ABSTRACT

Title of thesis: SIMULTANEOUS MULTI-VIEW FACE TRACKING AND RECOGNITION IN VIDEO USING PARTICLE FILTERING

Naotoshi Seo, Master of Science, 2009

Thesis directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Recently, face recognition based on video has gained wide interest especially due to its role in surveillance systems. Video-based recognition has superior advantages over image-based recognition because a video contains image sequences as well as temporal information. However, surveillance videos are generally of low-resolution and contain faces mostly in non-frontal poses.

We propose a multi-view, video-based face recognition algorithm using the Bayesian inference framework. This method represents an appearance of each subject by a complex nonlinear appearance manifold expressed as a collection of simpler pose manifolds and the connections, represented by transition probabilities, among them. A Bayesian inference formulation is introduced to utilize the temporal information in the video via the transition probabilities among pose manifolds. The Bayesian inference formulation realizes video-based face recognition by progressively accumulating the recognition confidences in frames. The accumulation step possibly enables to solve face recognition problems in low-resolution videos, and the progres-

sive characteristic is especially useful for a real-time processing. Furthermore, this face recognition framework has another characteristic that does not require processing all frames in a video if enough recognition confidence is accumulated in an intermediate frame. This characteristic gives an advantage over batch methods in terms of a computational efficiency.

Furthermore, we propose a simultaneous multi-view face tracking and recognition algorithm. Conventionally, face recognition in a video is performed in tracking-*then*-recognition scenario that extracts the best facial image patch in the tracking and then recognizes the identity of the facial image. Simultaneous face tracking and recognition works in a different fashion, by handling both tracking and recognition simultaneously. Particle filter is a technique for implementing a Bayesian inference filter by Monte Carlo simulation, which has gained prevalence in the visual tracking literature since the Condensation algorithm was introduced. Since we have proposed a video-based face recognition algorithm based on the Bayesian inference framework, it is easy to integrate the particle filter tracker and our proposed recognition method into one, using the particle filter for both tracking and recognition simultaneously. This simultaneous framework utilizes the temporal information in a video for not only tracking but also recognition by modeling the dynamics of facial poses. Although the time series formulation remains more general, only the facial pose dynamics is utilized for recognition in this thesis.

SIMULTANEOUS MULTI-VIEW FACE TRACKING AND RECOGNITION IN VIDEO USING PARTICLE FILTERING

by

Naotoshi Seo

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2009

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Shuvra S. Bhattacharyya
Professor Min Wu

Acknowledgments

First and foremost I would like to express my gratitude to my advisor, Professor Rama Chellappa, for his valuable guidance on research, financial supports, and giving me an opportunity to work on interesting projects. I owe my gratitude to my thesis committee members: Professor Shuvra S. Bhattacharyya, and Professor Min Wu. I would like to address my appreciation to Professor Shuvra S. Bhattacharyya for his financial supports.

I had a pleasant stay at the CfAR (Center for Automation Research). I am indebted to my lab roommates and friends: Soma Biswas, Ming Du, Ruonan Li, Ming-Yu Liu, Dikpal Reddy, and Pavan Turaga. I want to express my deeply-felt thanks to my senior graduate student Pavan Turaga for his valuable advice and excellent comments. I really enjoyed my fruitful discussions with Ming Du.

I owe my deepest thanks to my family - my father, mother, grandmother, and brothers. Words cannot express the gratitude I owe them. I greatly thank my eldest brother for his computer supports in Japan.

I would also like to thank UMIACS (University of Maryland Institute for Advanced Computer Studies) computer technical staffs for their quick responses and exceptional helps.

It is impossible to remember all, and I apologize to those I've inadvertently left out.

Lastly, thank you all!

Table of Contents

List of Figures

Chapter 1

Introduction

For decades human face recognition has been an active topic in the field of object recognition. Many algorithms have been proposed to deal with image-based recognition where both the training and test set consist of still face images. Recently, face recognition based on video has gained wide interest especially due to its role in surveillance systems. Video-based recognition has superior advantages over image-based recognition because a video contains image sequences as well as temporal information. However, surveillance videos are generally of low-resolution and contain faces mostly in non-frontal poses. This thesis provides a solution to video-based face recognition.

In Chapter 2, we present a review of still-image based face recognition. The study of still-image based face recognition provides knowledge of several feature extraction methods and their probabilistic models which can also be utilized in video-based face recognition. Subspace methods are pattern recognition techniques widely invoked in various face recognition approaches. Well-known appearance-based recognition schemes utilize principal component analysis (PCA). In this chapter, the "Eigenface" [74], the face-specific subspace (FSS) [70], the probabilistic density estimation in eigenspaces [53], [54], and the intrapesonal/extrapersonal subspace [52], [51], [49] are reviewed. In addition, we review the "Fisherface" method [3] which is

often compared with the "Eigenface" method.

In Chapter 3, we present a historical review of video-based face recognition algorithms, and propose a new video-based face recognition algorithm using Bayesian inference. Motivated by the previous work [43], this method represents an appearance manifold of each subject by a complex nonlinear appearance manifold express as a collection of simpler pose manifolds and the connections among them. We express the simpler pose manifolds as linear PCA subspaces, and perform similarity measurements between images and the PCA subspaces using the probabilistic density estimation in eigenspaces method [53] reviewed in Chapter 2. The Bayesian inference formulation realizes video-based face recognition by progressively accumulating the recognition results in frames, and enables to solve face recognition problems with high accuracy in low-resolution videos.

In Chapter 4, we review variants of Bayesian inference filter such as the Kalman filter, the grid-based filter, and the particle filter. The Kalman filter and the grid-based filter solve optimal solutions in limited situations. Often, such limitations do not hold, so we use approximation strategies to the optimal solution using the particle filters. Particle filter is a technique for implementing a Bayesian inference filter by Monte Carlo simulation, which has gained popularity in the visual tracking literature since the Condensation algorithm was introduced.

In Chapter 5, we propose a simultaneous multi-view face tracking and recognition algorithm using particle filtering. Since we have proposed a video-based face recognition algorithm which works in the Bayesian inference framework, it is easy to integrate the particle filter tracker and the proposed recognition method into one,

using the particle filter for both tracking and recognition simultaneously. Unlike the previous work [82], the proposed framework utilizes the temporal information in a video for not only tracking but also recognition by modeling the dynamics of facial poses. We also discuss and propose an adaptive particle filtering framework for the simultaneous tracking and recognition problem. This simultaneous multi-view framework successfully tracks multi-view faces in low-resolution videos and concurrently achieves accurate face recognition.

In Chapter 6, we provide conclusions and suggestions for future study.

Chapter 2

Review of Still Image-Based Face Recognition

As one of the most successful applications of image analysis and understanding, face recognition has received significant attention and has been an active topic for decades. Face recognition technologies have a variety of potential applications in public security, law enforcement, and commerce, such as mug-shot database matching, identity authentication from credit cards or driver licenses, access control, information security, and intelligent surveillance. In addition, there are many emerging fields that can benefit from face recognition technology, such as the new generation intelligent human-compute interfaces and e-services, including e-home, teleshopping, and tele-banking. Related research activities have significantly increased over the past decades [65], [12], [16], [81].

During the nineties, geometric feature-based methods and template matching methods used to be popular technologies, which were compared by Brunelli and Poggio [12]. They concluded that template matching outperforms the geometric feature-based ones. Therefore, since the 1990s, appearance-based methods have been investigated, from which two categories of face recognition technology have evolved: holistic appearance feature-based and analytic local feature-based. Popular methods belonging to the former paradigm include eigenface [74], Fisherface [3], [24], Probabilistic and Bayesian matching and Active Shape/Appearance Models

(ASM/AAM); [42], [17], [23]. Local feature analysis (LFA) [59] and Elastic Bunch Graph Matching (EBGM) [78] are typical examples of the latter category. LFA has been used in successful commercial face recognition system known as FaceIt by Visionics Corp. Support Vector Machines (SVM) have also been successfully applied to face recognition [28]. FERET evaluation has provided extensive comparisons of these algorithms [63], as well as several evaluation protocols for face recognition systems. Subsequently, FRVT 2000 [8], 2002 [62], 2006 [64], and FRGC [61] efforts have established a rich history of evaluating face recognition algorithms.

Now we briefly describe eigenface [74], Fisherface [3], the face-specific subspace (FSS) [70], probabilistic density estimation in eigenspaces [53], [54], and intrapesonal/extrapersonal subspace [52], [51], [49] methods. In addition, we offer further analyses on the intrapersonal subspace.

2.1 Eigenfaces

The *eigenface* method is based on linearly projecting the image space to a low dimensional feature space [71], [74], [75]. The eigenface method, which uses principal components analysis (PCA) for dimensionality reduction, yields projection directions that minimize the total mean square error in reconstruction.

Let us consider a set of N sample images $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ taking values in an D-dimensional image space, and assume that each image belongs to one of c classes $\{X_1, X_2, \ldots, X_c\}$. Let us also consider a linear transformation mapping the original D-dimensional image space into an M-dimensional feature space, where $M < D$.

The new feature vectors $\mathbf{y}_j \in \mathbb{R}^M$ are defined by the following linear transformation:

$$\mathbf{y}_j = W^T \mathbf{x}_j \qquad\qquad j = 1, 2, \ldots, N \tag{2.1}$$

where $W \in \mathbb{R}^{D \times M}$ is a matrix with orthonormal columns.

If the total scatter matrix S_T is defined as

$$S_T = \sum_{j=1}^{N} (\mathbf{x}_j - \mu)(\mathbf{x}_j - \mu)^T \tag{2.2}$$

where $\mu \in \mathbb{R}^D$ is the mean image of all samples, then after applying the linear transformation W^T, the scatter of the transformed feature vectors $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$ is $W^T S_T W$. In PCA, the projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projected samples, i.e.,

$$W_{\text{opt}} = \arg\max_W |W^T S_T W| \tag{2.3}$$

$$= [\mathbf{w}_1 \mathbf{w}_2 \ldots \mathbf{w}_M] \tag{2.4}$$

where $\{\mathbf{w}_i | i = 1, 2, \ldots, M\}$ is the set of D-dimensional eigenvectors of S_T corresponding to the M largest eigenvalues $\{\lambda_i | i = 1, 2, \ldots, M\}$ [22], i.e.,

$$S_T \mathbf{w}_i = \lambda_i \mathbf{w}_i, \qquad i = 1, 2, \cdots, M. \tag{2.5}$$

Since these eigenvectors have the same dimension as the original images, they are referred to as Eigenpictures in [71] or eigenfaces in [74], [75]. Classification is then performed using a nearest neighbor classifier in the reduced feature space consisting of coefficients that result from projecting the face images onto eigenvectors.

2.2 Fisherfaces

Since the learning set is labeled, it makes sense to use this information to build a more reliable method for reducing the dimensionality of the feature space. *Fisherface* method uses a class specific linear method, Fisher's Linear Discriminant (FLD) [25], for dimensionality reduction and simple classifiers in the reduced feature space. This method selects W in [16] in such a way that the ratio of the between-class scatter and the within class scatter is maximized.

Again, let us consider a set of N sample images $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ taking values in an D-dimensional image space, and assume that each image belongs to one of c classes $\{X_1, X_2, \ldots, X_c\}$. Let the between-class scatter matrix be defined as

$$S_B = \sum_{k=1}^{c} N_k(\mu_k - \mu)(\mu_k - \mu)^T \tag{2.6}$$

and the within-class scatter matrix be defined as

$$S_W = \sum_{k=1}^{c} \sum_{\mathbf{x}_j \in X_k} (\mathbf{x}_j - \mu_k)(\mathbf{x}_j - \mu_k)^T \tag{2.7}$$

where μ_k is the mean image of class X_k, N_k is the number of samples in class X_k, and μ is the mean image of all samples. If S_W is nonsingular, the optimal projection W_{opt} is chosen as the matrix with orthonormal columns which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples, i.e.,

$$\begin{aligned} W_{\text{opt}} &= \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} \\ &= [\mathbf{w}_1 \mathbf{w}_2 \ldots \mathbf{w}_M] \end{aligned} \tag{2.8}$$

where $\{\mathbf{w}_i | i = 1, 2, \ \ldots, \ M\}$ is the set of generalized eigenvectors of S_B and S_W corresponding to the M largest generalized eigenvalues $\{\lambda_i | i = 1, 2, \ \ldots, \ M\}$, i.e.,

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \qquad i = 1, 2, \ldots, M. \tag{2.9}$$

Note that there are at most $c - 1$ nonzero generalized eigenvalues, and so an upper bound on M is $c - 1$, where c is the number of classes. See [21].

To illustrate the benefits of class specific linear projection, a low dimensional analogue to the classification problem in which the samples from each class lie near a linear subspace is shown. Figure 2.1 is a comparison of PCA and FLD for a two-class problem in which the samples from each class are randomly perturbed in a direction perpendicular to a linear subspace. For this example, $N = 20, \ D = 2$, and $M = 1$. So, the samples from each class lie near a line passing through the origin in the 2D feature space. Both PCA and FLD have been used to project the points from 2D down to 1D. Comparing the two projections in the figure, *PCA actually smears the classes together* so that they are no longer linearly separable in the projected space. It is clear that, although PCA achieves larger total scatter, FLD achieves greater between-class scatter, and, consequently, classification is improved.

In the face recognition problem, one is confronted with the difficulty that the within-class scatter matrix $S_W \in \mathbb{R}^{D \times D}$ is always singular. This stems from the fact that the rank of S_W is at most $N - c$, and, in general, the number of images in the learning set N is much smaller than the number of pixels in each image D. This means that it is possible to choose the matrix W such that the within-class scatter of the projected samples can be made exactly zero.

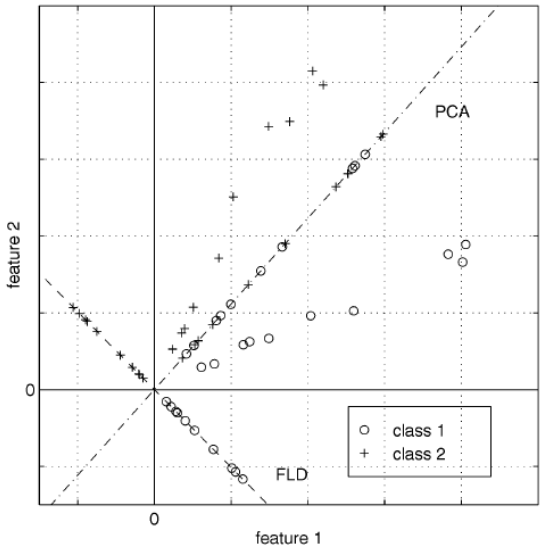


Figure 2.1: A comparison of principal component analysis (PCA) and Fisher's linear discriminant (FLD) for a two class problem where data for each class lies near a linear subspace [3].

In order to overcome the complication due to a singular S_W, an alternative to the criterion in (2.8) was proposed [3]. This method, named *Fisherfaces*, avoids this problem by projecting the image set to a lower dimensional space so that the resulting within-class scatter matrix S_W is nonsingular. This is achieved by using PCA to reduce the dimension of the feature space to $N - c$, and then applying the standard FLD defined (2.8) to reduce the dimension to $c - 1$. More formally, W_{opt} is given by

$$W_{\text{opt}}^T = W_{\text{fld}}^T W_{\text{pca}}^T \tag{2.10}$$

where

$$W_{\text{pca}} = \arg\max_W |W^T S_T W| \tag{2.11}$$

$$W_{\text{fld}} = \arg\max_{W} \frac{|W^T W_{\text{pca}}^T S_B W_{\text{pca}} W|}{|W^T W_{\text{pca}}^T S_W W_{\text{pca}} W|} \tag{2.12}$$

Note that the optimization for W_{pca} is performed over $D \times (N - c)$ matrices with orthonormal columns, while the optimization for W_{fld} is performed over $(N-c) \times M$ matrices with orthonormal columns. In computing W_{pca}, we have thrown away only the smallest $c - 1$ principal components.

2.3 Face-Specific Subspace (FSS)

The eigenface method applies PCA for "all" face images to construct a low dimensional face subspace, and discrimination of identities is performed based on Euclidean distance in the PCA subspace using the nearest neighbor classifier. A well-known fact of the PCA that it is optimal in the sense of minimizing the mean squared error (MMSE), that is, the PCA extracts the most expressive features of faces. However, the most expression features do not mean that they are also the most discriminative features for identity discrimination.

Therefore, many efforts to seek good features for discrimination have been done using Fisherfaces based on LDA. However, the LDA gives the optimal Bayesian discrimination in which data in each class is distributed on Gaussian with the same covariance, which is not true in general. To overcome the problem, Shan et al. [70] introduced a method which models each identity face with an identity specific PCA subspace named a "face-specific subspace" (FSS), and exploits distance from the PCA subspace, i.e., the reconstruction error as a dissimilarity measure for identification.

2.3.1 Observations on Eigenfaces

An experiment to illustrate the effects of eigenfaces is now presented. The PCA is applied to "all" face images and the obtained eigenfaces are shown in Figure 2.2. Two new images where one is a face image and another is a non-face image are projected into the face PCA subspace once and reconstructed back into the full space. These images are shown in Figure 2.3.

As shown in Figure 2.3, the reconstructed image of the non-face image is not similar to the original non-face image, i.e., the reconstruction error is large. This fact tells us that the PCA subspace trained for "all" face images would rather be used for discrimination of face and non-face images. This motivates us that for discrimination of identities, we should train a PCA subspace for images of each face identity, and identify a person by finding the minimum distance from the identity specific PCA subspaces, i.e., face-specific subspace (FSS).

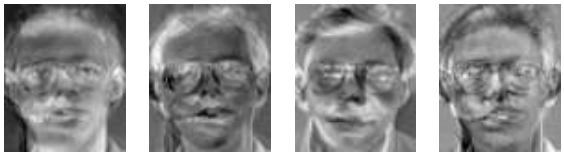


Figure 2.2: The eigenfaces trained for facial pictures taken from the ORL face dataset [66]. The left-most eigenface is the most principal one.

2.3.2 Construction of FSS

In this section, a procedure for constructing FSS is described. Let FSS be $\{\Omega_k | k = 1, 2, \ldots, c\}$ where c is the number of classes to be recognized. The basis

(a) (b) (c) (d)

Figure 2.3: (a) The input face image [66]. (b) The input non-face image taken from [77]. (c) The reconstructed image of the face image (d) The reconstructed image of the non-face image. The difference between the original non-face image and its reconstructed image is large.

functions in Karhunen-loeve Transform for kth face class are obtained by solving the eigenvalue problem

$$\Lambda_k = \Phi_k^T \Sigma_k \Phi_k \tag{2.13}$$

where Σ_k is the covariance matrix of the kth face data, Λ_k is the diagonal matrix of eigenvalues of Σ_k in the descendant order, and Φ_k is the corresponding eigenvector matrix. In PCA, a partial KLT is performed to identify the largest-eigenvalue eigenvectors and obtain a principal component feature vector $\mathbf{y}_k = (\Phi_k^{M_k})^T \tilde{\mathbf{x}}_k$, where $\tilde{\mathbf{x}}_k = \mathbf{x} - \overline{\mathbf{x}}_k$ is the mean-normalized image vector and $\Phi_k^{M_k}$ is a submatrix of Φ_k containing the principal eigenvectors. To sum up, the kth FSS is represented as a 4-tuple by

$$\Omega_k = (\Lambda_k, \Phi_k, \overline{\mathbf{x}}_k, M_k) \tag{2.14}$$

In practice, M_k is set to be identical in all identities, i.e., $M_k = M,\ \forall k$ to provide fairness.

2.3.3 Identify Faces in FSS

In this section, how to identify a face identity using the FSS is illustrated. In a partial KL expansion, the residual reconstruction error or the distance from the kth subspace is defined as

$$d(\mathbf{x}, \Omega_k) = \sum_{i=M_k+1}^{D} y_{ki}^2 = ||\tilde{\mathbf{x}}_k||^2 - \sum_{i=1}^{M_k} y_{ki}^2 \tag{2.15}$$

and can be easily computed from the first M_k principal components and the L_2-norm of the mean-normalized image $\tilde{\mathbf{x}}_k$. Typically M_k is identical in all classes, i.e., $M_k = M \ \forall k$ to achieve fairness.

The recognition task is performed in the nearest subspace sense: for a test image $\mathbf{x}$, the identity k^* can be determined by finding by the subspace Ω_k with minimal distance to $\mathbf{x}$, i.e.,

$$k^* = \arg\min_k d(\mathbf{x}, \Omega_k) \tag{2.16}$$

2.3.4 Practical Issue: Face Recognition from Single Example Image

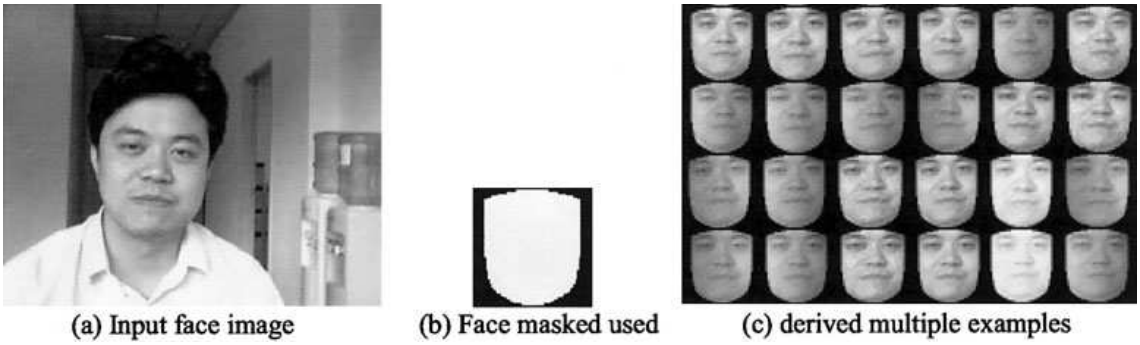


Figure 2.4: Deriving multiple samples from single image [70].

Typically, to learn a face subspace, multiple training example images are required. For a FSS, it means more than one example per face is needed to train his/her FSS. But for some face recognition applications, such as mug shot matching, suspect identification, etc., only few (often one) face image is available for each subject involved; therefore, the FSS-based method cannot be applied to these problems directly. To solve this problem, multiple samples are derived from a single example image.

The technique is based on the following two intuitive propositions [69]:

1. Proper geometric transforms, such as translation, rotation in image plane, scale changes, etc., do not change the identity attribute of a face image.

2. Proper gray-level transforms, such as simulative directional lighting, man-made noise, etc., do not change the identity attribute of a face image.

In this technique, two kinds of transforms are combined to derive tens of training examples from a single example image, which are then fed into the FSS learning procedure. Figure 2.4(c) illustrates some normalized "virtual" example images derived from one face image as shown in Figure 2.4(a).

2.4 Density Estimation in Eigenspaces

Moghaddam and Pentland [53], [54] proposed an approach to automatic visual learning based on density estimation. Instead of applying estimation techniques directly to the original high-dimensional space spanned by images, this method uses an eigenspace decomposition to yield a computationally feasible estimate. Specifically,

given a set of training images $\{\mathbf{x}^t\}_{t=1}^{N_T}$, from an object class Ω, this method estimates the class membership or a *likelihood* function for this data, i.e., $P(\mathbf{x}|\Omega)$. Here, we examine a density estimation technique for visual learning of high-dimensional data which is based on the assumption of a Gaussian distribution.

2.4.1 Principal Component Imagery

Given a set of *m-by-n* images $\{I^t\}_{t=1}^{N_T}$, we can form a training set of vectors $\{\mathbf{x}^t\}$, where $\mathbf{x} \in \mathcal{R}^{D=mn}$, by lexicographic ordering of the pixel elements of each image I^t. The basis functions in a PCA are obtained by solving the eigenvalue problem

$$\Lambda \ = \Phi^T \Sigma \Phi \tag{2.17}$$

where Σ is the covariance matrix of the data, Φ is the eigenvector matrix of Σ and Λ is the corresponding diagonal matrix of eigenvalues. A partial PCA is performed to identify the largest-eigenvalue eigenvectors and obtain a principal component feature vector $\mathbf{y} = \Phi_{\mathbf{M}}^T \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \mathbf{x} - \overline{\mathbf{x}}$ is the mean-normalized image vector and Φ_M is a submatrix of Φ containing the principal eigenvectors. PCA can be seen as a linear transformation $\mathbf{y} = \mathcal{I}(\mathbf{x}) : \mathcal{R}^D \rightarrow \mathcal{R}^M$ which extracts a lower-dimensional subspace corresponding to the maximal eigenvalues. This corresponds to an orthogonal decomposition of the vector space $\mathcal{R}^D$ into two mutually exclusive and complementary subspaces: the principal subspace (or feature space) $F = \{\Phi_i\}_{i=1}^{M}$ containing the principal components and its orthogonal complement $\bar{F} = \{\Phi_i\}_{i=M+1}^{D}$, as illustrated in Figure 2.5.

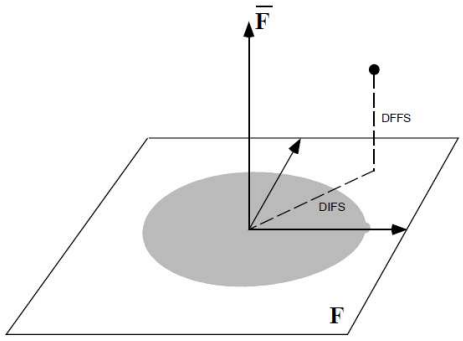


Figure 2.5: The principal subspace F and its orthogonal complement $\bar{F}$ for a Gaussian density [53].

In a partial PCA expansion, the residual reconstruction error is defined as

$$\epsilon^2(\mathbf{x}) = \sum_{i=M+1}^{D} y_i^2 = ||\tilde{\mathbf{x}}||^2 - \sum_{i=1}^{M} y_i^2 \tag{2.18}$$

and can be easily computed from the first M principal components and the L_2-norm of the mean-normalized image $\tilde{\mathbf{x}}$. Consequently the L_2 norm of every element $\mathbf{x} \in \mathcal{R}^D$ can be decomposed in terms of its projections in these two subspaces. We refer to the component in the orthogonal subspace F as the "distance-from-feature-space" (DFFS) which is a simple Euclidean distance and is equivalent to the residual error $\epsilon^2(\mathbf{x})$ in (2.18). The component of $\mathbf{x}$ which lies *in* the feature space F is referred to as the "distance-in-feature-space" (DIFS) but is generally not a distance-based norm, but can be interpreted in terms of the probability distribution of y in F.

2.4.2 Gaussian Densities

We begin by considering an optimal approach for estimating high-dimensional Gaussian densities. We assume that we have (robustly) estimated the mean $\overline{\mathbf{x}}$ and

covariance Σ of the distribution from the given training set $\{\mathbf{x}_t\}_{t=1}^{N_T}$. Under this assumption, the likelihood of a input pattern $\mathbf{x}$ is given by

$$P(\mathbf{x}|\Omega) = \frac{\exp[-\frac{1}{2}(\mathbf{x}-\overline{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\overline{\mathbf{x}})]}{(2\pi)^{D/2}|\Sigma|^{1/2}} \tag{2.19}$$

The sufficient statistic for characterizing this likelihood is the *Mahalanobis* distance

$$d(\mathbf{x}) = \tilde{\mathbf{x}}^T\Sigma\tilde{\mathbf{x}}^{-1} \tag{2.20}$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \overline{\mathbf{x}}$. Using the eigenvectors and eigenvalues of Σ we can rewrite Σ^{-1} in the diagonalized form

$$\begin{aligned} d(\mathbf{x}) &= \tilde{\mathbf{x}}^T\Sigma\tilde{\mathbf{x}}^{-1} \\ &= \tilde{\mathbf{x}}^T[\Phi\Lambda^{-1}\Phi^T]\tilde{\mathbf{x}} \\ &= \mathbf{y}^T\Lambda^{-1}\mathbf{y} \end{aligned} \tag{2.21}$$

where $\mathbf{y} = \Phi^T\tilde{\mathbf{x}}$ are the new variables obtained by the change of coordinates. Because of the diagonalized form, the *Mahalanobis* distance can also be expressed in terms of the sum

$$d(\mathbf{x}) = \sum_{i=1}^{D}\frac{y_i^2}{\lambda_i} \tag{2.22}$$

We now seek to estimate $d(\mathbf{x})$ using only the M principal projections. Therefore, we formulate an estimator for $d(\mathbf{x})$ as follows

$$\begin{aligned} \hat{d}(\mathbf{x}) &= \sum_{i=1}^{M}\frac{y_i^2}{\lambda_i} + \frac{1}{\rho}\left[\sum_{i=M+1}^{D} y_i^2\right] \\ &= \sum_{i=1}^{M}\frac{y_i^2}{\lambda_i} + \frac{1}{\rho}\epsilon^2(\mathbf{x}) \end{aligned} \tag{2.23}$$

where the term in the brackets is the DFFS $\epsilon^2(\mathbf{x})$, which as we have seen can be computed using the first M principal components. We can therefore write the

form of the likelihood estimate based on $\hat{d}(\mathbf{x})$ as the product of two marginal and independent Gaussian densities

$$\hat{P}(\mathbf{x}|\Omega) = \left[\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}}\right]\left[\frac{\exp\left(-\frac{\epsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(D-M)/2}}\right] \tag{2.24}$$

$$= P_F(\mathbf{x}|\Omega)\hat{P}_{\bar{F}}(\mathbf{x}|\Omega) \tag{2.25}$$

where $P_F(\mathbf{x}|\Omega)$ is the true marginal density in F-space and $\hat{P}_F(\mathbf{x}|\Omega)$ is the estimated marginal density in the orthogonal complement $\bar{F}$-space. The optimal value of ρ can now be determined by minimizing a suitable cost function $J(\rho)$. From an information-theoretic point of view, this cost function could be the Kullback-Leibler divergence between the true density $P(\mathbf{x}|\Omega)$ and its estimate $\hat{P}(\mathbf{x}|\Omega)$

$$J(\rho) = \int P(\mathbf{x}|\Omega)\log\frac{P(\mathbf{x}|\Omega)}{\hat{P}(\mathbf{x}|\Omega)}d\mathbf{x} = \mathbf{E}\left[\log\frac{P(\mathbf{x}|\Omega)}{\hat{P}(\mathbf{x}|\Omega)}\right] \tag{2.26}$$

Using the diagonalized forms of the *Mahalanobis* distance $d(\mathbf{x})$ and its estimate $\hat{d}(\mathbf{x})$ and the fact that $\mathbf{E}[y_i^2] = \lambda_i$, it can be easily shown that

$$J(\rho) = \frac{1}{2}\sum_{i=M+1}^{D}\left[\frac{\lambda_i}{\rho} - 1 + \log\frac{\rho}{\lambda_i}\right] \tag{2.27}$$

The optimal weight ρ^* can be then found by minimizing this cost function with respect to ρ. Solving the equation $\dfrac{\partial J}{\partial \rho} = 0$ yields

$$\rho^* = \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i \tag{2.28}$$

which is simply the arithmetic average of the eigenvalues in the orthogonal subspace $\bar{F}$. In addition to its optimality, ρ^* also results in an *unbiased* estimate of the

Mahalanobis distance – i.e., $\mathbf{E}[\hat{d}(\mathbf{x};\rho^*)] = \mathbf{E}[d(\mathbf{x})]$. What this derivation shows is that once we select the M-dimensional principal subspace F (as indicated, for example, by PCA), the optimal density estimate $\hat{P}(\mathbf{x}|\Omega)$ has the form of (2.25) with ρ given by (2.28).

This derivation of ρ is a special case of a more recent and general factor analysis model, called Probabilistic PCA (PPCA), proposed by Tipping and Bishop [12], In their formulation, the expression for ρ is the maximum likelihood solution of a latent variable role as opposed to the minimal-divergence solution derived in [53], For a more general expectation-maximization (EM) approach to factor analysis, the reader is referred to [37].

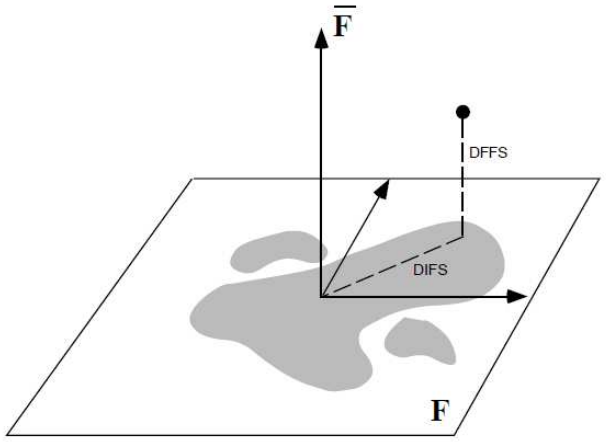


Figure 2.6: The principal subspace F and its orthogonal complement $\bar{F}$ for an arbitrary density [53].

In actual practice, it often happens that all D eigenvalues are not available because the number of training samples is fewer than the number of the feature dimension plus one, i.e., $N_T \leq D + 1$. But, they can be estimated, for example,

by fitting a nonlinear function to the available portion of the eigenvalue spectrum. Fractal power law spectra of the form $f(n) = an^k$ where a and k are constants, are thought to be typical of "natural" phenomenon and are a good fit to the decaying nature of the eigenspectrum–see Figure 2.7.

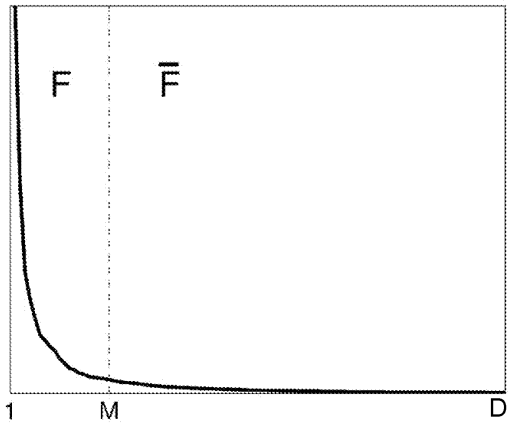


Figure 2.7: A typical eigenvalue spectrum and its division onto the two orthogonal subspaces [53].

2.5 Intrapersonal/Extrapersonal Subspace

Moghaddam et al. [52], [51], [49] proposed an intrapersonal/extrapersonal classifier. This method examines the variations in the difference images of same individuals to form an intrapersonal space, and the variations in the difference images of different individuals to form an extrapersonal space. The estimation of the intrapersonal and the extrapersonal distributions is based on the assumption that the intrapersonal distribution is Gaussian.

The input visual data (or equivalently its manifold representation) can form

the basis for simple recognition strategies using Euclidean metrics or normalized correlation. For example, in its simplest form, the similarity measure $S(I_1,\ I_2)$ between two images I_1 and I_2 (or their manifold projections) can be set to be inversely proportional to the norm $||I_1 - I_2||$ which corresponds to a template-matching approach to recognition [12], [37]. Such a formulation suffers from a major drawback: it does not exploit knowledge of which types of variations are critical (as opposed to incidental) in expressing similarity. However, one can formulate a *probabilistic* similarity measure which is based on the probability that the image intensity (or equivalently manifold vector) differences, denoted by $\triangle = I_1 - I_2$, are characteristic of typical variations in appearance of the *same* object. For example, for purposes of face recognition, one can define two classes of facial image variations: *intrapersonal* variations Ω_I (corresponding, for example, to different facial expressions of the *same* individual) and *extrapersonal* variations Ω_E (corresponding to variations between *different* individuals).

The similarity measure $S(\triangle)$ can then be expressed in terms of the intrapersonal a posteriori probability given by Bayes rule:

$$S(\triangle) = P(\Omega_I|\triangle) = \frac{P(\triangle|\Omega_I)P(\Omega_I)}{P(\triangle|\Omega_I)P(\Omega_I) + P(\triangle|\Omega_E)P(\Omega_E)}. \tag{2.29}$$

The likelihoods $P(\triangle|\Omega_I)$ and $P(\triangle|\Omega_E)$ can be estimated by traditional means (given enough training data) or, alternatively, with subspace density estimation techniques [54], [50] when faced with very high-dimensional data or with data shortage (insufficient number of samples). Furthermore, the priors $P(\Omega)$ can be set to reflect specific operating conditions (e.g., number of test images versus the size of the database) or

other sources of a priori knowledge regarding the two images being matched.

This particular Bayesian formulation casts the standard face recognition task (essentially an c-ary classification problem for c individuals) into a *binary* pattern classification problem with Ω_I and Ω_E. This simpler problem is then solved using the maximum a posteriori (MAP) rule–i.e., two images are determined to belong to the same individual if

$$P(\Omega_I|\triangle) > P(\Omega_E|\triangle) \quad \text{or, equivalently, if} \quad S(\triangle) > \frac{1}{2}. \tag{2.30}$$

Alternatively, a simplified similarity measure based only on the Ω_I likelihood can be used. This *maximum-likelihood* (ML) similarity measure ignores extrapersonal variations altogether and is given by $S'(\triangle) = P(\triangle|\Omega_I)$. Typically, the Ω_I density in (2.29) carries the greater weight in modeling the posterior similarity used for MAP recognition. The extrapersonal Ω_E density serves a secondary role and its accurate modeling is less critical. In the extreme case, by dropping the Ω_E likelihood in favor of a ML similarity, one obtains $S'(\triangle)$, which typically suffers only a minor deficit (3-4 percent) in accuracy as compared to $S(\triangle)$ [51].

2.5.1 Subspace Density Estimation

To deal with the inevitably high-dimensionality of $\triangle$ (which is the same as that of the images), we make use of the efficient density estimation method explained in Section 2.4 which divides the vector space $\mathcal{R}^D$ into two complementary subspaces as shown in Figure 2.5 using an eigenspace decomposition. This method uses PCA to obtain a principal subspace F whose principal components $\mathbf{y}$ can be used to form an

optimal (minimal divergence) low-dimensional estimate of the complete likelihood using only the first M principal components $\{y_1,\ y_2,\ y_3,\ \ldots,\ y_M\}$, where $M << D$.

As derived in Section 2.4, the complete likelihood estimate can be written as the product of two independent marginal Gaussian densities

$$\begin{aligned} \hat{P}(\triangle|\Omega) &= \left[\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}}\right]\left[\frac{\exp\left(-\frac{\epsilon^2(\triangle)}{2\rho}\right)}{(2\pi\rho)^{(D-M)/2}}\right] \\ &= P_F(\triangle|\Omega)\hat{P}_{\bar{F}}(\triangle|\Omega;\rho), \end{aligned} \tag{2.31}$$

where $P_F(\triangle|\Omega)$ is the true marginal density in F, $\hat{P}_{\bar{F}}(\triangle|\Omega;\rho)$ is the estimated marginal density in the orthogonal complement $\bar{F}$, λ_i are the eigenvalues, y_i are the principal components, and $\epsilon^2(\triangle)$ is the PCA residual (reconstruction error).

The information-theoretic optimal value for the density parameter ρ is derived by minimizing the Kullback-Leibler (KL) divergence and is found to be simply the average of the $\bar{F}$ eigenvalues

$$\rho = \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i. \tag{2.32}$$

Referring back to (2.29) we see that this approach requires two projections of the difference vector $\triangle$, from which likelihoods can be estimated for the Bayesian similarity measure $S(\triangle)$. The projection steps are linear, while the posterior computation is nonlinear. Because of the double PCA projections required, this approach has been called a dual eigenspace technique [54], [12], [51] in contrast to standard PCA-based "eigenfaces" in Figure 2.8. Note the projection of the difference vector $\triangle$ onto the "dual eigenfaces" (Ω_I and Ω_E) for computation of the posterior in

(2.29). In the following section, we will show that in practice, each input vector $\mathbf{x}$ will have two (precomputed) linear PCA projections $\mathbf{y}_{\Phi_I}$ and $\mathbf{y}_{\Phi_E}$ and that the posterior similarity $S(\triangle)$ between any pair of vectors can be expressed in terms of a pair of difference norms between their corresponding dual projections.

2.5.2 Efficient Similarity Computation

Consider a feature space of $\triangle$ vectors, the differences between two images (I_j and I_k). The two classes of interest in this space correspond to intrapersonal and extrapersonal variations and each is modeled as a high-dimensional Gaussian density as in (2.29). The densities are zero-mean since for each $\triangle = I_j - I_k$ there exists a $\triangle = I_k - I_j$.

$$P(\triangle|\Omega_E) = \frac{e^{\frac{1}{2}\triangle^T \Sigma_E^{-1} \triangle}}{(2\pi)^{D/2}|\Sigma_E|^{1/2}}$$
$$P(\triangle|\Omega_I) = \frac{e^{\frac{1}{2}\triangle^T \Sigma_I^{-1} \triangle}}{(2\pi)^{D/2}|\Sigma_I|^{1/2}}. \tag{2.33}$$

By PCA, the Gaussians are known to only occupy a subspace of image space (face-space) and, thus, only the top few eigenvectors of the Gaussian densities are relevant for modeling. These densities are used to evaluate the similarity in (2.29). Computing $S(\triangle)$ involves first subtracting a candidate image I_j from a database entry I_k. The resulting $\triangle$ image is then projected onto the eigenvectors of the extrapersonal Gaussian and also the eigenvectors of the intrapersonal Gaussian. The exponentials are computed, normalized, and then combined as in (2.29). This operation is iter-

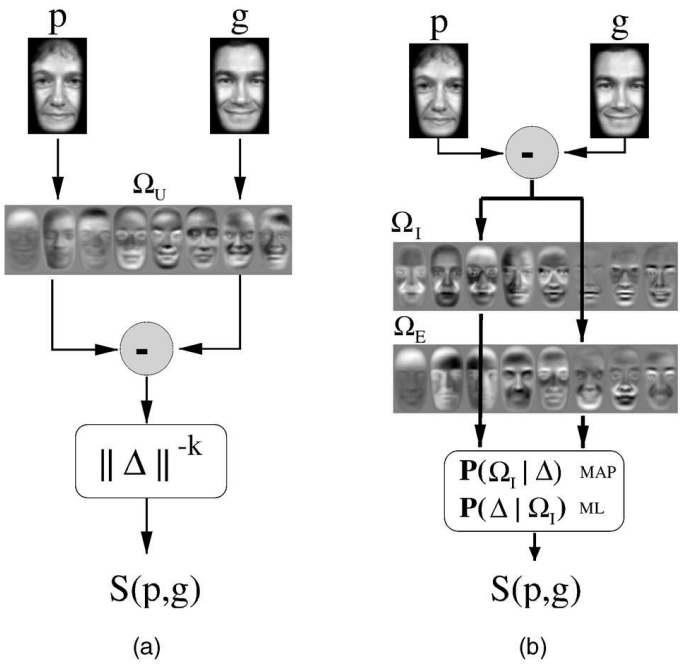


Figure 2.8: Signal flow diagrams for computing similarity $\mathbf{g}$ between two images: (a) Eigenface similarity and (b) Probabilistic similarity. The difference mage is projected through both sets of (intra/extra) eigenfaces in order to obtain the two likelihoods [52].

ated over all members of the database (many I_k images) until the maximum score is found (i.e., the match). Thus, for large databases, such evaluations are expensive and must be simplified by offline transformations.

To compute the likelihoods $P(\triangle|\Omega_I)$ and $P(\triangle|\Omega_E)$, one preprocess the I_k images with whitening transformations. Each image is converted and stored as a set of two whitened subspace coefficients, $\mathbf{y}_{\Phi_I}$ for intrapersonal space and $\mathbf{y}_{\Phi_E}$ for extrapersonal space (see (2.34)). Here, Λ and V are matrices of the largest

eigenvalues and eigenvectors of Σ_E or Σ_I.

$$\mathbf{y}^j_{\Phi_I} = \Lambda_I^{-\frac{1}{2}} V_I I_j \qquad\qquad \mathbf{y}^j_{\Phi_E} = \Lambda_E^{-\frac{1}{2}} V_E I_j \tag{2.34}$$

After this preprocessing, evaluating the Gaussians can be reduced to simple Euclidean distances as in (2.35). Denominators are of course precomputed. These likelihoods are evaluated and used to compute the MAP similarity $S(\triangle)$ in (2.29). Euclidean distances are computed between the M_I-dimensional $\mathbf{y}_{\Phi_I}$ vectors, as well as the M_E-dimensional $\mathbf{y}_{\Phi_E}$ vectors. Thus, roughly $2 \times (M_E + M_I)$ arithmetic operations are required for each similarity computation, avoiding repeated image differencing and projections.

$$P(\triangle|\Omega_I) = P(I_j - I_k|\Omega_I) = \frac{e^{-\|\mathbf{y}^j_{\Phi_I} - \mathbf{y}^k_{\Phi_I}\|^2/2}}{(2\pi)^{D/2}|\Sigma_I|^{1/2}}$$

$$P(\triangle|\Omega_E) = P(I_j - I_k|\Omega_E) = \frac{e^{-\|\mathbf{y}^j_{\Phi_E} - \mathbf{y}^k_{\Phi_E}\|^2/2}}{(2\pi)^{D/2}|\Sigma_E|^{1/2}}. \tag{2.35}$$

The ML similarity matching is even simpler since only the intrapersonal class is evaluated, leading to the following modified form for the similarity measure

$$S'(\triangle) = P(\triangle|\Omega_I) = -\|\mathbf{y}^j_{\Phi_I} - \mathbf{y}^k_{\Phi_I}\|^2/2 \tag{2.36}$$

by dropping the common denominator and taking log.

2.5.3 Recognition

Let us consider a set of N_P prototype sample images $\{I_1,\ I_2,\ \ldots,\ I_{N_P}\}$, and assume that each image belongs to one of c class subsets $\{X_1, X_2,\ \ldots, X_c\}$. The

prototype sample images are possibly taken as all training sample images or a subset of all training sample images. Recognition task is solved by finding the maximum similarity between a test image I and all prototypes, i.e.,

$$k^* = \arg\max_k \max_{I_j \in X_k} S(I - I_j) \tag{2.37}$$

or with ML similarity measure $S'(\triangle)$

$$k^* = \arg\max_k \max_{I_j \in X_k} S'(I - I_j). \tag{2.38}$$

as briefly described in Section 2.5.2. Although an efficient similarity computation was introduced, this exhaustive matching is still computationally too expensive when a large number of prototypes is used.

Therefore, one may solve the recognition task by finding the maximum similarity between a test image I and a representative image $\bar{I}_k$ from class k, i.e.,

$$k^* = \arg\max_k S(I - \bar{I}_k) \tag{2.39}$$

or with ML similarity measure $S'(\triangle)$

$$k^* = \arg\max_k S'(I - \bar{I}_k). \tag{2.40}$$

The representative image $\bar{I}_k$ could be taken manually or randomly from training sample images of a class k, or could be an estimated mean image of a class k obtained in the training sample images. Furthermore, one may use the MAP rule introduced in (2.30) as a reject option, i.e.,

$$\text{Reject } I \text{ if} \quad S(I - \bar{I}_k) < \frac{1}{2} \text{ for all } k = 1, \ldots, c. \tag{2.41}$$

2.5.4 Practical Approaches to Form Subspace

In practice, one suffers from computational costs of training PCA for the intrapersonal subspace and the extrapersonal subspace because of a huge number of difference images created. Let N_k be the number of training images available for identity k, and c be the number of identities. The possible number of intrapersonal difference images, N_I, is given by

$$N_I = \sum_{k=1}^{c} N_k^2 \tag{2.42}$$

because there exists a $\triangle = I_k - I_j$ for each $\triangle = I_j - I_k$ and we count $\triangle = I_j - I_j$ also and the possible number of extrapersonal difference images, N_E, is given by

$$N_E = 2 \sum_{k=1}^{c} \left[N_k \sum_{i \neq k}^{c} N_i \right]. \tag{2.43}$$

For example, when $N_k = 200$ images are available for $c = 20$ identities (this is a realistic number in a video database), $N_I = 800,000$ and $N_E = 30,400,000$. If we use 40×40 sized images and express one value by double data type (4 bytes) in C language, the number of bytes required for a feature matrix used to form an intrapersonal subspace is $800,000 * 40 * 40 * 4 = 5,120,000,000 = 5.12$GB. This is extremely huge and intractable because a 32-bit CPU can allocate a memory space upto $2^{32} = 4.295$GB. To overcome this problem, the authors of this thesis propose following four different approaches:

1. Pick fewer number of samples K_k randomly or manually from the total N_k samples,

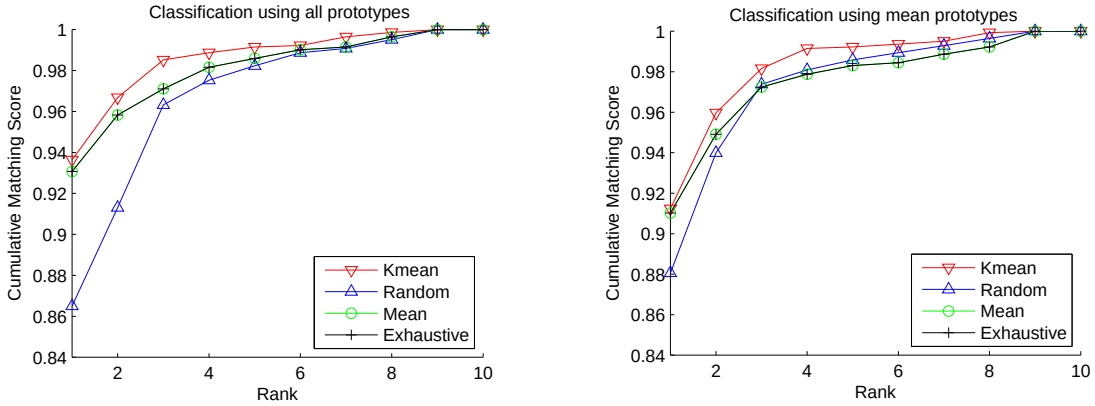
2. Apply K-means algorithm for N_k samples and use the obtained K_k means,

3. Use variants of incremental PCA algorithms [30], [31], [15], [10], [41] and

4. Create intrapersonal difference images in each class k by subtracting an estimated mean image $\bar{I}_k$ from sample images such that $N_I = \sum_{k=1}^{c} N_k$.

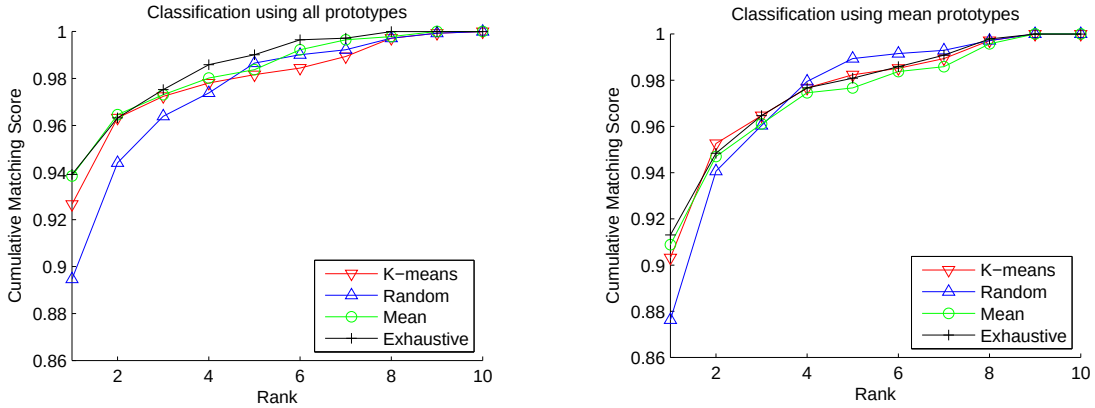
Notice that the 4th approach is different with others because this approach does not take combinations of sample images to create intrapesonal difference images, and does not propose how to form an extrapersonal subspace.

We performed two experiments to compare these approaches in terms of recognition using the intrapersonal similarity measure $S'(\triangle)$ and these results are shown in Figure 2.9 and Figure 2.10 using the cumulative match curves [63]. We used subsets of images obtained from the Honda/UCSD video database [43]. We used 20 frontal facial images for each person in the 1st experiment (Figure 2.9), and various number of images from $20 \sim 30$ for each person in the 2nd experiment (Figure 2.10). $20 \sim 30$ is not a huge number, therefore, we could compare these approaches with the original approach which takes full difference images exhaustively. Each image size was downsampled to 24×24 and the number of feature dimensions were reduced into 20 by projecting images onto eigenspaces. The number of subjects is 10. In both the random approach and the K-means approach, we chose $K_k = 10\ \forall k$ out of $20 \sim 30$ images. Figure 2.9(a) and 2.10(a) show comparisons of these approaches with the classification method using all the training images as prototypes. Figure 2.9(b) and 2.10(b) show comparisons of these approaches with the classification method using estimated mean class images as prototypes. We did not plot the experimental results using the 3rd incremental PCA approach because they resulted



(a) (b)

Figure 2.9: Comparisons of four intrapersonal approaches when the numbers of training sample images in each class are the same, 20. The number of feature dimensions used is 20. In both the random approach and the K-means approach, we chose $K_k = 10\ \forall k$ out of 20 images. (a) shows comparisons based on the classification method using all the training images as prototypes. (b) shows comparisons based on the classification method using estimated mean class images as prototypes.



(a) (b)

Figure 2.10: Comparisons of four intrapersonal approaches when the numbers of training sample images in each class are varying from $20 \sim 30$. Other parameters are the same as in the experiments shown in Figure 2.9.

in the similar performances with the original exhaustive approach as we expected.

The 1st random approach performed worse than others in our experiments although it would result in better than others sometimes because of its randomness. The 2nd K-means approach achieved better results than the original exhaustive approach (Figure 2.9) and similar results as the original exhaustive approach (Figure 2.10). The 4th mean subtraction approach achieved the exactly same results with the original exhaustive approach (Figure 2.9), and achieved similar results with the original exhaustive approach (Figure 2.10). Theoretically, the 4th mean subtraction approach obtains identical PCA eigenvectors with the original exhaustive approach when the number of samples in each class is the same as in the experiments shown in Figure 2.9. The proof is written in Section 2.5.5. The classification method which uses estimated mean class images as prototypes always performes worse than the classification method which uses all training images as prototypes, but gave a decent approximation.

2.5.5 Comparison with FSS

In this section, we describe a comparison between the intrapersonal subspace and the face-specific subspaces (FSS). As the first step, we give a proof of the following proposition.

Proposition. The mean subtraction approach described in Section 2.5.4 and the original exhaustive approach for modeling an intrapesonal subspace achieves identical eigenvectors when the number of samples in each class is the same.

Proof. Let c be the number of classes, N_k be the total number of samples in a

class k, and I_{ki} be the ith column vectorized sample image in a class k. First, we examine the total scatter matrix (2.2) $S_T^{(1)}$ of difference images created in the mean subtraction approach, i.e., difference images created by subtracting the empirical mean μ_k from the training samples in each class k. Because the mean of the created difference images is zero, $S_T^{(1)}$ essentially forms a within-class scatter matrix (2.7) and is given by

$$\begin{aligned}
S_T^{(1)} &= \sum_{k=1}^{c}\sum_{i=1}^{N_k}(I_{ki}-\mu_k)(I_{ki}-\mu_k)^T \\
&= \sum_{k=1}^{c}\sum_{i=1}^{N_k}(I_{ki}-\frac{1}{N_k}\sum_{j=1}^{N_k}I_{kj})(I_{ki}-\frac{1}{N_k}\sum_{j=1}^{N_k}I_{kj})^T \\
&= \sum_{k=1}^{c}\sum_{i=1}^{N_k}\left[I_{ki}I_{ki}^T-\frac{2}{N_k}I_{ki}\left(\sum_{j=1}^{N_k}I_{kj}\right)^T+\frac{1}{N_k^2}\left(\sum_{j=1}^{N_k}I_{kj}\right)\left(\sum_{j=1}^{N_k}I_{kj}\right)^T\right] \\
&= \sum_{k=1}^{c}\sum_{i=1}^{N_k}\left[I_{ki}I_{ki}^T-\frac{2}{N_k}\sum_{j=1}^{N_k}I_{ki}I_{kj}^T+\frac{1}{N_k^2}\sum_{\ell=1}^{N_k}\sum_{j=1}^{N_k}I_{k\ell}I_{kj}^T\right] \\
&= \sum_{k=1}^{c}\left[\sum_{i=1}^{N_k}I_{ki}I_{ki}^T-\frac{2}{N_k}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}I_{ki}I_{kj}^T+\frac{1}{N_k}\sum_{\ell=1}^{N_k}\sum_{j=1}^{N_k}I_{k\ell}I_{kj}^T\right] \\
&= \sum_{k=1}^{c}\left[\sum_{i=1}^{N_k}I_{ki}I_{ki}^T-\frac{1}{N_k}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}I_{ki}I_{kj}^T\right].
\end{aligned} \tag{2.44}$$

Next, we examine the total scatter matrix $S_T^{(2)}$ of intrapersonal difference images created in the original exhaustive approach. Because the mean of the difference

images is zero, the total scatter matrix $S_T^{(2)}$ is given by

$$\begin{aligned}
S_T^{(2)} &= \sum_{k=1}^{c}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}(I_{ki}-I_{kj})(I_{ki}-I_{kj})^T \\
&= \sum_{k=1}^{c}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}\left[I_{ki}I_{ki}^T - 2I_{ki}I_{kj}^T + I_{kj}I_{kj}^T\right] \\
&= \sum_{k=1}^{c}\sum_{i=1}^{N_k}\left[N_k I_{ki}I_{ki}^T - 2\sum_{j=1}^{N_k} I_{ki}I_{kj}^T + \sum_{j=1}^{N_k} I_{kj}I_{kj}^T\right] \\
&= \sum_{k=1}^{c}\left[N_k\sum_{i=1}^{N_k} I_{ki}I_{ki}^T - 2\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} I_{ki}I_{kj}^T + N_k\sum_{j=1}^{N_k} I_{kj}I_{kj}^T\right] \\
&= \sum_{k=1}^{c}\left[2N_k\sum_{i=1}^{N_k} I_{ki}I_{ki}^T - 2\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} I_{ki}I_{kj}^T\right] \\
&= \sum_{k=1}^{c} 2N_k\left[\sum_{i=1}^{N_k} I_{ki}I_{ki}^T - \frac{1}{N_k}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k} I_{ki}I_{kj}^T\right]. \qquad (2.45)
\end{aligned}$$

From (2.44) and (2.45), if $N_k = N\ \forall k$,

$$S_T^{(2)} = 2NS_T^{(1)}. \qquad (2.46)$$

Since these scatter matrices are proportional each other, their eigenvectors obtained by solving the eigensystem (2.5) are identical and their corresponding eigenvalues are proportional each other with factor $2N$. □

Their recognition results using the ML similarity measure are also same because the proportionality of eigenvalues does not affect the magnitude relationship of the ML similarity measure.

This investigation also tells that when the number of samples in each class is identical, the intrapersonal subspace method chooses a projection matrix W_I by solving an optimization problem that maximizes the determinant of the within-class

scatter matrix S_W (2.7) of the projected samples, i.e.,

$$W_I = \arg\max_W |W^T S_W W| \tag{2.47}$$

instead of maximizing the determinant of the total scatter matrix S_T of the projected samples as in PCA (2.4).

PCA is optimal in the sense of minimizing the mean squared error (MMSE), which analytically means that PCA can model data distributed as Gaussian well. The above analysis tells that the intrapersonal subspace method can model data well if data in each class is distributed as Gaussian with the same covariance. In contract, the FSS method introduced in Section 2.3 is more general and can model data well even when data in each class is distributed as Gaussian with different covariance. Therefore, we suggest modeling FSS rather than the intrapersonal subspace if sufficient number of data is available for each class. In this thesis, we use the FSS model with the density estimation measure in eigenspaces described in Section 2.4.

Chapter 3

Video-Based Face Recognition

3.1 Historical Review

For decades human face recognition has been an active topic in the field of object recognition. Many algorithms have been proposed to deal with image-based recognition where both the training and test sets consist of still face images. Recently, face recognition based on video has become popular. Video-based recognition has superior advantages over image-based recognition that the temporal and motion information of faces can be utilized to facilitate the recognition task.

Zhou et al. [83] used particle filter for simultaneous face tracking and recognition in video. Their recognition task was performed by creating many image patches (particles for tracking states) in one frame and marginalizing joint probabilities respect to these image patches. Turaga et al. [73] applied statistical analysis on Stiefel and Grassmann manifolds for video-based face recognition. Edwards et al. [23] proposed an adaptive framework for learning human identity by using the motion information along the video sequence, which improves both face tracking and recognition. Li et al. [44] presented a method to construct identity surfaces using shape and texture models as well as kernel feature extraction algorithms. This approach estimates pose angle first in order to select an appropriate shape model for tracking and recognition, and video-based recognition is performed using a weighted

temporal voting scheme. Liu et al. [46] learned temporal statistics of a face from a video using adaptive Hidden Markov Models to perform video-based face recognition. Park et al. [58] proposed a 3D model-based face recognition in video. They constructed a 3D face model from probe video image sequences, and compared it with frontal faces in the gallery.

Lee et al. [43] proposed a probabilistic face recognition model in video which utilizes temporal information to estimate the facial pose at the current frame. However, this method performs face recognition on a frame-by-frame basis, and did not propose a method to perform face recognition on an entire video basis. We propose a method to perform video-based face recognition using Bayesian inference. This method propagates recognition results in the previous frames to the recognition at the current frame, and realizes face recognition using the entire video basis.

In the following, we first give a description of the probabilistic appearance manifold (PAM) method proposed by Lee et al. [43]. Then, we give a description of our video-based face recognition method using Bayesian inference.

3.2 Video-Based Face Recognition using Probabilistic Appearance Manifolds

Lee et al. [43] represents each registered person by a low-dimensional appearance manifold in the ambient image space. The complex nonlinear appearance manifold is expressed as a collection of subsets (named pose manifolds), and the connections among them. Each pose manifold is approximated by an affine plane.

To construct this representation, exemplars are sampled from videos, and these exemplars are clustered using a K-means algorithm; each cluster is represented as a plane computed using principal component analysis. Connectivities between the pose manifolds are modeled using the transition probabilities between images in each of the pose manifold and learned from a training video sequences. A maximum a posteriori formulation is presented for face recognition in test video sequences by integrating the likelihood that the input image comes from a particular pose manifold and the transition probability to this pose manifold from the previous frame.

3.2.1 Probabilistic Appearance Manifolds

Consider a recognition problem with c objects where the images of an object are acquired by varying the viewpoint. It is well understood that the set of images of an object under varying viewing conditions can be treated as a low-dimensional manifold in the image space as demonstrated in parametric appearance manifold work [56] or view-based eigenspace approach [60]. The recognition task is straightforward if the appearance manifold $\mathcal{M}_k$ for each individual k is known: for a test image I, the identity k^* can be determined by finding the manifold $\mathcal{M}_k$ with minimal "distance" to I, i.e.,

$$k^* = \arg\min_k d_H(I, \mathcal{M}_k). \tag{3.1}$$

Here, d_H denotes the L^2-Hausdorff distance between the image I and $\mathcal{M}_k$. Let $x \in \mathcal{M}_k$ denote a point on a manifold $\mathcal{M}_k$ where $dim(\mathcal{M}_k) \leq dim(I)$. Given a point $x \in \mathcal{M}_k$, let the corresponding reconstructed face image be denoted as $\hat{I}_x$

where $dim(I) = dim(\hat{I}_x)$. If x^* is the point on $\mathcal{M}_k$ at minimal L^2 distance to I, then $d_H(I, \mathcal{M}_k) = d(I, x^*)$ where $d(\cdot, \cdot)$ denotes the L^2 distance. Alternatively, x^* can be regarded as the result of some nonlinear projection of I onto $\mathcal{M}_k$.

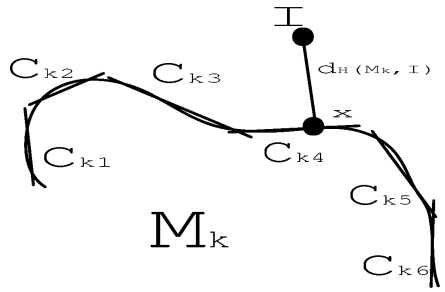


Figure 3.1: Appearance manifold. A complex and nonlinear manifold can be approximated as the union of several simpler pose manifolds. Here, each pose manifold is represented by a PCA plane [43].

Probabilistically, (3.1) is the result of defining the conditional probability $p(k|I)$ as

$$p(k|I) = \frac{1}{\Lambda} \exp\left(\frac{-1}{\sigma^2} d_H^2(I, \mathcal{M}_k)\right). \tag{3.2}$$

where Λ is a normalization term, and for a given image I

$$k^* = \arg\max_k p(k|I). \tag{3.3}$$

In order to implement this recognition scheme, one must be able to estimate the projected point $x^* \in \mathcal{M}_k$, and then the image to model distance, $d_H(I, \mathcal{M}_k)$, can be computed for a given I and for each $\mathcal{M}_k$. However, such distances can be computed accurately only if $\mathcal{M}_k$ is known exactly. In our case, $\mathcal{M}_k$ is usually not known and can only be approximated with samples. The main part of our algorithm is to

provide a probabilistic framework for estimating x^* and $d_H(x^*, I)$. Note that if we define the conditional probability $p_{\mathcal{M}_k}(x|I)$ to be the probability that among points on $\mathcal{M}_k$, $\hat{I}_{x^*}$ has the smallest L^2-distance to I, then

$$d_H(I, \mathcal{M}_k) = \int_{\mathcal{M}_k} d(x, I) p_{\mathcal{M}_k}(x|I) dx, \tag{3.4}$$

and (3.1) is equivalent to

$$k^* = \arg\min_k \int_{\mathcal{M}_k} d(x, I) p_{\mathcal{M}_k}(x|I) dx. \tag{3.5}$$

The above mentioned formulation shows that $d_H(I, \mathcal{M}_k)$ can be viewed as the expected distance between a single image frame I and a complex appearance manifold $\mathcal{M}_k$. Clearly, if $\mathcal{M}_k$ were fully known or well-approximated (e.g., described by some algebraic equations), then $p_{\mathcal{M}_k}(x|I)$ could be treated as a δ-function at the set of points with minimal distance to I. When sufficiently many samples are drawn from $\mathcal{M}_k$, the expected distance $d(I, \mathcal{M}_k)$ will be a good approximation to the true distance. The reason is that $p_{\mathcal{M}_k}(x|I)$ in the integrand in (3.4) will approach a delta function with its "energy" concentrated on the set of points with minimal distance to I. In our case, $\mathcal{M}_k$, at best, is approximated through a sparse set of samples, and so we will model $p_{\mathcal{M}_k}(x|I)$ with a Gaussian distribution.

Since the appearance manifold $\mathcal{M}_k$ is complex and nonlinear, it is reasonable to decompose $\mathcal{M}_k$ into a collection of m simpler disjoint manifolds, $\mathcal{M}_k = C^{k1} \cup \cdots \cup C^{km}$ where C^{ki} is called a pose manifold. Each pose manifold is further approximated by an affine plane computed through principal component analysis (called a PCA plane). We define the conditional probability $p(C^{ki}|I)$ for $1 \leq i \leq m$

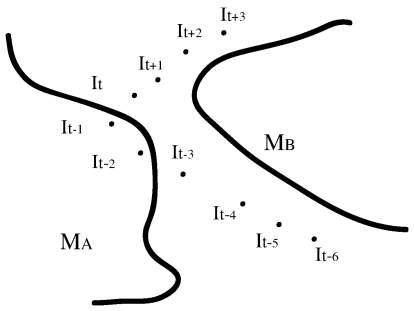


Figure 3.2: Difficulty of frame-based recognition: The two solid curves denote two different appearance manifolds, $\mathcal{M}_A$ and $\mathcal{M}_B$ It is difficult to reach a decision on the identity from frame I_{t-3} to frame I_t because these frames have smaller L^2 distance to appearance manifolds $\mathcal{M}_A$ than $\mathcal{M}_B$. However, by looking at the sequence of images $I_{t-6} \ldots I_{t+3}$, it is apparent that the sequence has most likely originated from appearance manifold $\mathcal{M}_B$ [43].

as the probability that C^{ki} contains a point x with minimal distance to I for the identity k. Since $p_{\mathcal{M}_k}(x|I) = \sum_{i=1}^{m} p(C^{ki}|I) p_{C^{ki}}(x|I)$, we have,

$$\begin{aligned} d_H(I, \mathcal{M}_k) &= \int_{\mathcal{M}_k} d(x, I) p_{\mathcal{M}_k}(x|I) dx \\ &= \sum_{i=1}^{m} p(C^{ki}|I) \int_{C^{ki}} d_H(x, I) p_{C^{ki}}(x|I) dx \\ &= \sum_{i=1}^{m} p(C^{ki}|I) d_H(I, C^{ki}). \end{aligned} \tag{3.6}$$

The above equation shows that the expected distance $d(I, \mathcal{M}_k)$ can be also treated as the average of expected distance between I and each pose manifold C^{ki} weighted by probabilities of each pose given the identity k and I. In addition, this equation transforms the integral to a finite summation which is feasible to compute numerically.

For face recognition from video sequences, we can exploit temporal coherence between consecutive image frames. As shown in Figure 3.2, the L^2 norm may occasionally be misleading during recognition. But if we consider previous frames in an image sequence rather than just one, then the set of closest points x^* will trace a curve on a pose manifold. In our framework, this is embodied by the term $p(C^{ki}|I)$ in (3.6). In Section 3.2.2, we will apply Bayesian inference to incorporate temporal information to provide a better estimation of $p(C^{ki}|I)$, and thus $d_H(I, \mathcal{M}_k)$ to achieve better recognition performance.

3.2.2 Computing $p(C_t^{ki}|I_t)$

For recognition from a video sequence, we need to estimate $p(C_t^{ki}|I_t)$ for each i at time t. To incorporate temporal information, $p(C_t^{ki}|I_t)$ should be taken as the joint conditional probability $p(C_t^{ki}|I_t, I_{0:t-1})$ where $I_{0:t-1}$ denotes the frames from the beginning up to time $t-1$. We further assume I_t and $I_{0:t-1}$ are independent given C_t^{ki}, as well as C_t^{ki} and $I_{0:t-1}$ are independent given C_{t-1}^{ki}. Using Bayes' rule we have the following recursive formulation:

$$\begin{aligned} p(C_t^{ki}|I_t, I_{0:t-1}) &= \alpha p(I_t|C_t^{ki}, I_{0:t-1}) p(C_t^{ki}|I_{0:t-1}) \\ &= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C_t^{ki}|C_{t-1}^{kj}, I_{0:t-1}) p(C_{t-1}^{kj}|I_{0:t-1}) \\ &= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C_t^{ki}|C_{t-1}^{kj}) p(C_{t-1}^{kj}|I_{t-1}, I_{0:t-2}) \end{aligned} \tag{3.7}$$

where α is a normalization term to ensure $\sum_{i=1}^{m} p(C_t^{ki}|I_t, I_{0:t-1}) = 1$.

The temporal dynamics of the video sequence is captured by the *transition probability* between the manifolds, $p(C_t^{ki}|C_{t-1}^{kj})$. Note that $p(C_t^{ki}|C_{t-1}^{kj})$ is the proba-

bility of $x_t \in C^{ki}$ given $x_{t-1} \in C^{kj}$. For two consecutive frames I_{t-1} and I_t, because of temporal coherency, we expect that their projected points x^*_{t-1} and x^*_t should have small geodesic distance on M (See Figure 3.2). That is the transition probability $p(C^{ki}_t | C^{kj}_{t-1})$ is related implicitly to the geodesic distance between C^{ki} and C^{kj}.

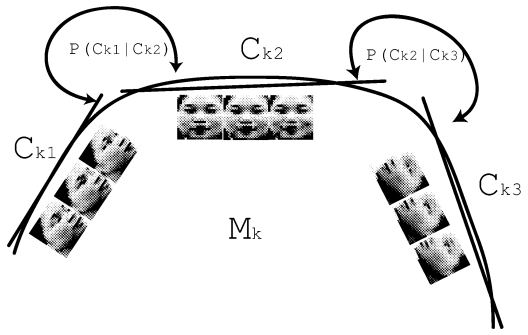


Figure 3.3: Dynamics among pose manifolds. The dynamics among the pose manifolds are learned from training videos which describes the probability of moving from one manifold to another at any time instance [43].

3.2.3 Learning Manifolds and Dynamics

For each person k, we collect at least one video sequence containing l consecutive images $S_k = \{I_1, \cdots I_l\}$. We further assume that each training image is a fair sample drawn from the appearance manifold $\mathcal{M}_k$. There are three steps in the algorithm. We first partition these samples into m disjoint subsets $\{S_{k1}, \cdots S_{km}\}$. For each collection S_{ki}, we can consider it as containing points drawn from some pose manifold C^{ki} of $\mathcal{M}_k$, and from the images in S_{ki}, we construct a linear approximation to the C^{ki} of the true manifold $\mathcal{M}_k$. After all the C^{ki} have been computed,

we estimate the transition probabilities $p(C^{ki}|C^{kj})$ for $i \neq j$.

In the first step, we apply a K-means clustering algorithm to the set of images in the video sequences. We initialize m seeds by finding m frames from the training videos with the largest L^2 distance to each other. Then the general K-means algorithm is used to assign images to the m clusters. As our goal in performing clustering is to approximate the data set rather than to derive semantically meaningful cluster centers, it is worth noting that the resulting clusters are no worse than twice what the optimal center would be if they could be easily found [32].

Second, for each S_{ki} we obtain a linear approximation of the underlying subset $C^{ki} \subset \mathcal{M}_k$ by computing a PCA plane L_{ki} of fixed dimension for the images in S_{ki}. Since the PCA planes approximate appearance manifold M_i, their dimension is the intrinsic dimension of M, and therefore all PCA planes L_i have the same dimension.

Finally, the transition probability $p(C^{ki}|C^{kj})$ is defined by counting the actual transitions between different S_i observed in the image sequence:

$$p(C^{ki}|C^{kj}) = \frac{1}{\Lambda} \sum_{q=2}^{l} \delta(I_{q-1} \in S_{ki}) \delta(I_q \in S_{kj}) \tag{3.8}$$

where $\delta(I_q \in S_{kj}) = 1$ if $I_q \in S_{kj}$ and otherwise it is 0. The normalizing constant Λ_{ki} ensures that

$$\sum_{j=1}^{m} p(C^{ki}|C^{kj}) = 1. \tag{3.9}$$

where we set $p(C^{ki}|C^{ki})$ to a constant κ. A graphic representation of a transition matrix with $m = 5$ learned from a training video is depicted in Figure 3.4.

With C^{ki} and its linear approximation L_{ki} defined, we can calculates $p(I|C^{ki})$. We can compute the L^2 distances $\hat{d}_{ki} = d_H(I, L_{ki})$ from I to each L_{ki}. We treat $\hat{d}_{ki}$ as

an estimate of the true distance from I to C^{ki}, i.e., $d_H(I, C^{ki}) = d_H(I, L_{ki})$. $p(I|C^{ki})$ is defined as

$$p(I|C^{ki}) = \frac{1}{\Lambda_k} \exp\left(\frac{-1}{2 * \sigma_{ki}^2} \hat{d}_{ki}^2\right) \tag{3.10}$$

with $\Lambda_k = \sum_{i=1}^{m} \exp\left(\frac{-1}{2*\sigma_{ki}^2} \hat{d}_{ki}^2\right)$. The variance σ_{ki}^2 is learned from the distribution of the distances from the training image sets to C^{ki}.

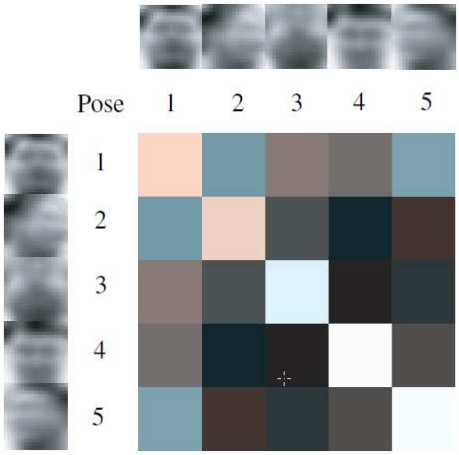


Figure 3.4: Graphic representation of a transition matrix learned from a training video. In this example, the appearance manifold is approximated by 5 pose subspaces. The reconstructed center image of each pose subspace is shown at the top raw and column. The transition probability matrix is drawn by the 5 $\times$5 block diagram. The brighter block means a higher transition probability. It is easy to see that the frontal pose (pose 1) has higher probability to change to other poses; the right pose (pose 2) has almost zero probability to directly change to the left pose (pose 3) [43].

3.2.4 Face Recognition from Video

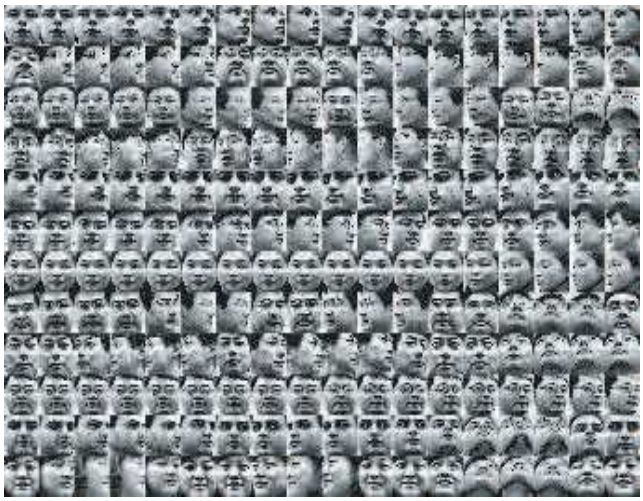


Figure 3.5: Sample gallery videos used in the experiments. Note the pose variation changed is rather large in this data set [43].

Given an image I from a video sequence, we compute for each person k the distance $d_H(I, \mathcal{M}_k)$ using (3.6). Note that $p(C^{ki}|I)$ has a temporal dependency, and it is computed recursively using (3.7). Once all the $d_H(I, \mathcal{M}_k)$ have been computed, the posterior $p(k|I)$ is computed by (3.2) with appropriate σ, and the human identity is decided by (3.5).

It is also worth mentioning that the proposed framework exploits the temporal coherence in the appearance of consecutive face images by integrating the manifold transition at the previous and current time instance. For face recognition with varying pose, our method ensures that the transitions between pose manifolds do not occur arbitrarily but rather in a constrained order. For example the appearance of one person's face cannot change immediately from left profile to right profile in

two consecutive frames, but rather it must pass through some intermediate pose or orientation (See Figure 3.5). This process can also be considered as putting a first order Markov process or finite state machine over a piecewise linear structure. In contrast, simple temporal voting scheme has been commonly adopted in most video-based face recognition methods [44] [68].

3.2.5 Recognizing Partially Occluded Faces

Similar to the formulation exploiting temporal information for recognition, the same approach can be easily extended to deal with partial occlusion of a face by considering the previous frame as prior information. The original formulation for $d_H(C_t^{ki}, I_t)$ treats every pixel in image I_t with equal weight assuming that there is no occlusion anywhere in the image sequence. If we knew which pixels corresponded to occlusions, we would put lower weights on those pixels

$$W_t^{(1)} = \exp\left(\frac{-1}{2 * \sigma^2}(\hat{I}_{x^*} - I_t). * (\hat{I}_{x^*} - I_t)\right) \tag{3.11}$$

in the first iteration. Alliteratively, W_t can be iteratively updated based on $W_t^{(1)}$ and $\hat{I}_{x^*}^{(1)}$ (i.e., the reconstructed image based on $W_t^{(1)}$ and $d_H(M_{k*}, W_t^{(1)}. * I_t)$)

$$W_t^{(i+1)} = \exp\left(\frac{-1}{2 * \sigma^2}(\hat{I}_{x^*}^{(i)} - I_t). * (\hat{I}_{x^*}^{(i)} - I_t)\right) \tag{3.12}$$

until the difference between $W_t^{(i)}$ and $W_t^{(i-1)}$ is below a threshold value at the i-th iteration.

Both the appearance manifold and mask information at previous frames are utilized to estimate the current occlusion mask in the equations above. We first perform the weighted projection to find a reconstructed image using the corresponding

pose manifold and iteratively estimate the occlusion areas in the current frame. Once we get an updated mask W_t in frame I_t by (3.11), we evaluate (3.6) for face recognition by replacing $d_H(C_t^{ki}, I_t)$ with $d_H(C_t^{ki}, W_t. * I_t)$.

Figure 3.6 shows an example where a face is partially occluded by an object (lower left). The reconstructed image using the corresponding pose manifold is shown in the lower center. The updated mask is shown in the lower right where the values have been thresholded –a dark pixel denotes a probability of occlusion. Note that the updated mask matches the occluded region reasonably well. Note also that the mask predicts that several pixels are occluded though in fact they are not. This is caused by the disagreement between the input image and the reconstructed image. Nevertheless, the regions that matter most for recognition (i.e., the central face region and the occluded region) are weighted appropriately. Our experimental results, presented in the next section, also demonstrate that the mask scheme is effective in recognizing partially occluded faces.

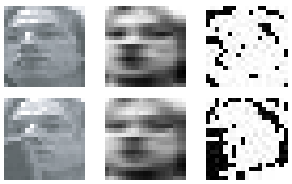


Figure 3.6: Top row: (left) an unoccluded face image, (center) a reconstructed image using corresponding pose manifold, and (right) a corresponding mask). Bottom row: (left) a face image partially occluded by one hand, (center) a reconstructed image using corresponding pose manifold, and (right) an updated mask.

3.3 Video-Based Face Recognition in Bayesian Inference

Lee et al. [43] proposed a probabilistic face recognition model in video which utilizes temporal information to estimate the facial pose at the current frame for improving face recognition. However, this method performs face recognition on a frame-by-frame basis, and they did not propose a method to perform face recognition using entire video sequence. We propose a method to perform video-based face recognition using entire video and Bayesian inference. This method propagates recognition results in the previous frames to the recognition at the current frame, and utilizes the temporal information in a video sequence to model dynamics of facial poses.

This method represents an appearance manifold of each person by a complex nonlinear appearance manifold expressed as a collection of subsets (named pose manifolds) and the connectivity among them as in [43]. Facial images are partitioned into m disjoint pose subsets manually and pose manifolds are constructed and approximated by modeling the pose images by the PCA plane. The connectivity between the pose manifolds is represented by transition probabilities between pose subsets. It is modeled by discretizing the pose transition probabilities of facial images in roll and pitch directions that are assumed to be distributed as Gaussian when no prior knowledge about the test video is available. A Bayesian inference formulation is presented to utilize the temporal information in the video, i.e., the transition probabilities between pose manifolds and to accumulate the recognition results. A maximum a posteriori is applied for face recognition after marginalizing

the posterior probabilities of pose manifolds of a particular person.

3.3.1 Problem Formulation

Consider a recognition problem with c objects where the images of an object are acquired by varying the viewpoint. Given a test set of temporal observation image sequences $\{I_t, t = 0, 1, \ldots, T-1\}$ where T is the number of frames, the recognition task is straight-forward if the manifold $\mathcal{M}_k$ for each individual k is known: identity k^* can be determined by finding the most probable manifold $\mathcal{M}_k$ that a given set of images $I_{0:T-1}$ belongs to, i.e.,

$$k^* = \arg\max_k p(\mathcal{M}_k | I_{0:T-1}). \tag{3.13}$$

where $p(\mathcal{M}_k | I_{0:T-1})$ denotes the probability that a given set of images $I_{0:T-1}$ belongs to the manifold $\mathcal{M}_k$.

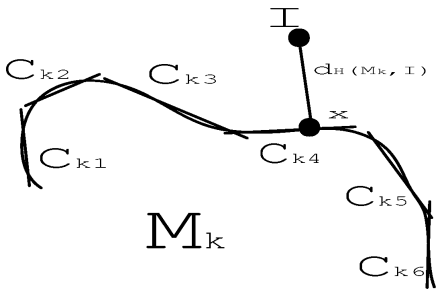


Figure 3.7: Appearance manifold. A complex and nonlinear manifold can be approximated as the union of several simpler pose manifolds; here, each pose manifold is represented by a PCA plane.

Since the appearance manifold $\mathcal{M}_k$ is complex and nonlinear, it is reasonable to decompose $\mathcal{M}_k$ into a collection of m simpler disjoint manifolds, $\mathcal{M}_k = C^{k1} \cup$

$\cdots \cup C^{km}$ where C^{ki} is called as a pose manifold and the pose manifolds $\{C^{ki}, 1 \leq k \leq c\}$ represent a corresponding pose in all c objects. Each pose manifold is further approximated by an affine plane computed through principal component analysis (called a PCA plane). We define the conditional probability $p(C^{ki}|I_{0:T-1})$ as the probability that a set of images $I_{0:T-1}$ belongs into C^{ki} manifold where we have

$$p(\mathcal{M}_k|I_{0:T-1}) = \sum_{i=1}^{m} p(C^{ki}|I_{0:T-1}). \tag{3.14}$$

For recognition from a video sequence, we need to estimate $p(C_t^{ki}|I_{0:t})$ for each i and k at time t. We assume I_t and $I_{0:t-1}$ are independent given C_t^{ki}, as well as C_t^{ki} and $I_{0:t-1}$ are independent given C_{t-1}^{ki}. Using Bayes' rule we have the following recursive formulation:

$$\begin{aligned}
p(C_t^{ki}|I_{0:t}) &= p(C_t^{ki}|I_t, I_{0:t-1}) \\
&= \alpha p(I_t|C_t^{ki}, I_{0:t-1}) p(C_t^{ki}|I_{0:t-1}) \\
&= \alpha p(I_t|C_t^{ki}) p(C_t^{ki}|I_{0:t-1}) \\
&= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C_t^{ki}, C_{t-1}^{kj}|I_{0:t-1}) \\
&= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C_t^{ki}|C_{t-1}^{kj}, I_{0:t-1}) p(C_{t-1}^{kj}|I_{0:t-1}) \\
&= \alpha p(I_t|C_t^{ki}) \sum_{j=1}^{m} p(C_t^{ki}|C_{t-1}^{kj}) p(C_{t-1}^{kj}|I_{0:t-1})
\end{aligned} \tag{3.15}$$

where α is a normalization term to ensure $\sum_{k=1}^{c} \sum_{i=1}^{m} p(C_t^{ki}|I_{0:t}) = 1$. The initial probabilities are assumed to be uniform, i.e., $p(C_0^{ki}|I_{0:-1}) = p(C_0^{ki}) = \frac{1}{mc}$. The temporal dynamics of the video sequence is captured by the *transition probability* between the pose manifolds, $p(C_t^{ki}|C_{t-1}^{kj})$.

3.3.2 Learning Manifolds

For each person k, collect a set of training images $\{I_{kt}\}_{t=1}^{N_T}$ and form a training set of vectors $\{\mathbf{x}_{kt}\}_{t=1}^{N_T}$ where $\mathbf{x}_k \in \mathcal{R}^D$ and D is the number of dimensions of the vector, by lexicographic ordering of the pixel elements of each image I_{kt}. We partition these samples into m facial pose disjoint subsets $\{S_{k1}, \cdots, S_{km}\}$ manually so that subsets $\{S_{ki}, 1 \leq k \leq c\}$ are partitioned into the same facial pose. From the images in S_{ki}, we construct a linear PCA approximation L_{ki} to the C^{ki} of the true manifold $\mathcal{M}_k$. With C^{ki} and its linear PCA approximation L_{ki} defined, we can define how $p(I|C^{ki})$ can be calculated. Moghaddam et al. [53] proposed a method for density estimation in eigenspaces.

This method uses PCA to obtain a principal subspace L_{ki} whose principal components $\mathbf{y}_{ki}$ can be used to form an optimal (minimal divergence) low-dimensional estimate of the complete likelihood using only the first M_{ki} principal components $\{y_{ki1}, y_{ki2}, y_{ki3}, \ldots, y_{kiM_{ki}}\}$, where $M_{ki} \ll D$. We define $p(I|C^{ki})$ as the likelihood density estimate $\hat{P}(\mathbf{x}|L_{ki})$ of a test image $\mathbf{x}$ in L_{ki} which can be written as the product of two independent marginal Gaussian densities, i.e.,

$$p(I|C^{ki}) \triangleq \hat{P}(\mathbf{x}|L_{ki}) = \left[\frac{\exp\left(-\frac{1}{2}\sum_{\ell=1}^{M_{ki}}\frac{y_{ki\ell}^2}{\lambda_{ki\ell}}\right)}{(2\pi)^{M_{ki}/2}\prod_{\ell=1}^{M_{ki}}\lambda_{ki\ell}^{1/2}}\right]\left[\frac{\exp\left(-\frac{1}{2}\frac{\epsilon_{ki}^2(\mathbf{x})}{\rho_{ki}}\right)}{(2\pi\rho_{ki})^{(D-M_{ki})/2}}\right] \quad (3.16)$$

where $\{\lambda_{ki\ell}\}_{\ell=1}^{M_{ki}}$ are the eigenvalues of L_{ki}, $\{y_{ki\ell}\}_{\ell=1}^{M_{ki}}$ are the principal components in L_{ki}, and $\epsilon_{ki}^2(\mathbf{x})$ is the PCA residual (reconstruction error) for L_{ki}. The density parameter ρ_{ki} is derived by minimizing the Kullback-Leibler (KL) divergence and is

found to be

$$\rho_{ki} = \frac{1}{D - M_{ki}} \sum_{\ell=M_{ki}+1}^{D} \lambda_{ki\ell}. \tag{3.17}$$

In practice, M_{ki} is set to be identical for all identities and poses, i.e., $M_{ki} = M$, $\forall k$ and $\forall i$ to provide fairness.

3.3.3 Modeling Dynamics

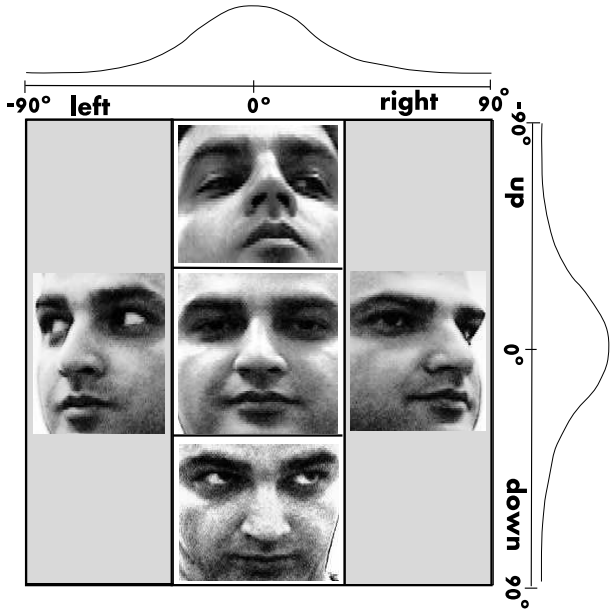


Figure 3.8: Examples of face images from front, left, right, up, and down pose subsets. The curves represent the pose transition probabilities given the frontal face pose which are distributed as Gaussians. The images are collected from Honda/UCSD Video Database [43].

When no prior knowledge for dynamics in a probe video is available, we assume that the transition probability between poses is independent from the identity k, and further assume that the continuous conditional probabilities of pose transitions in

both roll (up-down profile) and pitch (left-right profile) direction are distributed on Gaussian densities and independent of each other, i.e.,

$$p(\phi_t^x|\phi_{t-1}^x) = \frac{1}{\sqrt{2\pi}\sigma_{\phi^x}} \exp\left(-\frac{(\phi_t^x - \phi_{t-1}^x)^2}{2\sigma_{\phi^x}^2}\right), \quad (3.18)$$

$$p(\phi_t^y|\phi_{t-1}^y) = \frac{1}{\sqrt{2\pi}\sigma_{\phi^y}} \exp\left(-\frac{(\phi_t^y - \phi_{t-1}^y)^2}{2\sigma_{\phi^y}^2}\right), \quad (3.19)$$

$$p(\phi_t^x, \phi_t^y|\phi_{t-1}^x, \phi_{t-1}^y) = p(\phi_t^x|\phi_{t-1}^x)p(\phi_t^y|\phi_{t-1}^y) \quad (3.20)$$

where ϕ_t^x and ϕ_t^y are the continuous roll and pitch rotation angles at time t, and $\sigma_{\phi^x}^2$ and $\sigma_{\phi^y}^2$ are variances in the roll and pitch directions respectively.

We partition the joint set of continuous pose parameters (ϕ^x, ϕ^y) into m disjoint facial pose subsets $\{\mathcal{P}^1, \ldots, \mathcal{P}^m\}$ and define the transition probability between pose subsets as

$$p(\mathcal{P}_t^i|\mathcal{P}_{t-1}^j) = \alpha \int_{(\phi_t^x, \phi_t^y) \in \mathcal{P}_t^i} p(\phi_t^x, \phi_t^y|\phi_{t-1}^x, \phi_{t-1}^y = \overline{\phi^x}^j{}_{t-1}, \overline{\phi^y}^j{}_{t-1})\mathrm{d}(\phi_t^x, \phi_t^y) \quad (3.21)$$

where $\overline{\phi^x}^j$ and $\overline{\phi^y}^j$ are the center angles in the pose subset $\mathcal{P}^j$ for the roll and pitch directions respectively and α is a normalization constant to ensure $\sum_{i=1}^m p(\mathcal{P}_t^i|\mathcal{P}_{t-1}^j) = 1$. From the assumption that the transition probability between poses is independent from the identity k, $p(C_t^{ki}|C_{t-1}^{kj})$ is defined as

$$\begin{aligned} p(C_t^{ki}|C_{t-1}^{kj}) &= p(\omega_t^k|\omega_{t-1}^k, \mathcal{P}_{t-1}^j)p(\mathcal{P}_t^i|\omega_{t-1}^k, \mathcal{P}_{t-1}^j) \\ &= p(\mathcal{P}_t^i|\mathcal{P}_{t-1}^j). \end{aligned} \quad (3.22)$$

where ω denotes an identity random variable and $p(\omega_t^k|\omega_{t-1}^k, \mathcal{P}_{t-1}^j) = p(\omega_t^k|\omega_{t-1}^k) = 1$. A graphic example of pose subsets where $m = 5$ and the pose transition probabilities given the frontal face pose is illustrated in Figure 3.8.

3.3.4 Face Recognition from Video

Given an image I_t from a video sequence, we compute for each person k the likelihood probability $p(I_t|C^{ki})$ using (3.16). The posterior probability $p(C^{ki}|I_{0:t})$ is computed recursively by propagating results from previous frames using (3.15). Once all the $p(C^{ki}|I_{0:t})$ is computed, the posterior $p(\mathcal{M}_k|I_{0:t})$ is computed using (3.14).

We recursively repeat the above steps until the maximum identity confidence exceeds a given confidence threshold τ, i.e, $\max_k p(\mathcal{M}_k|I_{0:t}) \geq \tau$. Finally, the human identity k^* is determined by (3.13).

This method has four useful characteristics: (1) it utilizes temporal information in a video, (2) it accumulates recognition results in frames, (3) it progressively obtains the recognition confidence, and (4) it does not require to process all frames in a video. The 2nd accumulation characteristic possibly enables to solve face recognition problems in low-resolution videos. The 3rd progressive characteristic is useful especially in a real-time processing because we do not need to wait to receive an entire probe video before processing. The 4th characteristic results in a computational efficiency over batch methods.

3.3.5 Experiments and Results

In this section, experimental results are given. We use the Honda/UCSD video dataset [43] for experiments. The Honda/UCSD video dataset consists of a set of 45 videos of 20 different people. Each individual in the database has at

least two videos where each person moves in a different combination of 2-D and 3-D rotation, expressions, and speed. Each video lasting for 20 seconds was recorded in an indoor environments (with 30 color frames of 640 × 480 pixels per second). We cropped facial image patches manually in the video as shown in Figure 3.9, and then each image was downsampled to 20 × 20 pixels, to imitate the image quality in surveillance systems. The pixels in each image were normalized to have zero mean and unit variance.

Figure 3.9: Examples from Honda/UCSD video dataset [43]

3.3.5.1 Cropping Facial Images

To crop facial image patches in a image sequence manually fast, we have created a software named *imageclipper* and the software is available online [67]. This software is useful not only for facial images but also for any kinds of images and works under multi-platform (Windows and Linux). Using this software, we can (1) open images in a directory sequentially, (2) open a video, frame by frame, (3)

crop (save) an image patch and go to the next image by pressing one button, (4) move and resize the rectangle region to crop by hotkeys or right mouse button, and (5) rotate and shear deform, i.e., affine transform, the rectangle region. A snapshot of the software is shown in Figure 3.10.

Figure 3.10: A snapshot of the software *imageclipper*

3.3.5.2 Semi-Automation of Pose Clustering

We need to partition a facial image sequence in a training video into different pose subsets. In this experiment we partitioned the image set into the front, left, right, up, and down pose subsets, i.e., $m = 5$ as in [43]. This partitioning was performed semi-automatically using the K-means clustering method. In the Honda/UCSD video dataset, 2-D appearance variations produced by pose variations are much higher than the one caused by illumination and facial expression

variations. Therefore, the unsupervised K-means clustering algorithm can partition an image set into different pose clusters in most cases. Utilizing this characteristic, we partitioned a facial image sequence by applying the K-means algorithm to the raw image vectors, and then inspected the results manually. Examples in the generated pose subsets are shown in Figure 3.11.

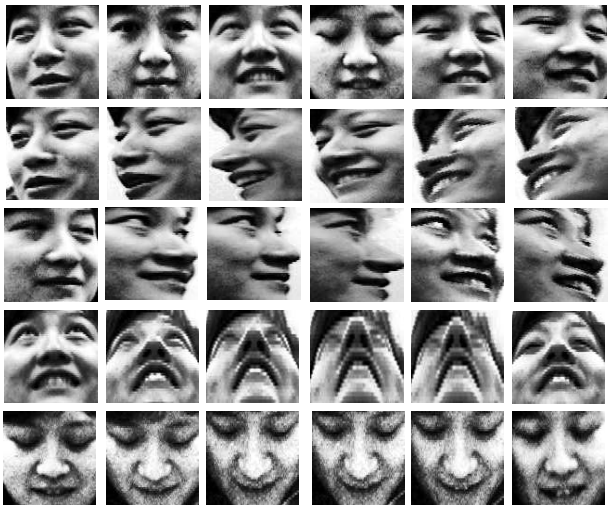


Figure 3.11: Examples in the pose subsets. The top row presents frontal faces, the 2nd top row presents left-profile faces, the middle row presents right-profile faces, the 2nd bottom row presents up-profile faces, and the bottom row presents the bottom-profile faces. The left/right-up and left/right-down profile faces are included in the left/right-profile subsets respectively. Images were resized to have a square size.

3.3.5.3 Confidence Update

Figure 3.12 presents a plot of the posterior probability $p(\mathcal{M}_k|I_{0:t})$ versus time t for a test video. This experimental result shows a behavior of our proposed method

such that the confidence in recognition is gradually increased as time proceeds.

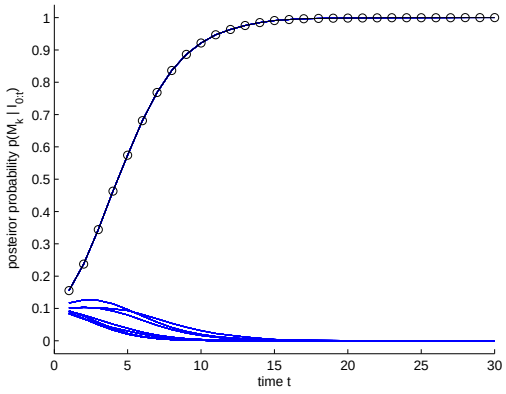
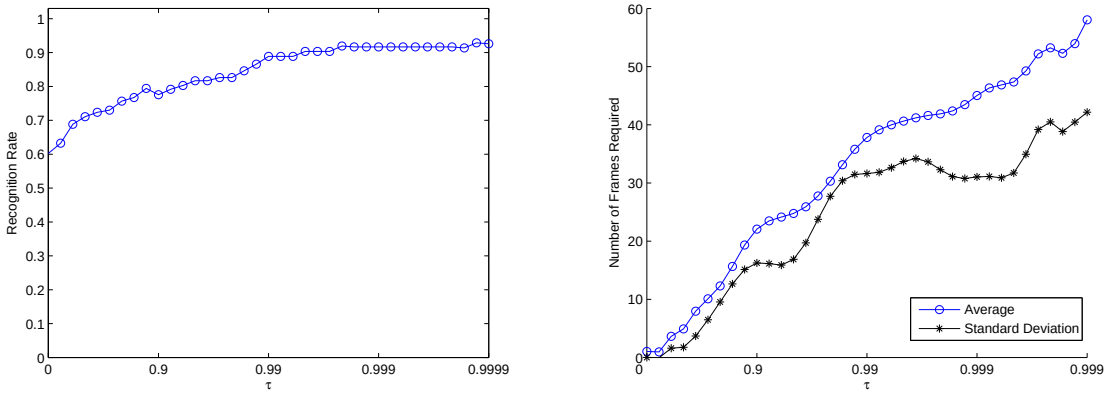


Figure 3.12: Posterior probability $p(\mathcal{M}_k|I_{0:t})$ against time t obtained by the proposed algorithm. Notice that the confidence $p(\mathcal{M}_k|I_{0:t})$ is gradually increased as time proceeds.

3.3.5.4 Good Choice of the Confidence Threshold

Our proposed face recognition algorithm can stop to process video frames at time t when the posterior confidence $p(\mathcal{M}_k|I_{0:t})$ achieves a confidence threshold τ. In this experiment we examine a good number of the confidence threshold τ. Figure 3.13(a) presents a plot of recognition rate versus τ, and Figure 3.13(b) presents a plot of the average number of frames required to process and its standard deviation versus τ. There exists 20 test videos, i.e., one video for each individual in the Honda/UCSD dataset. We split each test video in every 150 frames, where the video originally have $300 \sim 500$ frames, to increase the number of test videos. We downsampled each facial image to 20×20 and reduced the number of feature dimensions to 20 by

projecting them onto PCA subspaces. Figure 3.13(a) shows that the recognition rate increases as the confidence threshold increases. However, Figure 3.13(b) shows that the number of frames required to process also increases as the confidence threshold increases. The practical choice seems to be $\tau = 0.999$ from this experiment.



(a) (b)

Figure 3.13: (a) Face recognition rate versus τ. (b) The average number of frames required to process and its standard deviation versus τ. Each image size was 20×20 and these images were reduced into 20 dimensions by projecting onto PCA subspace. Face recognition at $\tau = 0$ tells a result using one image, i.e., a result of image-based face recognition.

3.3.5.5 Face Recognition

Finally, the recognition results in the HONDA/UCSD test videos are presented here. The facial image patches in the 20 training videos and 20 test videos, i.e., one video for each individual respectively, were manually cropped, and the appearance manifolds were trained using the facial images in the training videos. The number of

test videos was increased to 78 by dividing the original test video sequence to have 150 frames, where original videos typically have $300 \sim 500$ frames. Each cropped facial image was downsampled to 20×20 pixels and the pixels in each image were normalized to have zero mean and unit variance. Each image was then projected onto the PCA subspaces to reduce the number of feature dimension. The confidence threshold $\tau = 0.999$ was used. Figure 3.14 shows the cumulative match curves [63] when 20 dimensional features were used. The recognition rate was about 92%. We also applied a frame-by-frame strategy using the same appearance model and likelihood measurement, and performed recognition in a video by temporal majority voting. The recognition rate was 83% in this case. When 40 dimensional features were used, our proposed algorithm achieved the 100% recognition rate by processing 37.95 frames on average.

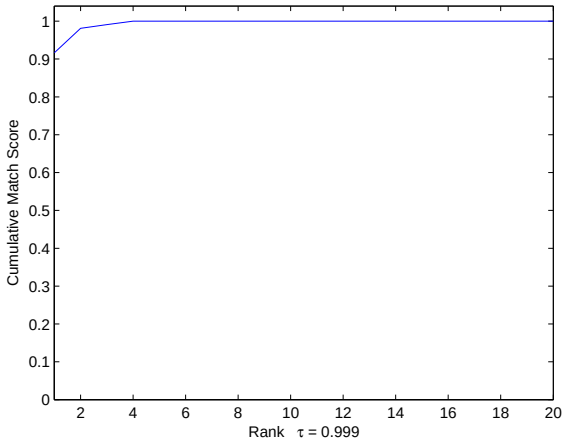


Figure 3.14: The cumulative matching score versus rank where the 1st rank shows the recognition rate. 20 identities are in the database, and the number of test videos is 78. The confidence threshold $\tau = 0.999$ which processes 45.04 frames in average was used. Each image size was 20×20 and these images were reduced to 20 dimensions by projecting onto PCA subspace. When 40 dimensional features were used, 100% recognition rate was achieved.

Chapter 4

Object Tracking using Bayesian Filtering

4.1 Introduction

Many problems require estimation of the state of a system that changes over time, using a sequence of noisy measurements. In order to analyze and make inference about a dynamic system we need two models, a **transition model** describing the evolution of the state $\{\mathbf{x}_k,\ k \in \mathbb{N}\}$ with time, and an **observation model** relating the noisy measurements $\{\mathbf{z}_k,\ k \in \mathbb{N}\}$ to the state.

In Bayesian framework, all relevant information about $\{\mathbf{x}_0,\ \mathbf{x}_1,\ ..,\ \mathbf{x}_k\}$ given observations up to and including time k, can be obtained from the posterior distribution $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. In many applications we are interested in recursively estimating this distribution, and particularly the marginals, the so-called filtering distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. This problem is known as the Bayesian filtering problem or the optimal filtering problem [1].

A recursive filtering approach means that received data can be processed sequentially rather than in a batch mode, so that it is neither necessary to store the complete data set nor to reprocess existing data if a new measurement becomes available. Such a filter consists of essentially two stages: *prediction* and *update*. The prediction stage uses the transition model to predict the state distribution forward from one measurement time to the next. Since the state is usually subject to

unknown disturbances (modeled as random noise), the prediction generally translates, deforms, and spreads the state distribution. The update operation uses the latest measurement to modify the prediction distribution. This is achieved using the Bayes theorem, which is a mechanism for updating knowledge about the target state in the light of additional information from new data.

In the following, we give a description of the nonlinear tracking problem and its optimal solutions. When certain constraints hold, this optimal solutions are tractable. Often, the optimal solution is intractable so we take approximation strategies to the optimal solution using particle filters.

4.2 Nonlinear Bayesian tracking

To define the problem of tracking, consider the evolution of the state sequence $\{\mathbf{x}_k,\ k \in \mathbb{N}\}$ of a target given by

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \tag{4.1}$$

where $\mathbf{f}_k(.)$ is a possibly nonlinear function of the state $\mathbf{x}_{k-1}$ and $\{\mathbf{v}_k,\ k \in \mathbb{N}\}$ is an i.i.d. process noise sequence. The objective of tracking is to recursively estimate the posterior distribution from measurements:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k,\ \mathbf{n}_k), \tag{4.2}$$

where $\mathbf{h}_k(.)$ is a possibly nonlinear function, $\{\mathbf{v}_k,\ k \in \mathbb{N}\}$ is an i.i.d. measurement noise sequence. In particular, we seek filtered estimates of the posterior based on the set of all available measurements $\mathbf{z}_{1:k} = \{\mathbf{z}_i,\ i = 1,\ \ldots,\ k\}$, up to time k.

From a Bayesian perspective, the tracking problem is one of recursively calculating some degree of belief in the state $\mathbf{x}_k$ at time k, given the data $\mathbf{z}_{1:k}$ up to time k. Thus, it is required to construct the probability density function (pdf) $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. It is assumed that the initial pdf $p(\mathbf{x}_0|\mathbf{z}_0) \equiv p(\mathbf{x}_0)$, of the state vector, also known as the prior, is available. Then, in principle, the pdf $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ may be obtained recursively in two stages: prediction and update.

Suppose that the required pdf $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ at time $k-1$ is available. The prediction stage involves using the transition model (4.1) to obtain the prior pdf of the state at time k via the Chapman-Kolmogorov equation [1]:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}. \tag{4.3}$$

Note that a Markov process of order one $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$ has been used. The probabilistic model of the state evolution, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, is defined by the system equation (4.1) and the known statistics of $\mathbf{v}_{k-1}$.

At time step k, a measurement $\mathbf{z}_k$ becomes available, and this may be used to update the prior (update stage) via Bayes rule

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{4.4}$$

where the normalizing constant

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})\mathrm{d}\mathbf{x}_k \tag{4.5}$$

depends on the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ defined by the measurement model (4.2) and the known statistics of $\mathbf{n}_k$. In the update stage, the measurement is used to modify the prior density to obtain the required posterior density of the current state.

The recurrence relations (4.3) and (4.4) form the basis for the optimal Bayesian solution [1]. This recursive propagation of the posterior density is only a conceptual solution in that, generally, it cannot be analytically determined. However, solutions do exist for restrictive set of cases, including the Kalman filter and grid-based filters.

4.3 Optimal Algorithms

4.3.1 Kalman Filter

The Kalman filter [39] assumes that the posterior density at every time step is Gaussian distributed and, hence, represented by a mean and covariance. If $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is Gaussian, it can be shown that $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is also Gaussian, provided that some assumptions hold:

- $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are drawn from Gaussian distributions with known parameters.
- $\mathbf{f}_k(\mathbf{x}_{k-1},\ \mathbf{v}_{k-1})$ is known, and is a linear function of $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$.
- $\mathbf{h}_k(\mathbf{x}_k,\ \mathbf{n}_k)$ is a known linear function of $\mathbf{x}_k$ and $\mathbf{n}_k$.

That is, system equations (4.1) and (4.2) can be rewritten as:

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{4.6}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{n}_k \tag{4.7}$$

[1] For clarity, the optimal Bayesian solution solves the problem of recursively calculating the exact posterior density. An optimal algorithm is a method for deducing this solution.

where $\mathbf{F}_k$ and $\mathbf{H}_k$ are the known matrices that define the linear functions. The covariances of $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are respectively Q_{k-1} and R_k. Here we consider the case when $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ have zero mean and are statistically independent. Note that the system and measurement matrices F_k and H_k, as well as noise parameters Q_{k-1} and R_k, are allowed to be time variant.

The Kalman filter algorithm, derived using (4.3) and (4.4), can then be viewed as the following recursive relationship:

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}\ ;\ m_{k-1|k-1},\ P_{k-1|k-1}) \tag{4.8}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k\ ;\ m_{k|k-1},\ P_{k|k-1}) \tag{4.9}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k\ ;\ m_{k|k},\ P_{k|k}) \tag{4.10}$$

where

$$m_{k|k-1} = F_k m_{k-1|k-1} \tag{4.11}$$

$$P_{k|k-1} = Q_{k-1} + F_k P_{k-1|k-1} F_k^T \tag{4.12}$$

$$m_{k|k} = m_{k|k-1} + K_k(\mathbf{z}_k - H_k m_{k|k-1}) \tag{4.13}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{4.14}$$

and where $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m and covariance P and:

$$S_k = H_k P_{k|k-1} H_k^T + R_k, \tag{4.15}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \tag{4.16}$$

are the covariance of the innovation term $\mathbf{z}_k - H_k m_{k|k-1}$, and the Kalman gain, respectively.

This is the optimal solution to the tracking problem –if the (highly restrictive) assumptions hold. The implication is that no algorithm can ever do better than a Kalman filter in this linear Gaussian environment.

4.3.2 Grid-based Filter

Grid-based methods [1] provide the optimal recursion of the filtered density, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, if the state space is discrete and consists of a finite number of states. Suppose the state space at time $k-1$ consists of discrete states $\mathbf{x}_{k-1}^i,\ i = 1,\ \ldots,\ N_s$. For each state $\mathbf{x}_{k-1}^i$, let the conditional probability of that state, given measurements up to time $k-1$ be denoted by $w_{k-1|k-1}^i$, that is, $\Pr(\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^i|\mathbf{z}_{1:k-1}) = w_{k-1|k-1}^i$. Then, the posterior pdf at $k-1$ can be written as

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \sum_{i=1}^{N_s} w_{k-1|k-1}^i \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^i) \tag{4.17}$$

where $\delta(\cdot)$ is the Dirac delta measure. Substitution of (4.17) into (4.3) and (4.4) yields the prediction and update equations, respectively:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \sum_{i=1}^{N_s} w_{k|k-1}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{4.18}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \sum_{i=1}^{N_s} w_{k|k}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{4.19}$$

where

$$w_{k|k-1}^i \triangleq \sum_{j=1}^{N_s} w_{k-1|k-1}^j p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^j), \tag{4.20}$$

$$w_{k|k}^i \triangleq \frac{w_{k|k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i)}{\sum_{j=1}^{N_s} w_{k|k-1}^j p(\mathbf{z}_k|\mathbf{x}_k^j)} \tag{4.21}$$

The above assumes that $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^j)$ and $p(\mathbf{z}_k|\mathbf{x}_k^i)$ are known, but does not constrain the particular form of these discrete densities. Again, this is the optimal solution if the assumptions hold.

4.4 Particle Filtering

4.4.1 Sequential Importance Sampling (SIS)

The Sequential Importance Sampling (SIS) particle filter algorithm is a Monte Carlo (MC) method that forms the basis for most sequential MC filters developed over the past decades [20]. The sequential MC (SMC) approach is known variously as bootstrap filtering [27], the condensation algorithm [48], particle filtering [13], interacting particle approximations [19], [55] and survival of the fittest [40]. It is a technique for implementing a recursive Bayesian filter using MC simulations. The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, this MC characterization becomes an equivalent representation to the usual functional description of the posterior pdf, and the SIS filter approaches the optimal Bayesian estimate.

In order to develop the details of the algorithm, let $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_s}$ denote a random measure that characterizes the posterior pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, where $\{\mathbf{x}_{0:k}^i,\ i = 1,\ \ldots,\ N_s\}$ is a set of support points with associated weights $\{w_k^i,\ i = 1,\ \ldots,\ N_s\}$, and $\mathbf{x}_{0:k} = \{\mathbf{x}_0,\ \mathbf{x}_1,\ \ldots,\ \mathbf{x}_k\}$, is the set of all states up to time k. The weights are normalized such that $\sum_{i=1}^{N_s} w_k^i = 1$. Then, the posterior density at time k can be

approximated as:

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i). \tag{4.22}$$

Therefore, we have a discrete weighted approximation to the true posterior, $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. The weights are chosen using the principle of *Importance Sampling* [20]. This principle relies on the following assumptions. Suppose that $p(\mathbf{x}) \propto \pi(\mathbf{x})$ is a probability density from which it is difficult to draw samples, but for which $\pi(\mathbf{x})$ can be evaluated (as well as $p(\mathbf{x})$ up to proportionality). Also, let $\mathbf{x}^i \sim q(\mathbf{x}), i = 1, \cdots, N_s$ be samples that are easily generated from a proposal $q(.)$ called an *importance density*. Then, a weighted approximation to the density $p(.)$ is given by

$$p(\mathbf{x}) \approx \sum_{i=1}^{N_s} w^i \delta(\mathbf{x} - \mathbf{x}^i) \tag{4.23}$$

where

$$w^i \propto \frac{\pi(\mathbf{x}^i)}{q(\mathbf{x}^i)} \tag{4.24}$$

is the normalized weight of the i^{th} particle.

Therefore, if the samples $\mathbf{x}_{0:k}^i$ were drawn from an importance density $q(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$, then the weights are defined to be:

$$w_k^i \propto \frac{p(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}. \tag{4.25}$$

Returning to the sequential case, at each iteration, one could have samples constituting an approximation to $p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$ and it is required to approximate $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ with a new set of samples. If the importance density is chosen to factorize such that

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}), \tag{4.26}$$

then one can obtain samples $\mathbf{x}^i_{0:k} \sim q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ by augmenting each of the existing samples $\mathbf{x}^i_{0:k-1}$ with the new sample $\mathbf{x}^i_k \sim q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$. To derive the weight update equation, $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ is first expressed in terms of $p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$, $p(\mathbf{z}_k|\mathbf{x}_k)$, and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$:

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}). \tag{4.27}$$

By substituting, the weight update equation can then be shown to be

$$w^i_k \propto w^i_{k-1}\frac{p(\mathbf{z}_k|\mathbf{x}^i_k)p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k|\mathbf{x}^i_{0:k-1}, \mathbf{z}_{1:k})}. \tag{4.28}$$

Furthermore, if $q(\mathbf{x}_k|\mathbf{x}_{0:k-1},\ \mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1},\ \mathbf{z}_k)$, then the importance density becomes only dependent on $\mathbf{x}_{k-1}$ and $\mathbf{z}_k$. This is useful in the common case when only a filtered estimate of the posterior distribution is required at each time step. In such scenarios, only $\mathbf{x}^i_k$ need to be stored; therefore, one can discard the path $\mathbf{x}^i_{0:k}$ and the history of observations. The modified weight becomes then

$$w^i_k \propto w^i_{k-1}\frac{p(\mathbf{z}_k|\mathbf{x}^i_k)p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k|\mathbf{x}^i_{k-1}, \mathbf{z}_k)}, \tag{4.29}$$

and the posterior filtered density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be approximated as:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w^i_k \delta(\mathbf{x}_k - \mathbf{x}^i_k) \tag{4.30}$$

where the weights are defined in (4.29). It can be shown that as $N_s \to \infty$ the approximation approaches the true posterior density, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ [18]. The SIS algorithm thus consists of recursive propagation of the weights and support points, as each measurement is sequentially received.

The state estimate $\hat{\mathbf{x}}_k$ can either be the minimum mean square error (MMSE) estimate

$$\hat{\mathbf{x}}_k^{\text{MMSE}} = E[\mathbf{x}_k|\mathbf{z}_{1:k}] \approx \sum_{j=1}^{N_s} w_k^i \mathbf{x}_k^i \tag{4.31}$$

or the maximum *a posteriori* (MAP) estimate

$$\hat{\mathbf{x}}_k^{\text{MAP}} = \arg\max_{\mathbf{x}_k} p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \arg\max_{\mathbf{x}_k^i} w_k^i \tag{4.32}$$

or other forms based on $p(\mathbf{x}_k|\mathbf{z}_{1:k})$.

Degeneracy Problem

A common problem with the SIS particle filter is the degeneracy phenomenon, where after a few iterations, all but one particle have negligible weight. It has been shown that the variance of the importance weights can only increase over time and, thus, it is impossible to avoid the degeneracy phenomenon. This degeneracy implies that a large computational effort is devoted to update particles whose contribution to the approximation is almost zero. A suitable measure of degeneracy of the algorithm is the effective sample size N_{eff} introduced in [45], and defined as

$$N_{\text{eff}} = \frac{N_s}{1 + Var(w_k^{*i})} \tag{4.33}$$

where $w_k^{*i} = p(\mathbf{x}_k^i|\mathbf{z}_{1:k})/q(\mathbf{x}_k^i|\mathbf{x}_{1:k-1}^i,\ \mathbf{z}_k)$, is referred to as the "true weight". This cannot be evaluated exactly, but an estimate $\hat{N}_{\text{eff}}$ of N_{eff} can be obtained by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_s}(w_k^i)^2} \tag{4.34}$$

where w_k^i is the normalized weight obtained using (4.29). Notice that $N_{\text{eff}} \leq N_s$, and a small N_{eff} indicates severe degeneracy. Clearly, the degeneracy problem is

undesirable. The brute force approach reduce this effect, uses a very large value for N_s. Since this is often impractical, usually one relies on a technique called *Resampling* described next.

4.4.2 Resampling

The basic idea of resampling is to eliminate particles that have small weights and to concentrate on particles with large weights. The resampling step involves generating a new set $\{\mathbf{x}_k^{i*}\}_{i=1}^{N_s}$ by resampling (with replacement) N_s times from an approximate discrete representation of $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ given by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i). \tag{4.35}$$

so that $\Pr(\mathbf{x}_k^{i*} = \mathbf{x}_k^j) = w_k^j$. The resulting sample is in fact an i.i.d. sample from the discrete density above; therefore, the weights are now reset to $w_k^i = 1/N_s$.

Although the resampling step reduces the effects of the degeneracy problem, it introduces other practical problems. First, it limits the opportunity to parallelize since all the particles must be combined. Second, the particles that have high weights are, statistically, selected many times. This leads to a loss of diversity among the particles as the resultant sample will contain many repeated points. This problem, which is known as *sample impoverishment*, is severe in the case of small process noise. In fact, for the case of very small process noise, all particles will collapse to a single point within a few iterations.

The sequential importance sampling algorithm presented in Section 4.4.1 forms the basis for most particle filters that have been developed so far. The various ver-

sions of particle filters proposed in the literature can be regarded as special cases of this general SIS algorithm. These special cases can be derived from the SIS algorithm by an appropriate choice of importance sampling density and/or modification of the resampling step. Below, one of these approach proposed in [27] is presented.

4.4.3 Sampling Importance Resampling

The Sampling Importance Resampling (SIR) Filter [27] is a Monte Carlo method that can be applied to solve recursive Bayesian filtering problems. The assumptions required to use the SIR filter are very weak. The state dynamics $\mathbf{f}_k(\cdot,\cdot)$ and measurement functions $\mathbf{h}_k(\cdot,\cdot)$ need to be known, and it is required to be able to sample realizations from the process noise distribution of $\mathbf{v}_{k-1}$ and from the prior distribution $p(\mathbf{x}_k|\mathbf{z}_{1_k-1})$. Finally, the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ needs to be available for pointwise evaluation (at least up to proportionality). The SIR algorithm can be easily derived from the SIS algorithm by an appropriate choice of: (i) The importance density, where $q(\mathbf{x}_k|\mathbf{x}^i_{k-1},\ \mathbf{z}_k)$ is chosen to be the density $p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$, and (ii) Resampling step, which is to be applied at every time index.

The above choice of importance density implies that we need samples from $p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$. A sample $\mathbf{x}^i_k \sim p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$ can be created by first generating a process noise sample $\mathbf{v}^i_{k-1}$, and setting $\mathbf{x}^i_k = \mathbf{f}_k(\mathbf{x}^i_{k-1}, \mathbf{v}^i_{k-1})$. For this particular choice of importance density, it is evident that the weights are given by:

$$w^i_k \propto w^i_{k-1} p(\mathbf{z}_k|\mathbf{x}^i_k). \tag{4.36}$$

However, noting that resampling is applied at every time index, we have $w^i_{k-1} =$

$1/N \ \ \forall i$; therefore

$$w_k^i \propto p(\mathbf{z}_k|\mathbf{x}_k^i). \tag{4.37}$$

The weights given by the proportionality in (4.37) are normalized before the resampling stage.

As the importance sampling density for the SIR filter is independent of measurement $\mathbf{z}_k$, the state space is explored without any knowledge of the observations. Therefore, this filter can be inefficient and sensitive to outliers[2]. Furthermore, as resampling is applied at every iteration, this can result in rapid loss of diversity in particles, that is a problem introduced as a *sample impoverishment*. However, the SIR method does have the advantage that the importance weights are easily evaluated, and that the importance density can be easily sampled.

4.5 Condensation

Sequential Monte Carlo algorithms have gained prevalence in the visual tracking literature due in part to the Condensation (**Con**ditional **Den**sity propag**ation**) algorithm [34], which belongs to the class of SIR filters.

Spatio-temporal estimation has been dealt with thoroughly by Kalman filtering in the relatively clutter-free case, in which $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be modeled as Gaussian. These solutions work poorly in clutter that causes the posterior density to be multi-modal and, therefore, non-Gaussian. In the simple Gaussian case, the density function evolves as a Gaussian pulse that translates, spreads and is reinforced,

[2]In statistics, an outlier is a single observation far away from the rest of the data.

remaining throughout Gaussian, as in Figure 4.1(a).

The random component of the dynamical model leads to spreading, increasing uncertainty, while the deterministic component causes the density function to drift bodily. The effect of an external observation $\mathbf{z}_k$ is to superimpose a reactive effect and, consequently, the density tends to peak in the vicinity of observations. In clutter there are typically several competing observations, and these tend to encourage a non-Gaussian state density (Figure 4.1(b)).

In Condensation the output of an iteration will be a weighted, time-stamped sample-set, denoted $\{\mathbf{s}_k^{(i)},\ i = 1,\ ..,\ N_s\}$, with weights w_k^i approximately representing the conditional state-density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ at time k. Clearly the process must begin with a prior density and the effective prior for time step k should be $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$. This prior is, of course, multi-modal in general and no functional representation of it is available. It is derived from the sample set representation $\{\mathbf{s}_{k-1}^{(i)},\ \pi_{k-1}^{(i)},\ i = 1, 2,\ ..,\ N_s\}$ of $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ the output from the previous time step, to which the prediction

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{4.38}$$

must then be applied. The iterative process, as applied to sample sets, is depicted in Figure 4.2 and mirrors the continuous diffusion process in Figure 4.1(b). At the top of the diagram, the output from time step $k-1$ is the weighted sample set $\{\mathbf{s}_{k-1}^{(i)}, w_{k-1}^{(i)}, i = 1, .., N_s\}$. The first operation, therefore, is to sample from the set, with replacement, N_S times, choosing a given element with probability w_{k-1}^i.

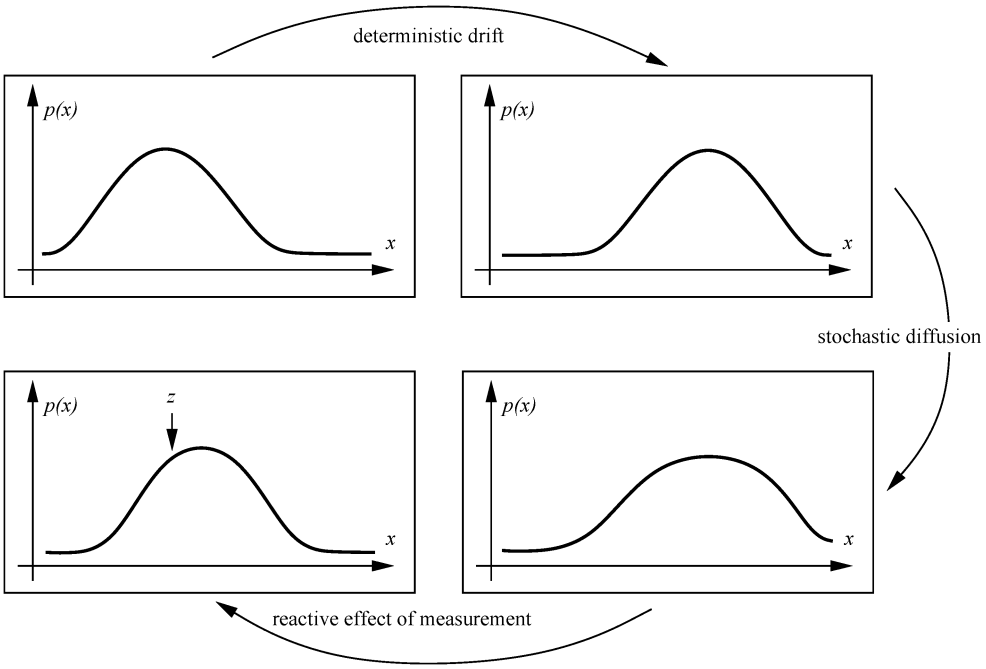
Some elements, especially those with high weights, may be chosen several

times, leading to identical copies of elements in the new set. Others with relatively low weights, may not be chosen at all. Each element chosen from the new set is now subjected to the predictive steps. The predictive step is random and identical elements now split because each one undergoes its own independent Brownian motion step. At this stage, the sample set $\{\mathbf{s}_i^k\}$ for the new time step has been generated, but without its weights. Finally, the observation step is applied, generating weights from the observation density to obtain the sample set representation of state density for time k.

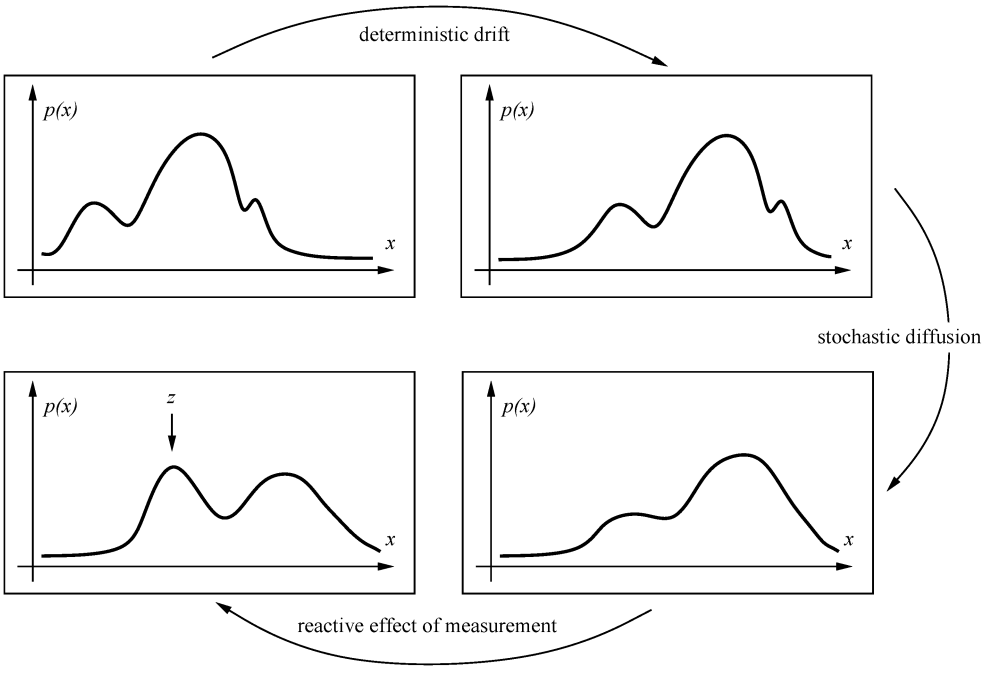
Figure 4.3 gives a synopsis of the algorithm in its original formulation [9].

4.5.1 ICondensation

The Condensation can be extended to incorporate the statistical technique of Importance Sampling. Importance sampling offers a mathematically principled way of directing search combining prediction information based on the previous object position and motion with any additional knowledge available. In the standard formulation of the Condensation algorithm (see Figure 4.3) positions of samples $\mathbf{s}_k^{(n)}$ are fixed in the prediction stage using only the previous approximation of the state density $\{\mathbf{s}_k^{(n)},\ w_k^{(n)}\}$ and the motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The portions of state space which are to be examined in the measurement stage are therefore determined before any measurements are made. This is appropriate when the sample set approximation to the state density is accurate. In principle as the state density evolves over time the random nature of the motion model induces some non-zero probability everywhere in



(a) Gaussian case



(b) Non-Gaussian case

Figure 4.1: Probability density propagation [34].

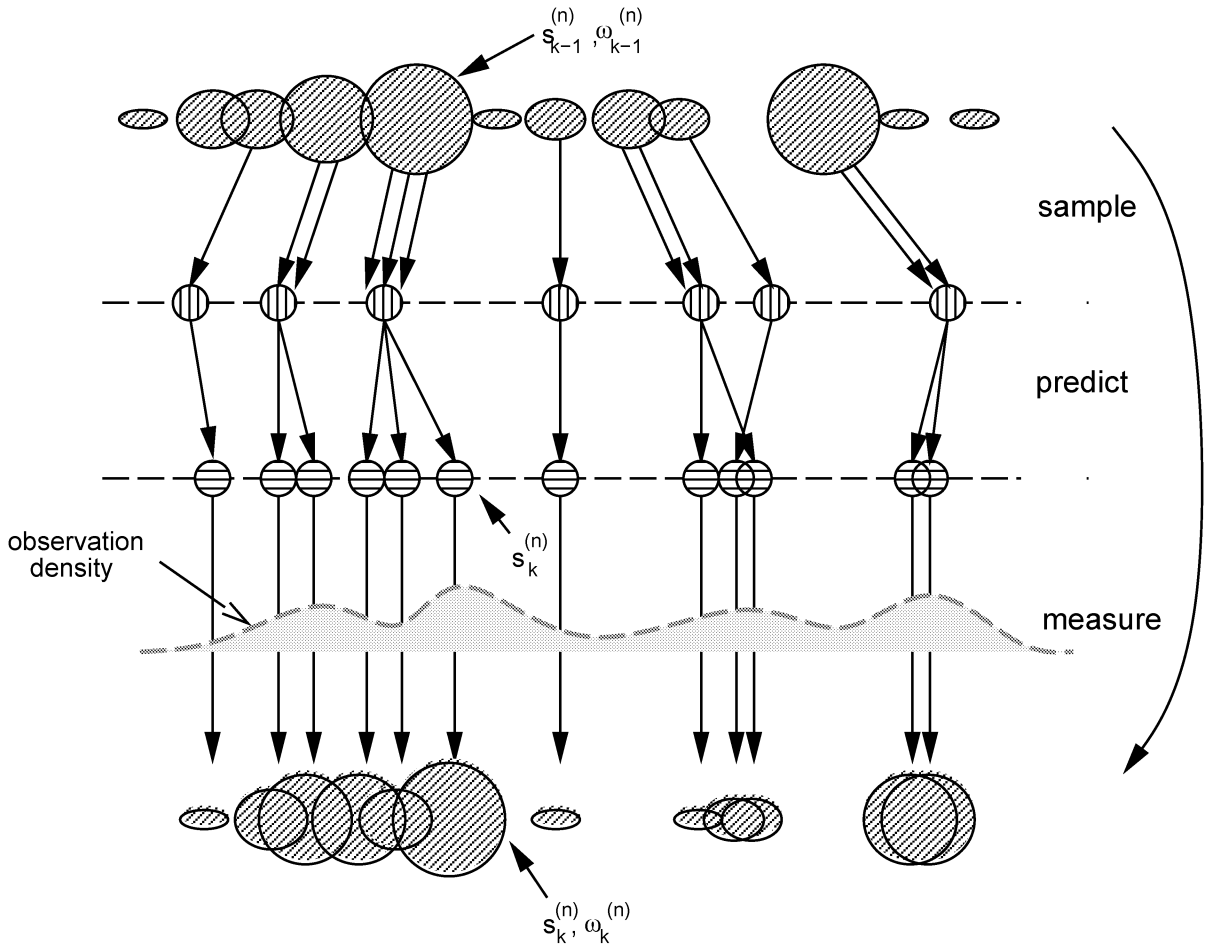


Figure 4.2: Condensation Steps. One time-step in the Condensation algorithm [34].

Iterate

From the "old" sample-set $\{\mathbf{s}_{t-1}^{(n)},\ \pi_{t-1}^{(n)},\ c_{t-1}^{(n)},\ n\ = 1,\ \ldots,\ N\}$ at time-step $t-1$, construct a "new" sample-set $\{\mathbf{s}_t^{(n)},\ \pi_t^{(n)},\ c_t^{(n)}\}$, $n = 1,\ \ldots,\ N$ for time t. Construct the $n^{\mathbf{th}}$ of N new samples as follows:

1. **Resample** a sample $\mathbf{s}_t^{\prime(n)}$ as follows:
   (a) generate a random number $r \in [0, 1]$, uniformly distributed.
   (b) find, by binary subdivision, the smallest j for which $c_{t-1}^{(j)} \geq r$
   (c) set $\mathbf{s}_t^{\prime(n)} = \mathbf{s}_{t-1}^{(j)}$
2. **Predict** by sampling from
   $$p(\mathbf{x}_t|\mathbf{x}_{t-1} = \mathbf{s}_t^{\prime(n)})$$
   to choose each $\mathbf{s}_t^{(n)}$. For instance, in the case that the dynamics are governed by a 1inear stochastic differential equation, the new sample value maybe generated as: $\mathbf{s}_t^{(n)} = \mathbf{A}\mathbf{s}_t^{(n)} + B\mathbf{w}_t^{(n)}$ where $\mathbf{w}_t^{(n)}$ is a vector of standard normal random variables, and BB^T is the process noise covariance.
3. **Measure** and weight the new position in terms of the measured features $\mathbf{z}_t$:
   $$\pi_t^{(n)} = p(\mathbf{z}_t|\mathbf{x}_t = \mathbf{s}_t^{(n)})$$
   then normalise so that $\sum_n \pi_t^{(n)} = 1$ and store together with cumulative probability as $(\mathbf{s}_t^{(n)},\ \pi_t^{(n)},\ c_t^{(n)})$ where
   $$c_t^{(0)} = 0, c_t^{(n)} = c_t^{(n-1)} + \pi_t^{(n)} (n = 1, \ldots, N).$$

Once the N samples have been constructed: estimate, if desired, moments of the tracked position at time-step t as

$$\mathcal{E}[f(\mathbf{x}_t)] = \sum_{n=1}^{N} \pi_t^{(n)} f(\mathbf{s}_t^{(n)})$$

obtaining, for instance, a mean position using $f(\mathbf{x}) = \mathbf{x}$.

Figure 4.3: Condensation Algorithm in its original formulation [34].

state space that the object is present at that point. With a sufficiently good sample set approximation this would tend to cause all areas of state space to lie near some samples so even motions which were extremely unlikely given the model would be detected and could therefore be tracked. In practice, however, the finite nature of the sample set approximation means that all of the samples will be concentrated near the most likely object positions. There may be several such clusters corresponding to multiple hypotheses but in general each cluster will be fairly localised, which is precisely the behavior which permits an efficient discrete representation of high dimensional state spaces. The result is that large areas of state space contain no samples at all. In order to robustly track sudden movements the process noise of the motion model must be artificially high thus increasing the extent of each predicted cluster in state space. To populate these larger clusters with enough samples to permit effective tracking the sample set size must be increased and the algorithm therefore runs more slowly. Various techniques have been proposed to improve the efficiency of the representation in random sampling filters [27]. Importance sampling applies when auxiliary knowledge is available in the form of an importance function $q(\mathbf{x})$ describing which areas of state-space contain most information about the posterior. Importance sampling can be applied in the context of Condensation sampling and this extension is called ICondensation [35]. The idea is to concentrate samples in those areas of state space by generating sample positions $\mathbf{s}_k^{(n)}$ from an importance function $q(\mathbf{x}_k)$. The desired effect is to avoid as far as possible generating any samples which have low weights, since they are "wasted" as they provide a negligible contribution to the posterior. A correction term f/q must be added to

the sample weights giving:

$$w_k^{(n)} = \frac{f(\mathbf{s}_k^{(n)})}{q(\mathbf{x}_k^{(n)})} p(\mathbf{z}_k|\mathbf{x}_k = \mathbf{s}_k^{(n)}) \quad \text{where} \quad f(\mathbf{s}_k^{(n)}) = p(\mathbf{x}_k = \mathbf{s}_k^{(n)}|\mathbf{z}_{1_k-1}) \tag{4.39}$$

to compensate for the uneven distribution of sample positions. This correction term ensures that, for large N_S, importance sampling has no effect on the consistency of the approximation. The effect of the correction ratio is to preserve the information about motion coherence which is present in the dynamical model. Although the samples are positioned according to $q(\mathbf{x}_k)$ the distribution approximated by $\{\mathbf{s}_k^{(n)},\ w_k^{(n)}\}$ still generates $p(\mathbf{x}_k|\mathbf{z}_k)$. Importance sampling is again intended to improve the efficiency of the sample set representation but does not change the probabilistic model.

Chapter 5

Simultaneous Multi-View Face Tracking and Recognition using Particle Filtering

5.1 Introduction

Conventionally, face recognition in a video is performed in tracking-*then*-recognition scenario that extracts the best facial image patch in the tracking stage and then recognizes the given the facial image. The scenario handles uncertainties in both tracking and recognition separately using different appearance models. Simultaneous face tracking and recognition works by handling both uncertainties in tracking and recognition simultaneously using common appearance models. The simultaneous tracking and recognition framework is shown to be an effective scenario that improves both tracking and recognition accuracies over the tracking-*then*-recognition scenario [83].

The Bayesian recursive filtering method, which estimates the state of a system that changes over time, is often used for object tracking in the computer vision community. Early works [11], [2] used the Kalman filter for object tracking. Recently, the particle filtering method which is also called as the sequential Monte Carlo (SMC) algorithms have gained popularity in the visual tracking literature since the Condensation algorithm [33] was introduced.

We use the Bayesian filtering framework to solve the tracking problem. For recognition, we propose a video-based face recognition algorithm using the Bayesian filtering framework in Section 3.3. Using the proposed video-based face recognition algorithm, we can easily integrate the tracking and recognition methods into one solving both problem simultaneously because both use the Bayesian filtering framework. We use the particle filter to solve the simultaneous tracking and recognition problem. In this simultaneous framework the temporal information in a video is utilized not only for tracking but also for recognition because our proposed video-based face recognition utilizes the temporal information to model the dynamics of facial poses. Although the time series formulation is more general, only the facial pose dynamics is utilized for recognition in this thesis.

5.2 Overview of Particle Filtering

Particle filtering [1] is a technique for implementing a recursive Bayesian filter by Monte Carlo simulation to estimate the unknown state $\mathbf{x}_t$ from a noisy collection of observations $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ arriving in a sequential fashion. A state space model is often employed to accommodate such a time series. Two important components of this approach are state transition and observation models whose most general forms can be defined as follows:

$$\text{State transition model}: \quad \mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{v}_t) \tag{5.1}$$

$$\text{Observation model}: \quad \mathbf{z}_t = \mathbf{h}_t(\mathbf{x}_t, \mathbf{n}_t) \tag{5.2}$$

where $\mathbf{v}_{t-1}$ is the system noise, $\mathbf{f}_t(\cdot,\cdot)$ characterizes the kinematics, $\mathbf{n}_t$ is the observation noise, and $\mathbf{h}_t(\cdot,\cdot)$ models the observer. The particle filter approximates the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ by a set of weighted particles $\{\mathbf{x}_t^{(j)}, w_t^{(j)}\}_{j=1}^{N_s}$ where N_s is the number of particles, $\{\mathbf{x}_t^{(j)}\}_{j=1}^{N_s}$ is a set of support states, and $\{w_t^{(j)}\}_{j=1}^{N_s}$ is a set of associated weights that are normalized such that $\sum_{j=1}^{N_s} w_t^{(j)} = 1$. The general particle filter algorithm is written in Figure 5.1.

Initialize a sample set $\{\mathbf{x}_0^{(j)}, 1\}_{j=1}^{N_s}$ according to prior distribution $p(\mathbf{x}_0)$.
For $t = 1, 2, \cdots$
  Resample $\{\mathbf{x}_{t-1}^{(j)}, w_{t-1}^{(j)}\}$ based on $w_{t-1}^{(j)}$ and obtain a new sample $\{\mathbf{x}_{t-1}^{'(j)}, 1\}$ $\forall j$
  Predict $\mathbf{x}_t^{(j)}$ by sampling from $\mathbf{x}_t^{(j)} = \mathbf{f}_t(\mathbf{x}_{t-1}^{'(j)}, \mathbf{v}_t)$ $\forall j$
  Measure and update weights from $w_t^{(j)} = p(\mathbf{z}_t|\mathbf{x}_t^{(j)})$ $\forall j$
  Normalize weights using $w_t^{(j)} = w_t^{(j)} / \sum_{i=1}^{N_s} w_t^{(i)}$ $\forall j$
End

Figure 5.1: General particle filter algorithm [1].

5.3 Simultaneous Framework

In the simultaneous tracking and recognition framework, the state vector $\mathbf{x}_t$ is composed of the identity state denoted by ω_t and the other states denoted by $\mathbf{s}_t$, i.e., $\mathbf{x}_t = \{\mathbf{s}_t, \omega_t\}$. The essence of the simultaneous framework is the computation of the posterior probability $p(\mathbf{s}_t, \omega_t|\mathbf{z}_{1:t})$, whose marginal posterior probability $p(\omega_t|\mathbf{z}_{1:t})$ solves the recognition task.

In our framework, we further partition the state vector $\mathbf{s}_t$ into the 2D tracking state vector θ_t and the appearance state vector ψ_t, i.e., $\mathbf{x}_t = \{\mathbf{s}_t, \omega_t\} = \{\theta_t, \psi_t, \omega_t\}$. The 2D tracking state vector θ_t describes the location of the object in the 2D ob-

served image that is used to extract an image patch displaying the object, and the appearance state vector ψ_t describes the parameters used in modeling the appearance.

Although the time series formulation is general, in this thesis, we set $\theta_t = \{x_t, y_t, s_t^x, s_t^y, \phi_t^z\}$ where x_t and y_t denote the central coordinate of the object in x-coordinate and the y-coordinate (in terms of pixels) respectively, s_t^x and s_t^y denote the width and height of the object respectively, and ϕ_t^z denotes the counterclockwise rotation angle of the object in the observed 2D image at time t. ϕ_t^z also expresses the 3D yaw rotation angle of the object. In addition we set $\psi_t = \{\phi_t^x, \phi_t^y\}$ where ϕ_t^x and ϕ_t^y denote 3D roll and pitch rotation angles of the object at time t. The video-based face recognition algorithm described at Section 3.3 is used to solve a simultaneous multi-view face tracking and recognition problem. As described in Section 3.3.2, the roll and pitch rotation angles are used to model the appearance manifold.

5.4 Observation Model

The observation model is based on the appearance manifold described in Section 3.3.2. This model approximates a complex and nonlinear manifold of a face as the union of several simpler pose manifolds where each pose manifold is represented by a PCA plane, and captures the temporal dynamics in the video sequence by the *transition probability* between the pose manifolds as shown in Figure 5.2.

For the observation model, we first construct a linear PCA approximation L_{ki} to the pose manifold C^{ki} of the true manifold $\mathcal{M}_k$ modeling a face of the

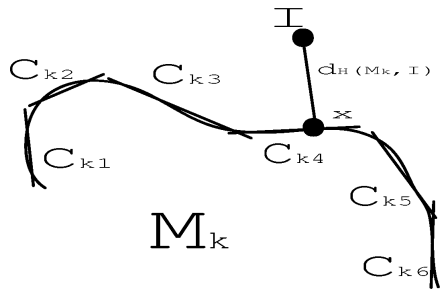


Figure 5.2: Appearance manifold. A complex and nonlinear manifold can be approximated as the union of several simpler pose manifolds; here, each pose manifold is represented by a PCA plane [43].

identity k where i denotes the discrete pose label, and the likelihood probability $p(I|C^{ki})$ of a test image I in C^{ki} is used for an observation measurement. The calculation of $p(I|C^{ki})$ was described in Section 3.3.2. With L_{ki}, its principal components $\mathbf{y}_{ki}$ can be used to form an optimal (minimal divergence) low-dimensional estimate of the complete pdf using only the first M_{ki} principal components $\{y_{ki1}, y_{ki2}, y_{ki3}, \ldots, y_{kiM_{ki}}\}$, where $M_{ki} \ll D$ and D is the number of feature dimension. The $p(I|C^{ki})$ is defined as the likelihood density estimate $\hat{P}(I|L_{ki})$ which can be written as the product of two independent marginal Gaussian densities, i.e.,

$$p(I|C^{ki}) \triangleq \hat{P}(I|L_{ki}) \quad = \quad \left[\frac{\exp\left(-\frac{1}{2}\sum_{\ell=1}^{M_{ki}}\frac{y_{ki\ell}^2}{\lambda_{ki\ell}}\right)}{(2\pi)^{M_{ki}/2}\prod_{\ell=1}^{M_{ki}}\lambda_{ki\ell}^{1/2}}\right]\left[\frac{\exp\left(-\frac{1}{2}\frac{\epsilon_{ki}^2(\mathbf{I})}{\rho_{ki}}\right)}{(2\pi\rho_{ki})^{(D-M_{ki})/2}}\right] \tag{5.3}$$

where $\{\lambda_{ki\ell}\}_{\ell=1}^{M_{ki}}$ are the eigenvalues of L_{ki}, $\{y_{ki\ell}\}_{\ell=1}^{M_{ki}}$ are the principal components in L_{ki}, and $\epsilon_{ki}^2(I)$ is the PCA residual (reconstruction error) for L_{ki}, and ρ_{ki} is given

by

$$\rho_{ki} = \frac{1}{D - M_{ki}} \sum_{\ell=M_{ki}+1}^{D} \lambda_{ki\ell}. \tag{5.4}$$

In practice, M_{ki} is set to be identical for all identities and poses, i.e., $M_{ki} = M,\ \forall k$ and $\forall i$ to provide fairness.

Finally, the likelihood function $p(\mathbf{z}_t|\mathbf{x}_t) = p(\mathbf{z}_t|\theta_t, \psi_t, \omega_t)$ which indicates the likelihood probability that the hypothesized state $\mathbf{x}_t$ gives rise to the observed data is defined as

$$p(\mathbf{z}_t|\theta_t, \psi_t, \omega_t^k) = p(I_t|C^{ki}) \tag{5.5}$$

where $I_t = I(\mathbf{z}_t, \theta_t)$ is the image patch sampled at the hypothesized tracking state θ_t, k is the identity, and i is the discrete pose label given by $i = g(\psi)$ where $g(\cdot)$ is a discretizing function of the joint set of continuous pose parameter $\psi = (\phi^x, \phi^y)$ described in following.

5.4.1 Discretization of Poses

The observation model introduced in the previous section needs to discretize the joint set of continuous pose parameter $\psi = (\phi^x, \phi^y)$ into the discrete pose label i. As described in 3.3.3, the joint set of continuous pose parameters (ϕ^x, ϕ^y) is partitioned into m disjoint facial pose subsets $\{\mathcal{P}^1, \cdots, \mathcal{P}^m\}$, thus the discretizing function $g(\cdot)$ is defined as

$$g(\psi_t) \triangleq \arg\max_l I^{(l)}(\psi_t) \tag{5.6}$$

where $I^{(l)}(\psi_t)$ is an indicator function

$$I^{(l)}(\psi_t) = \begin{cases} 1 & \text{if } (\phi_t^x, \phi_t^y) \in \mathcal{P}^l \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

5.5 State Transition Model

The state transition model characterizes the dynamic behavior of states between frames. The state space equation describing the evolution of the state sequence $\{\mathbf{x}_t, t \in \mathbb{N}\}$ can be written in its general form as:

$$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{v}_t) \tag{5.8}$$

where $\mathbf{f}_t(\cdot)$ is a possibly nonlinear function of the state $\mathbf{x}_{t-1}$ and $\{\mathbf{v}_t, t \in \mathbb{N}\}$ is a process noise sequence.

In practice, however, choosing an appropriate function $\mathbf{f}(\cdot)$ is not an easy task and depends on the particular situation in a video, thus an approximate model is used. There are three types of approximations commonly found in the literature:

1. A fixed constant-velocity model with fixed noise variance as in [7, 6, 80, 83]:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_t \tag{5.9}$$

where $\mathbf{v}_t$ has a fixed noise variance of the form $\mathbf{v}_t = R_0 * \mathbf{v}_0$ with R_0 a fixed constant measuring the extent of noise and $\mathbf{v}_0$ a standard normal random variable/vector[1]. If R_0 is small, it is very hard to model rapid movements;

[1]Consider the scalar case, for example. If $\mathbf{v}_t$ is distributed as $N(0, \sigma^2)$, we can write $\mathbf{v}_t = \sigma\mathbf{v}_0$, where $\mathbf{v}_0$ is standard normal $N(0, 1)$. This also applies to multivariate cases.

if R_0 is large, it is computationally inefficient, since many more particles are needed to accommodate the large noise variance. These factors make such a model ineffective. Therefore, an adaptive velocity model is needed.

2. An adaptive velocity model with fixed/adaptive noise variance as in [82]

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathrm{d}\mathbf{x}_t + \mathbf{v}_t \tag{5.10}$$

where $\mathrm{d}\mathbf{x}_t$ is a velocity state vector. In practice, the velocity is estimated as $\hat{\mathrm{d}}\mathbf{x}_t = \mathbf{x}_{t-1} - \mathbf{x}_{t-2}$ using a 1st order linear approximation [29, 38, 4, 47], thus the (5.10) forms a special case of the 2nd order AR model:

$$\mathbf{x}_t = 2\mathbf{x}_{t-1} - \mathbf{x}_{t-2} + \mathbf{v}_t \tag{5.11}$$

The 2nd order AR model was also used in [33].

3. Nth order AR model. North et al. [57] identified an Nth order AR transition model from a training video. However, such a model may overfit the training data and may not necessarily succeed when presented with testing videos containing objects arbitrarily moving at different times and places. Also, one cannot always rely on the availability of training data.

We use both an adaptive velocity model and a fixed-constant velocity model with adaptive noise variance. The framework and the modeling of the adaptive noise variance is described in Section 5.6. We further assume that the noise is distributed as independent Gaussian with zero mean. Please note that angle parameters are in fact assumed to be distributed as "wrapped" Gaussian distribution so that $\phi \in [0, 360)$.

5.5.1 Identity State Transition Model

As a special case, we assume that the identity state ω_t does not change as time proceeds, i.e., the identity state equation is given by

$$\omega_t = \omega_{t-1}. \tag{5.12}$$

However, it often happens that one identity dominates identity states in all particles after the resampling process because particles with the same identity have high likelihood probabilities concurrently in observation measurements. This behavior causes the simultaneous face tracking and recognition method to track faces using a model of a fixed identity in succeeding frames. This results in improved tracking especially when the identity estimation is correct, however, this may adversely affect face recognition accuracy in succeeding frames and may also cause poor tracking especially when the identity estimation is incorrect. Therefore, we propose to diffuse identity states uniformly random with $\beta\%$ chances. Specifically, we diffuse identity states uniformly random with 50% chances in our experiments.

5.6 Adaptive Particle Filter

The efficiency and accuracy of the particle filter depends on the number of particles and noise variance in the state transition equation. We need large noise variance to track rapid movements, and more particles are required to achieve accuracy. Recently, several approaches have been introduced to adapt the number of particles over time [82], [26], [72]. The adaptive approach in [26], [72] adjusts the number of particles based on the likelihood of observations by generating particles

until the sum of the non-normalized likelihoods exceeds a pre-specified threshold. Zhou et al. [82] proposed an algorithm to adapt noise variance based on the quality of prediction, i.e., the residual error measurements of observations and adapt the number of particles based on the adaptation of noise variance. We basically follow the approach of Zhou et al. [82].

5.6.1 Measure of Prediction Quality

To adapt number of particle filters and noise variance, we first determine a measure of prediction quality. We propose to determine the prediction quality ε_t by the distances between an image and the observation models, which is defined by:

$$\varepsilon_t = \min_j \hat{d}_{ki}(I_t) = \min_j \left[\sum_{\ell=1}^{M_{ki}} \frac{y_{ki\ell}^2}{\lambda_{ki\ell}} + \frac{\epsilon_{ki}^2(I_t)}{\rho_{ki}} \right]. \tag{5.13}$$

where $I_t = I(\mathbf{z}_t, \theta_t^{(j)})$ is the image patch sampled at the tracking state $\theta_t^{(j)}$, i is the discrete pose label given by $i = g(\psi_t^{(j)})$, k is the identity given by $k = \omega_t^{(j)}$. The distance metric $\hat{d}_{ki}(I_t)$ is sum of the distance-*in*-feature-space and the distance-*from*-feature-space [53] between the image patch I_t and the PCA subspace L_{ki} that are calculated in (5.3).

5.6.2 Adaptive Noise

Zhou et al. [82] proposed an adaptive noise model based on the quality of prediction. The adaptive noise is given by the form:

$$\mathbf{v}_t = R_t \mathbf{v}_0 \tag{5.14}$$

where R_t is a measure of the extent of noise and $\mathbf{v}_0$ is a standard Gaussian random variable/vector. Following their work [82], we define R_{t+1} by

$$R_{t+1} = \max(\min(R_0\sqrt{\varepsilon_t/\eta}, R_{\max}), R_{\min}) \tag{5.15}$$

where $R_{\min}$ is the lower bound to maintain a reasonable sample coverage, $R_{\max}$ is the upper bound to constrain the computational load, ε_t is a prediction error defined in (5.13), and η is an acceptable prediction error threshold. The acceptable threshold value η is determined by trial and error. We use the square root because ε_t is a variance-type measure.

The meaning of such a choice is explained as follows: if the quality of prediction is good, i.e., $\varepsilon_t < \eta$, we need noise with small variance to absorb the residual motion; if the quality of prediction is poor, i.e., $\varepsilon_t > \eta$, we then need noise with large variance to cover potentially large jumps in the motion state.

5.6.3 Adaptive Number of Particles

If the noise variance R_t is large, we need more particles, while, conversely, fewer particles are needed for noise with small variance R_t. Based on the principle of asymptotic relative efficiency (ARE) [14], we should adjust the particle number $N_{s(t)}$ in a similar fashion, i.e.

$$N_{s(t)} = \frac{N_{s(0)}R_t}{R_0} \tag{5.16}$$

5.6.4 One Frame Iteration

Furthermore, we propose to iterate particle filtering in one frame until the simultaneous tracker achieves small enough prediction error ε_t, i.e., $\varepsilon_t < \eta$ where η is an acceptable prediction error determined by trial and error. This iteration adapts the number of particles by resampling new particles and performing observation measurements with different states in the same observation, and adapts noise variance by repeating the diffusion process. The inequality $\varepsilon_t < \eta$ provides a verification of that the tracking has succeeded. This verification process considerably improves the tracking and recognition accuracy because it is often difficult for a tracker to recover in succeeding frames when it has failed tracking objects once. Although this iteration process attempts to adapt the number of samples and noise variance, still using the adaptive noise model and adaptive number of particles in each iteration is attractive.

Moreover, we propose to switch the state transition model from an adaptive velocity model to a fixed-constant velocity model in this iteration because we use a fixed time observation in this iteration. The previous state $\mathbf{x}_{t-1}$ should be retained for use in the adaptive velocity model at the next frame. One may realize this strategy by formulating the adaptive velocity model as

$$\begin{aligned} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-1} \end{pmatrix} &= \begin{pmatrix} 2\mathbf{x}_{t-1} - \mathbf{x}_{t-2} + \mathbf{v}_t \\ \mathbf{x}_{t-1} \end{pmatrix} \\ &= \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_{t-1} \\ \mathbf{x}_{t-2} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_t \\ 0 \end{pmatrix} \end{aligned} \tag{5.17}$$

and the fixed-constant velocity model as

$$\begin{pmatrix} \mathbf{x}_{t+1} \\ \mathbf{x}_t \end{pmatrix} = \begin{pmatrix} \mathbf{x}_t + \mathbf{v}_{t+1} \\ \mathbf{x}_{t-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_{t-1} \end{pmatrix} + \begin{pmatrix} \mathbf{v}_{t+1} \\ 0 \end{pmatrix} \tag{5.18}$$

with an abuse of notation. In practice, one may limit the number of iterations in one frame. A state estimation at time t such as estimation of a tracking state estimation should be performed after this iteration step is done.

5.7 Face Recognition

The main objective of the simultaneous framework is face recognition. Face recognition is performed using the maximum *a posteriori* (MAP) estimate as

$$k^* = \arg\max_k p(\omega_t^k | \mathbf{z}_{1:t}). \tag{5.19}$$

Therefore, our goal is to obtain the posterior probability $p(\omega_t^k|\mathbf{z}_{1:t})$, which is in fact a probability mass function (PMF) because ω_t is a discrete random vector indicating the person identity. Also, the joint posterior probability $p(\mathbf{x}_t|\mathbf{z}_{1:t}) = p(\mathbf{s}_t, \omega_t|\mathbf{z}_{1:t})$ is a mixed distribution.

Using particle filtering, the posterior pdf $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is approximated as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \sum_{j=1}^{N_s} w_t^{(j)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(j)}), \quad \text{equivalently,} \tag{5.20}$$

$$p(\mathbf{s}_t, \omega_t|\mathbf{z}_{1:t}) \approx \sum_{j=1}^{N_s} w_t^{(j)} \delta(\mathbf{s}_t - \mathbf{s}_t^{(j)}) \delta(\omega_t - \omega_t^{(j)}). \tag{5.21}$$

where $\delta(\cdot)$ the Dirac delta function. The posterior probability $p(\omega_t^k|\mathbf{z}_{1:t})$ is then obtained by marginalizing the posterior probability $p(\mathbf{s}_t, \omega_t^k|\mathbf{z}_{1:t})$ respect to the state $\mathbf{s}_t$, i.e.,

$$\begin{aligned} p(\omega_t^k|\mathbf{z}_{1:t}) &= \int p(\mathbf{s}_t, \omega_t^k|\mathbf{z}_{1:t})\mathrm{d}\mathbf{s}_t \\ &\approx \int \sum_{j=1}^{N_s} w_t^{(j)} \delta(\mathbf{s}_t - \mathbf{s}_t^{(j)})\delta(\omega_t^k - \omega_t^{(j)})\mathrm{d}\mathbf{s}_t \\ &\approx \sum_{j=1}^{N_s} w_t^{(j)} \delta(\omega_t^k - \omega_t^{(j)}). \end{aligned} \tag{5.22}$$

As described in Section 3.3, we progressively process frames in a video until the maximum identity confidence exceeds a given confidence threshold τ, i.e., $\max_k p(\omega_t^k|\mathbf{z}_{1:t}) \geq \tau$ for the recognition purpose.

5.7.1 Facial Pose Estimation

Similarly, facial pose estimation is performed using the MAP estimate. The discrete pose label $i = g(\psi)$ is estimated as

$$i^* = g(\psi_t)^* = \arg\max_{g(\psi_t)} p(g(\psi_t)|\mathbf{z}_{1:t}) \tag{5.23}$$

where the posterior probability $p(g(\psi_t)|\mathbf{z}_{1:t})$ is given by

$$p(g(\psi_t)|\mathbf{z}_{1:t}) \approx \sum_{j=1}^{N_s} w_t^{(j)} \delta(g(\psi_t) - g(\psi_t^{(j)})) \tag{5.24}$$

and $g(\cdot)$ denotes the discretizing function of the joint set of continuous pose parameter $\psi = (\phi^x, \phi^y)$.

5.7.2 Tracking State Estimation

The simultaneous framework does not aim to estimate a particular tracking state in a video sequence because face recognition, which is the main objective, is performed simultaneously unlike the tracking-*then*-recognition framework. Yet, a particular tracking state can also be estimated.

The tracking state estimation is exampled as the MAP estimate as

$$\theta_t^* = \arg\max_{\theta_t} p(\theta_t|\mathbf{z}_{1:t}). \tag{5.25}$$

The posterior pdf $p(\theta_t|\mathbf{z}_{1:t})$ is obtained by marginalizing the posterior probability $p(\theta_t, \psi_t, \omega_t|\mathbf{z}_{1:t})$ with respect to the identity ω_t and the appearance state ψ_t, where it results in,

$$p(\theta_t|\mathbf{z}_{1:t}) \approx \sum_{j=1}^{N_s} w_t^{(j)} \delta(\theta_t - \theta_t^{(j)}). \tag{5.26}$$

However, in practice the tracking particle states $\{\theta_t^{(j)}, j = 1, \ldots, N_s\}$ are varying each other because $\{\theta_t^{(j)}, j = 1, \ldots, N_s\}$ are finite number of samples drawn from a continuous random vector θ_t. Therefore, the posterior probability $p(\theta_t|\mathbf{z}_{1:t})$ results in and is further approximated as

$$p(\theta_t|\mathbf{z}_{1:t}) \approx w_t^{(j)} \delta(\theta_t - \theta_t^{(j)}), \quad j = 1, \ldots, N_s \tag{5.27}$$

in practice. Therefore, we propose to obtain the MAP estimate as

$$\theta_t^* = \theta_t^{(j^*)} \text{ where } j^* = \arg\max_j w_t^{(j)}. \tag{5.28}$$

5.8 Final Algorithm

Our algorithm is summarized in Figure 5.3.

Initialize a sample set $\{\mathbf{x}_0^{(j)}, 1\}_{j=1}^{N_{s(0)}}$ according to prior distribution $p(\mathbf{x}_0)$.
Initialize $n = 1$, $R_1 = R_0$, $N_{s(1)} = N_{s(0)}$, and $p(\omega_0^k|\mathbf{z}_0) = 1/c \ \ \forall k$
For $t = 1, 2, \cdots$, and $\max_k p(\omega_{n-1}^k|\mathbf{z}_{1:t-1}) < \tau$
    Set the state transition model to an adaptive velocity model.
    Initialize $\varepsilon_n = \infty$ and $iter = 0$
    While $\varepsilon_n > \eta$ and $iter++ < maxiter$
        Resample $\{\mathbf{x}_{n-1}^{(j)}, w_{n-1}^{(j)}\}_{j=1}^{N_{s(n-1)}}$ and obtain a new sample $\{\mathbf{x}_{n-1}^{'(j)}, 1\}_{j=1}^{N_{s(n)}}$
        Predict $\mathbf{x}_n^{(j)}$ by sampling from $\mathbf{x}_n^{(j)} = \mathbf{f}_n(\mathbf{x}_{n-1}^{'(j)}, \mathbf{v}_n) \ \forall j$
        Measure and update the weights from $w_n^{(j)} = p(\mathbf{z}_t|\mathbf{x}_n^{(j)}) \ \forall j$
        Normalize the weights using $w_n^{(j)} = w_n^{(j)} / \sum_{j=1}^{N_{s(n)}} w_n^{(j)} \ \ \forall j$
        Update ε_n by (5.13)
        Adapt the noise variance R_{n+1} by (5.15)
        Adapt the number of particles $N_{s(n+1)}$ by (5.16)
        Set the state transition model to a fixed constant-velocity model.
        $n++$
    End
    Obtain $p(\omega_{n-1}^k|\mathbf{z}_{1:t}) \ \forall k$ by marginalizing weights (5.22)
    Estimate pose (5.23) and tracking state (5.28) by MAP if needed.
End
Estimate identity (face recognition) by MAP (5.19)
where c indicates the number of identities, n indicates the index of iterations in total, t indicates the time index, *iter* indicates the index of iterations in one frame, *maxiter* indicates the maximum allowable number of iterations in one frame, τ is the confidence threshold, and η is the acceptable prediction error threshold.

Figure 5.3: Proposed simultaneous tracking and recognition algorithm

5.9 Experimental Results

In this section, experimental results are provided. We use the Honda/UCSD video dataset [43] for experiments as we have used in Section 3.3. The Honda/UCSD video dataset consists of a set of 45 videos of 20 different people. Each individual in the database has at least two videos as each person moves in a different combination of 2-D and 3-D rotation, expressions, and speed. Each video was recorded in an indoor environment and each one lasted for 20 seconds (with 30 color frames of 640 $\times$ 480 pixels per second).

To train appearance manifolds, we manually cropped facial image patches and classified into the front, left, right, up, down pose subsets. The examples are shown in Figure 5.4. Then, we downsampled each image to 20×20 pixels, to imitate the image quality in surveillance systems. The pixels in each image were normalized to have zero mean and unit variance.

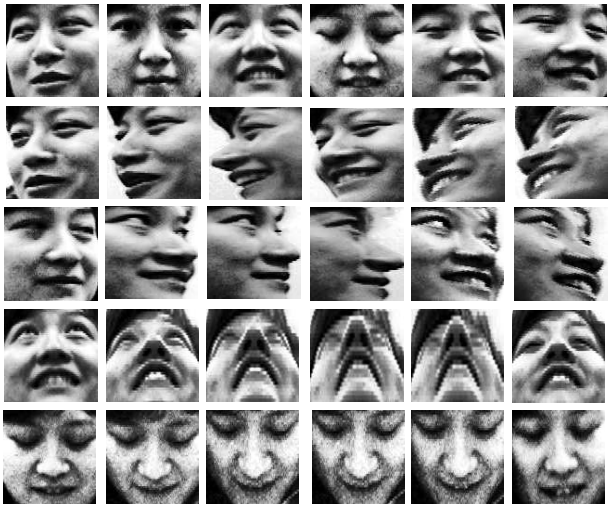


Figure 5.4: Examples of facial images in pose subsets. Images were resized to have a square size. The top row presents frontal faces, the 2nd top row presents left-profile faces, the middle row presents right-profile faces, the 2nd bottom row presents up-profile faces, and the bottom row presents the bottom-profile faces. The left/right-up and left/right-down profile faces are included in the left/right-profile subsets respectively.

5.9.1 Tracking Result

The tracking results in a test video having multi-view faces are presented. Figure 5.5 presents tracking results for a video of identity 1. In Figure 5.5, all particle configurations of tracking states are shown with white rectangles. Figure 5.6 presents tracking results for a video of identity 2, and Figure 5.7 presents tracking results for the same video without the one frame iteration process. In Figure 5.6 and Figure 5.7, only the MAP estimate of a tracking state is shown with a white rectangle. The comparison between Figure 5.6 and Figure 5.7 shows the effectiveness of the one frame iteration process. In contrast to the tracking state ϕ^z, we assigned large standard deviations to $\psi = (\phi^x, \phi^y)$ such as 20° to allow possible transitions between pose manifolds.

(a) frame 3 / front

(b) frame 23 / right

(c) frame 53 / left

(d) frame 93 / up

(e) frame 104 / down

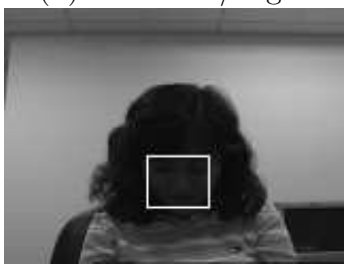
Figure 5.5: Tracking results showing all particles' tracking states.

(a) frame 4 / front

(b) frame 39 / right

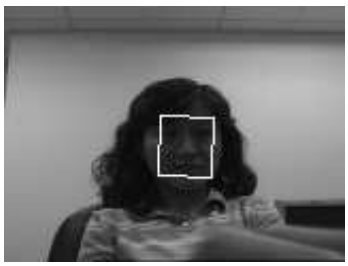
(c) frame 68 / left

(d) frame 94 / up



(e) frame 122 / down

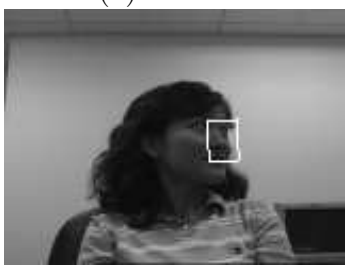
Figure 5.6: Tracking results showing the maximum *a posteriori* estimate of the tracking states.



(a) frame 4

(b) frame 32

(c) frame 33



(d) frame 34

(e) frame 35

(e) frame 39

Figure 5.7: Tracking results for the same video with Figure 5.6 without iterations in one frame. The tracker lost a target face in the 35th frame (we can see symptoms of failure from the 33rd frame), and could not recover. Iterations in one frame are especially helpful for tracking such intermediate poses between two modeled poses, i.e., front and right profile poses in this example.

5.9.2 Tracking Performance Measure

We measure the performance of the tracking algorithm by comparing with the *ground truth*. Let us denote the rectangle tracking state $\theta = (x, y, s^x, s^y, \psi^z)$ as $\theta = (\theta^1, \theta^2, \theta^3, \theta^4, \theta^5)$. We define a dissimilarity measure of two rectangle regions as

$$d(\hat{\theta}, \theta) = \sum_{i=1}^{5} \alpha^i (\hat{\theta}^i - \theta^i)^2. \tag{5.29}$$

where $\hat{\theta}$ denotes the MAP estimate of the tracking state, θ denotes the ground truth, α^i denotes weights especially used to ensure a balance between the importance of the angle parameter and the position parameter. Note that the angle difference must be in $[0, 180)$ because the angle parameter is a wrapped parameter. Figure 5.9 shows a plot of tracking error versus time t for a test video of identity 2 with iterations in one frame. We accomplished a correct identity recognition in this video. We furthermore perfomed discrete pose estimation frame-by-frame in this video, and achieved 94.32% correct rate. Figure 5.8 shows a comparison between simultaneous trackers with and without iterations in one frame in terms of the tracking error. We used $\alpha^i = 1 \; \forall i$ in this experiment. From here, we always use iterations in one frame.

5.9.3 Identity Confidence Convergence

Figure 5.10(a) presents a plot of the posterior probability $p(\omega_t|\mathbf{z}_{1:t})$ versus time t obtained by the proposed simultaneous tracking and recognition algorithm for a test video. Figure 5.10(b) presents the same plot obtained by the tracking-*then*-recognition scenario for the same video. The tracking results, i.e., the facial image patches, were obtained manually, which is considered as the best object tracking

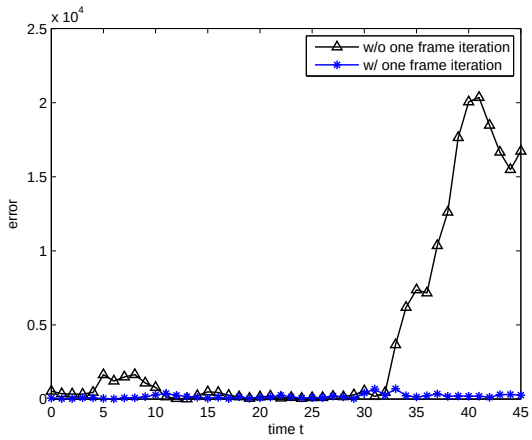


Figure 5.8: The tracking error $d(\hat{\theta}_t, \theta_t)$ versus time t. Tracking with iterations in one frame version has less error than the tracking without iterations in one frame version. In fact, the tracking without iterations failed to track a face from the 33rd frame onwards as shown in 5.7.

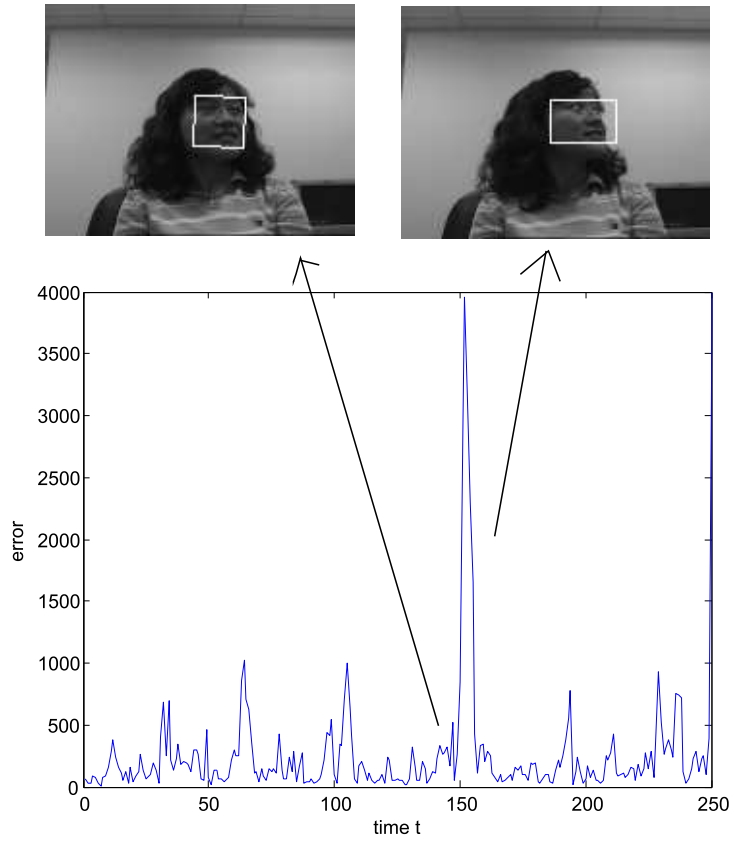
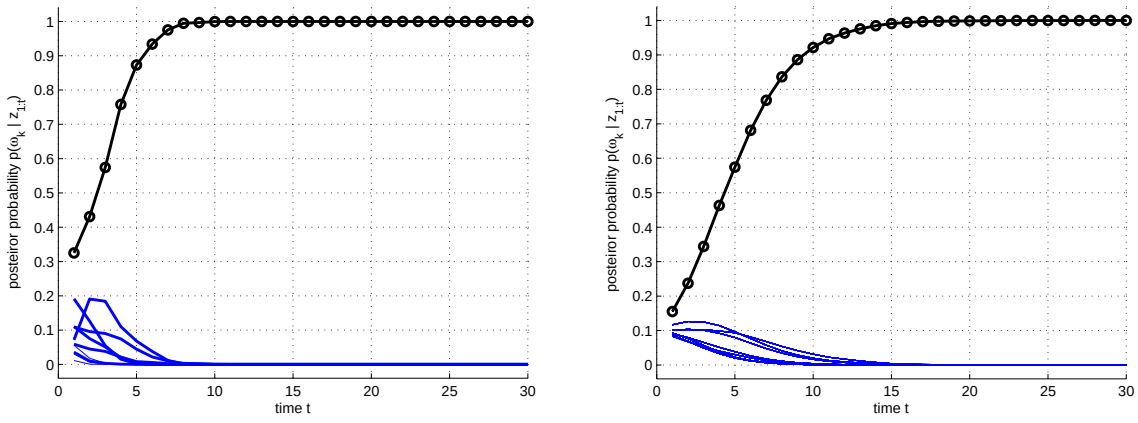


Figure 5.9: The tracking error $d(\hat{\theta}_t, \theta_t)$ versus time t. We accomplished a correct identity recognition, and a correct discrete pose estimation rate of 94.32% perfomed on frame-by-frame in this video.

method. The video-based face recognition algorithm proposed in Section 3.3 was used for face recognition. As shown in the two plots, the simultaneous scenario achieved faster convergence . This is because of the marginalization of posterior probability $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ with respect to observations of many image patches in a video frame.



(a) (b)

Figure 5.10: Posterior probability $p(\omega_t|I_{0:t})$ versus time t. (a) Result of the simultaneous tracking and recognition. (b) Result of the tracking-*then*-recognition. The probability confidence of the correct identity exceeded 0.999 at time $t = 8$ in the simultaneous algorithm although it took $t = 15$ in the tracking-*then*-recognition scenario.

5.9.4 Face Recognition

Finally, we performed a face recognition experiment for 20 test videos. We initialized the simultaneous face tracking and recognition processes from frame num-

bers, 0, 150, and 300 to increase the number of experiments. We achieved 78 experiments using the original 20 test videos with $300 \sim 500$ frames. We downsampled each video frame to 160×120 where the original size is 640×480 so that each facial image patch has $20 \times 20 \sim 30 \times 30$ resolution in the video, which is considered as a low resolution. There are 20 different subjects in the video dataset. We resized each cropped facial image patch into 20×20 pixels and normalized the pixels in each image to have zero mean and unit variance. Then, we projected the image feature vector onto the PCA subspaces and reduced the number of dimension to 20. We achieved 100% recognition rate by processing 8 frames of a video on the average.

Chapter 5

Conclusions and Future Research Directions

5.1 Conclusions

In this work, we presented a simultaneous multi-view face tracking and recognition algorithm using particle filtering. To utilize temporal information in a video sequence for not only tracking but also recognition, a video-based face recognition algorithm based on Bayesian inference was also proposed.

The proposed video-based face recognition method interprets temporal information in a video as transition probabilities between facial poses. Compared to the previous work [43], this Bayesian inference algorithm realized a face recognition algorithm using the full video rather than in a frame-by-frame basis by progressively accumulating the face recognition confidences in frames. Thanks to the accumulation characteristic, the algorithm achieved a face recognition rate 100% even in low resolution videos where each facial image has only 20×20 resolution. Furthermore, this face recognition framework has another useful characteristic in that it allows to stop processing the frames in a video progressively at an itermediate frame if enough recognition confidence is accumulated yet. This characteristic gives an advantage over batch methods in terms of computational efficiency.

Then, the video-based face recognition algorithm based on Bayesian inference was tightly coupled with a face tracking algorithm which is also based on the

Bayesian inference framework, and the face recognition problem was solved as a simultaneous face tracking and recognition problem. To solve the Bayesian inference problem, the particle filtering method was used. Unlike [83], this simultaneous framework utilized the temporal information in a video for not only tracking but also recognition by modeling the dynamics of facial poses.

5.2 Future Works

Here, we briefly list some potential ideas to be explored in the context of face tracking and recognition.

- **Multi-View Face Detector**. Although we used the adaptive noise variance model to enlarge a searching range of a tracker, it was still difficult for the tracker to continue tracking in successive frames when it has failed to track objects once. In such case, re-initialization, i.e., detection is helpful. To incorporate a detector in our multi-view supported tracker, we have to study multi-view face detection methods, and investigate when the re-initialization process should be triggered. We might propose to use the measure of prediction error ε_t as one criterion to trigger the re-initialization process. We might propose a face detection method which performs an exhaustive search with the appearance models used in this thesis, however, it is computationally too expensive and slow. We might use a fast multi-view face detection algorithm proposed by Jones and Viola [36] which first estimates facial poses and then rapidly detects faces in an image.

- **Rapid Simultaneous Face Tracking and Recognition**. To exploit the evidence accrual behavior of the proposed algorithm, which is especially useful in real-time processing, rapid real-time feature extraction methods must be investigated. Viola and Jones [76] introduced a rapid face detection algorithm using Haar-like features. Using such features jointly with particle filtering, we can possibly develop a fast simultaneous face tracking and recognition algorithm. To do this, we have to study the following three central issues:

  1. **Probabilistic model of Haar-like features.** To use the Haar-like features in particle filtering, we have to provide a probabilistic model of Haar-like features.

  2. **Face recognition using Haar-like features.** To construct a rapid simultaneous tracking and recognition framework, face recognition using Haar-like features must be studied.

  3. **Feature selection method.** Viola and Jones [76] originally used the Adaboost algorithm to select discriminative features from the Haar-like features for two-class problems, i.e., classification between face and non-face. To select the Haar-like features for multi-class problem, we must provide another feature selection method. Furthermore, the training method developed by them requires considerable time, e.g., two weeks to obtain a valid face detector. Developing a fast training framework must be an interesting issue. For example, Wu [79] recently proposed an asymmetric fast training for the Haar-like features using the Forward

Feature Selection (FFS) method.

- **Handling Illumination Variations** Illumination variation has enormously complex effects on the image of an object. The changes induced by illumination variations are often larger than the differences between individuals. To achieve a robust face recognition algorithm, illumination variations must be handled. Soma et al. [5] proposed a method to estimate albedo, i.e., the fraction of light that a surface point reflects when it is illuminated. Unlike image intensity, albedo is invariant to changes in illumination conditions which makes it useful for illumination-invariant matching of objects.

Bibliography

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[2] A. Azarbayejani and A. Pentland. Recursive estimation of motion, stmcture, and focal length. *IEEE Trans. Pattern Anal. Machine Intell.*, 17:562–575, June 1995.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs Fisherfaces: Recognition using class specific linear projection. *European Conf. Computer Vision*, pages 45–58, 1996.

[4] A. Bergen, P. Anadan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. Eur. Conf. Computer Vision*, pages 237–252, 1992.

[5] S. Biswas, G. Aggarwal, and R. Chellappa. Robust estimation of albedo for illumination-invariant matching and shape recovery. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[6] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion discontinuities. In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 551–558, 1999.

[7] M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 176–181, 1999.

[8] D. Blackburn, M. Bone, and P. J. Phillips. Facial recognition vendor test 2002 evaluation report, Feb. 2001.

[9] A. Blake and M. Isard. The CONDENSATION algorithm - conditional density propagation and applications to visual tracking. In *Advances in Neural Information Processing Systems*, pages 36–1. The MIT Press, 1996.

[10] M. E. Brand. Incremental singular value decomposition of uncertain data with missing values. *European Conference on Computer Vision (ECCV)*, 2350:707–720, May 2002.

[11] T. J. Broida, S. Chandra, and R. Chellappa. Recursive techniques for estimation of 3-d translation and rotation parameters from noisy image sequences. *IEEE Trans. Aerosp. Electron. Syst.*, 26:639–656, Apr. 1990.

[12] R. Brunelli and T. Poggio. Face recognition: Features vs templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10), Oct. 1993.

[13] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. *In IEEE Proceedings on Radar and Sonar Navigation*, 1999.

[14] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, 2002.

[15] S. Chandrasekaran, B. S.Manjunath, Y. F.Wang, J.Winkeler, and H. Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59(5):321–332, Sept. 1997.

[16] R. Chellappa, C. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proc. IEEE*, 83(5):705–740, 1995.

[17] T. F. Cootes, G. J. Edwards, and C.J.Taylor. Active appearance models. In *Proc. European Conference on Computer Vision 1998 (H.Burkhardt and B. Neumann Ed.s)*, volume 2, pages 484–498. Springer, 1998.

[18] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746, Mar 2002.

[19] D. Crisan, P. D. Moral, and T. J. Lyons. Non-linear filtering uing branching and interacting particle systems. *Markov Processes and Related Fields*, 5, 1999.

[20] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[21] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[22] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2001.

[23] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Improving identification performance by integrating evidence from sequence. *In Proc. IEEE Conf. on Computer Vision and Pattem Recognition*, pages 486–491, 1999.

[24] K. Etemed and R. Challeppa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America*, pages 1724–1733, 1997.

[25] R. A. Fisher. The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7, 1936.

[26] D. Fox. Kld-sampling: adaptive particle filters and mobile robot localization. *Neur. Inform. Process. Syst.*, 2001.

[27] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr 1993.

[28] G. Guo, S. Z. Li, and K. Chan. Face recognition by support vector machines. In *Proc. of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, pages 196–201, 2000.

[29] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Anal. Machine Intell.*, 20:1025–1039, Oct. 1998.

[30] P. Hall, D. Marshall, and R. Martin. Incremental eigenalysis for classification. In *British Machine Vision Conference*, volume 1, pages 286–295, Sept. 1998.

[31] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1048, 2000.

[32] D. Hochbaum and D. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10:180–184, 1985.

[33] M. Isard and A. Blake. Contour tracking by stochatic propagation of conditional density. In *Proc. Eur. Conf. Computer Vision*, pages 343–356, 1996.

[34] M. Isard and A. Blake. Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, Aug. 1998.

[35] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. *Lecture Notes in Computer Science*, 1406:893–908, 1998.

[36] M. Jones and P. Viola. Fast multi-view face detection. Technical Report TR2003-096, MERL, 2003.

[37] M. J. Jones and T. Poggio. Model-based matching by linear combination of prototypes. Technical Report AI Memo No. 1583, Artificial Intelligence Laboratory, Massachusetts Inst. of Technology, Nov. 1998.

[38] F. Jurie and M. Dhome. A simple and efficient template matching algorithm. In *Proc. Int'l Conf. Computer Vision*, volume 2, pages 544–549, 2001.

[39] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[40] K. Kanazawa, D. Koller, and S. J. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in AI (UAI '95)*, pages 346–351, 1995.

[41] J. T. Kwok and H. Zhao. Incremental eigen decomposition. In *Proc. ICANN*, pages 270–273, 2003.

[42] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Pattern Analysis and Machine Intelligence*, 19(7):743–756, July 1997.

[43] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:313, 2003.

[44] Y. Li, S. Gong, and H. Liddell. Constructing facial identity surface in a nonlinear discriminating space. *In Proc. IEEE Conf. on Computer Vision and Pattem Recognition*, 2:258–263, 2001.

[45] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamical systems. *Journal of the American Statistical ssociation*, 93:1032–1044, 1998.

[46] X. Liu and T. Chen. Video-based face recognition using adaptive hidden markov models. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 340–345, 2003.

[47] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int'l Joint Conf. Artificial Intelligence*, 1981.

[48] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. Int'l. Conf. Computer Vision*, pages 572–578, 1999.

[49] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(6):780–788, Jun 2002.

[50] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian modeling of facial similarity. *Advances in Neural Information Processing Systems 11*, pages 910–916, 1998.

[51] B. Moghaddam, T. Jebara, and A. Pentland. Efficient MAP/ML similarity matching for face recognition. In *Proc. Int'l Conf. Pattern Recognition*, Aug. 1998.

[52] B. Moghaddam, C. Nastar, and A. Pentland. A Bayesian similarity measure for direct image matching. In *Matching, M.I.T Media Laboratory Perceptual Computing Section*, pages 350–358. IEEE Computer Society Press, 1996.

[53] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. *lnt'l Conf. Computer Vision*, pages 786–793, 1995.

[54] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *IEEE Trans. Pattem Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[55] P. D. Moral. Non-linear filtering: Interacting particle solution. *Markov Processes and Related Fields*, 2:555–580, 1996.

[56] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int'l. J. Computer Vision*, 14:5–24, 1995.

[57] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Trans. Pattern Anal. Machine Intell.*, 22:1016–1034, Oct. 2000.

[58] U. Park and A.K.Jain. 3D model-based face recognition in video. In *Proc. of 2nd Intl. Conf. on Biometrics (ICB)*, pages 1085 – 1094, Seoul, South Korea, August 2007.

[59] P. Penev and J. Atick. Local feature analysis: A general statistical theory for object representation. *Neural Systems*, 6:477–500, 1996.

[60] A. Pentland, B. Moghaddam, and Starner. View-based and modular eigenspaces for face recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 84–91, 1994.

[61] P. Phillips, T. S. P.J. Flynn, K. Bowyer, and W. Worek. Preliminary face recognition grand challenge results. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, FGR06*, pages 15–24, Apr. 2006.

[62] P. J. Phillips, P. Grother, R. J. Micheals, D. M. Blackburn, and E. Tabassi. Facial recognition vendor test 2002: Evaluation report, Mar. 2003.

[63] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, Oct. 2000.

[64] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 large-scale results. Technical Report NISTIR 7408, National Institute of Standards and Technology, Mar. 2007.

[65] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, 1992.

[66] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. *2nd IEEE Workshop on Applications of Computer Vision*, Dec. 1994.

[67] N. Seo. imageclipper: A tool to clip images manually fast. `http://code.google.com/p/imageclipper/`, 2008.

[68] G. Shakhnarovich, J. W. Fisher, and T. Darrell. Face recognition from long-term observations. *In Proc. European Conf. on Computer Vision*, 3:851–865, 2002.

[69] S. Shan, W. Gao, and D. Zhao. Face identification from a single example based on face-specific subspace (FSS). *IEEE 2002 International Conference on Acoustic, Speech and Signal Processing*, May 2002.

[70] S. Shan, W. Gao, and D. Zhao. Face recognition based on face-specific subspace. *International Journal of Imaging Systems and Technology*, 13(1):23–32, 2003.

[71] L. Sirovitch and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Optical Soc of Am. A*, 2, 1987.

[72] A. Soto. Self adaptive particle filter. In *Proceedings of International Joint Conference on Artificial Intelligence*, 2005.

[73] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[74] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1), 1991.

[75] M. Turk and A. Pentland. Face recognition using eigenfaces. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[76] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[77] A. G. Weber. The USC-SIPI image database. `http://sipi.usc.edu/services/database/Database.html`, Oct. 1997.

[78] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Pattern Analysis and Machine Intelligence*, 19(7):775–779, jul 1997.

[79] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg. Fast asymmetric learning for cascade face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):369–382, Mar. 2008.

[80] Y. Wu and T. S. Huang. A co-inference approach to robust visual tracking. In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 26–33, 2001.

[81] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfield. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):400–459, 2003.

[82] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 11:1434–1456, Nov 2004.

[83] S. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding (CVIU) (special issue on Face Recognition)*, 91:214–245, 2003.