Chapter 13
Face Tracking and Recognition in Video

Rama Chellappa, Ming Du, Pavan Turaga, and Shaohua Kevin Zhou

13.1 Introduction

Faces are expressive three dimensional objects. Information useful for recognition tasks can be found both in the geometry and texture of the face and also facial motion. While geometry and texture together determine the 'appearance' of the face, motion encodes behavioral cues such as idiosyncratic head movements and gestures which can potentially aid in recognition tasks. Traditional face recognition systems have relied on a gallery of still images for learning and a probe of still images for recognition. While the advantage of using motion information in face videos has been widely recognized, computational models for video based face recognition have only recently gained attention.

In this chapter, we consider applications where one is presented with a video sequence—either in a single camera setting or a multi-camera setting—and the goal is to recognize the person in the video. The gallery could consist of either still-images or could be videos themselves.

Video is a rich source of information in that it can lead to potentially better representations by offering more views of the face. Further, the role of facial motion for face perception has been well documented. Psychophysical studies [26] have

R. Chellappa (✉) · M. Du · P. Turaga
Department of Electrical and Computer Engineering, Center for Automation Research, University of Maryland, College Park, MD 20742, USA
e-mail: rama@umiacs.umd.edu

M. Du
e-mail: mingdu@umiacs.umd.edu

P. Turaga
e-mail: pturaga@umiacs.umd.edu

S.K. Zhou
Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540, USA
e-mail: kzhou@scr.siemens.com

S.Z. Li, A.K. Jain (eds.), *Handbook of Face Recognition*,
DOI 10.1007/978-0-85729-932-1_13,

found evidence that when both structure and dynamics information is available, humans tend to rely more on dynamics under nonoptimal viewing conditions (such as low spatial resolution, harsh illumination conditions etc.). Dynamics also aids in recognition of familiar faces [31]. If one were to ignore temporal dependencies, a video sequence can be considered as a collection of still images; so still-image-based recognition algorithms can always be applied. The properties of video sequences that can be exploited are (1) temporal correlations, (2) idiosyncratic dynamic information, and (3) availability of multiple views. Video thus proves useful in various tasks—it can be used to generate better appearance models, mitigate effects of noncooperative viewing conditions, localize a face using motion, model facial behavior for improved recognition, generate better models of face shape from multiple views, etc.

The rest of the chapter is organized as follows. In Sect. 13.2, we describe the utility of videos in enhancing performance of image-based recognition tasks. In Sect. 13.3, we discuss a joint tracking-recognition framework that allows for using the motion information in a video to better localize and identify the person in the video using still galleries. In Sect. 13.4, we discuss how to jointly capture facial appearance and dynamics to obtain a parametric representation for video-to-video recognition. In Sect. 13.5, we discuss recognition in multi-camera networks where the probe and gallery both consist of multi-camera videos. Finally in Sect. 13.6, we present concluding remarks and directions for future research.

13.2 Utility of Video

Frame-Based Fusion An immediate possible utilization of temporal information for video-based face recognition is to fuse the results obtained by a 2D face recognition algorithm on each frame of the sequence. The video sequence can be seen as an unordered set of images to be used for both training and testing phases. During testing one can use the sequence as a set of probes, each of them providing a decision regarding the identity of the person. Appropriate fusion techniques can then be applied to provide the final identity. Perhaps the most frequently used fusion strategy in this case is majority voting [24, 34].

In [28], Park et al. adopt three matchers for frame-level face recognition: FaceVACS, PCA and correlation. They use the sum rule (with min-max normalization) to fuse results obtained from the three matchers and the maximum rule to fuse results of individual frames. In [21], the concept of identity surface is proposed to represent the hyper-surface formed by projecting face patterns of an individual to the feature vector space parameterized with respect to pose. This surface is learned from gallery videos. In testing stage, model trajectories are synthesized on the identity surfaces of enrolled subjects after the pose parameters of probe video have been estimated. Every point on the trajectory corresponds to a frame of the video and trajectory distance is defined as a weighted sum of point-wise distances. The model trajectory that yields minimum distance to the probe video's trajectory gives the final identification result. Based on the result that images live approximately in a bilinear

space of motion and illumination variables, Xu et al. estimate these parameters for each frame of a probe video sequence with a registered 3D generic face model [38]. They then replace the generic model with a person-specific model of each subject in the gallery to synthesize video sequences with the estimated illumination and motion parameters. Frame-wise comparison is conducted between the synthesized videos and the probe video. A synthesized video is considered as a winner if one of its frames yield the smallest distance across all frames and all the subjects in the gallery.

Ensemble Matching Without recourse to modeling temporal dynamics, one can consider a video as an ensemble of images. Recent methods have focused on utilizing image-ensembles for object and face recognition [4, 15, 17, 41]. For example, it was shown by Jacobs et al. that the illumination cone of a convex Lambertian surface can be approximated by a 9-dimensional linear subspace [5]. Motivated by this, the set of face images of the same person under varying illumination conditions is frequently modeled as a linear subspace of 9-dimensions [19]. In such applications, an object 'category' consists of image-sets of several 'instances'. A common approach in such applications is to approximate the image-space of a single face/object under these variations as a linear subspace [14, 15]. A simplistic model for object appearance variations is then a mixture of subspaces. In [41], Zhou and Chellappa study the problem of measuring similarity between two ensembles by projecting the data into a Reproducing Kernel Hilbert Space (RKHS). The ensemble distance is then characterized as the probabilistic distance (Chernoff distance, Bhattacharyya distance, Kullback–Leibler (KL) divergence etc.) in RKHS.

Appearance Modeling Most face recognition approaches rely on a model of appearance for each individual subject. The simplest appearance model is a static image of the person. Such appearance models are rather limited in utility in video-based face recognition tasks where subjects may be imaged under varying viewpoints, illuminations, expressions etc. Thus, instead of using a static image as an appearance model, a sufficiently long video which encompasses several variations in facial appearance can lend itself to building more robust appearance models. Several methods have been proposed for extracting more descriptive appearance models from videos. For example, a facial video is considered as a sequence of images sampled from an 'appearance manifold' in [20]. In principle, the appearance manifold of a subject contains all possible appearances of the subject. In practice, the appearance manifold for each person is estimated from training data of videos. For ease of estimation, the appearance manifold is considered to be a collection of affine subspaces, where each subspace encodes a set of similar appearances of the subject. Temporal variations of appearances in a given video sequence are then modeled as transitions between the appearance subspaces. This method is robust to large appearance changes if sufficient 3D view variations and illumination variations are available in the training set. Further, the tracking problem can be integrated into this framework by searching for a bounding-box on the test image that minimizes the distance of the cropped region to the learnt appearance manifold.

In a related work, [3] represents the appearance variations due to shape and illumination on human faces, using the assumption that the 'shape-illumination manifold' of all possible illuminations and head poses is generic for human faces. This means that the shape-illumination manifold can be estimated using a set of subjects exclusive of the test set. They show that the effects of face shape and illumination can be learnt using Probabilistic PCA from a small, unlabeled set of video sequences of faces in randomly varying lighting conditions. Given a novel sequence, the learnt model is used to decompose the face appearance manifold into albedo and shape-illumination manifolds, producing the classification decision using robust likelihood estimation.

13.3 Still Gallery vs. Video Probes

Following Phillips et al. [29], we define a still-to-video scenario as follows. The gallery consists of still facial templates, and the probe set consists of video sequences containing the facial region. Though significant research has been conducted on still-to-still recognition, research efforts on still-to-video recognition are relatively fewer owing to the following challenges [40] in typical surveillance applications: poor video quality, significant illumination and pose variations, and low image resolution. Most existing video-based recognition systems [9, 40] attempt the following: The face is first detected and then tracked over time. Only when a frame satisfying certain criteria (size, pose) is acquired, recognition is performed using still-to-still recognition technique. For this, the face part is cropped from the frame and transformed or registered using appropriate transformations. This tracking-then-recognition approach attempts to resolve uncertainties in tracking and recognition sequentially and separately and requires a criterion for selecting good frames and estimation of parameters for registration. Also, still-to-still recognition does not effectively exploit temporal information.

We will assume that a certain feature representation for spatio-temporal patterns of moving faces has been made. We will also assume that there exists a set of hidden parameters, constituting the state vector, which govern how the spatio-temporal patterns evolve in time. The state vector encodes information such as motion parameters which can be used for tracking and identity parameters that can be used for recognition. Given a set of features, we need inference algorithms for estimating these hidden parameters. The three basic components of the model are the following.

- A motion equation governing the kinematic behavior of the tracking motion vector
- An identity equation governing the temporal evolution of the identity variable
- An observation equation establishing a link between the motion vector and the identity variable

We denote the gallery as $\mathscr{I} = \{I_1, I_2, \ldots, I_N\}$, indexed by the identity variable n, which lies in a finite sample space $\mathscr{N} = \{1, 2, \ldots, N\}$. And we denote the identity,

motion vector, and the observation at time t as n_t, θ_t and z_t, respectively. Using the Sequential Importance Sampling (SIS) [12, 18, 22] technique, the joint posterior distribution of the motion vector and the identity variable [i.e., $p(n_t, \theta_t \mid z_{0:t})$] is estimated at each time instant and then propagated to the next time instant governed by motion and identity equations. The marginal distribution of the identity variable [i.e., $p(n_t \mid z_{0:t})$] is estimated to provide the recognition result.

The recognition model consists of the following components.

- *Motion equation*
  In its most general form, the motion model can be written as

$$\theta_t = g(\theta_{t-1}, u_t); \quad t \geq 1 \tag{13.1}$$

  where u_t is noise in the motion model, whose distribution determines the motion state transition probability $p(\theta_t \mid \theta_{t-1})$. The function $g(.,.)$ characterizes the evolving motion, and it could be a function learned offline or given a priori. One of the simplest choices is an additive function (i.e., $\theta_t = \theta_{t-1} + u_t$), which leads to a first-order Markov chain.

  The choice of θ_t is dependent on the application. Affine motion parameters are often used when there is no significant pose variation available in the video sequence. However, if a three-dimensional (3D) face model is used, 3D motion parameters should be used accordingly.
- *Identity equation*
  Assuming that the identity does not change as time proceeds, we have

$$n_t = n_{t-1}; \quad t \geq 1. \tag{13.2}$$

  In practice, one may assume a small transition probability between identity variables to increase the robustness.
- *Observation equation*
  By assuming that the transformed observation is a noise-corrupted version of some still template in the gallery, the observation equation can be written as

$$\mathscr{T}_{\theta_t}\{z_t\} = I_{n_t} + v_t; \quad t \geq 1 \tag{13.3}$$

  where v_t is observation noise at time t, whose distribution determines the observation likelihood $p(z_t \mid n_t, \theta_t)$, and $\mathscr{T}_{\theta_t}\{z_t\}$ is a transformed version of the observation z_t. This transformation could be geometric, photometric, or both. However, when confronting difficult scenarios, one should use a more sophisticated likelihood function as discussed in [43].
- *Statistical independence*
  We assume statistical independence between all noise variables u_t and v_t.
- *Prior distribution*
  The prior distribution $p(n_0 \mid z_0)$ is assumed to be uniform.

$$p(n_0 \mid z_0) = \frac{1}{N}; \quad n_0 = 1, 2, \ldots, N. \tag{13.4}$$

In our experiments, $p(\theta_0|z_0)$ is assumed to be Gaussian: its mean comes from an initial detector or manual input and its covariance matrix is manually specified.

Using an overall state vector $x_t = (n_t, \theta_t)$, (13.1) and (13.2) can be combined into one state equation (in a normal sense) that is completely described by the overall state transition probability

$$p(x_t \mid x_{t-1}) = p(n_t \mid n_{t-1})p(\theta_t \mid \theta_{t-1}). \tag{13.5}$$

Given this model, our goal is to compute the posterior probability $p(n_t \mid z_{0:t})$. It is in fact a probability mass function (PMF), as n_t only takes values from $\mathcal{N} = \{1, 2, \ldots, N\}$, as well as a marginal probability of $p(n_t, \theta_t \mid z_{0:t})$, which is a mixed distribution. Therefore, the problem is reduced to computing the posterior probability.

13.3.1 Posterior Probability of Identity Variable

The evolution of the posterior probability $p(n_t \mid z_{0:t})$ as time proceeds is interesting to study, as the identity variable does not change by assumption [i.e., $p(n_t \mid n_{t-1}) = \delta(n_t - n_{t-1})$, where $\delta(.)$ is a discrete impulse function at zero, that is, $\delta(x) = 1$ if $x = 0$; otherwise $\delta(x) = 0$]. Using time recursion, Markov properties, and statistical independence embedded in the model, one can derive the following expressions:

$$\begin{aligned}
&p(n_{0:t}, \theta_{0:t} \mid z_{0:t}) \\
&\quad = p(n_{0:t-1}, \theta_{0:t-1} \mid z_{0:t-1}) \frac{p(z_t \mid n_t, \theta_t)p(n_t \mid n_{t-1})p(\theta_t \mid \theta_{t-1})}{p(z_t \mid z_{0:t-1})} \\
&\quad = p(n_0, \theta_0 \mid z_0) \prod_{i=1}^{t} \frac{p(z_i \mid n_i, \theta_i)p(n_i \mid n_{i-1})p(\theta_i \mid \theta_{i-1})}{p(z_i \mid z_{0:i-1})} \\
&\quad = p(n_0 \mid z_0)p(\theta_0 \mid z_0) \prod_{i=1}^{t} \frac{p(z_i \mid n_i, \theta_i)\delta(n_i - n_{i-1})p(\theta_i \mid \theta_{i-1})}{p(z_i \mid z_{0:i-1})}.
\end{aligned} \tag{13.6}$$

Therefore, by marginalizing over $\theta_{0:t}$ and $n_{0:t-1}$, we obtain the marginal posterior distribution for the identity j.

$$\begin{aligned}
p(n_t = j \mid z_{0:t}) = p(n_0 = j \mid z_0) &\int_{\theta_0} \cdots \int_{\theta_t} p(\theta_0 \mid z_0) \\
&\times \prod_{i=1}^{t} \frac{p(z_i \mid j, \theta_i)p(\theta_i \mid \theta_{i-1})}{p(z_i \mid z_{0:i-1})} \, d\theta_t \cdots d\theta_0.
\end{aligned} \tag{13.7}$$

Thus, $p(n_t = j \mid z_{0:t})$ is determined by the prior distribution $p(n_0 = j \mid z_0)$ and the product of the likelihood functions $\prod_{i=1}^{t} p(z_i \mid j, \theta_i)$. If a uniform prior is assumed, then $\prod_{i=1}^{t} p(z_i \mid j, \theta_i)$ is the only determining factor.

13.3.2 Sequential Importance Sampling Algorithm

Consider a general time series state space model fully determined by (1) the overall state transition probability $p(x_t \mid x_{t-1})$; (2) the observation likelihood $p(z_t \mid x_t)$; and (3) prior probability $p(x_0)$ and statistical independence among all noise variables. We wish to compute the posterior probability $p(x_t \mid z_{0:t})$.

If the model is linear with Gaussian noise, it is analytically solvable by a Kalman filter, which essentially propagates the mean and variance of a Gaussian distribution over time. For nonlinear and non-Gaussian cases, an extended Kalman filter and its variants have been used to arrive at an approximate analytic solution [2]. Recently, the SIS technique, a special case of the Monte Carlo method [12, 18, 22] has been used to provide a numerical solution and propagate an arbitrary distribution over time.

The essence of the Monte Carlo method is to represent an arbitrary probability distribution $\pi(x)$ closely by a set of discrete samples. It is ideal to draw i.i.d. samples $\{x^{(m)}\}_{m=1}^{M}$ from $\pi(x)$. However, it is often difficult to implement, especially for nontrivial distributions. Instead, a set of samples $\{x^{(m)}\}_{m=1}^{M}$ is drawn from an importance function $g(x)$; then a weight

$$w^{(m)} = \pi\left(x^{(m)}\right)/g\left(x^{(m)}\right) \tag{13.8}$$

is assigned to each sample. This technique is called importance sampling. It can be shown [22] that the importance sample set $\mathscr{S} = \{(x^{(m)}, w^{(m)})\}_{m=1}^{M}$ is properly weighted to the target distribution $\pi(x)$. To accommodate a video, importance sampling is used in a sequential fashion, which leads to SIS. SIS propagates $\mathscr{S}_{t-1}$ according to the sequential importance function, say $g(x_t \mid x_{t-1})$, and calculates the weight using

$$w_t = w_{t-1}\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1})/g(x_t \mid x_{t-1}). \tag{13.9}$$

In the CONDENSATION algorithm, $g(x_t \mid x_{t-1})$ is taken to be $p(x_t \mid x_{t-1})$ and (13.9) becomes

$$w_t = w_{t-1}\, p(z_t \mid x_t). \tag{13.10}$$

In fact, (13.10) is implemented by first resampling the sample set $\mathscr{S}_{t-1}$ according to w_{t-1} and then updating the weight w_t using $p(z_t \mid x_t)$. For a complete description of the SIS method, refer to Doucet et al. [12] and Liu and Chen [22].

In the context of video-based face recognition, the posterior probability $p(n_t, \theta_t \mid z_{0:t})$ is represented by a set of indexed and weighted samples

$$\mathscr{S}_t = \left\{\left(n_t^{(m)}, \theta_t^{(m)}, w_t^{(m)}\right)\right\}_{m=1}^{M} \tag{13.11}$$

with n_t as the above index. We can sum the weights of the samples belonging to the same index n_t to obtain a proper sample set $\{n_t, \beta_{n_t}\}_{n_t=1}^{N}$ with respect to the posterior PMF $p(n_t|z_{0:t})$. Straightforward implementation of the CONDENSATION algorithm for simultaneous tracking and recognition is not efficient in terms of its computational load. We refer the reader to [42] for a more detailed treatment of this issue.

13.3.3 Experimental Results

In this section, we describe the still-to-video scenarios used in our experiments and model choices, followed by a discussion of results. Two databases are used in the still-to-video experiments.

Database-0 was collected outside a building. We mounted a video camera on a tripod and requested subjects to walk straight toward the camera to simulate typical scenarios for visual surveillance. Database-0 includes one face gallery and one probe set. The probe contains 12 videos, one for each individual.

In Database-1, we have video sequences with subjects walking in a slant path toward the camera. There are 30 subjects, each having one face template. The face gallery is shown in Fig. 13.1. The probe contains 30 video sequences, one for each subject. Figure 13.1 shows some frames extracted from one probe video. As far as imaging conditions are concerned, the gallery is quite different from the probe, especially in terms of lighting. This is similar to the "FC" test protocol of the FERET test [29]. These images/videos were collected as part of the HumanID project by the National Institute of Standards and Technology and University of South Florida researchers.

13.3.3.1 Results for Database-0

We now consider affine transformation. Specifically, the motion is characterized by $\theta = (a_1, a_2, a_3, a_4, t_x, t_y)$, where $\{a_1, a_2, a_3, a_4\}$ are deformation parameters and $\{t_x, t_y\}$ are 2D translation parameters. It is a reasonable approximation because there is no significant out-of-plane motion as the subjects walk toward the camera. Regarding the photometric transformation, only the zero-mean-unit-variance operation is performed to compensate partially for contrast variations. The complete transformation $\mathscr{T}_\theta\{z\}$ is processed as follows. Affine transform z using $\{a_1, a_2, a_3, a_4\}$, crop out the interested region at position $\{t_x, t_y\}$ with the same size as the still template in the gallery, and perform the zero-mean-unit-variance operation.

A time-invariant first-order Markov Gaussian model with constant velocity is used for modeling motion transition. Given that the subject is walking toward the camera, the scale increases with time. However, under perspective projection, this increase is no longer linear, causing the constant-velocity model to be not optimal. However, experimental results show that so long as the samples of θ can cover the motion, this model is sufficient.

The likelihood measurement is simply set as a "truncated" Laplacian:

$$p_1(z_t \mid n_t, \theta_t) = \mathsf{L}\big(\big\| \mathscr{T}_{\theta_t}\{z_t\} - I_{n_t} \big\|; \sigma_1, \tau_1\big) \tag{13.12}$$

where $\|.\|$ is sum of absolute distance, σ_1 and λ_1 are manually specified, and

$$\mathsf{L}(x; \sigma, \tau) = \begin{cases} \sigma^{-1} \exp(-x/\sigma) & \text{if } x \le \tau\sigma, \\ \sigma^{-1} \exp(-\tau) & \text{otherwise.} \end{cases} \tag{13.13}$$

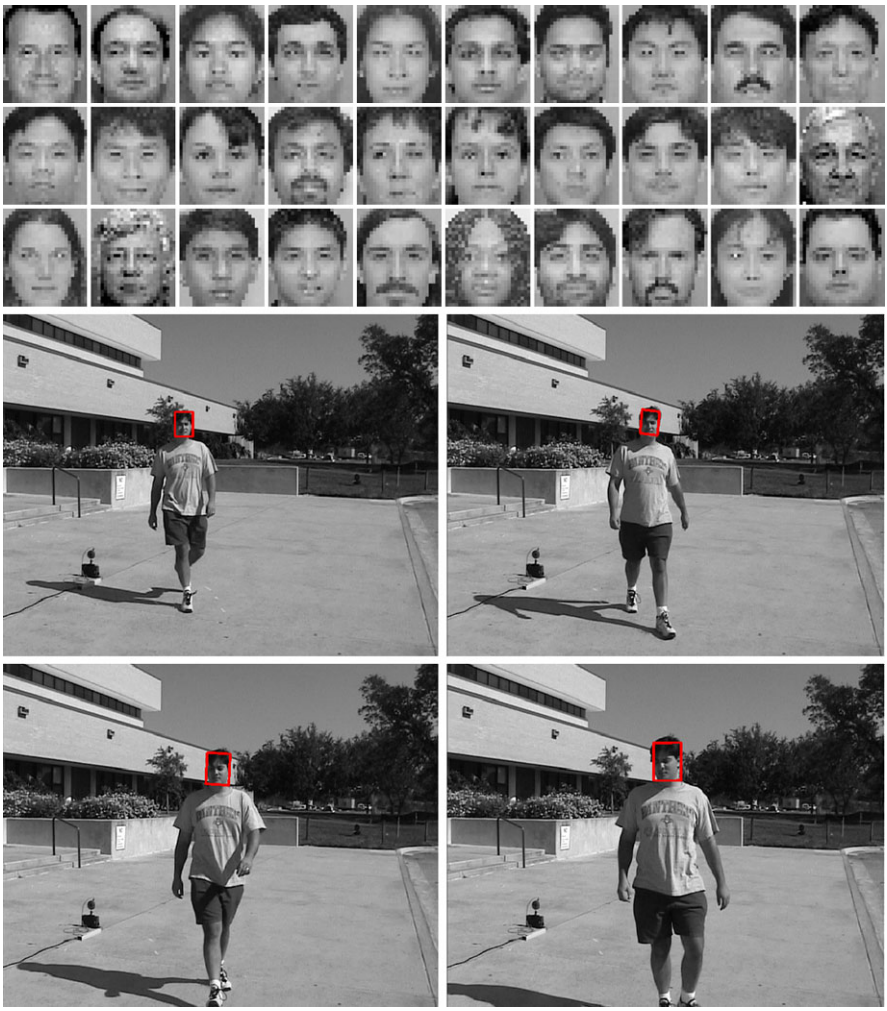

Fig. 13.1 Database-1. *First row*: the face gallery with image size of 30×26. *Second* and *third rows*: four frames in one probe video with image size of 720×480; the actual face size ranged from approximately 20×20 in the first frame to 60×60 in the last frame. Note the significant illumination variations between the probe and the gallery

Gaussian distribution is widely used as a noise model, accounting for sensor noise and digitization noise among others. However, given the observation equation: $v_t = \mathscr{T}_{\theta_t}\{z_t\} - I_{n_t}$, the dominant part of v_t becomes the high-frequency residual if θ_t is not proper; and it is well known that the high-frequency residual of natural images is more Laplacian-like. The "truncated" Laplacian is used to give a "surviving" chance for samples to accommodate abrupt motion changes.

Table 13.1 summarizes the average recognition performance and computational time of the CONDENSATION and the proposed algorithm when applied to

Table 13.1 Recognition performance of algorithms when applied to Database-0

Algorithm	CONDENSATION	Proposed
Recognition rate within top one match	100%	100%
Time per frame	7 seconds	0.5 seconds

Table 13.2 Performances of algorithms when applied to Database-1

Case	Case 1	Case 2	Case 3	Case 4	Case 5
Tracking accuracy	83%	87%	93%	100%	NA
Recognition within top 1 match	13%	NA	83%	93%	57%
Recognition within top 3 matches	43%	NA	97%	100%	83%

Database-0. Both algorithms achieved 100% recognition rate with top match. However, the proposed algorithm is more than 10 times faster than the CONDENSATION algorithm.

13.3.3.2 Results on Database-1

Case 1: Tracking and Recognition Using Laplacian Density We first investigate the performance using the same setting as described in Sect. 13.3.3.1. Table 13.2 shows that the recognition rate is poor: only 13% are correctly identified using the top match. The main reason is that the "truncated" Laplacian density is not able to capture the appearance difference between the probe and the gallery, indicating a need for more effective appearance modeling. Nevertheless, the tracking accuracy is reasonable, with 83% successfully tracked because we are using multiple face templates in the gallery to track the specific face in the probe video. After all, faces in both the gallery and the probe belong to the same class of human face, and it seems that the appearance change is within the class range.

Case 2: Pure Tracking Using Laplacian Density In Case 2, we measure the appearance change within the probe video as well as the noise in the background. To this end, we introduce a dummy template T_0, a cut version in the first frame of the video. Define the observation likelihood for tracking as

$$q(z_t \mid \theta_t) = \mathsf{L}\big(\big\| \mathscr{T}_{\theta_t}\{z_t\} - T_0 \big\|; \sigma_2, \tau_2\big) \tag{13.14}$$

where σ_2 and τ_2 are set manually. The other setting, such as motion parameter and model, is the same as in Case 1. We still can run the CONDENSATION algorithm to perform pure tracking. Table 13.2 shows that 87% are successfully tracked by this simple tracking model, which implies that the appearance within the video remains similar.

Case 3: Tracking and Recognition Using Probabilistic Subspace Density As mentioned in Case 1, we need a new appearance model to improve the recognition accuracy. Of the many approaches suggested in the literature, we decided to use the approach suggested by Moghaddam et al. [25] because of its computational efficiency and high recognition accuracy. However, here we model only the intrapersonal variations.

We need at least two facial images for one identity to construct the intrapersonal space (IPS). Apart from the available gallery, we crop out the second image from the video ensuring no overlap with the frames actually used in probe videos.

We then fit a probabilistic subspace density on top of the IPS. It proceeds as follows: A regular PCA is performed for the IPS. Suppose the eigensystem for the IPS is $\{(\lambda_i, e_i)\}_{i=1}^{d}$, where d is the number of pixels and $\lambda_1 \geq \cdots \geq \lambda_d$. Only top r principal components corresponding to top r eigenvalues are then kept while the residual components are considered isotropic. The density is written as follows

$$\mathsf{Q}(x) = \left\{ \frac{\exp(-\frac{1}{2}\sum_{i=1}^{r} \frac{y_i^2}{\lambda_i})}{(2\pi)^{r/2} \prod_{i=1}^{r} \lambda_i^{1/2}} \right\} \left\{ \frac{\exp(-\frac{\varepsilon^2}{2\rho})}{(2\pi\rho)^{(d-r)/2}} \right\} \tag{13.15}$$

where the principal components y_i, the reconstruction error ε^2, and the isotropic noise variance ρ are defined as

$$y_i = e_i^{\mathrm{T}} x, \qquad \varepsilon^2 = \|x\|^2 - \sum_{i=1}^{r} y_i^2, \qquad \rho = (d-r)^{-1} \sum_{i=r+1}^{d} \lambda_i. \tag{13.16}$$

It is easy to write the likelihood as follows:

$$p_2(z_t \mid n_t, \theta_t) = \mathsf{Q}_{\mathrm{IPS}}(\mathscr{T}_{\theta_t}\{z_t\} - I_{n_t}). \tag{13.17}$$

Table 13.2 lists the performance using this new likelihood measurement. It turns out that the performance is significantly better than in Case 1, with 93% tracked successfully and 83% correctly recognized within the top match. If we consider the top three matches, 97% are correctly identified.

Case 4: Tracking and Recognition Using Combined Density In Case 2, we studied appearance changes within a video sequence. In Case 3, we studied the appearance change between the gallery and the probe. In Case 4, we attempt to take advantage of both cases by introducing a combined likelihood defined as follows.

$$p_3(z_t \mid n_t, \theta_t) = p_2(z_t \mid n_t, \theta_t) q(z_t \mid \theta_t). \tag{13.18}$$

Again, all other settings are the same as in Case 1. We now obtain the best performance so far: no tracking error, 93% are correctly recognized as the first match, and no error in recognition when the top three matches are considered.

Case 5: Still-to-Still Face Recognition We also performed an experiment for still-to-still face recognition. We selected the probe video frames with the best frontal face view (i.e., biggest frontal view) and cropped out the facial region by normalizing with respect to the eye coordinates manually specified. It turns out that the recognition result is 57% correct for the top match and 83% for the top three matches. Clearly, Case 4 is the best among all.

13.4 Video Gallery vs. Video Probes

Here we describe a parametric model for appearance and dynamics to understand the manifold structures of these models, which are then used to devise joint appearance and dynamic based recognition algorithms.

13.4.1 Parametric Model for Appearance and Dynamic Variations

A wide variety of spatio-temporal data have often been modeled as realizations of dynamical models. Examples include dynamic textures [11], human joint angle trajectories [6] and silhouettes [37]. A well-known dynamical model for such time-series data is the autoregressive and moving average (ARMA) model. Linear dynamical systems represent a class of parametric models for time-series. A wide variety of time series data such as dynamic textures, human joint angle trajectories, shape sequences, video based face recognition etc., are frequently modeled as autoregressive and moving average (ARMA) models [1, 6, 11, 37]. Let $f(t)$ be a sequence of features extracted from a video indexed by time t. The ARMA model parametrizes the evolution of the features $f(t)$ using the following equations:

$$f(t) = Cz(t) + w(t) \quad w(t) \sim N(0, R), \tag{13.19}$$

$$z(t+1) = Az(t) + v(t) \quad v(t) \sim N(0, Q) \tag{13.20}$$

where, $z \in \mathbb{R}^d$ is the hidden state vector, $A \in \mathbb{R}^{d \times d}$ the transition matrix and $C \in \mathbb{R}^{p \times d}$ the measurement matrix. $f \in \mathbb{R}^p$ represents the observed features while w and v are noise components modeled as normal with 0 mean and covariances $R \in \mathbb{R}^{p \times p}$ and $Q \in \mathbb{R}^{d \times d}$, respectively.

For high-dimensional time-series data (dynamic textures etc), the most common approach is to first learn a lower-dimensional embedding of the observations via PCA, and learn temporal dynamics in the lower-dimensional space. Closed form solutions for learning the model parameters (A, C) from the feature sequence $(f_{1:T})$ have been proposed by [11, 27] and are widely used in the computer vision community. Let observations $f(1), f(2), \ldots, f(\tau)$, represent the features for the time indices $1, 2, \ldots, \tau$. Let $[f(1), f(2), \ldots, f(\tau)] = U \Sigma V^{\mathrm{T}}$ be the singular value decomposition of the data. Then $\hat{C} = U$, $\hat{A} = \Sigma V^{\mathrm{T}} D_1 V (V^{\mathrm{T}} D_2 V)^{-1} \Sigma^{-1}$, where $D_1 = [00; I_{\tau-1}\ 0]$ and $D_2 = [I_{\tau-1} 0; 00]$.

The model parameters (A, C) do not lie in a vector space. The transition matrix A is only constrained to be stable with eigenvalues inside the unit circle. The observation matrix C is constrained to be an orthonormal matrix. For comparison of models, the most commonly used distance metric is based on subspace angles between column-spaces of the observability matrices [10]. For the ARMA model of (13.20), starting from an initial condition $z(0)$, it can be shown that the *expected* observation sequence is given by

$$E\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ . \\ . \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ . \\ . \end{bmatrix} z(0) = O_\infty(M)z(0). \tag{13.21}$$

Thus, the expected observation sequence generated by a time-invariant model $M = (A, C)$ lies in the column space of the extended *observability* matrix given by

$$O_\infty^{\mathrm{T}} = \left[C^{\mathrm{T}}, (CA)^{\mathrm{T}}, \left(CA^2\right)^{\mathrm{T}}, \ldots, \left(CA^n\right)^{\mathrm{T}}, \ldots\right]. \tag{13.22}$$

In experimental implementations, we approximate the extended observability matrix by the finite observability matrix as is commonly done [33]

$$O_m^{\mathrm{T}} = \left[C^{\mathrm{T}}, (CA)^{\mathrm{T}}, \left(CA^2\right)^{\mathrm{T}}, \ldots, \left(CA^{m-1}\right)^{\mathrm{T}}\right]. \tag{13.23}$$

The size of this matrix is $mp \times d$. The column space of this matrix is a d-dimensional subspace of $\mathbb{R}^{mp}$, where d is the dimension of the state-space z in (13.20). d is typically of the order of 5–10.

Thus, given a database of videos, we estimate the model parameters as described above for each video. The finite observability matrix is computed as in (13.23). To represent the subspace spanned by the columns of this matrix, we store *an* orthonormal basis computed by Gram-Schmidt orthonormalization. Since, a subspace is a point on a *Grassmann* manifold [35, 36], a linear dynamical system can be alternately identified as a point on the Grassmann manifold corresponding to the column space of the observability matrix. The goal now is to devise methods for classification and recognition using these model parameters. Given a set of videos for a given class, we would like to compute a parametric or non-parametric class-conditional density. Then, the maximum likelihood classification for each test instance can be performed using these class conditional distributions. To enable these, we need to understand the geometry of the Grassmann manifold.

13.4.2 The Manifold Structure of Subspaces

The set of all d-dimensional linear subspaces of $\mathbb{R}^n$ is called the Grassmann manifold which will be denoted as $\mathcal{G}_{n,d}$. The set of all $n \times d$ orthonormal matrices is

called the Stiefel manifold and shall be denoted as $\mathscr{S}_{n,d}$. As discussed in the applications above, we are interested in computing statistical models over the Grassmann manifold. Let $U_1, U_2, \ldots, U_k$ be some previously estimated points on $\mathscr{S}_{n,d}$ and we seek their sample mean, an average, for defining a probability model on $\mathscr{S}_{n,d}$. Recall that these U_is are tall, orthogonal matrices. It is easy to see that the Euclidean sample mean $\frac{1}{k}\sum_{i=1}^{k} U_i$ is not a valid operation, because the resultant mean does not have the property of orthonormality. This is because $\mathscr{S}_{n,d}$ is not a vector space. Similarly, many of the standard tools in estimation and modeling theory do not directly apply to such spaces but can be adapted by accounting for the underlying nonlinear geometry.

A subspace is stored as an orthonormal matrix which forms a basis for the subspace. As mentioned earlier, orthonormal matrices are points on the Stiefel manifold. However, since the choice of basis for a subspace is not unique, any notion of distance and statistics should be invariant to this choice. This requires us to interpret each point on the Grassmann manifold as an equivalence of points on the Stiefel manifold, where all orthonormal matrices that span the same subspace are considered equivalent. This interpretation is more formally described as a *quotient* interpretation that is, the Grassmann manifold is considered a quotient space of the Stiefel manifold. Quotient interpretations allow us to extend the results of the base manifold such as tangent spaces, geodesics etc to the new quotient manifold. In our case, it turns out that the Stiefel manifold itself can be interpreted as a quotient of a more basic manifold—the special orthogonal group $\mathrm{SO}(n)$. A quotient of Stiefel is thus a quotient of $\mathrm{SO}(n)$ as well.

A point U on $\mathscr{S}_{n,d}$ is represented as a tall-thin $n \times d$ orthonormal matrix. The corresponding equivalence class of $n \times d$ matrices $[U] = UR$, for $R \in GL(d)$ is called the Procrustes representation of the Stiefel manifold. Thus, to compare two points in $\mathscr{G}_{n,d}$, we simply compare the smallest squared distance between the corresponding equivalence classes on the Stiefel manifold according to the Procrustes representation. Given matrices U_1 and U_2 on $\mathscr{S}_{n,d}$, the smallest squared Euclidean distance between the corresponding equivalence classes is given by

$$d^2_{\text{Procrust}}\big([U_1],[U_2]\big) = \min_R \operatorname{tr}(U_1 - U_2R)^{\mathrm{T}}(U_1 - U_2R) \tag{13.24}$$

$$= \min_R \operatorname{tr}\big(R^{\mathrm{T}}R - 2U_1^{\mathrm{T}}U_2R + I_k\big). \tag{13.25}$$

When R varies over the orthogonal group $O(d)$, the minimum is attained at $R = H_1H_2^{\mathrm{T}} = A(A^{\mathrm{T}}A)^{-1/2}$, where $A = H_1DH_2^{\mathrm{T}}$ is the singular value decomposition of A. We refer the reader to [8] for proofs and alternate cases. Given several examples from a class $(U_1, U_2, \ldots, U_n)$ on the manifold, the class conditional density can be estimated using an appropriate kernel function. We first assume that an appropriate choice of a divergence on the manifold has been made such as the one above. For the Procrustes measure, the density estimate is given by [8] as

$$\hat{f}(U; M) = \frac{1}{n}C(M)\sum_{i=1}^{n} K\big[M^{-1/2}\big(I_k - U_i^{\mathrm{T}}UU^{\mathrm{T}}U_i\big)M^{-1/2}\big] \tag{13.26}$$

where $K(T)$ is the kernel function, M is a $d \times d$ positive definite matrix which plays the role of the kernel width or a smoothing parameter. $C(M)$ is a normalizing factor chosen so that the estimated density integrates to unity. The matrix valued kernel function $K(T)$ can be chosen in several ways. We have used $K(T) = \exp(-\mathrm{tr}(T))$ in all the experiments reported in this chapter. In this non-parametric method for density estimation, the choice of kernel width M becomes important. Thus, though this is a non-iterative procedure, the optimal choice of the kernel width can have a large impact on the final results. In general, there is no standard way to choose this parameter except for cross-validation. In the experiments reported here, we use $M = I$, the $d \times d$ identity matrix.

In addition to such nonparametric methods, there are principled methods to devise parametric densities on manifolds. Here, we simply refer the reader to [36] for mathematical details. In brief, using the tangent structure of the manifold, it is possible to define the well-known parametric densities such as multi-variate Gaussian, mixture-of-Gaussians etc., on the tangent spaces and wrap them back to the manifold. Densities defined in such a manner are called 'wrapped'-densities. In the experiments section, we use a wrapped-Gaussian to model class-condition densities on the Grassmann manifold. This is compared to the simpler nonparametric method described above.

13.4.3 Video-Based Face Recognition Experiments

We performed a recognition experiment on the NIST's Multiple Biometric Grand Challenge (MBGC) dataset. The MBGC Video Challenge dataset consists of a large number of subjects walking towards a camera in a variety of illumination conditions. Face regions are manually tracked and a sequence of cropped images is obtained. There were a total of 143 subjects with the number of videos per subject ranging from 1 to 5. In our experiments, we took subsets of the dataset which contained at least 2 sequences per person denoted as S_2, at least 3 sequences per person denoted as S_3 etc. Each of the face-images was first preprocessed to zero-mean and unity variance. In each of these subsets, we performed a leave-one-out testing. The results of the leave one out testing are shown in Table 13.3. Also reported are the total number of distinct subjects and the total number of video sequences in each of the subsets. In the comparisons, we show results using the 'arc-length' metric between subspaces [13]. This metric computes the subspace angles between two subspaces and takes the Frobenius norm of the angles as a distance measure [13]. We also show comparisons with the Procrustes measure, the Kernel density estimate with $M = I$ and a parametric wrapped Gaussian density on the manifold. The wrapped Gaussian is estimated on the tangent-plane centered at the mean-point of the dataset. The mean, more formally defined as the Karcher mean, is defined as the point that minimizes the sum of squared geodesic distances to all other points. The tangent-plane being a vector space allows the use of multi-variate statistics to define class-conditional densities. We refer the reader to [36] for mathematical details.

Table 13.3 Comparison of video based face recognition approaches using (a) Subspace Angles + Arc-length metric, (b) Procrustes Distance, (c) kernel density, (d) Wrapped Normal on Tangent Plane

Subset	Distinct Subjects	Total Sequences	Arc-length Metric	Procrustes Metric	Kernel density	Wrapped Normal
S_2	143	395	38.48	43.79	39.74	**63.79**
S_3	55	219	48.85	53.88	50.22	**74.88**
S_4	54	216	48.61	53.70	50.46	**75**
Avg.			45.31%	50.45%	46.80%	**71.22%**

As can be seen, statistical methods outperform nearest-neighbor based approaches. As one would expect, the results improve when more examples per class are available. Since the optimal kernel-width is not known in advance, this might explain the relatively poor performance of the kernel density method. More examples of statistical inference on the Grassmann manifold for image and video-based recognition can be found in [35].

13.5 Face Recognition in Camera Network

Video-based face recognition algorithms exploit information temporally across the video sequence to improve recognition performance. With camera networks, we can capture multi-view videos which allow us to further integrate information spatially across view angles. It is worth noting that this is different from traditional face recognition of single-camera videos in which various face poses exhibit. In that case, one usually needs to model the dynamics of pose changes in the training phase and estimate pose in the testing phase. For example, in [20], Lee et al. train a representation for the face appearance manifold. The manifold consists of locally linear subspaces for different poses. A transition probability matrix is also trained to characterize the temporal dynamics for this representation. In [23], the dynamics are encoded in the learned Hidden Markov Models (HMMs). The mean observations of hidden states are shown to represent facial images at various poses. These approaches are designed to work with a single camera.

On the other hand, in camera network deployments there are multiple images of the face in different poses at a given time instant. These images could include a mix of frontal and nonfrontal images of the face, or, in some cases, a mix of nonfrontal images (see Fig. 13.2). Videos captured in such a mode have natural advantages in providing persistent sensing over a large area and stronger cues for handling pose variations. Nonetheless, if we do not leverage the collaboration among cameras, the power of multi-view data over single-views cannot be fully exploited. For example, if we extend the single-view video-based methods, such as [20] and [23], to a camera network, they have to function in such a mode that cameras do not collaborate with each other except at the final fusion stage.

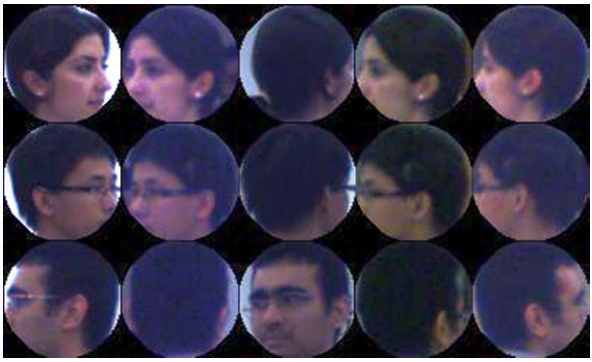

Fig. 13.2 Images acquired by a multi-camera network. Each column corresponds to a different camera, and each row corresponds to a different time instant and subject. Note that, under unconstrained acquisition, it is entirely possible that none of the images are frontal in spite of using five cameras to observe the subject [32]

In general, there are some principles one should follow in developing a video-based face recognition algorithm for camera networks: First, the method should be able to collaboratively utilize information collected by multiple cameras and arrive at a multi-view representation from it, as opposed to perform recognition for each view individually and then fusing the result. Second, the method should be able to tackle pose variations effectively, as this is the major concern of a multi-view face recognition system. Third, the method should work on data whose acquisition conditions are as close to practical surveillance situations as possible. These conditions include: reasonable distance between subject and cameras, relatively low resolution in the face region, uncontrolled pose variations, uncontrolled subject motion, and possible interruptions in acquisition (say, the subject moves out of the field of view of a camera) etc.

Next, we will introduce a video-based face tracking and recognition framework following these principles. The system first tracks a subject's head from multi-view videos and back-projects textures to a spherical head-model. Then a rotation-invariant feature based on spherical harmonic (SH) transform is constructed from the texture maps. Finally, video-based recognition is achieved through measurement of ensemble similarity.

13.5.1 Face Tracking from Multi-view Videos

The tracker is set in a Sequential Importance Resampling (SIR) (particle filtering) framework, which can be broken down into a description of its state space, the state transition model and the observation model. To fully describe the position and pose of a 3D object, we usually need a 6-D representation ($\mathbb{R}^3 \times \mathrm{SO}(3)$), where the 3-D real vector space is used to represent the object's location, and the special orthogonal group SO(3) is used to represent the object's rotation. In our work, we model the human head as a sphere and perform pose-robust recognition. This enables us to explore in 3-D state space $\mathscr{S} = \mathbb{R}^3$. Each state vector $\mathbf{s} = [x, y, z]$ represents the 3-D position of a sphere's center, disregarding the orientation. The radius of the sphere is assumed to be known through an initialization step. The low dimensionality of

the state space contributes to the reliability of the tracker, since for SIR, even a large number of particles will necessarily be sparse in high dimensional space.

The state transition model $P(\mathbf{s}_t \mid \mathbf{s}_{t-1})$ is set as a Gaussian distribution $\mathcal{N}(\mathbf{s}_t \mid \mathbf{s}_{t-1}, \sigma^2\mathbf{I})$. We have found that the tracking result is relatively insensitive to the specific value of σ and fixed it at 50 mm (our external camera calibration is metric). The observations for the filter are histograms extracted from the multi-view video frames I_t^j, where j is the camera index and t is the frame index. Histogram features are invariant to rotations and thus fit the circumstance of reduced state space. To adopt this feature, we need to back-project I_t^j onto the spherical head model and establish the histogram over the texture map. The observation likelihood is modeled as follows:

$$P\big(O_t \mid \mathbf{s}_\mathbf{t}^{(\mathbf{i})}\big) = P\big(I_t^1, I_t^2, \ldots, I_t^K \mid \mathbf{s}_\mathbf{t}^{(\mathbf{i})}\big) \propto 1 - D\big(H(M_{t,i}), H_{\text{template}}\big), \tag{13.27}$$

where $\mathbf{s}_t^{(i)}$ is the ith particle at the tth frame; $H(M_{t,i})$ is the histogram of the texture map built from the particle $\mathbf{s}_t^{(i)}$; H_{template} is the histogram of template texture map. The template texture map is computed after initializing the head position in the first frame, then updated by back-projecting the head region in the image, which is fixed by the maximum a posteriori (MAP) estimate onto the sphere model. The $D(H_1, H_2)$ function calculates the Bhattacharyya distance between two normalized histograms.

We now describe the procedure for obtaining texture map on the surface of the head model. First, we uniformly sample the spherical surface. Then for the jth camera, the world coordinates of sample points $[x_n, y_n, z_n]$, $n = 1, 2, \ldots, N$ are transformed into coordinates in that camera's reference frame $[x_n^{C_j}, y_n^{C_j}, z_n^{C_j}]$ to determine their visibility in that camera's view. Only unoccluded points (i.e., those satisfying $z_n^{C_j} \leq z_0^{C_j}$, where $z_0^{C_j}$ is the distance from the head center to the jth camera center) are projected onto the image plane. By relating these model surface points $[x_n, y_n, z_n]$ to the pixels at their projected image coordinates $I(x_n^{P_j}, y_n^{P_j})$, we build the texture map M^j of the visible hemisphere for the jth camera view. This continues until we have transformed the texture maps obtained from all camera views to the spherical model. Points in the overlapped region are fused using a weighting strategy, based on representing the texture map of the jth camera view as a function of locations of surface points $M^j(x, y, z)$. We assign the function value at point $[x_n, y_n, z_n]$ a weight $W_{n,j}$, according to the point's proximity to the projection center. This is based on the fact that, on the rim of a sphere, a large number of surface points tend to project to the same pixel, so image pixels corresponding to those points are not suitable for back-projection. The intensity value at the point $[x_n, y_n, z_n]$ of the resulting texture map will be:

$$M(x_n, y_n, z_n) = M^{j_{\max}}(x_n, y_n, z_n), \tag{13.28}$$

where

$$j_{\max} = \arg\max W_{n,j}, \quad j = 1, 2, \ldots, K. \tag{13.29}$$

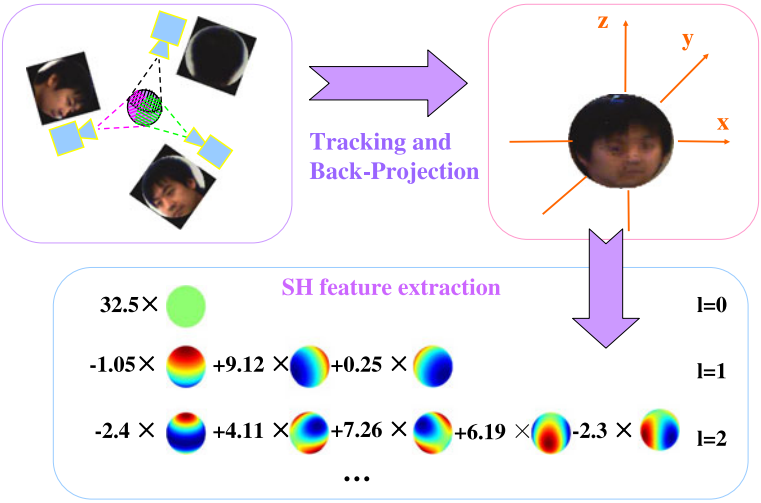

Fig. 13.3 Feature extraction. We first obtain the texture map of the human head on the surface of a spherical model through back projection of multi-view images captured by the camera network, then represent it with spherical harmonics

The texture mapping and back-projection processes are illustrated in the left part of Fig. 13.3.

Figure 13.4 shows an example of our pose-free tracking result for a multi-view video sequence. The video sequence has 500 frames. The tracker is able to stably track all the frames without failure, despite the considerably abrupt motions and the frequent occurrences of rotation, translation and scaling of the human head as shown. Sometimes the subject's head is outside the field-of-view of certain cameras. Though subjects usually do not undergo such extreme motion in real-world surveillance videos, this example clearly illustrates the reliability of our tracking algorithm. In our experiments, the tracker handles all the captured videos without difficulty. The occasionally observed inaccuracies in bounding circles are mostly due to the difference between sphere and the exact shapes of human heads. Successful tracking enables the subsequent recognition task.

13.5.2 Pose-Free Feature Based on Spherical Harmonics

In this section, we describe the procedure for extracting a rotation-invariant feature from the texture map obtained in Sect. 13.5.1. The process is illustrated in Fig. 13.3. According to the Spherical Harmonics (SH) theory, SHs form a set of orthonormal basis functions over the unit sphere, and can be used to linearly expand any square-integrable function on S^2. SH representation has been used for matching 3D shapes [16] due to its properties related to the rotation group. In the vision community,

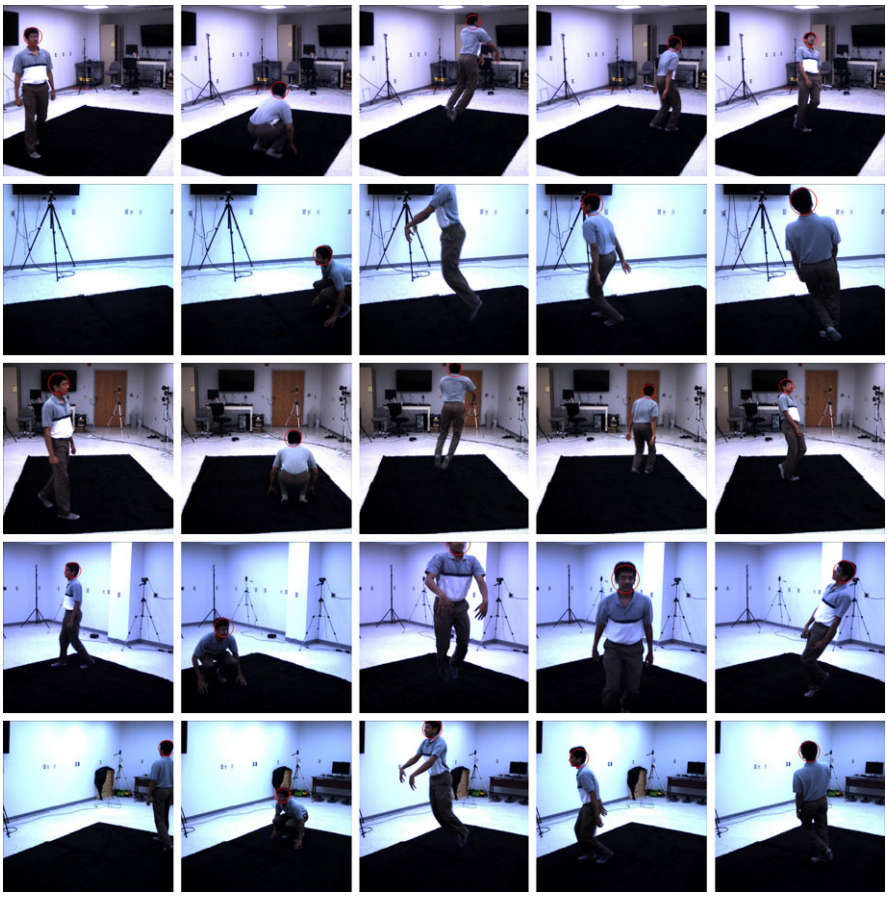

Fig. 13.4 Sample tracking results for a multi-view video sequence. 5 views are shown here. Each row of images is captured by the same camera. Each column of images corresponds to the same time-instant

following the work of Basri and Jacobs [5], researchers have used SH to understand the impact of illumination variations in face recognition [30, 39].

The general SH representation is used to analyze complex functions (For description of general SH, please refer to [5] or [30]). However, the spherical function determined by the texture map are real functions, and thus we consider real spherical harmonics (or Tesseral SH):

$$Y_l^m(\theta,\phi) = \begin{cases} Y_{l0} & \text{if } m = 0, \\ \frac{1}{\sqrt{2}}(Y_{lm} + (-1)^m Y_{l,-m}) & \text{if } m > 0, \\ \frac{1}{\sqrt{2}i}(Y_{l,-m} - (-1)^m Y_{lm}) & \text{if } m < 0 \end{cases} \tag{13.30}$$

where $Y_{lm}(\cdot,\cdot)$ denotes the general SH basis function of degree $l \geq 0$ and order m in $(-l, -l+1, \ldots, l-1, l)$. Note that here we are using the spherical coordinate system. $\theta \in (0, \pi)$ and $\phi \in (0, 2\pi)$ are the zenith angle and azimuth angle, respectively. The Real SHs are also orthonormal and they share most of the major properties of the general Spherical Harmonics. From now on, the word "Spherical Harmonics" shall refer only to the Real SHs. As in Fourier expansion, the SH expansion coefficients f_l^m of function $f(\theta, \phi)$ can be computed as:

$$f_l^m = \int_\theta \int_\phi f(\theta, \phi) Y_l^m(\theta, \phi)\, d\theta\, d\phi. \tag{13.31}$$

The expansion coefficients have a very important property which is directly related to our 'pose-free' face recognition application:

Proposition *If two functions defined on* S^2: $f(\theta, \phi)$ *and* $g(\theta, \phi)$ *are related by a rotation* $R \in \mathrm{SO}(3)$, *that is,* $g(\theta, \phi) = R(f(\theta, \phi))$, *and their SH expansion coefficients are* f_l^m *and* g_l^m ($l = 0, 1, \ldots$ *and* $m = -l \ldots l$), *respectively, the following relationship exists*:

$$g_l^m = \sum_{m'=-l}^{l} D_{mm'}^l f_l^{m'} \tag{13.32}$$

and the $D_{mm'}^l$*s satisfy*:

$$\sum_{m'=-l}^{l} \left(D_{mm'}^l\right)^2 = 1. \tag{13.33}$$

In other words, after rotation, the SH expansion coefficients at a certain degree l are actually linear combinations of those before the rotation, and coefficients at different degrees do not affect each other. This proposition is a direct result of the following lemma [7, 16]:

Lemma *Denote* E_l *the subspace spanned by* $Y_l^m(\theta, \phi)$, $m = -l \ldots l$, *then* E_l *is an irreducible representation for the rotation group SO*(3).

Thus, given a texture map $f(\theta, \phi)$ and its corresponding SH coefficient $\{f_l^m, l = 0, 1, \ldots, m = -l, \ldots, l\}$, we can formulate the energy vector associated with $f(\theta, \phi)$ as $e_f = (\|f_0\|_2, \|f_1\|_2, \|f_l\|_2, \ldots)$, where f_l is the vector of all f_l^m at degree l. Equation (13.33) guarantees that e_f keeps unchanged when the texture map is rotated, and this enables pose-robust face recognition. We refer to e_f as the SH Energy feature. Note that this is different from the energy feature defined in [16]. In practice, we further normalize the SH energy feature with regard to total energy. This is the same as assuming that all the texture maps have the same total energy, and somehow function as an illumination-normalized signature. Although this also means that skin color information is not used for recognition, it proves to work very well in experiments.

Fig. 13.5 Comparison of the reconstruction qualities of head/face texture map with different number of spherical harmonic coefficients. The images from left to right are: the original 3D head/face texture map, the texture map reconstructed from 40-degree, 30-degree and 20-degree SH coefficients, respectively [32]

The remaining issue concerns with obtaining a suitable band-limited approximation with SH for our application. In Fig. 13.5, we show a 3D head texture map and its reconstructed version with 20, 30 and 40 degree SH transform, respectively. The ratio of computation time for the 3 cases is roughly 1:5:21. (The exact time varies with configuration of the computer, for example, on a PC with Xeon 2.13 GHz CPU, it takes roughly 1.2 seconds to do a 20 degree SH transform for 18 050 points.) We have observed that the 30-degree transform achieves the best balance between approximation precision and computational cost.

13.5.3 Measure Ensemble Similarity

Given two multi-view video sequences with m and n frames (Every "frame" is actually a group of images, each captured by a camera in the network.), respectively, we generate 2 ensembles of feature vectors, respectively. They may contain different number of vectors. To achieve video-level recognition, we are interested in measuring the similarity between these two sets of vectors. Now, we calculate the ensemble similarity as the limiting Bhattacharyya distance in RKHS following [41]. In experiments, we measure the ensemble similarity between feature vectors of a probe video and those of all the gallery videos. The gallery video with the shortest distance to the probe is considered as the best match. For detailed derivations and explanation of limiting Bhattacharyya distance in the RKHS, please refer to [41].

13.5.4 Experiments

Most existing "multi-view" still or video face databases, such as PIE, Yale-B, the oriental face data, M2VTS etc., target recognition-across-pose algorithms, so they are not applicable to our multi-view to multi-view matching algorithm. The data we used in this work are multi-view video sequences captured with 4 or 5 video cameras in an indoor environment, collected at 3 different sessions: one for building

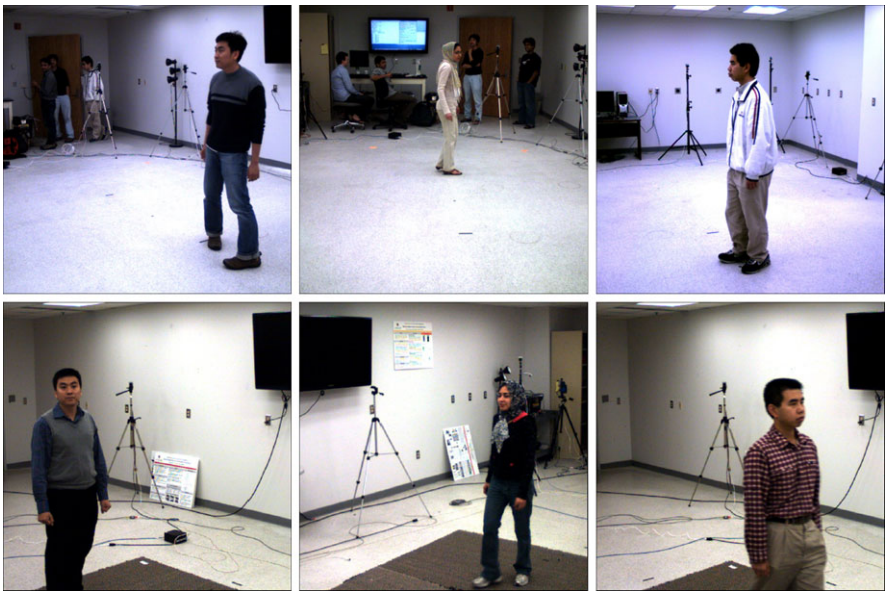

Fig. 13.6 Example of gallery and probe video frames. Images on the top are gallery frames, and those on the bottom are probe frames of the same subjects. Many subjects look differently in gallery and probe

a gallery and the other two for constructing probes. To test the robustness of our recognition algorithm, we arranged the second session to be one week after the first one, and the third 6 months after the second. The appearance of some subjects change significantly between the sessions. The database enrolls 25 subjects. Each subject has 1 gallery video and most subjects have 2 probe videos. Each video is 100 to 200 frames in length. Since each video sequence is captured with multiple cameras, it is equivalent to 4 or 5 videos in the single camera case. Figure 13.6 shows some example frames from gallery and probe video sequences. This data set poses great challenges to multi-view face recognition algorithms.

13.5.4.1 Feature Comparison

We associate 5 different kinds of features with different classifiers to compare their performance in image-based face recognition systems. By "image-based face recognition" we mean that each frame is treated as gallery or probe individually and no video-level fusion of results is performed. As a result, the recognition rate is computed by counting the number of correctly classified frames, not videos. The inputs to all these face recognition systems are based on the same tracking results. For any system based on feature of raw image intensity value, we use only the head region that is cropped by a circular mask as provided by the tracking result. All the head images are scaled to 30×30. For the PCA features, Eigenvectors that preserve the

Table 13.4 Comparison of recognition performance

Feature	NN	KDE	SVM-Linear	SVM-RBF
Intensity PCA	49.7%	39.0%	49.2%	57.8%
Intensity LDA	50.5%	27.2%	33.1%	40.7%
SH PCA	33.6%	30.9%	31.2%	44.2%
SH Energy	55.3%	47.9%	50.3%	67.1%
Normalized SH Energy	**60.8%**	**64.7%**	**78.2%**	**86.0%**

Table 13.5 KL divergence of in-class and between-class distances for different features

Intensity	Intensity + PCA	SH + PCA	SH Energy	Normalized SH Energy
0.1454	0.1619	0.2843	0.1731	**1.1408**

top 95% energy are kept. For the SH-based feature, we perform a 30-degree SH transform. Here, we would like to emphasize that since both gallery and probe are captured when subjects are performing free motion, the poses exhibited in images of any view are arbitrary and keep changing. This is significantly different from the settings of most existing multi-view face databases. The results are shown in Table 13.4. As we can see, the performance of the proposed feature exceeds that of other features by a large margin in all cases. Note that we do not fuse the results of different views for non-SH-based features.

To quantitatively verify the proposed feature's discrimination power, we then conducted the following experiment. We calculate distances for each unordered pair of feature vectors $\{x_i, x_j\}$ in the gallery. If $\{x_i, x_j\}$ belongs to the same subject, then the distance is categorized as being *in-class*. Otherwise, the distance is categorized as being *between-class*. We approximate the distribution of the two kinds of distances as histograms.

Intuitively, if a feature has good discrimination power, then the in-class distances evaluated using that feature tends to take smaller values compared to the between-class distances. If the two distributions mix together, then this feature is not good for classification. We use the symmetric KL divergence $\mathrm{KL}(p\|q) + \mathrm{KL}(q\|p)$ to evaluate the difference between the two distributions. We summarize the values of KL divergence for the 5 features in Table 13.5 and plot the distributions in Fig. 13.7. As clearly shown, the in-class distances for normalized SH energy feature are concentrated in the low value bins, while the between-class ones tend to have higher values, and their modes are obviously separated from each other. For all other features, the between-class distances do not show a clear trend of being larger than the in-class ones, and their distributions are just mixed. The symmetric KL-divergence also suggests the same.

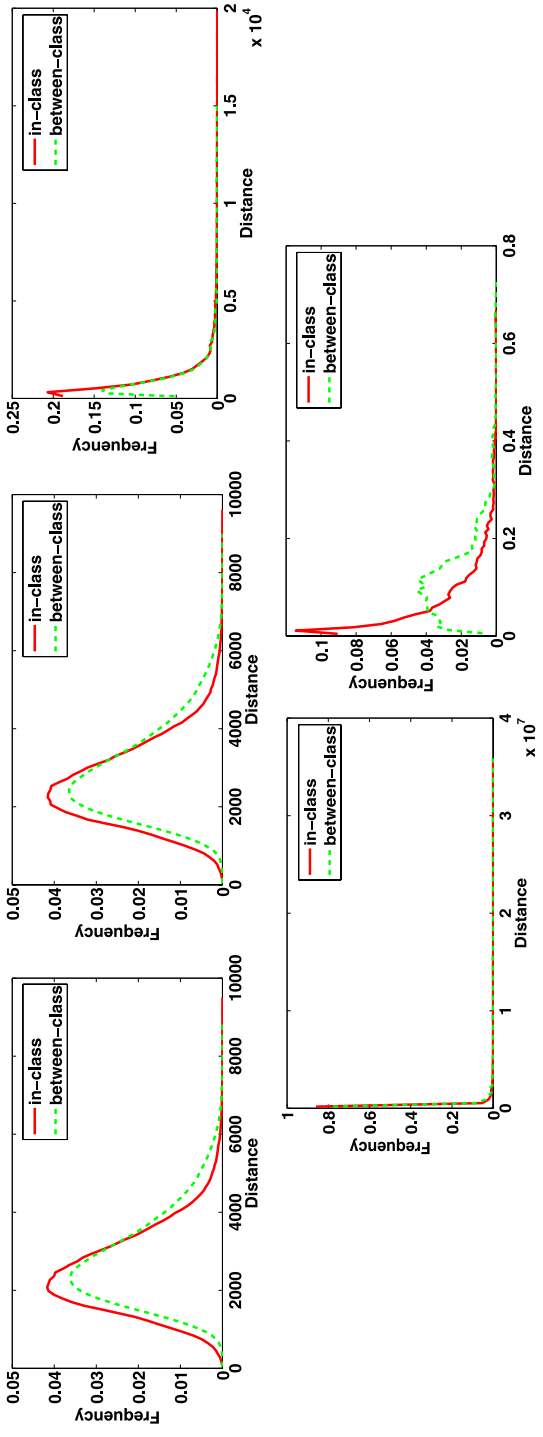

Fig. 13.7 Comparison of the discriminant power of the 5 features. *First row*, from left to right: intensity value, intensity value + PCA, SH, SH + PCA. *Second row*, from left to right: SH Energy, Normalized SH energy. The *green curve* is between-class distance distribution and the red one is in-class distance distribution. Number of bins is 100 [32]

13.5.4.2 Video-Based Recognition

In this experiment, we compare the performance of 4 video-level recognition systems: (1) Ensemble-similarity-based algorithm as proposed in [41] for cropped face images. The head images in a video are automatically cropped by a circular mask as provided by the tracking results and scaled to 30 by 30. Then we calculate the limiting Bhattacharyya distance between gallery and probe videos in RKHS for recognition. The kernel is RBF. If a video has n frames and it is captured by k cameras, then there are $k \times n$ head (face) images in the ensemble. (2) View-selection-based algorithm. We first train a PCA subspace for frontal-view face. The training images are a subset of the Yale B database and are scaled to 30 by 30. We then use this subspace to pick frontal-view face images from our gallery videos. We construct a frontal-view face PCA subspace for each individual. For every frame of a probe video, we first compute the "frontalness" of the subject's face in each view according to its distance to the general PCA model. The view which best matches the model is selected and fitted to the individual PCA subspaces of all the subjects. After classification of all the frames has been finished, recognition result for the video is obtained through majority voting. (3) video-based face recognition algorithm using probabilistic appearance manifold as proposed in [20]. We use 8 planes for local manifold model and set the probability of remaining the same pose to be 0.7 in the pose transition probability matrix. We first use this algorithm to process each view of a probe video. To fuse results of different views we use majority voting. If there is a tie in views' voting, we pick the one with smaller Hausdorff distance as the winner. (4) Normalized SH energy feature + ensemble similarity. This algorithm is as described in Sect. 13.5.2 and Sect. 13.5.3.

We plot the cumulative recognition rate curve in Fig. 13.8. Note that the numbers shown here should not be compared with those in the previous image-based recognition experiment to draw misleading conclusions, as these two sets of recognition rates are not convertible to each other. The view-selection method heavily relies on the availability of frontal-view face images, however, in the camera network case, the frontal pose may not appear in any view of the cameras. As a result, it does not perform well in this multi-view to multi-view matching experiment. Rather than the ad-hoc majority voting fusion scheme adopted by the view-selection algorithm, the manifold-based algorithm and the image-ensemble-based algorithm use more reasonable strategies to combine classification results of individual frames. Moreover, they both have certain ability to handle pose variations, especially the manifold-based one. However, because they are designed to work with a single camera, they are single-view in nature. Repeating these algorithms for each view does not fully utilize the multi-view information. On the other hand, the proposed method is multi-view in nature and is based on a pose-free feature, so it performs noticeably better than the other 3 algorithms in this experiment.

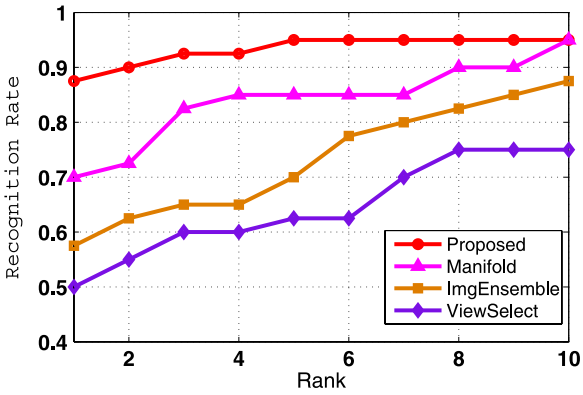

Fig. 13.8 Cumulative recognition rate of the 4 video-based face recognition algorithms

13.6 Conclusions

Video offers several advantages for face recognition, in terms of motion information and availability of more views. We reviewed several techniques that exploit video by either fusing information on a per-frame basis, considering them as image-ensembles, or by learning better appearance models. However, the availability of video opens interesting questions of how to exploit the temporal correlation for better tracking of faces, how to exploit behavioral cues available from video, and how to fuse the multiple views afforded by a camera network. Also, algorithms need to be derived that allow for matching a probe video to a still or video gallery. We showed applications involving such scenarios and discussed the issues involved in designing algorithms for such scenarios. There are several future research directions that are promising. While there are several studies that suggest that humans can recognize faces in non-cooperative conditions [26]—poor resolution, bad lighting etc.—if motion and dynamic information is available. This capability has been difficult to describe mathematically and replicate in an algorithm. If this phenomenon can be modeled mathematically, it could lead to more accurate surveillance and biometric systems. The role of familiarity in face recognition and the role that motion plays in recognition of familiar faces, while well known in psychology and neuroscience literature [31], is yet another avenue that has been challenging to model mathematically and replicate algorithmically.

Acknowledgements Supported by a MURI Grant N00014-08-1-0638 from the Office of Naval Research. The authors would like to thank Dr. Aswin Sankaranarayanan for helpful discussions related to Sect. 13.5.

References

1. Aggarwal, G., Roy-Chowdhury, A., Chellappa, R.: A system identification approach for video-based face recognition. In: International Conference on Pattern Recognition, Cambridge, UK, August 2004

2. Anderson, B., Moore, J.: Optimal Filtering. Prentice Hall, Englewood Cliffs (1979)
3. Arandjelovic, O., Cipolla, R.: Face recognition from video using the generic shape-illumination manifold. In: European Conference on Computer Vision, pp. 27–40 (2006)
4. Arandjelovic, O., Shakhnarovich, G., Fisher, J., Cipolla, R., Darrell, T.: Face recognition with image sets using manifold density divergence. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 581–588, San Diego, USA, June 2005
5. Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. In: Proceedings of IEEE International Conference on Computer Vision, vol. 2, pp. 383–390 (2001)
6. Bissacco, A., Chiuso, A., Ma, Y., Soatto, S.: Recognition of human gaits. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 52–57, Hawaii, USA, December 2001
7. Brocker, T., Dieck, T.: Representations of Compact Lie Groups. Springer, Berlin (2003)
8. Chikuse, Y.: Statistics on Special Manifolds. Lecture Notes in Statistics. Springer, New York (2003)
9. Choudhury, T., Clarkson, B., Jebara, T., Pentland, A.: Multimodal person recognition using unconstrained audio and video. In: Proc. of Intl. Conf. on Audio- and Video-Based Person Authentication, pp. 176–181 (1999)
10. Cock, K.D., Moor, B.D.: Subspace angles between ARMA models. Syst. Control Lett. **46**, 265–270 (2002)
11. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. Int. J. Comput. Vis. **51**(2), 91–109 (2003)
12. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering. Stat. Comput. **10**(3), 197–209 (2000)
13. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1999)
14. Fan, W., Yeung, D.-Y.: Locally linear models on face appearance manifolds with application to dual-subspace based classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1384–1390, New York, NY, USA, June 2006
15. Hamm, J., Lee, D.D.: Grassmann discriminant analysis: a unifying view on subspace-based learning. In: International Conference on Machine Learning, pp. 376–383, Helsinki, Finland, June 2008
16. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In: Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 156–164 (2003)
17. Kim, T.K., Kittler, J., Cipolla, R.: Discriminative learning and recognition of image set classes using canonical correlations. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1005–1018 (2007)
18. Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. **5**, 1–25 (1996)
19. Lee, K.-C., Ho, J., Kriegman, D.J.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005)
20. Lee, K.-C., Ho, J., Yang, M.-H., Kriegman, D.J.: Visual tracking and recognition using probabilistic appearance manifolds. Comput. Vis. Image Underst. **99**(3), 303–331 (2005)
21. Li, Y., Gong, S., Liddell, H.: Video-based online face recognition using identity surfaces. In: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp. 40–46 (2001)
22. Liu, J., Chen, R.: Sequential Monte Carlo for dynamic systems. J. Am. Stat. Assoc. **93**, 1031–1041 (1998)
23. Liu, X., Chen, T.: Video-based face recognition using adaptive hidden Markov model. In: Proceedings of Computer Vision and Pattern Recognition, vol. 1, pp. 340–345 (2003)
24. Liu, X., Chen, T., Thornton, S.M.: Eigenspace updating for non-stationary process and its application to face recognition. Pattern Recognit. **36**, 1945–1959 (2003)
25. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 696–710 (1997)

26. O'Toole, A.J., Roark, D., Abdi, H.: Recognizing moving faces: a psychological and neural synthesis. Trends Cogn. Sci. **6**, 261–266 (2002)
27. Overschee, P.V., Moor, B.D.: Subspace algorithms for the stochastic identification problem. Automatica **29**(3), 649–660 (1993)
28. Park, U., Jain, A.K., Ross, A.: Face recognition in video: adaptive fusion of multiple matchers. In: Proceedings of IEEE Computer Society Workshop on Biometrics (In Conjunction with CVPR), pp. 1–8 (2007)
29. Philipps, P., Moon, H., Rivzi, S., Ross, P.: The Feret evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1090–1104 (2000)
30. Ramamoorthi, R.: Analytic pca construction for theoretical analysis of lighting variability in images of a Lambertian object. IEEE Trans. Pattern Anal. Mach. Intell. **24**(10), 1322–1333 (2002)
31. Roark, D.A., Barrett, S.E., O'Toole, A.J., Abdi, H.: Learning the moves: The effect of familiarity and facial motion on person recognition across large changes in viewing format. Perception 761–773 (2006)
32. Ross, A.A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics. International Series on Biometrics. Springer, New York (2006)
33. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 58–63, Hawaii, USA, December 2001
34. Shakhnarovich, G., Fisher, J.W., Darrell, T.: Face recognition from long-term observations. In: Proceedings of the European Conference on Computer Vision, vol. 3, pp. 851–865, May 2002
35. Turaga, P., Veeraraghavan, A., Chellappa, R.: Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, Alaska, USA, June 2008
36. Turaga, P., Veeraraghavan, A., Srivastava, A., Chellappa, R.: Statistical analysis on manifolds and its applications to video analysis. In: Schonfeld, D., Shan, C., Tao, D., Wang, L. (eds.) Video Search and Mining. Studies in Computational Intelligence, Chap. 5. Springer, Berlin (2010)
37. Veeraraghavan, A., Roy-Chowdhury, A., Chellappa, R.: Matching shape sequences in video with an application to human movement analysis. IEEE Trans. Pattern Anal. Mach. Intell. **27**(12), 1896–1909 (2005)
38. Xu, Y., Roy-Chowdhury, A., Patel, K.: Pose and illumination invariant face recognition in video. In: Proceedings of IEEE Computer Society Workshop on Biometrics (In Conjunction with CVPR), pp. 1–7 (2007)
39. Zhang, L., Samaras, D.: Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 351–363 (2006)
40. Zhao, W.Y., Chellappa, R., Rosenfeld, A., Phillips, P.: Face recognition: a literature survey. ACM Comput. Surv. **35** (2003)
41. Zhou, S.K., Chellappa, R.: From sample similarity to ensemble similarity: probabilistic distance measures in reproducing kernel Hilbert space. IEEE Trans. Pattern Anal. Mach. Intell. **28**(6), 917–929 (2006)
42. Zhou, S., Krueger, V., Chellappa, R.: Probabilistic recognition of human faces from video. Comput. Vis. Image Underst. **91**, 214–245 (2003)
43. Zhou, S., Chellappa, R., Moghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans. Image Process. (2004)