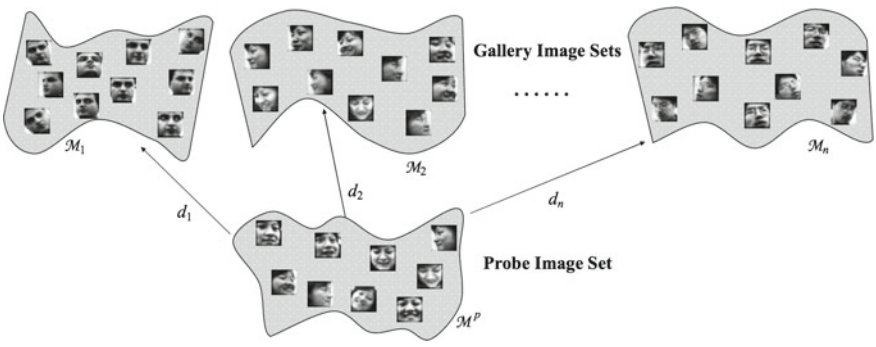

Fig. 7.3 Manifold-based face recognition [41]

In [41], the image set classification problem is based on the computation of manifold–manifold distance. It models the image sets cropped from videos as manifolds which consist of component linear subspaces. Then the manifold-to-manifold distance can be considered as the similarity between two videos.

Given face image set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ from a video, it is partitioned into a collection of disjoint Maximal Linear Patches $\{C_i\}$. Each video is considered as a manifold consisting of these local linear patches which can be obtained by Algorithm 1.

Algorithm 1 Local model construction.

1. Initialize that $i = 1$, $C_i = \emptyset$, $X_T = \emptyset$, $X_R = X$.
2. While($X_R \neq \emptyset$)

   2.1 Randomly select a seed point from X_R as $\mathbf{x}_1^{(i)}$, update $C_i = \{\mathbf{x}_1^{(i)}\}$, $X_R = X_R - \{\mathbf{x}_1^{(i)}\}$.

   2.2 For ($\forall \mathbf{x}_m^{(i)} \in C_i$)
   Identify each of its k-NNs $\mathbf{x}_c$ as *candidate*. If $\mathbf{x}_c$ satisfies simultaneously $\mathbf{x}_c \in X_R$ and

$$D_G(\mathbf{x}_c, \mathbf{x}_n^{(i)}) / D_E(\mathbf{x}_c, \mathbf{x}_n^{(i)}) < \theta, \ \forall \mathbf{x}_n^{(i)} \in C_i \tag{7.9}$$

   then update $C_i = C_i \cup \{\mathbf{x}_c\}$, $X_R = X_R - \{\mathbf{x}_c\}$, until no candidate point can be added into C_i.

   2.3 $X_T = \cup_{j=1}^{i} C_j$, $X_R = X - X_T$, $i = i + 1$, $C_i = \emptyset$.

Here, $D_E(\cdot)$ denotes the Euclidean distance and $D_G(\cdot)$ denotes the geodesic distance. Their ratio reflects the linear deviation of the local linear subspace. Threshold θ controls the degree of linear deviation. Thus larger θ implies fewer local structures but large linear deviation in each structure. After obtaining the local linear subspaces for each video, the manifold-manifold distance between two video manifolds M_1 and M_2 can be computed as

$$d(M_1, M_2) = \min_{C_i \in M_1} \min_{C_j \in M_2} d(C_i, C_j) \tag{7.10}$$

which is the distance between the closest local subspace pair.

Suppose $\mathbf{e}_i$, $\mathbf{e}_j$ and $\mathbf{P}_i \in \mathbb{R}^{D \times d_1}$, $\mathbf{P}_j \in \mathbb{R}^{D \times d_2}$ are the exemplars (means) and orthonormal bases of two subspaces C_i and C_j. $r = \min(d_i, d_j)$. The SVD of $\mathbf{P}_1^T \mathbf{P}_2$ is $\mathbf{P}_1^T \mathbf{P}_2 = \mathbf{Q}_{12} \mathbf{\Lambda} \mathbf{Q}_{21}^T$ and $\mathbf{\Lambda} = diag(\sigma_1, \ldots, \sigma_r)$. The distance between two local subspace is defined as

$$d(C_i, C_j) = (1 - \alpha) d_E(C_i, C_j) + \alpha d_V(C_i, C_j). \tag{7.11}$$

Here, $d_E(C_i, C_j) = \|\mathbf{e}_i\| \|\mathbf{e}_j\| / \mathbf{e}_i^T \mathbf{e}_j$ is called the *exemplar distance measure*, which measures how similar the two sets are. $d_V(C_i, C_j) = r / \sum_{k=1}^{r} \sigma_k$ is called the *variation distance measure* which measures how close the common variation modes of the two sets. By fusing these distance measures, the overall manifold–manifold

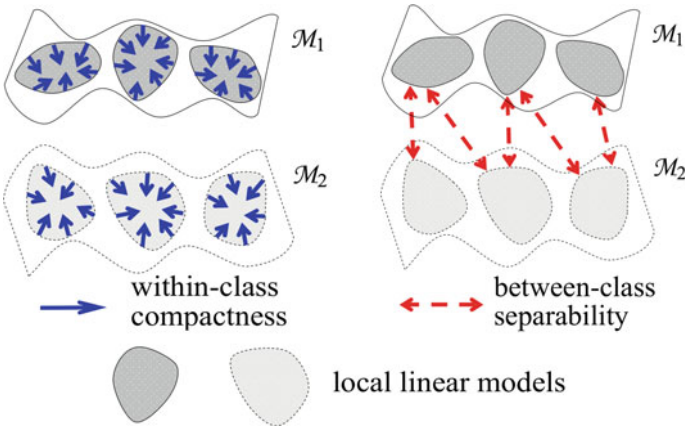

Fig. 7.4 Manifold discriminant analysis [39]

distance captures the difference of both average appearance and variation information between two sets.

Finally, for verification task, the similarity score between any gallery and probe video pair is the manifold–manifold distance between their corresponding manifolds. For identification task, decision is made by finding the video with the minimum manifold–manifold distance.

A manifold-based discriminative learning method called Manifold Discriminant Analysis for image set classification was proposed in [39]. It learns an embedding space where manifolds with different class labels are better separated and local data compactness within each manifold is enhanced. An overview of this method is shown in Fig. 7.4.

Like [41], given image sets considered as manifolds, local linear models are first extracted as $M_i = \{C_{i,k}\}$. The learning method is formulated as:

1. Two graphs are constructed, which are intrinsic graph G and penalty graph G'. In both graphs, nodes are all the images in the training set $\mathbf{X} = \{\mathbf{x}_m\}$. In G, nodes $\mathbf{x}_m$ and $\mathbf{x}_n$ are connected if $\mathbf{x}_m \in C_{i,k}$, $\mathbf{x}_n \in C_{j,l}$, $i = j$ and $k = l$, which means only the nodes come from the same local linear model are connected. In G', nodes $\mathbf{x}_m$ and $\mathbf{x}_n$ are connected if their class labels are different and $C_{i,k}$ is among the k'-nearest between-class neighbors of $C_{j,l}$.
2. The weight matrix $\mathbf{W} = \{w_{mn}\}$ for G is computed as

$$w_{mn} = \begin{cases} 1 & \text{if } \mathbf{x}_m \text{ and } \mathbf{x}_n \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{7.12}$$

$\mathbf{W}'$ for G' is computed in the same way. $\mathbf{D}$ and $\mathbf{D}'$ are diagonal matrices with diagonal elements $d_{mm} = \sum_n w_{mn}$ and $d'_{mm} = \sum_n w'_{mn}$. $\mathbf{L}_w = \mathbf{D} - \mathbf{W}$ and $\mathbf{L}_b = \mathbf{D}' - \mathbf{W}'$ are their Laplacian matrices, respectively.

3. A linear embedding $\mathbf{z} = \mathbf{V}^T\mathbf{x}$ based on linear projection is learned, where $\mathbf{V} \in \mathbb{R}^{d\times l}$ with $l \ll d$. For each column of $\mathbf{V}$, learning is fulfilled by maximizing the between-class scatter $S_b = \sum_{m,n} \|\mathbf{v}^T\mathbf{x}_m - \mathbf{v}^T\mathbf{x}_n\|^2 w'_{m,n} = 2\mathbf{v}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{v}$ and minimizing the within-class scatter $S_w = \sum_{m,n} \|\mathbf{v}^T\mathbf{x}_m - \mathbf{v}^T\mathbf{x}_n\|^2 w_{m,n} = 2\mathbf{v}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{v}$. This is equivalent to solving the optimization problem:

$$\underset{\mathbf{v}}{\text{maximize}} = \frac{S_b}{S_w} = \frac{\mathbf{v}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{v}}{\mathbf{v}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{v}} \tag{7.13}$$

The columns of the optimal $\mathbf{V}$ are the generalized eigenvectors corresponding to the l largest eigenvalues in

$$\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{v} = \lambda\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{v}. \tag{7.14}$$

Finally, for verification task, given two manifolds M_k and M_l, their distance is calculated as $d(M_k, M_l) = \min_{i,j} d(C_{i,k}, C_{j,l})$, which is same as the manifold-to-manifold distance proposed in [41]. $d(C_{i,k}, C_{j,l}) = \|\mathbf{e}_{i,k} - \mathbf{e}_{j,l}\|$ is the *empirical distance* between each pair of local linear models, where $\mathbf{e}_{i,k} = \frac{1}{N_{i,k}} \sum_{n=1}^{N_{i,k}} \mathbf{V}^T\mathbf{x}_{i,k}^n$ is the sample mean of $C_{i,k}$ and $\mathbf{e}_{j,l} = \frac{1}{N_{j,l}} \sum_{n=1}^{N_{j,l}} \mathbf{V}^T\mathbf{x}_{j,l}^n$ is the sample mean of $C_{j,l}$, both in the learned embedding space.

Similarly, Wang et al. [40] proposed a discriminative learning approach for image set classification by modeling the image set using its covariance matrix. The conceptual illustration of this method is shown in Fig. 7.5.

Given face images from a video, $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_n]$, the samples covariance of this image set is

$$\mathbf{C} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{s}_i - \bar{\mathbf{s}})(\mathbf{s}_i - \bar{\mathbf{s}})^T \tag{7.15}$$

where $\bar{\mathbf{s}}$ is the sample mean. The video is thus characterized by its covariance matrix $\mathbf{C}$. Since $\mathbf{C}$ is an SPD matrix, it lies on a Riemannian manifold. It is not easy to train a classifier on the manifold because most of the classic classifiers are

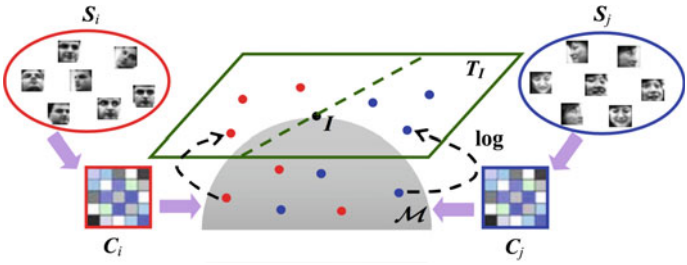

Fig. 7.5 Covariance discriminative learning [40]

designed for Euclidean metrics. In the paper, a distance metric, Log-Euclidean distance (LED) is introduced as $d_{LED}(\mathbf{C}_1, \mathbf{C}_2) = \|\log(\mathbf{C}_1) - \log(\mathbf{C}_2)\|_F$ where $\log(\cdot)$ here is the ordinary matrix logarithm operator. If $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$ is an SPD matrix, $\log(\mathbf{C}) = \mathbf{U}\log(\mathbf{\Sigma})\mathbf{U}^T$.

Given training videos $\{\mathbf{S}_i^{tr}\}$ from C different classes, first the covariance matrices $\{\mathbf{C}_i^{tr}\}$ are calculated. Then, two different learning methods are used:

1. Kernel LDA
   The KLDA optimization problem is:

$$\boldsymbol{\alpha}_{opt} = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \frac{\boldsymbol{\alpha}^T \mathbf{K}\mathbf{W}\mathbf{K}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{K}\mathbf{K}\boldsymbol{\alpha}}, \tag{7.16}$$

   where $\mathbf{K}_{ij} = k(\mathbf{S}_i^{tr}, \mathbf{S}_j^{tr}) = d_{LED}(\mathbf{C}_i^{tr}, \mathbf{C}_j^{tr})$. And $\mathbf{W}$ is defined as:

$$\mathbf{W}_{ij} = \begin{cases} 1/n_k & \text{if } \mathbf{S}_i^{tr}, \mathbf{S}_j^{tr} \text{ are both in the } k\text{th class} \\ 0 & \text{otherwise} \end{cases} \tag{7.17}$$

   and n_k is the number of videos in the kth class.
   The solution to (7.16) is the eigenvector corresponding to the largest eigenvalue of the problem $\mathbf{K}\mathbf{W}\mathbf{K}\boldsymbol{\alpha} = \lambda\mathbf{K}\mathbf{K}\boldsymbol{\alpha}$. Then given a testing video $\mathbf{S}^{te}$ with its covariance matrix $\mathbf{S}^{te}$, its projection in the $C-1$ dimensional discriminant subspace is:

$$\mathbf{z}^{te} = \mathbf{A}^T \mathbf{K}^{te} \tag{7.18}$$

   where $\mathbf{A} = \left[\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{C-1}\right]$ is the collection of $C-1$ largest eigenvectors and $\mathbf{K}^{te} = \left[k(\mathbf{S}_1^{tr}, \mathbf{S}^{te}), k(\mathbf{S}_2^{tr}, \mathbf{S}^{te}), \ldots\right]^T$. Nearest Neighbor classification in the discriminant subspace based on Euclidean distance is then performed.
2. Kernel PLS
   Different from KLDA, KPLS directly learns a regression model between training observations $\{\mathbf{S}_i^{tr}\}$ and their 1-of-K coding labels $\mathbf{Y}^{tr}$ (refer to [34] for more details). Then given testing video $\mathbf{S}^{te}$, its KPLS prediction is given by

$$\mathbf{y}^{te} = \mathbf{K}^{teT}\mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}^{tr} \tag{7.19}$$

   where $\mathbf{U}$ and $\mathbf{T}$ are regression parameters learned by KPLS, $\mathbf{K}$ and $\mathbf{K}^{te}$ are the same as in KLDA. The entry index with the largest response in $\mathbf{y}^{te}$ determines the label of the video.

Furthermore, [22] introduced a method that learns the projection metric directly on the Grassmann manifold rather than in Hilbert space. It performs a geometry-aware dimensionality reduction from the original Grassmann manifold to a lower dimensional, more discriminative Grassmann manifold. The method is demonstrated in Fig. 7.6.

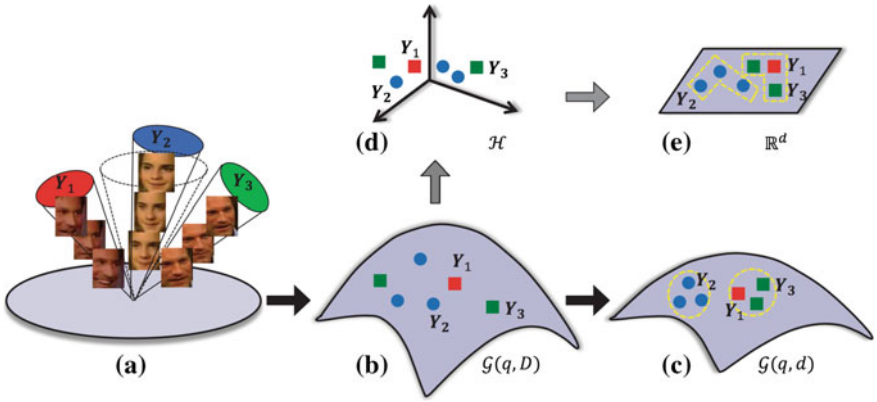

Fig. 7.6 Projection metric learning on Grassmann manifold [22]

Given face frames $\{\mathbf{X}_i\}$ from videos where $\mathbf{X}_i \in \mathbb{R}^{D\times n_i}$ describes a data matrix of the n_i frames from the ith video, $\mathbf{X}_i$ is represented by a q-dimensional linear subspace spanned by an orthonormal basis matrix $\mathbf{Y}_i \in \mathbb{R}^{D\times q}$. This is calculated by $\mathbf{X}_i\mathbf{X}_i^T \simeq \mathbf{Y}_i\mathbf{\Lambda}_i\mathbf{Y}_i^T$, $\mathbf{\Lambda}_i$ and $\mathbf{Y}_i$ correspond to the q largest eigenvalues and eigenvectors, respectively.

The linear subspace span $(\mathbf{Y}_i)$ lies on a Grassmann manifold $\mathcal{G}(q, D)$. It can be represented by the projection mapping $\Phi(\mathbf{Y}_i) = \mathbf{Y}_i\mathbf{Y}_i^T$ since there is a one-to-one mapping between each projection matrix and the point on the Grassmann manifold. The projection distance metric between $\mathbf{Y}_i\mathbf{Y}_i^T$ and $\mathbf{Y}_j\mathbf{Y}_j^T$ is defined as

$$d_p(\mathbf{Y}_i\mathbf{Y}_i^T, \mathbf{Y}_j\mathbf{Y}_j^T) = 2^{-1/2}\|\mathbf{Y}_i\mathbf{Y}_i^T - \mathbf{Y}_j\mathbf{Y}_j^T\|_F. \tag{7.20}$$

The method learns a mapping $f : \mathcal{G}(q, D) \to \mathcal{G}(q, d)$ which is defined as

$$f(\mathbf{Y}_i\mathbf{Y}_i^T) = \mathbf{W}^T\mathbf{Y}_i\mathbf{Y}_i^T\mathbf{W} = (\mathbf{W}^T\mathbf{Y}_i)(\mathbf{W}^T\mathbf{Y}_i)^T \tag{7.21}$$

where $\mathbf{W} \in \mathbb{R}^{D\times d}$ is the column full rank transformation matrix. Here, $\mathbf{W}^T\mathbf{Y}_i$ is not a orthonormal basis in general, which doesn't lie on a Grassmann manifold. Thus, $\mathbf{W}^T\mathbf{Y}_i$ is replace by $\mathbf{W}^T\mathbf{Y}'_i$, which is an orthonormal basis of $\mathbf{W}^T\mathbf{Y}_i$.

After transformation, the projection distance between $\mathbf{W}^T\mathbf{Y}'_i\mathbf{Y}'^T_i\mathbf{W}$ and $\mathbf{W}^T\mathbf{Y}'_j\mathbf{Y}'^T_j\mathbf{W}$ is

$$d_p^2(\mathbf{W}^T\mathbf{Y}'_i\mathbf{Y}'^T_i\mathbf{W}, \mathbf{W}^T\mathbf{Y}'_j\mathbf{Y}'^T_j\mathbf{W}) = \frac{1}{2}\|\mathbf{W}^T\mathbf{Y}'_i\mathbf{Y}'^T_i\mathbf{W} - \mathbf{W}^T\mathbf{Y}'_j\mathbf{Y}'^T_j\mathbf{W}\|_F^2 = \frac{1}{2}\mathrm{tr}(\mathbf{P}\mathbf{A}_{ij}\mathbf{A}_{ij}^T\mathbf{P}), \tag{7.22}$$

where $\mathbf{A}_{ij} = \mathbf{Y}'_i\mathbf{Y}'^T_i - \mathbf{Y}'_j\mathbf{Y}'^T_j$ and $\mathbf{P} = \mathbf{W}\mathbf{W}^T$, which is a rank-$d$ $D \times D$ PSD matrix.

The method learns $\mathbf{P}$ by minimizing the projection distances of any within-class subspace pairs and maximizing the projection distances of between-class subspace pairs. The corresponding objective function $J(\mathbf{P})$ is defined as

$$\mathbf{P}^* = \underset{\mathbf{P}}{\operatorname{argmin}} J(\mathbf{P}) = \underset{\mathbf{P}}{\operatorname{argmin}}(J_w(\mathbf{P}) - \alpha J_b(\mathbf{P})), \tag{7.23}$$

where α is the trade-off parameter between within-class scatter $J_w(\mathbf{P})$ and between-class scatter $J_b(\mathbf{P})$,

$$\begin{aligned} J_w(\mathbf{P}) &= \frac{1}{N_w} \sum_{i=1}^{m} \sum_{j:C_i=C_j} \operatorname{tr}(\mathbf{P}\mathbf{A}_{ij}\mathbf{A}_{ij}^T\mathbf{P}) = \operatorname{tr}(\mathbf{P}\mathbf{S}_w\mathbf{P}) \\ J_b(\mathbf{P}) &= \frac{1}{N_b} \sum_{i=1}^{m} \sum_{j:C_i\neq C_j} \operatorname{tr}(\mathbf{P}\mathbf{A}_{ij}\mathbf{A}_{ij}^T\mathbf{P})\operatorname{tr}(\mathbf{P}\mathbf{S}_b\mathbf{P}). \end{aligned} \tag{7.24}$$

Since $\mathbf{W}\mathbf{Y}_i'$ need to be orthogonal all the time, an algorithm is proposed to optimize $\mathbf{P}$ and solving $\mathbf{Y}_i'$ iteratively. In each iteration, $\mathbf{W}^T\mathbf{Y}_i$ is first decomposed into $\mathbf{W}^T\mathbf{Y}_i = \mathbf{Q}_i\mathbf{R}_i$ by QR-decomposition. $\mathbf{Y}_i'$ is normalized by $\mathbf{Y}_i' = \mathbf{Y}_i\mathbf{R}_i^{-1}$. Then $\mathbf{P}$ is solved using Riemannian Conjugate Gradient algorithm [1]. Finally, for verification task, given two videos, their projection distance in the low-dimensional space can be calculated using (7.22).

In [21] a hybrid metric learning method for image set-based face recognition was proposed, which is essentially an extension of [20]. The image sets are modeled simultaneously by mean, covariance matrix and Gaussian distribution and fused together for robustness. Another highlight of this paper is that the metrics are learned based on deep learning features. Combining set-based face recognition algorithm and the power of deep learning, the proposed method achieved state-of-the-art results in many challenging datasets. The conceptual illustration of the method is shown in Fig. 7.7.

Given an image set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^d$, the DCNN features $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ are first extracted. Here, according to [9], the DCNN network is trained on 256 by 256 pixel face images. The face images are normalized using detected eye positions. The network has 17 layers, including 14 convolution layers, 2 fully connected layers, and 1 soft-max layer. The training of the DCNN network consists of pretraining and fine-tuning. The pretraining is conducted on "Celebrities on the Web" (CFW) dataset [44]. The fine-tuning is carried using the training part of the given dataset. Finally, the output of the second fully connected layer of the trained DCNN network is used as the deep feature. All the network training and feature extraction are accomplished by the Caffe deep learning framework [24].

After the deep features are obtained, the first statistic, the sample mean is defined by $\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{y}_i$, which lies in Euclidean space $\mathbb{R}^d$. The second statistic, the covariance matrix, is defined by $\mathbf{C} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{y}_i - \mathbf{m})(\mathbf{y}_i - \mathbf{m})^T$, which lies on Riemannian manifold Sym_+^d. The third statistic, the Gaussian Mixture Model, is learned by Expectation Maximization algorithm. It can be written as

$$G = \sum_{i=1}^{M} w_i \mathcal{N}(\mathbf{y}|\tilde{\mathbf{m}}_i, \tilde{\mathbf{C}}_i), \tag{7.25}$$

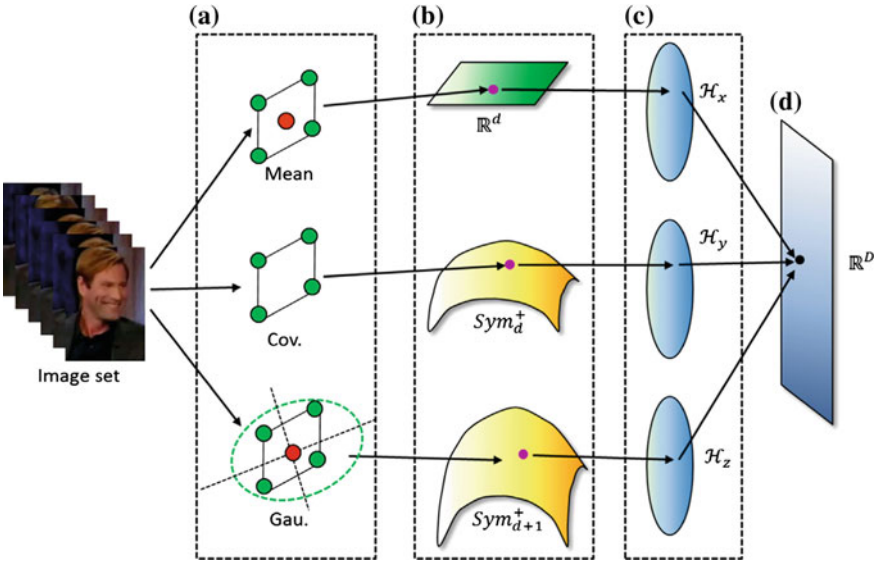

Fig. 7.7 Hybrid Euclidean-and-Riemannian metric learning [21]

where $\tilde{\mathbf{m}}_i$ and $\tilde{\mathbf{C}}_i$ are the mean and covariance matrix for the ith Gaussian component. According to the information geometry theory in [29], it can be embedded into Sym_+^{d+1} and represented by a $(d+1)\times(d+1)$-dimensional SPD matrix $\mathbf{P}$ as

$$\mathcal{N}(\tilde{\mathbf{m}}_i, \tilde{\mathbf{C}}_i) \sim \mathbf{P} = |\mathbf{Q}|^{-2/(d+1)} \begin{bmatrix} \mathbf{Q}\mathbf{Q}^T + \tilde{\mathbf{m}}_i\tilde{\mathbf{m}}_i^T & \tilde{\mathbf{m}}_i \\ \tilde{\mathbf{m}}_i^T & 1 \end{bmatrix}, \tag{7.26}$$

where $\tilde{\mathbf{C}} = \mathbf{Q}\mathbf{Q}^T$ and $|\mathbf{Q}| > 0$. For mean vectors, the linear kernel is directly used, which is

$$K_m(\mathbf{m}_i, \mathbf{m}_j) = \mathbf{m}_i^T\mathbf{m}_j. \tag{7.27}$$

For covariance matrices, the Log-Euclidean Distance is used, which is $d(\mathbf{C}_i, \mathbf{C}_j) = \|\log(\mathbf{C}_i) - \log(\mathbf{C}_j)\|_F$. It leads to the kernel

$$K_C(\mathbf{C}_i, \mathbf{C}_j) = \mathrm{tr}(\log(\mathbf{C}_i, \mathbf{C}_j)). \tag{7.28}$$

For GMMs, the LED metric is used as well. The kernel function is

$$K_G(\mathbf{G}_i, \mathbf{G}_j) = \sum_{a=1}^{M_a}\sum_{b=1}^{M_b} w_a w_b \mathrm{tr}(\log(\mathbf{P}_i^a)\log(\mathbf{P}_j^b))), \tag{7.29}$$

where $\mathbf{P}_i^a$ is the ath Gaussian component of the i th GMM.

Given training sets $\mathbf{X}_i$ and $\mathbf{X}_j$, let $\mathbf{\Phi}_i^r$ and $\mathbf{\Phi}_j^r$ be the high dimensional features in RKHS of the rth statistic feature. The distance metric is defined as

$$d_{A_r}(\mathbf{\Phi}_i^r, \mathbf{\Phi}_j^r) = \mathrm{tr}(\mathbf{A}_r(\mathbf{\Phi}_i^r - \mathbf{\Phi}_j^r)(\mathbf{\Phi}_i^r - \mathbf{\Phi}_j^r)^T), \tag{7.30}$$

where $\mathbf{A}_r$ is the learned Mahalanobis matrix for the rth statistic in the high dimensional RKHS ($r = 1, \ldots, 3$ here). Using the Information-Theoretic Metric Learning method proposed in [15], the objective function for learning $\{\mathbf{A}_r\}$ is formulated as

$$\begin{aligned} &\min_{\mathbf{A}_1 \geq 0, \ldots, \mathbf{A}_R \geq, \xi} \frac{1}{R} \sum_{r=1}^{R} D_{\ell d}(\mathbf{A}_r, \mathbf{A}_0) + \gamma D_{\ell d}(\mathrm{diag}(\boldsymbol{\xi}), \mathrm{diag}(\boldsymbol{\xi}_0)), \\ &s.t. \frac{\delta_{ij}}{R} \sum_{r=1}^{R} d_{\mathbf{A}_r}(\mathbf{\Phi}_i^r, \mathbf{\Phi}_j^r) \leq \boldsymbol{\xi}_{ij}, \ \forall i, j \end{aligned} \tag{7.31}$$

where $D_{\ell d}(\mathbf{A}_r, \mathbf{A}_0) = \mathrm{tr}(\mathbf{A}_r \mathbf{A}_0^{-1}) - \log\det(\mathbf{A}_r \mathbf{A}_0^{-1}) - d$, d is the dimensionality of the data. $\boldsymbol{\xi}$ is a vector of slack variables and is initialized to $\boldsymbol{\xi}_0$, where $\boldsymbol{\xi}_{0ij} = \boldsymbol{\delta}_{ij}\rho - \zeta\tau$, ρ is the threshold for distance comparison, τ is the margin, and ζ is the tuning scale of the margin. $\boldsymbol{\delta}_{ij} = \begin{cases} 1 & \text{if } \mathbf{X}_i \text{ and } \mathbf{X}_j \text{ come from the same class} \\ -1 & \text{otherwise} \end{cases}$

Learning $\mathbf{A}_r$ is equivalent to learning $\mathbf{W}_r$ such that $\mathbf{A}_r = \mathbf{W}_r \mathbf{W}_r^T$. By applying the kernel trick, explicit computation of $\mathbf{\Phi}^r$ can be avoided. Assume that every column of $\mathbf{W}_r$ is a linear combination of all the training samples in RKHS, $\mathbf{w}_k^r$ can be expressed by $\mathbf{w}_k^r = \sum_{j=1}^{N} \mathbf{u}_j^k \mathbf{\Phi}_j^r$, $\mathbf{u}^k$ are the expansion coefficients here. Let $\mathbf{U}_r = \left[\mathbf{u}^1, \ldots, \mathbf{u}^N\right]$, $\mathbf{W}_r = \mathbf{\Phi}^r \mathbf{U}_r$, instead of learning $\mathbf{W}_r$ directly, $\mathbf{U}_r$ can be learned. Then the objective function can be rewritten as

$$\begin{aligned} &\min_{\mathbf{B}_1 \geq 0, \ldots, \mathbf{B}_R \geq, \xi} \frac{1}{R} \sum_{r=1}^{R} D_{\ell d}(\mathbf{B}_r, \mathbf{B}_0) + \gamma D_{\ell d}(\mathrm{diag}(\boldsymbol{\xi}), \mathrm{diag}(\boldsymbol{\xi}_0)), \\ &s.t. \frac{\delta_{ij}}{R} \sum_{r=1}^{R} d_{\mathbf{B}_r}(\mathbf{K}_{\cdot i}^r, \mathbf{K}_{\cdot j}^r) \leq \boldsymbol{\xi}_{ij}, \ \forall i, j, \end{aligned} \tag{7.32}$$

where $\mathbf{B}_r = \mathbf{U}_r \mathbf{U}_r^T$ is the new Mahalanobis matrix. $d_{\mathbf{B}_r}(\mathbf{K}_{\cdot i}^r, \mathbf{K}_{\cdot j}^r) = \mathrm{tr}(\mathbf{B}_r(\mathbf{K}_{\cdot i}^r - \mathbf{K}_{\cdot j}^r)(\mathbf{K}_{\cdot i}^r - \mathbf{K}_{\cdot j}^r)^T)$. $\mathbf{K}_{\cdot i}^r$ is the ith column of $\mathbf{K}^r$. The proposed method adopted the cyclic Bregman projection method [10] to solve this problem.

After $\{\mathbf{B}_r\}_{r=1}^3$ are learned for all statistics, for verification task, given two image sets $\mathbf{X}_i$ and $\mathbf{X}_j$, their corresponding DCNN features are first calculated. Means, covariance matrices and GMMs are then computed. Then the kernels between these testing samples and the training samples are computed as $\mathbf{k}_i^r$ and $\mathbf{k}_j^r$. Finally, their distance is calculated by

$$d(\mathbf{X}_1, \mathbf{X}_2) = \sum_{r=1}^{3} d_{\mathbf{B}_r}(\mathbf{k}_i^r, \mathbf{k}_j^r) = \sum_{r=1}^{3} \mathrm{tr}(\mathbf{B}_r(\mathbf{k}_i^r - \mathbf{k}_j^r)(\mathbf{k}_i^r - \mathbf{k}_j^r)^T). \tag{7.33}$$

Besides the methods mentioned above, Wang et al. [42] proposed a face recognition method for image sets using Gaussian Mixture Model which lies on specific Riemannian manifold. Huang et al. [23] provided a image set-based metric learning method using Log-Euclidean metric on SPD manifold. Arandjelovic and Cipolla [6] built a pose-wise linear illumination manifold model for video-based face recognition. Arandjelovic and Cipolla [5] modeled the video faces by shape-illumination manifolds which are robust to different variations. Kim et al. [25] introduced canonical correlations between two subspaces for image set recognition. Huang et al. [19] proposed the Euclidean-to-Riemannian Metric for Point-to-Set Classification on Riemannian manifold.

7.4 Probabilistic Methods

Probabilistic methods provide flexibility so that the similarity scores can either be modeled as "distance" or as "likelihood".

In [27], a video-based face recognition algorithm based on probablistic appearance manifolds is introduced. The image set of a given object can be treated as a low-dimensional appearance manifold M_k in the image space. Given a testing image I, identity k^* is determined by finding the manifold M_k with minimal distance to I, which is

$$k^* = \underset{k}{\mathrm{argmin}}\, d_H(I, M_k), \tag{7.34}$$

where d_H denotes the L^2-Hausdorff distance between the image I and M_k. Probabilistically, let

$$P(k|I) = \frac{1}{\Lambda} \exp(-\frac{1}{2\sigma^2} d_H^2(I, M_k)), \tag{7.35}$$

where Λ is a normalization term. Thus, (7.34) turns into

$$k^* = \underset{k}{\mathrm{argmax}}\, P(k|I). \tag{7.36}$$

Since M_k is usually not known and can only be estimated by samples, $d_H(I, M_k)$ cannot be calculated directly. Let $p_{M_k}(x|I)$ be the probability that x is the point on M_k at minimal L^2 distance to I. Also, since the appearance manifold is complex and non-linear, it is decomposed into a collection of m simpler disjoint manifolds as $M_k = C^{k1} \cup \ldots \cup C^{km}$ where C^{ki} is called a pose manifold. Each pose manifold is further

approximated by an affine plane through PCA. $P(C^{ki}|I)$ denotes the probability that C^{ki} contains point x^* with minimal distance to I. Then

$$d_H(I, M_k) = \int_{M_k} d(x, I) p_{M_k}(x|I) dx = \sum_{i=1}^{m} P(C^{ki}|I) \int_{C^{ki}} d_H(x, I) p_{C^{ki}}(x|I) dx$$
$$= \sum_{i=1}^{m} P(C^{ki}|I) d_H(I, C^{ki}), \tag{7.37}$$

which is the average expected distance between I and each pose manifold C^{ki}. This is shown in Fig. 7.8.

For video-based face recognition, the temporal coherence between consecutive image frames can be exploited. As the example shown in Fig. 7.9, $\{I_t\}$ probably originate from M_B by looking at the whole sequence. But because of the appearance variations, some of the frames are closer to M_A. By considering the temporal coherence, the image-to-manifold can be estimated more robustly.

Given previous frames $I_{0:t-1}$ at time t, assume I_t and $I_{0:t}$ are independent given C_t^{ki}, C_t^{ki} and $I_{0:t-1}$ are independent given C_{t-1}^{ki}, $P(C_t^{ki}|I_t, I_{0:t-1})$ can be calculated as

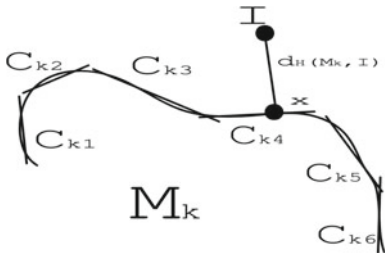

Fig. 7.8 $d_H(I, M_k)$ [27]

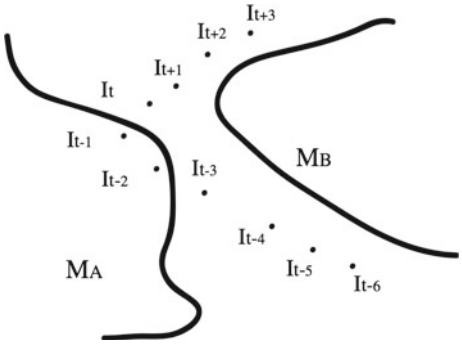

Fig. 7.9 Exploit temporal coherence [27]

$$P(C_t^{ki}|I_t, I_{0:t-1}) = \alpha P(I_t|C_t^{ki}, I_{0:t-1})P(C_t^{ki}|I_{0:t-1})$$
$$= \alpha P(I_t|C_t^{ki}) \sum_{j=1}^{m} P(C_t^{ki}|C_{t-1}^{kj}, I_{0:t-1})P(C_{t-1}^{kj}|I_{0:t-1})$$
$$= \alpha P(I_t|C_t^{ki}) \sum_{j=1}^{m} P(C_t^{ki}|C_{t-1}^{kj})P(C_{t-1}^{kj}|I_{t-1}, I_{0:t-2}), \tag{7.38}$$

where α is a normalization constant. $P(C_t^{ki}|C_{t-1}^{kj})$ is the probability of $x_t^* \in C^{ki}$ given $x_{t-1}^* \in C^{kj}$. Because of the temporal coherency between consecutive frames, x_{t-1}^* and x_t^* should have small geodesic distance on M_k. $P(C_t^{ki}|C_{t-1}^{kj})$ is thus related to their geodesic distance. Equation (7.38) can be computed recursively if $P(I_t|C_t^{ki})$ and $P(C_t^{ki}|C_{t-1}^{kj})$ $\forall i, j, t$ are known.

Given training image sets $\{S_k\}$ from videos, K-means algorithm is used to partition these sets into m disjoint subsets $\{S_{k1}, \ldots, S_{km}\}$. For each S_{ki}, a linear approximation L_{ki} of local manifold C^{ki} is obtained by PCA. $P(C^{ki}|C^{kj})$ is then calculated by

$$P(C^{ki}|C^{kj}) = \frac{1}{\Lambda_{kj}} \sum_{t=2}^{l} \delta(I_{t-1} \in S_{ki})\delta(I_t \in S_{kj}), \tag{7.39}$$

which is counting the actual transitions in the corresponding training set. Λ_{kj} is a normalization constant. $P(I|C^{ki})$ is calculated by

$$P(I|C^{ki}) = \frac{1}{\Lambda_{ki}} \exp(-\frac{1}{2\sigma^2} d_H(I, L_{ki})), \tag{7.40}$$

where L_{ki} is the low-dimensional linear approximation of manifold C^{ki}. Λ_{ki} is a normalization constant. $d_H(I, L_{ki}) = d_H(I, C^{ki})$ is the distance between I and C^{ki}. Finally, for identification task, given an image I_t from a testing video sequence $\{I_t\}$, $P(C_t^{ki}|I_t, I_{0:t-1}), \forall k, i$ are calculated recursively by (7.38). $d_H(I, M_k)$ is then obtained by (7.37). The decision is made by (7.36).

A probability distribution-based method for video-based face recognition was proposed in [7]. The Kullback–Leibler divergence is used as the distance measure between the distributions of videos. Given image sets collected from videos, Gaussian mixture models $\hat{p}$ are learned for each image set. This is done using the Expectation Maximization algorithm. EM is initialized by K-means clustering and constrained to diagonal covariance matrices. The number of components is selected according to the minimal description length criterion [8]. Then for each training and testing video pair (V^{te}, V^{tr}), the KL divergence between the learned distributions $\hat{p}^{te}$ and $\hat{p}^{tr}$ is used as the distance measure, which is

$$d(V^{te}, V^{tr}) = D_{KL}(\hat{p}^{te}||\hat{p}^{tr}) = \int \hat{p}^{te}(\mathbf{x}) \log \left(\frac{\hat{p}^{te}(\mathbf{x})}{\hat{p}^{tr}(\mathbf{x})} \right) d\mathbf{x}. \tag{7.41}$$

The KL divergence $D_{KL}(p||q)$ quantifies how well the distribution p describes samples from q. It is nonnegative and equal to zero if $p \equiv q$. Since the calculation of the KL divergence involves integration, there is no closed form when $\hat{p}^{te}$ and $\hat{p}^{tr}$ are GMMs. However, according to the law of large numbers, the KL divergence can still be approximated by sampling using Monte-Carlo simulation:

$$D_{KL}(\hat{p}^{te}||\hat{p}^{tr}) \approx \frac{1}{N}\sum_{k=1}^{N}\log\left(\frac{\hat{p}^{te}(\mathbf{x}_k)}{\hat{p}^{tr}(\mathbf{x}_k)}\right), \tag{7.42}$$

where $\mathbf{x}_k$ are samples drawn from distribution $\hat{p}^{te}$. Then for identification task, the similarity between every training and testing video pair is computed using (7.42).

Liu and Chen [28] proposed a Hidden Markov Models based method to perform video-based face recognition. When training, the statistics and the temporal information of training videos are learned by HMMs. During the recognition phase, the temporal characteristics of the testing videos are analyzed by the HMM corresponding to each subject. The decision is made by finding the highest likelihood scores provided by the HMMs.

A continuous HMM model is defined as the triplet $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$. $\mathbf{A} = \{a_{ij}\}$ is the transition probability matrix, where $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$, $1 \le i,j \le N$. $\mathbf{B} = \{b_i(\mathbf{o})\}$ is the observation probability density function, where $b_i(\mathbf{o}) = \sum_{k=1}^{M} c_{ik}$ $\mathcal{N}(\mathbf{o}; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$. $\boldsymbol{\pi} = \{\pi_i\}$ is the initial state distribution, where $\pi_i = P(q_1 = S_i)$, $1 \le i \le N$. Here $\mathbf{S} = \{S_1, S_2, \ldots, S_N\}$ is the set of states in the model. $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$ are the observations and $\mathbf{Q} = \{q_1, q_2, \ldots, q_T\}$ are the corresponding hidden state variables. Given state S_i, $b_i(\mathbf{o})$ is a Gaussian Mixture Model with M Gaussians. c_{ik}, $\boldsymbol{\mu}_{ik}$ and $\boldsymbol{\Sigma}_{ik}$ are the mixture coefficient, mean and covariance for the kth Gaussian, respectively.

Given training videos, the face images are first projected to a low-dimensional space using PCA. Then each video is modeled as an HMM with these low-dimensional features as observations $\mathbf{O}$, which is shown in Fig. 7.10.

The estimation for HMM parameter $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ is as follows:

1. $\mathbf{A}$, $\mathbf{B}$, and $\boldsymbol{\pi}$ are initialized (observations are clustered into M Gaussians. c_{ik}^0, $\boldsymbol{\mu}_{ik}^0$ and $\boldsymbol{\Sigma}_{ik}^0$ are estimated for each Gaussian). $n = 0$.
2. Do.

   2.1. Reestimate λ using the expectation maximization algorithm, in order to maximize the likelihood $p(\mathbf{O}|\lambda)$. The reestimation is defined as

$$\pi_i^{n+1} = \frac{P(\mathbf{O}, q_1 = i|\lambda^n)}{p(\mathbf{O}|\lambda^n)} \tag{7.43}$$

$$a_{ij}^{n+1} = \frac{\sum_{t=1}^{T} p(\mathbf{O}, q_{t-1} = i, q_t = j|\lambda^n)}{\sum_{t=1}^{T} p(\mathbf{O}, q_{t-1} = i|\lambda^n)} \tag{7.44}$$

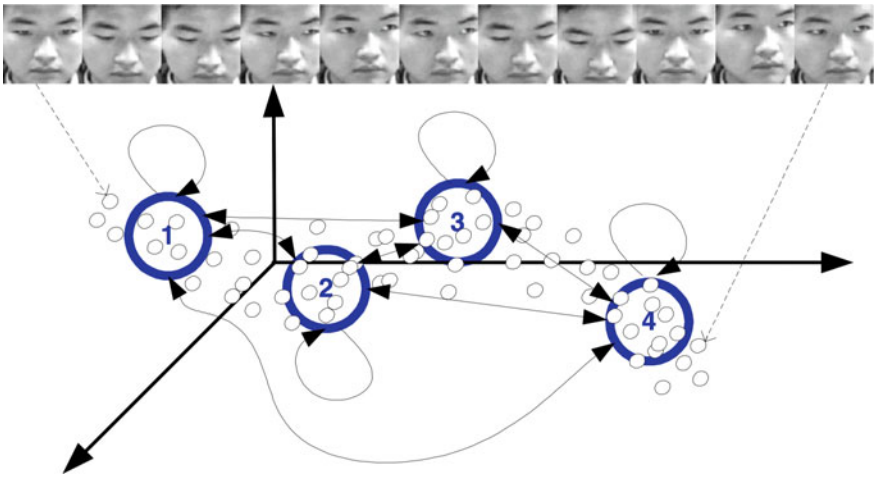

Fig. 7.10 Temporal HMM for modeling face sequences [28]

$$c_{ik}^{n+1} = \frac{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = k|\mathbf{O}, \lambda^n)}{\sum_{t=1}^{T}\sum_{k=1}^{M} P(q_t = i, m_{q_t t} = k|\mathbf{O}, \lambda^n)} \tag{7.45}$$

$$\boldsymbol{\mu}_{ik}^{n+1} = \frac{\sum_{t=1}^{T} \mathbf{o}_t P(q_t = i, m_{q_t t} = k|\mathbf{O}, \lambda^n)}{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = k|\mathbf{O}, \lambda^n)} \tag{7.46}$$

$$\boldsymbol{\Sigma}_{ik}^{n+1} = (1-\alpha)\mathbf{C} + \alpha \frac{\sum_{t=1}^{T} (\mathbf{o}_t - \boldsymbol{\mu}_{ik}^{n+1})(\mathbf{o}_t - \boldsymbol{\mu}_{ik}^{n+1})^T P(q_t = i, m_{q_t t} = k|\mathbf{O}, \lambda^n)}{P(q_t = i, m_{q_t t} = k|\mathbf{O}, \lambda^n)} \tag{7.47}$$

where $m_{q_t t}$ indicates the mixture component of state q_t and time t. $\mathbf{C}$ is a general model for the variance of all videos. α is a weighting factor, which prevents the estimated $\boldsymbol{\Sigma}$ to be singular.

2.2 $n = n + 1$

3. Until $p(\mathbf{O}|\lambda)$ converges.

For identification task, after the HMM models $\{\lambda_c^{tr}\}$ are estimated for training videos, given a testing video, the face images are projected onto the same low-dimensional space as the training samples and obtain the testing observation $\mathbf{O}^{te}$. Then the likelihood score $p(\mathbf{O}^{te}|\lambda_c^{tr})$ of the observation given the training testing HMM models are computed. The identification decision is made by $p = \text{argmax}_c$ $p(\mathbf{O}^{te}|\lambda_c^{tr})$, which finds the highest likelihood score.

In addition to the methods discussed above, Zhou et al. [45] introduced an appearance-adaptive model-based on particle filter to realize robust visual tracking and recognition. Zhou et al. [46] proposed a time series based method for video-based face recognition. Arandjelovic and Cipolla [4] provided another method based on kernelized distribution-to-distribution distance. Wang et al. [43] introduced a probablistic nearest neighbor search method for image set classification.

7.5 Geometrical Model-Based Methods

Geometrical model-based methods construct certain geometrical models for faces in the videos. Then the texture map of the faces are projected on to these models and features are extracted. The recognition will based on these features. The models can vary from the simple spherical head models to the human-specific 3D head models. Geometrical model based methods are more robust to illumination and pose variations because they exploit the geometrical structures from the faces.

Sankaranarayanan and Chellappa [17] proposed a novel feature for robust video-based face recognition in camera networks. It is developed using the spherical harmonic representation of the face texture mapped onto a spherical head model. Spherical harmonics are a set of orthonormal basis functions defined over the unit sphere, and can be used to linearly expand any square-integrable function on $\mathbb{S}^2$ as

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_{lm} Y_{lm}(\theta, \phi), \tag{7.48}$$

where $Y_{lm}(\cdot, \cdot)$ defines the SH basis function of degree $l \geq 0$ and order $m \in (-l, -l+1, \ldots, l-1, l)$. f_{lm} is the coefficient associated with the basis function Y_{lm} for the function f. The spherical coordinate system is used here. $\theta \in (0, \pi)$ and $\phi \in (0, 2\pi)$ are the zenith and azimuth angles, respectively. There are $2l + 1$ basis functions for a given order l. The SH basis function for degree l and order m has the following form (shown in Fig. 7.11):

$$Y_{lm}(\theta, \phi) = K_{lm} P_l^m(\cos\theta) \mathbf{e}^{jm\phi}, \tag{7.49}$$

where K_{lm} denotes a normalization constant such that

$$\int_0^{\pi} \int_0^{2\pi} Y_{lm} Y_{lm}^* d\phi d\theta = 1. \tag{7.50}$$

Here, $P_{lm}(x)$ are the associated Legendre functions. As with Fourier expansion, the SH expansion coefficients f_l^m can be computed as

$$f_l^m = \int_{\theta} \int_{\phi} f(\theta, \phi) Y_l^m(\theta, \phi) d\theta d\phi. \tag{7.51}$$

Given two multiview videos, the head centers in these videos are first obtained using a multiview tracking algorithm proposed in the paper. Then a spherical head model for each head is build. The SH spectrum features are extracted from the texture map projected on the models from all views. The projection of the texture map is shown in Fig. 7.12.

These features are projected into a reproducing kernel Hilbert space (RKHS), which is performed via an Radial Basis Function (RBF) kernel. The limiting

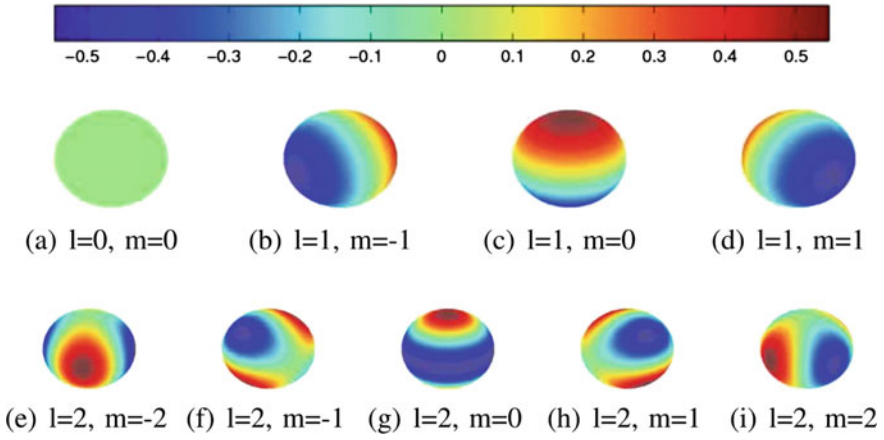

Fig. 7.11 Visualization of the first three degree of Spherical Harmonics [17]

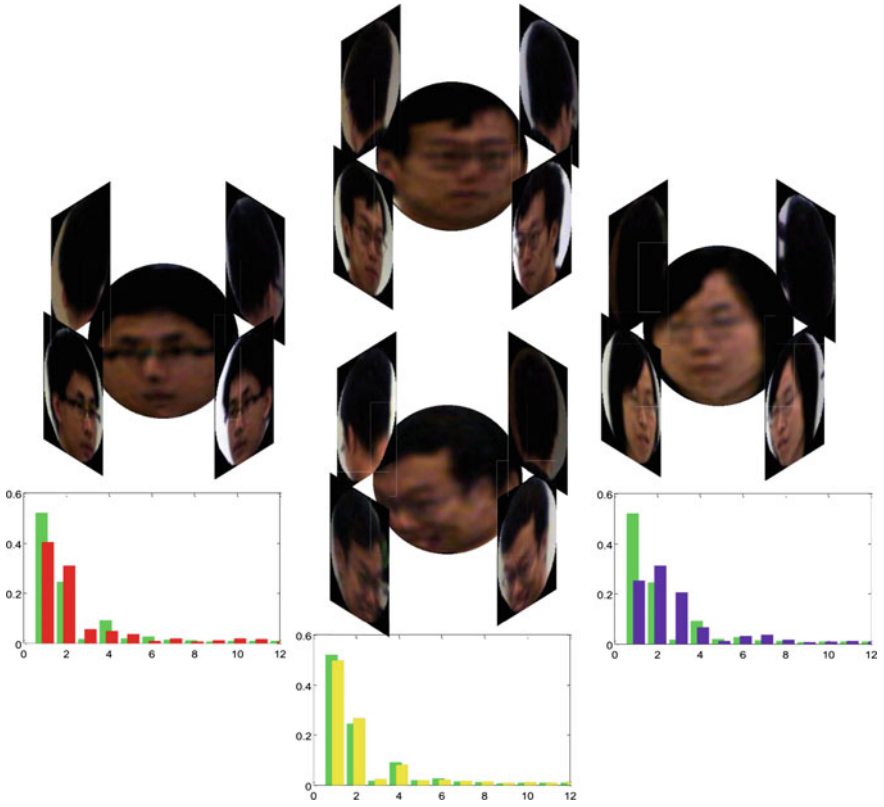

Fig. 7.12 Texture map projection [17]

Bhattacharyya distance between these probability distributions in RKHS (assume to be Gaussian) is considered as the distance measure. The limiting Bhattacharyya distance in this case is

$$D = \frac{1}{8}(\alpha_{11} + \alpha_{22} - 2\alpha_{12}), \tag{7.52}$$

where

$$\alpha_{ij} = \boldsymbol{\mu}_i^T (\frac{1}{2}\mathbf{C}_i + \frac{1}{2}\mathbf{C}_j)^{-1} \boldsymbol{\mu}_j. \tag{7.53}$$

$\boldsymbol{\mu}_i$ and $\mathbf{C}_i$ are the means and covariance matrices in RKHS which cannot be directly calculated. Denote the Gram matrix as $\mathbf{K}_{ij}$, where $i,j \in \{1,2\}$ are the indices of videos. $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$ are centered by

$$\mathbf{K}'_{ii} = \mathbf{J}_i^T \mathbf{K}_{ii} \mathbf{J}_i, \mathbf{J}_i = N_i^{-\frac{1}{2}} (\mathbf{I}_N - \mathbf{s}\mathbf{1}^T), \tag{7.54}$$

where $\mathbf{s} = N_i^{-1}\mathbf{1}$, $\mathbf{1}$ is a $N_i \times 1$ vector of 1s and N_i is the number of features from video i. Then α_{ij} is calculated by

$$\alpha_{ij} = \mathbf{s}_i^T \mathbf{K}_{ij} \mathbf{s}_j - \mathbf{s}_i^T \begin{bmatrix} \mathbf{K}_{i1} & \mathbf{K}_{i2} \end{bmatrix} \mathbf{B} \begin{bmatrix} \mathbf{K}_{j1} \\ \mathbf{K}_{j2} \end{bmatrix} \mathbf{s}_j, \tag{7.55}$$

where

$$\mathbf{B} = \mathbf{P}\mathbf{L}^{-1}\mathbf{P}, \mathbf{L} = \mathbf{P}^T \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \mathbf{P} \tag{7.56}$$

and

$$\mathbf{P} = \begin{bmatrix} \sqrt{\frac{1}{2}}\mathbf{J}_1\mathbf{V}_1 & 0 \\ 0 & \sqrt{\frac{1}{2}}\mathbf{J}_2\mathbf{V}_2 \end{bmatrix} \tag{7.57}$$

$\mathbf{V}_i$ is the matrix which stores the first r eigenvectors of $\mathbf{K}'_{ii}$ (i.e., corresponding to the r largest eigenvalues). For identification and verification tasks, the similarity between the two set of features is measured by the computed limiting Bhattacharyya distance between them.

Park and Jain [33] also provided a video-based face recognition method which reconstructs 3D face models from the videos and recognizes faces at frontal view.

7.6 Dynamical Model-Based Methods

Dynamical model-based methods are sequence-based methods. They consider videos as dynamical systems with video frames as the observation of these systems. The advantage of these methods is that the extra temporal information is exploited.

Dynamical models are often used to represent motions or activities, but there are some publications that use dynamical models for face recognition.

In [2], the video-to-video face recognition problem is transferred into a dynamical system identification and classification problem. Videos are modeled by dynamical systems. Here, the ARMA model is used for the dynamical system. The ARMA model is defined as

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{v}(t) \tag{7.58}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{w}(t), \tag{7.59}$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{y}(t)$ is the observation. $\mathbf{A}$ and $\mathbf{C}$ are transition matrix and observation matrix, respectively. The system is driven by the IID process $\mathbf{v}(t)$. $\mathbf{w}(t)$ is the observation noise.

Suppose $\mathbf{v}(t) \sim \mathcal{N}(0, \mathbf{Q})$ and $\mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{R})$, given a video sequence $\mathbf{Y}^\tau = \left[\mathbf{y}(1), \ldots, \mathbf{y}(\tau)\right]$, (7.59) can be rewritten as

$$\mathbf{Y}^\tau = \mathbf{C}\mathbf{X}^\tau + \mathbf{W}^\tau, \tag{7.60}$$

where $\mathbf{X}$ and $\mathbf{W}$ are similarly defined. Then the model parameters can be estimated by

$$\hat{\mathbf{C}}(\tau) = \mathbf{U} \tag{7.61}$$

$$\hat{\mathbf{X}}(\tau) = \mathbf{\Sigma}\mathbf{V}^T \tag{7.62}$$

$$\hat{\mathbf{A}}(\tau) = \mathbf{\Sigma}\mathbf{V}^T\mathbf{D}_1\mathbf{V}(\mathbf{V}^T\mathbf{D}_2\mathbf{V})^{-1}\mathbf{\Sigma}^{-1} \tag{7.63}$$

$$\hat{\mathbf{Q}}(\tau) = \frac{1}{\tau}\sum_{t=1}^{\tau} \hat{\mathbf{v}}(t)\hat{\mathbf{v}}^T(t), \tag{7.64}$$

where $\mathbf{Y}^\tau = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD of $\mathbf{Y}^\tau$. $\mathbf{D}_1 = \begin{bmatrix} 0 & 0 \\ \mathbf{I}_{\tau-1} & 0 \end{bmatrix}$ and $\mathbf{D}_2 = \begin{bmatrix} \mathbf{I}_{\tau-1} & 0 \\ 0 & 0 \end{bmatrix}$. $\hat{\mathbf{v}}(t) = \hat{\mathbf{x}}(t+1) - \hat{\mathbf{A}}(\tau)\hat{\mathbf{x}}(t)$.

Given video pairs $\mathbf{V}_1$ and $\mathbf{V}_2$, their model parameters M_1 and M_2 are first estimated, respectively. Then the distance between two ARMA models is calculated by

$$d_M(M_1, M_2)^2 = \ln \prod_{i=1}^{n} \frac{1}{\cos^2 \theta_i} \tag{7.65}$$

$$d_g(M_1, M_2) = \sin \theta_{max} \tag{7.66}$$

$$d_f(M_1, M_2)^2 = 2\sum_{i=1}^{n} \sin^2 \theta_i, \tag{7.67}$$

where $d_M(M_1, M_2)$ is the Martin distance, $d_g(M_1, M_2)$ is the gap distance and $d_f(M_1, M_2)$ is the distance based on Frobenius norm. θ_i's are the subspace angles between M_1

and M_2 (see [13] for more details). Different distances can be chosen for different scenarios or fused together to improve the performance.

Turaga [37] also considered videos as ARMA models and treated each video as a point on the Grassmann manifold for recognition.

7.7 Conclusion and Future Directions

As we saw in this chapter, most of the modeling approaches for video-based face recognition focus on how to define the similarity scores (or the "distances") between videos. Sparse coding-based methods model videos as dictionaries and use reconstruction error as the similarity score. Manifold-based methods use special kernels between manifolds as the similarity. Probabilistic methods are more flexible. The similarity scores can be the KL divergence between distributions, or the expected distance under some certain distributions. Dynamical model-based methods consider videos as dynamical systems. The similarity scores are the distance between two systems on a certain manifold. Geometrical model-based methods are slightly different from the others since their main objective is to construct geometrical models from videos and project texture maps onto them.

Since deep learning is becoming increasingly important recently, one of the future directions for video-based face recognition will be the classic methods combined with deep learning-based methods. The special statistical and geometrical properties of deep features will lead to new modeling techniques. Another possible direction would be to build 3D DCNN networks, where the convolutions are applied through the time-axis as well, in order to capture the temporal information between consecutive frames. Also, thanks to the fast developments in deep learning-based detection and landmark extraction techniques, face detection and alignment are becoming more and more precise, which can provide geometrical model-based methods with improved performance.

Acknowledgements This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

1. Absil PA, Mahony R, Sepulchre R (2007) Optimization algorithms on matrix manifolds. Princeton University Press, Princeton, NJ, USA
2. Aggarwal G, Chowdhury A, Chellappa R (2004) A system identification approach for video-based face recognition. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol 4, pp 175–178
3. Aharon M, Elad M, Bruckstein A (2006) K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311–4322
4. Arandjelovic O, Cipolla R (2004) Face recognition from face motion manifolds using robust kernel resistor-average distance. In: Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04, p 88
5. Arandjelovic O, Cipolla R (2006) Face Recognition from Video Using the Generic Shape-Illumination Manifold. In: Proceedings of Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006, Part IV. Springer, Berlin Heidelberg, pp 27–40
6. Arandjelovic O, Cipolla R (2009) A pose-wise linear illumination manifold model for face recognition using video. Comput Vis Image Underst 113(1):113–125
7. Arandjelovic O, Shakhnarovich G, Fisher J, Cipolla R, Darrell T (2005) Face recognition with image sets using manifold density divergence. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, CVPR 2005. vol 1 (2005), pp 581–588
8. Barron A, Rissanen J, Yu B (1998) The minimum description length principle in coding and modeling. IEEE Trans Inf Theory 44(6):2743–2760
9. Beveridge J, Zhang H, Draper B, Flynn P, Feng Z, Huber P, Kittler J, Huang Z, Li S, Li Y, Kan M, Wang R, Shan S, Chen X, Li H, Hua G, Struc V, Krizaj J, Ding C, Tao D, Phillips P (2015) Report on the fg 2015 video person recognition evaluation. In: 2015 11th IEEE international conference and workshops on Automatic Face and Gesture Recognition (FG), vol 1, pp 1–8
10. Bregman L (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Comput Math Math Phys 7(3):200–217
11. Chen YC, Patel V, Shekhar S, Chellappa R, Phillips P (2013) Video-based face recognition via joint sparse representation. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp 1–8
12. Chen YC, Patel VM, Phillips PJ, Chellappa R (2012) Dictionary-Based Face Recognition from Video. In: Computer Vision— ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 Oct 2012, Proceedings. Springer, Berlin, pp 766–779
13. Cock KD, Moor BD (2002) Subspace angles between arma models. Syst Control Lett 46(4):265–270
14. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), CVPR '05, vol 1. IEEE Computer Society, Washington, DC, USA, pp 886–893
15. Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. Proceedings of the 24th International Conference on Machine Learning., ICML '07ACM, New York, NY, USA, pp 209–216
16. Du M, Chellappa R (2014) Video-based face recognition using the intra-personal/extra-personal difference dictionary. In: Proceedings of the British Machine Vision Conference. BMVA Press
17. Du M, Sankaranarayanan A, Chellappa R (2014) Robust face recognition from multi-view videos. IEEE Trans Image Process 23(3):1105–1117
18. Hu Y, Mian A, Owens R (2011) Sparse approximated nearest points for image set classification. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 121–128

19. Huang Z, Wang R, Shan S, Chen X (2014) Learning euclidean-to-riemannian metric for point-to-set classification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1677–1684 (2014)
20. Huang Z, Wang R, Shan S, Chen X (2015) Hybrid Euclidean-and-Riemannian Metric Learning for Image Set Classification. In: Computer Vision—ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, 1–5 Nov 2014, Revised Selected Papers, Part III. Springer International Publishing, Cham, pp 562–577
21. Huang Z, Wang R, Shan S, Chen X (2015) Face recognition on large-scale video in the wild with hybrid euclidean-and-riemannian metric learning. Patt Recogn 48(10):3113–3124
22. Huang Z, Wang R, Shan S, Chen X (2015) Projection metric learning on grassmann manifold with application to video based face recognition. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 140–149
23. Huang Z, Wang R, Shan S, Li X, Chen X (2015) Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In: Blei D, Bach E (eds.) Proceedings of the 32nd International Conference on Machine Learning (ICML-15), JMLR Workshop and Conference Proceedings, pp 720–729
24. Jia Y, Shelhamer E, Donahue J, Karayev S, Long, J, Girshick, R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093
25. Kim TK, Kittler J, Cipolla R (2007) Discriminative learning and recognition of image set classes using canonical correlations. IEEE Trans Pattern Anal Mach Intell 29(6):1005–1018
26. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Pereira E, Burges C, Bottou L, Weinberger K (eds) Advances in neural information processing systems, vol 25. Curran Associates, Inc., pp 1097–1105
27. Lee KC, Ho J, Yang MH, Kriegman D (2003) Video-based face recognition using probabilistic appearance manifolds. Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'03. IEEE Computer Society, Washington, DC, USA, pp 313–320
28. Liu X, Chen T (2003) Video-based face recognition using adaptive hidden markov models. Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'03. IEEE Computer Society, Washington, DC, USA, pp 340–345
29. Lovri M, Min-Oo M, Ruh EA (2000) Multivariate normal distributions parametrized as a riemannian symmetric space. J Multivariate Anal 74(1):36–48
30. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60(2):91–110
31. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
32. Ortiz E, Wright A, Shah M (2013) Face recognition in movie trailers via mean sequence sparse representation-based classification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3531–3538
33. Park U, Jain AK (2007) 3D Model-based face recognition in video. In: Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, 27–29 Aug 2007. Proceedings. Springer, Berlin, pp 1085–1094
34. Rosipal R, Kramer N (2006) Overview and recent advances in partial least squares. In: Proceedings of the 2005 International Conference on Subspace, Latent Structure and Feature Selection, SLSFS'05. Springer, Berlin, pp 34–51
35. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556**
36. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
37. Turaga P, Veeraraghavan A, Srivastava A, Chellappa R (2011) Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. IEEE Trans Pattern Anal Mach Intell 33(11):2273–2286

38. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vision 57(2):137–154
39. Wang R, Chen X (2009) Manifold discriminant analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009, CVPR 2009. pp 429–436
40. Wang R, Guo H, Davis L, Dai Q (2012) Covariance discriminative learning: a natural and efficient approach to image set classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2496–2503
41. Wang R, Shan S, Chen X, Gao W (2008) Manifold-manifold distance with application to face recognition based on image set. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, pp 1–8
42. Wang W, Wang R, Huang Z, Shan S, Chen X (2015) Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2048–2057
43. Wang W, Wang R, Shan S, Chen X (2015) Probabilistic nearest neighbor search for robust classification of face image sets. In: 2015 11th IEEE international conference and workshops on automatic Face and Gesture Recognition (FG) (Vol. 1, pp. 1–7)
44. Zhang X, Zhang L, Wang XJ, Shum HY (2012) Finding celebrities in billions of web images. IEEE Trans Multimedia 14(4):995–1007
45. Zhou S, Chellappa R, Moghaddam B (2004) Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans Image Process 13(11):1491–1506
46. Zhou S, Krueger V, Chellappa R (2003) Probabilistic recognition of human faces from video. Comput Vis Image Understand 91(12):214–245. Special Issue on Face Recognition
47. Zhu X, Ramanan D (2012) Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2879–2886