ABSTRACT

Title of Dissertation: UNCONSTRAINED FACE RECOGNITION

Shaohua Zhou, Doctor of Philosophy, 2004

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

Although face recognition has been actively studied over the past decade, the state-of-the-art recognition systems yield satisfactory performance only under controlled scenarios and recognition accuracy degrades significantly when confronted with unconstrained situations due to variations such as illumintion, pose, etc. In this dissertation, we propose novel approaches that are able to recognize human faces under unconstrained situations.

Part I presents algorithms for face recognition under illumination/pose variations. For face recognition across illuminations, we present a generalized photometric stereo approach by modeling all face appearances belonging to all humans under all lighting conditions. Using a linear generalization, we achieve a factorization of the observation matrix consisting of face appearances of different individuals, each under a different illumination. We resolve ambiguities in factorization using surface integrability and symmetry constraints. In addition, an illumination-invariant identity descriptor is provided to perform face recognition across illuminations. We further extend the generalized photometric stereo approach to an illuminating light field approach, which is able to recognize faces under pose and illumination variations.

Face appearance lies in a high-dimensional nonlinear manifold. In Part II, we introduce machine learning approaches based on reproducing kernel Hilbert space (RKHS) to capture higher-order statistical characteristics of the nonlinear appearance manifold. In particular, we analyze principal components of the RKHS in a probabilistic manner and compute distances such as the Chernoff distance, the Kullback-Leibler divergence between two Gaussian densities in RKHS.

Part III is on face tracking and recognition from video. We first present an enhanced tracking algorithm that models online appearance changes in a video sequence using a mixture model and produces good tracking results in various challenging scenarios. For video-based face recognition, while conventional approaches treat tracking and recognition separately, we present a simultaneous tracking-and-recognition approach. This simultaneous approach solved using the sequential importance sampling algorithm improves accuracy in both tracking and recognition. Finally, we propose a unifying framework called probabilistic identity characterization able to perform face recognition under registration/illumination/pose variation and from a still image, a group of still images, or a video sequence.

UNCONSTRAINED FACE RECOGNITION

by

Shaohua Zhou

Dissertation 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
Doctor of Philosophy
2004

Advisory Committee:

Professor Rama Chellappa, Chairman
Professor Larry S. Davis
Professor David W. Jacobs
Professor Adrian Papamarcou
Professor Min Wu

DEDICATION

To Chunhui

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisor, Professor Rama Chellappa, for his sustained financial support, his valuable guidance on research, and his scholarly and honest attitude toward life.

I am grateful to my committee members, Professors Larry S. Davis, David W. Jacobs, Adrian Papamarcou, and Min Wu. I enjoyed my fruitful discussions with Professor David W. Jacobs. I also thank Professor Eric V. Slud in the Mathematics department for educating me and sharing with me his broad knowledge on statistics and Dr. Baback Moghaddam at Mitsubishi Electric Research Labs (MERL) for hosting me as a summer intern in 2002. I also would like to express my appreciation of Professor Azriel Rosenfeld, who was in my proposal examination committee and edited two of my technical reports.

I had a pleasant stay at the Center for Automation Research (CfAR). I am indebted to my lab colleagues: Amit R. Chowdhury, Naresh Contoor, Jian Li, Jian Liang, Haiying Liu, Amit Kale, Gang Qian, Jie Shao, Namrata Vaswani, Zhanfen Yue, and Qinfen Zheng. I really enjoyed my collaborations and discussions with these brilliant guys.

I take this special occasion to thank my parents and parents-in-law back in China for their support and to wish them best. Finally, I thank my wife, Chunhui, for her patience, her encouragement, and her lifelong love. I dedicate my thesis to her.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Overview

Identifying people from faces is an effortless task for humans. Is it the same for computers? This defines the very question for the field of automatic face recognition [20, 21, 22, 23, 24, 25, 26, 27, 191] (also referred to as face recognition in the present dissertation), one of the most active research areas in computer vision, pattern recognition, and image understanding.

Over the past decade, face recognition has attracted substantial attention from various disciplines and contributed to a skyrocketing growth in the literature. Below, we mainly emphasize the biometric, experimental, and theoretic perspectives of face recognition.

1.1.1 Biometric perspective

Face is a biometric [31]. As a consequence, face recognition finds wide applications related to authentication, security, and so on. One striking example is recent deployment of the US-VISIT system [30] by the Department of Homeland Security

(DHS), collecting foreign passengers' fingerprints and face images.

Biometrics enable automatic identification of a person based on physiological or behavioral characteristics [29, 28]. Physiological biometrics are biological/chemical traits that are innate or naturally grown, while behavioral biometrics are mannerisms or traits that are learned or acquired. Table 1.1 lists commonly used biometrics. Some introductory discussions on biometrics may be found in [28, 29, 31, 32].

Type	Examples
Physiological biometrics	Body odor, DNA, face, fingerprint, hand geometry, iris, pulse, retinal
Behavioral biometrics	Face, gait, handwriting, signature, voice

Table 1.1: A list of biometrics.

Biometrics technologies are becoming the foundations of an extensive array of highly secure identification and personal verification solutions. Compared with conventional identification and verification methods based on personal identification numbers (PINs) or passwords, biometrics technologies offer some unique advantages. First, biometrics are individualized traits while passwords may be used or stolen by someone other than the authorized user. Also, a biometric is very convenient since there is nothing to carry or remember. In addition, biometric technology is becoming more accurate and inexpensive.

Among all biometrics listed in Table 1.1, face biometric is a very unique one because face is the only biometric belonging to both physiological and behavioral categories. While the physiological part of the face biometric is widely researched in the literature, the behavioral part is not yet fully investigated. In addition, as reported in [33, 34], face has advantage over other biometrics because it is a natural, non-intrusive, and easy-to-use biometric. For example [33], among the

six biometrics of face, finger, hand, voice, eye, and signature in Figure 1.1, face biometric ranks the first in the compatibility evaluation of a machine readable travel document (MRTD) system in terms of six criteria: enrollment, renewal, machine-assisted identity verification requirements, redundancy, public perception, and storage requirements and performance. Probably the most important feature of a biometric is its ability to collect the signature from non-cooperating subjects.

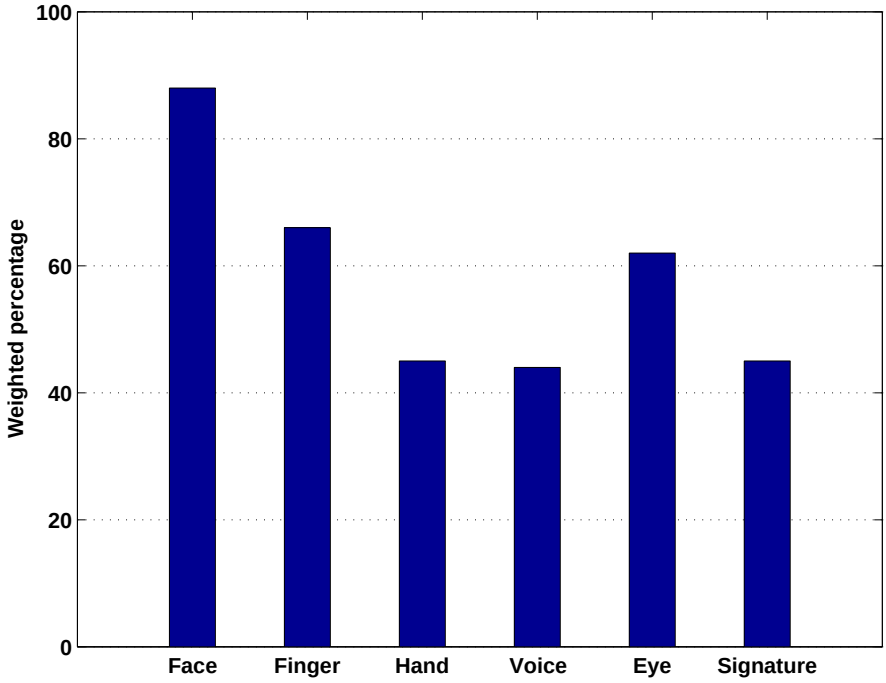


Figure 1.1: Comparison of various biometric features based on MRTD compatibility (from [33]).

Besides applications related to identification and verification such as access control, law enforcement, ID and licensing, surveillance, etc., face recognition is also useful in human-computer interaction, virtual reality, database retrieval, multimedia, computer entertainment, etc. See [27, 45] for a review of face recognition applications.

1.1.2 Experimental perspective

Face recognition mainly involves the following three tasks [59]:

- Verification. The recognition system determines if the query face image and the claimed identity match.
- Identification. The recognition system determines the identity of the query face image by matching it with a database of images with known identities, assuming that the identity is inside the database.
- Watch list. The recognition system first determines if the identity of the query face image is on the stored watch list and, if yes, then identifies the individual.

Figure 1.2 illustrates the above three tasks and corresponding statistics used for evaluation. Among three tasks, the watch list task is the most difficult one.

The present thesis focuses only on the identification task. We introduce a face recognition test protocol FERET [58] widely observed in the face recognition literature. FERET stands for 'facial recognition technology'. In most experiments conducted in the thesis, we follow the FERET protocol.

FERET assumes availability of the following three sets, namely one training set, one gallery set, and one probe set. The training set is provided for the recognition algorithm to learn the characteristic features. The gallery and probe sets are used in the testing stage. The gallery set contains images with known identities and the probe set with unknown identities. The algorithm associates descriptive features with images in the gallery and probe sets and determines the identities of the probe images by comparing their associated features with those features associated with gallery images.

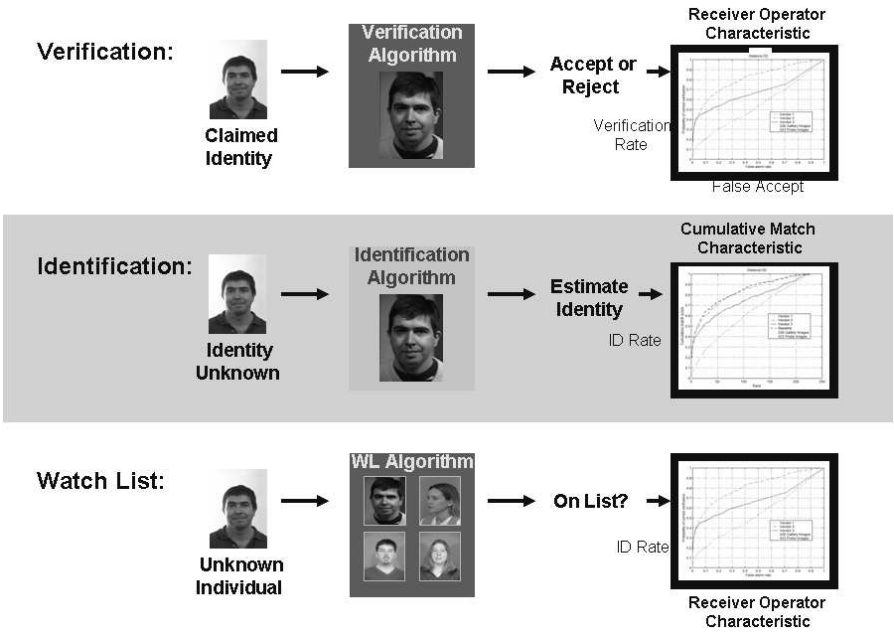


Figure 1.2: Three face recognition tasks: verification, identification, watch list (courtesy of P.J.Phillips [59]).

1.1.3 Theoretic perspective

Face recognition is by nature an interdisciplinary research area, tied to an array of research fields, ranging from pattern recognition, computer vision and graphics, and image processing/understanding to statistical computing and machine learning. In addition, automatic face recognition designs are often guided by the psychophysical and neural studies. A good summary of research on face perception is presented in [27, 35, 38]. We now focus on the theoretical implications of pattern recognition for the special task of face recognition.

We present a three-level structure for understanding the face recognition problem. The three levels forming the pyramid are: pattern, visual pattern, and face pattern, each associated with a corresponding theory of recognition. Accordingly, face recognition approaches can be grouped into three categories.

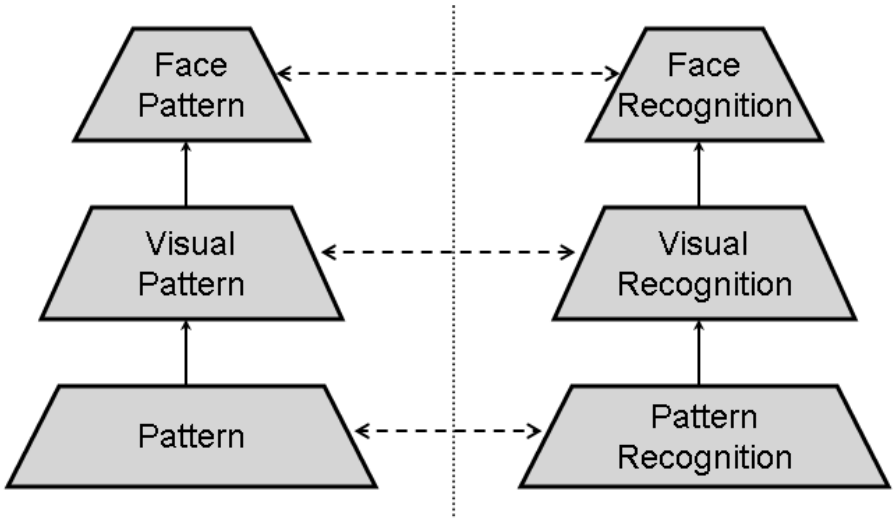


Figure 1.3: A hierarchy of face pattern and face recognition.

Pattern and recognition

On the base of the pyramid lies a general pattern. Because face is first a pattern, any pattern recognition theory [7] can be directly applied to a face recognition problem. In general, a vector representation is used in pattern recognition. A common way of deriving a vector representation from a 2D face image, say of size $M \times N$, is through a 'vectorization' operator that stacks the pixels in a particular order, say a raster-scanning order, to an $MN \times 1$ vector. Obviously, given an arbitrary $MN \times 1$ vector, it can be decoded into an $M \times N$ image by reversing the above 'vectorization' operator. Such a vector representation corresponds to a holistic-based viewpoint in the psychophysics literature [36, 37].

Subspace methods are pattern recognition techniques widely invoked in various face recognition approaches. Two well-known appearance-based recognition schemes utilize principal component analysis (PCA) [12] and linear discriminant analysis (LDA) [7]. PCA performs an eigen-decomposition of the covariance matrix and consequently minimizes the reconstruction error in the mean square sense.

LDA minimizes the within-class scatter while maximizing the between-class scatter. The PCA approach used in face recognition is called the 'Eigenface' approach [62]. Another work using PCA earlier than 'Eigenface' is [47]. The LDA approach used in face recognition is called the 'Fisherface' approach [41] since LDA is also commonly referred to as Fisher discriminant analysis. LDA for face recognition was also independently proposed in [44]. Further PCA and LDA are combined (LDA after PCA) as in [64] to yield a better recognition scheme. Other subspace methods such as independent component analysis (ICA) [20, 40, 155], local feature analysis (LFA) [164], probabilistic subspace [54, 55, 56], multi-exemplar discriminant analysis [211] have been used in face recognition. A comparison of these subspace methods is reported in [56, 200]. Other than the subspace methods, classical pattern recognition tools such as neural networks [51], learning methods [57], and evolutionary pursuit/genetic algorithms [52] have also been applied to face recognition.

One concern in a general pattern recognition problem is the 'curse of dimensionality' since usually M and N themselves are quite large. In face recognition, because of limitations of image acquisition, practical face recognition systems store only a small number of samples per subject. This further worsens the 'curse of dimensionality' problem.

Face recognition also differs from general pattern recognition problem in various aspects. Some of the differences are illustrated below.

Visual pattern and visual recognition

In the middle of the pyramid in Figure 1.3 sits the visual pattern layer. A face is a visual pattern in the sense that it is a 2D appearance of a 3D object captured by

an imaging system. Certainly, visual appearance is affected by the configuration of an imaging system. An illustration of the imaging system is presented in Figure 1.4.

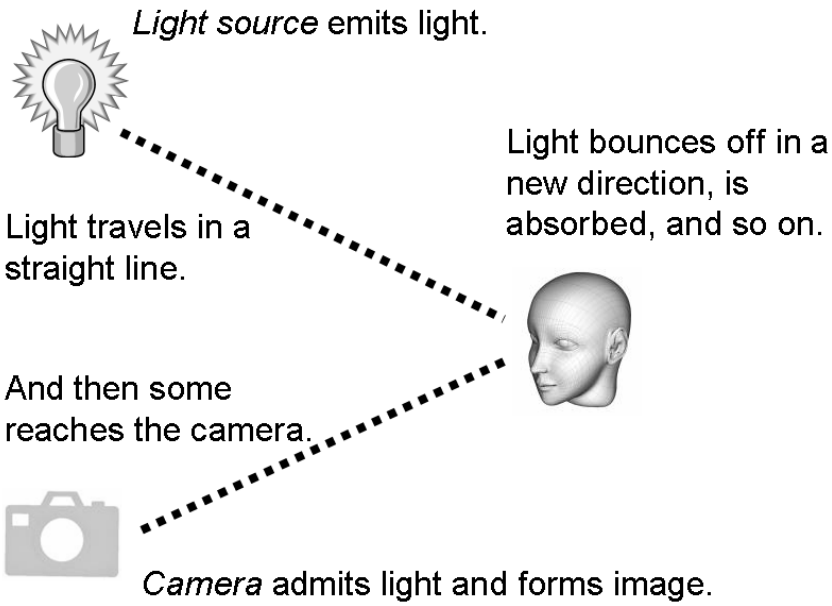


Figure 1.4: An illustration of the imaging system.

There are two distinct characteristics of the imaging system: photometric and geometric.

- Photometric characteristics are related to the light sources distributed in the scene. Figure 1.5 shows the face images of one object captured under varying illumination conditions. Numerous models have been proposed to describe the illuminating phenomenon, i.e., how the light travels when it hits the object. In addition to its relationship with the light distribution such as the light direction and intensity, an illumination model is in general also related to the object surface material properties.

- Geometric characteristic is about the camera properties and the relative po-

sitioning of the camera and the object. Camera properties include camera intrinsic paramters and camera imaging models. The imaging models widely studied in the computer vision literature are orthographic, scaled orthographic, and perspective models. Because the perspective model is difficult to deal with as it requires the depth information, the orthographic or scaled orthograhic model is more used in the face recognition community. The relative positioning of the camera and the object results in pose variation, a key factor determining how the 2D appearances are produced. Figure 1.5 shows the face images of one object captured at different poses.

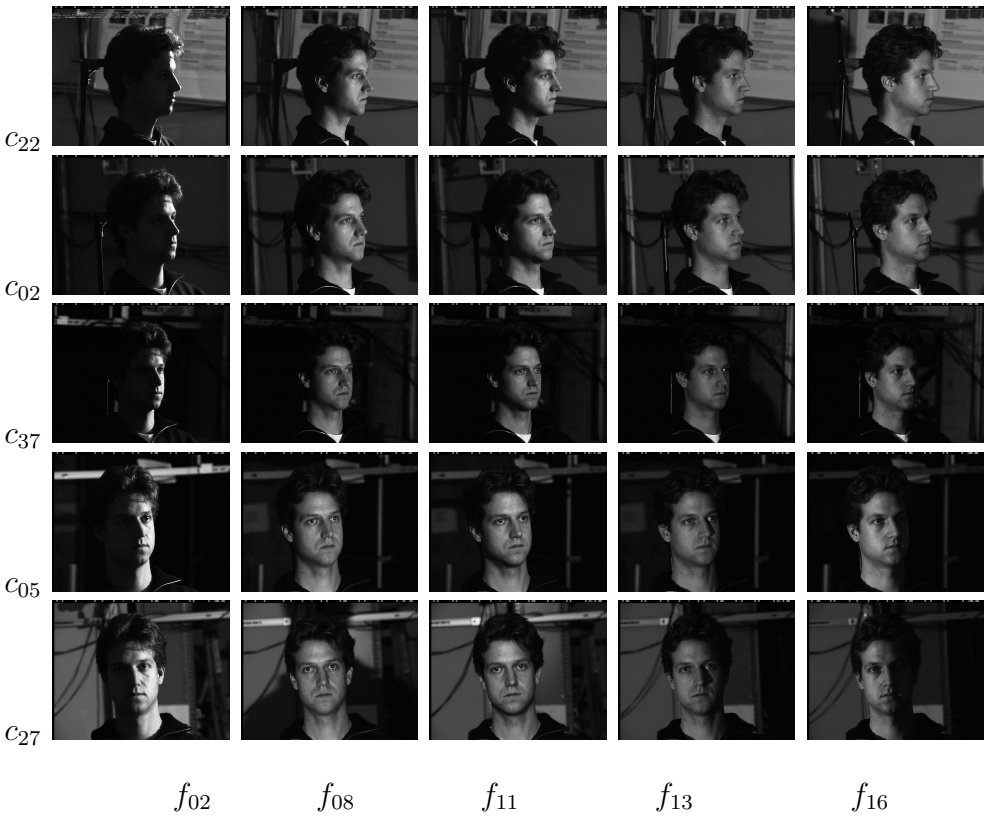


Figure 1.5: One PIE [75] individual under different illumination and poses.

Studying photometric and geometric characteristics is the key problem in the

computer vision literature and consequently visual recognition under illumination and pose variations is the main challenge in the recognition community. A full review of the visual recognition literature is beyond the scope of the thesis. However, face recognition methods that address the photometric and geometric characteristics are still in a nascent stage and needs to be fully explored.

Approaches to face recognition under illumination variation are usually treated as extensions of research efforts on illumination models. For example, if a simplified Lambertian reflectance model ignoring shadow pixels [96, 101, 103] is used, a rank-3 subspace can be constructed to cover the appearances arbitrarily illuminated by a distant point source. Similarly low-dimensional subspaces [94, 95] can be found using a Lambertian model with attached shadows. Face recognition can be performed by checking if a query face image lies in the object-specific illumination subspace. To generalize from the object-specific illumination subspace to a class-specific illumination subspace, bilinear models are used in [74, 138, 204]. Most face recognition approaches across pose variation use view-based appearance representation [67, 69, 72]. Face recognition across illumination and poses is more difficult compared with recognition across one single modality. Proposed approaches in the literature include [66, 70, 208], among which the 3D morphable model [66] yields the best recognition performance. The feature-based approach [48] is reported to be partially robust to illumination and pose variations.

An important feature of a visual pattern is its presence in video. The ubiquitousness of video sequences calls upon recognition algorithms based on videos. Because a video sequence is a collection of still images, face recognition from still images certainly applies. However, an important property of a video sequence is its temporal dimension. Recent psychophysical and neural studies [37, 39] demon-

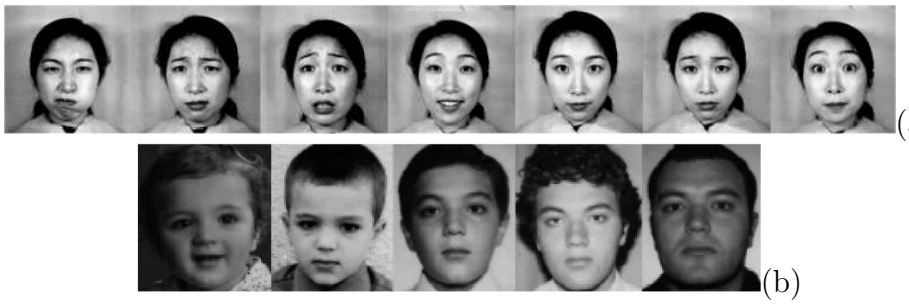
strate the role of movement in face recognition: Famous faces are easier to recognize when presented in moving sequences than in still photographs, even under a range of different types of degradations. Computational approaches utilizing such temporal information include [86, 193, 194, 185, 186, 190]. Figure 1.7 shows the tracked face appearance in a video sequence captures in an office environment [84]. Clearly, due to free movement of the human face and an uncontrolled environment, issues like illumination and pose variations still exist. Besides these issues, localizing faces or face segmentation in a cluttered environment in video sequences is very challenging.

In surveillance scenarios, further challenges include poor video quality and lower resolution. For example, the face region can be as small as 15×15, while most feature-based approaches [48, 66] need big face images of size as large as 128×128. However, video provides multiple observations linked by their temporal continuity.

Face pattern and face recognition

At the top of the pyramid lies the face pattern. The face pattern specializes the visual pattern by letting the object be a human face. Therefore, face-specific properties or characteristics should be taken into account when performing face recognition.

- *Deformation.* Humans express emotions through facial expressions, yielding patterns under nonrigid deformations. The non-rigidity is of very high degree of freedom and perplexes the recognition task. Figure 1.6(a) shows the face images of a person exhibiting different expressions. While face expression analysis attracts a lot of attention [42, 60, 61], recognition under facial expression variation has not been fully explored.


(a)

(b)

Figure 1.6: (a) Appearances of one individual with different facial expression (from [53]). (b) Appearances of one individual at different ages (from [50]).

- *Aging.* Face appearances vary significantly with aging and such variations are specific to an individual. As a result, theoretical modeling of aging [50] is very difficult due to the individualized variation. Figure 1.6(b) shows the face images of a person at different ages.

- *Face surface.* One speciality of face surface is its bilateral symmetry. Symmetry constraint has been widely exploited in [102, 104, 204]. In addition, surface integrability is an inherent property of any surface, which has also been used in [99, 103, 137, 204].

- *Self-similarity.* There is a strong visual similarity among face images of different individuals. Geometric positioning of facial features such as eyes, noses, mouths, etc. are alike across individuals. Early face recognition approaches in the 70's [24, 46] used the distances between feature points to describe the face and achieved some success. Also, face surface materials properties are similar within the same race. As a consequence of visual similarity, the 'shapes' of the face appearance manifolds belonging to different subjects are similar. This is the foundation of approaches [55, 56, 211] that attempt to capture the 'shape' characteristics by constructing the so-called intra-person

space.

- *Makeup, cosmetic, etc.* There factors are specific to an individual and so are unpredictable. Except that the effect of glasses has been studied in [41], effects induced by other factors have not been widely investigated.

Face appearances of the same individual under variations in illumination, pose, deformation, aging, etc. lie in a nonlinear manifold. Figure 1.7 visualizes such a manifold by projecting the appearances of the top row into top three principal components. Manifold characterization can be done in various ways. One way is to embed a manifold in a low-dimensional space [162, 166]. The other way is to learn the nonlinearity using machine learning techniques [9, 19, 63, 172, 177, 179, 181, 189, 198].

1.2 Unconstrained Face Recognition

State-of-the-art face recognition systems yield satisfactory performance under controlled conditions. To be specific, the face images are typically acquired in frontal views and are often illuminated by a frontal light source. These conditions pose strong restrictions on patterns possibly acquired. In other words, the clustering nature of the produced patterns (usually tightly clustered) is amenable for classical pattern analysis. Therefore, most face recognition approaches lie in the first level of the hierarchy. Unfortunately, recognition performance degrades significantly when face recognition systems are presented with patterns that go beyond these controlled conditions.

Recently, researchers have begun to investigate face recognition under unconstrained conditions. Examples of unconstrained conditions include illumination

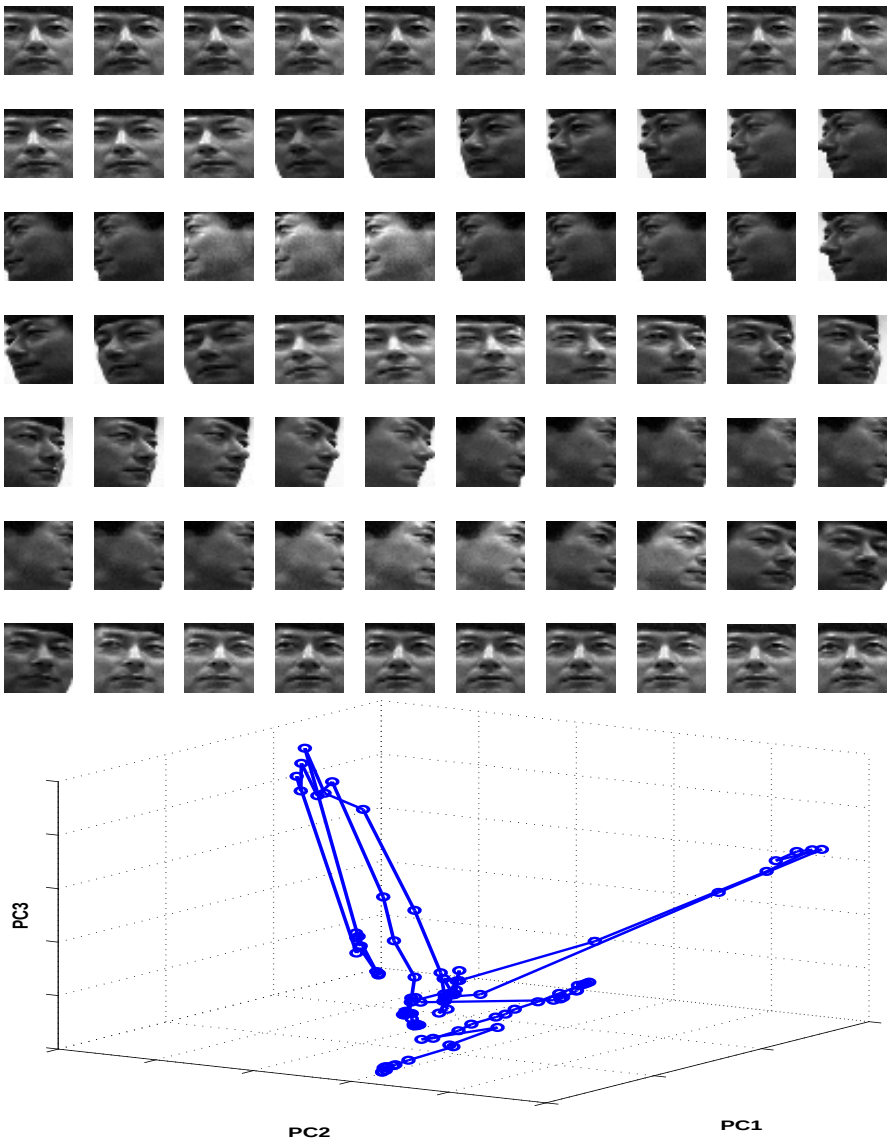


Figure 1.7: Face appearances in a video sequences, forming a nonlinear manifold.

and pose variations, video sequences, expression, aging, and so on. In general, recognition approaches addressing the second and third levels of the hierarchy can be considered in the category of unconstrained face recognition.

The present thesis presents several unconstrained face recognition approaches. It consists of three parts: Part I is on *Face Recognition under Variations*, Part

II on *Face Recognition via Kernel Learning*, and Part III on *Face Tracking and Recognition from Videos*.

1.2.1 Face recognition under variations

Part I of the thesis studies face recognition under illumination and pose variations. Pose and illumination are related to the second level of Figure 1.3. In Chapter 2, we present a *generalized photometric stereo* algorithm for recognizing faces under illumination variation and then in Chapter 3 an *illuminating light field* algorithm for recognizing faces under illumination and pose variations.

Most photometric stereo algorithms employ a Lambertian reflectance model with a varying albedo field and involve the appearances of only one object. The recovered albedos and surface normals are *object-specific* and appearances not belonging to the object cannot be easily handled. In Chapter 2, we generalize photometric stereo algorithms to handle all appearances of all objects in a class, in particular the human face class, by assuming that albedos and surface normals of all objects in the class be rank-constrained, i.e. lie in a subspace. Rank constraints lead us to a factorization of an observation matrix that consists of exemplar images of different objects under different illuminations. To fully recover the subspace bases or *class-specific* albedos and surface normals, we employ integrability and face symmetry constraints and propose a linearized algorithm. This algorithm takes into account the effects of varying albedo field by approximating the integrability terms using only the surface normals. We then apply our generalized photometric stereo algorithm for recognizing faces under illumination variations. As far as recognition is concerned, we can utilize a bootstrap set which is just a collection of 2D image observations to avoid an explicit requirement that 3D infor-

mation be available. We obtain good recognition results using the PIE database [187, 202, 204].

The *illuminating light field* algorithm presented in Chapter 3 is an image-based method for face recognition across different illumination and different poses, where the term image-based means that no explicit prior 3D models are needed. As face recognition under illumination and pose variations involves three factors, namely identity, illumination, and pose, generalizations in all these three factors are desired. The *illuminating light field* approach is able to generalize in identity and illumination and handle a given set of poses. The proposed approach derives an identity signature that is illumination- and pose-invariant, where the identity is tackled using subspace encoding, the illumination is characterized using a Lambertian reflectance model, and the given set of poses is treated as a whole. Experimental results using the PIE database demonstrate the effectiveness of the proposed approach [188, 208].

1.2.2 Face recognition via kernel learning

As mentioned earlier, the visual pattern lies in a nonlinear manifold, which is further complicated by face-specific characteristics. Nonlinear data modeling is an important research topic in machine learning. While linear data modeling such as PCA and LDA utilizes first- and second-order statistics, higher-order statistics play essential roles in nonlinear data modeling. Kernel learning methods (or kernel methods) are able to capture the higher-order statistical information.

In the core of kernel learning methods lie two important components: a learning algorithm using linear geometry and a nonlinear feature space induced by a kernel function. Such a space is referred as reproducing kernel Hilbert space (RKHS)

in the literature. Kernel methods are linear learning algorithms operating on the nonlinear feature space. In Part II, we introduce two kernel learning methods.

Chapter 4 presents a probabilistic approach to analyze kernel principal components by naturally combining in one treatment the theory of probabilistic principal component analysis and that of kernel principal component analysis. In this formulation, the kernel component enhances the nonlinear modeling power, while the probabilistic structure offers (i) a mixture model for nonlinear data structure containing nonlinear sub-structures, and (ii) an effective classification scheme. It also turns out that the original loading matrix [15] is replaced by the newly defined empirical loading matrix. The expectation/maximization algorithm for learning parameters of interest is then developed. Computation of reconstruction error and Mahalanobis distance is also discussed. Finally, we apply this approach to face recognition [198, 209].

Probabilistic distance measures are important quantities in many research areas. For example, the Chernoff distance (or the Bhattarchayya distance as its special example) is often used to bound the Bayes error in a pattern classification task and the Kullback-Leibler (KL) distance is a key quantity in information theory literature. However, computing these distances is a difficult task and analytic solutions are not available except under some special conditions. One popular example is the Gaussian density. The Gaussian density employs only up to second-order statistics and its modeling capacity is linear and hence rather limited. In Chapter 5, we enhance this capacity through a nonlinear mapping from original data space to RKHS, which is implemented using kernel embedding. Since this mapping is nonlinear, we achieve a new paradigm to study these distances whose feasibility and efficiency are demonstrated using experiments on synthetic and face

recognition examples [189].

1.2.3 Face tracking and recognition from videos

Video sequences are becoming ubiquitous due to the advances in digital imaging devices and the advent of internet era. A face in video sequences presents further challenges to recognition algorithms besides those common to face recognition from still images.

In Chapter 6, we present an approach called *adaptive visual tracking* that incorporates appearance-adaptive models in a particle filter to realize robust visual tracking. Tracking needs modeling of inter-frame motion and appearance changes whereas recognition needs modeling of appearance changes between frames and gallery images. In conventional tracking algorithms, the appearance model is either fixed or rapidly changing, and the motion model is simply a random walk with fixed noise variance. Also, the number of particles is typically fixed. All these factors make the visual tracker unstable. To stabilize the tracker, we propose the following features: an observation model arising from an adaptive appearance model, an adaptive velocity motion model with adaptive noise variance, and an adaptive number of particles. The adaptive-velocity model is derived using a first-order linear predictor based on the appearance difference between the incoming observation and the existing particle configuration. Occlusion analysis is implemented using robust statistics. Experimental results [186, 201, 203] on tracking visual objects in long outdoor and indoor video sequences demonstrate the effectiveness and robustness of our tracking algorithm.

In Chapter 7, recognition of human faces using a gallery of still images and a probe set of videos is systematically investigated using a probabilistic framework

called *simultaneous tracking and recognition*. In still-to-video recognition, where the gallery consists of still images, a time series state space model is proposed to fuse temporal information in a probe video, which simultaneously characterizes the kinematics and identity using a motion vector and an identity variable, respectively. The joint posterior distribution of the motion vector and the identity variable is estimated at each time instant and then propagated to the next time instant. Marginalization over the motion vector yields a robust estimate of the posterior distribution of the identity variable. A computationally efficient sequential importance sampling (SIS) algorithm is developed to estimate the posterior distribution. Empirical results demonstrate that, due to the propagation of the identity variable over time, a degeneracy in posterior probability of the identity variable is achieved to give improved recognition. We perform experiments [192, 193, 194, 195, 196, 197, 199] using images/videos with pose/illumination variations to illustrate the effectiveness of this approach for the still-to-video scenario with appropriate model choices.

In Chapter 8, we present the most general framework for characterizing the face identity in a single image or a group of images with each image containing a transformed version of the object. In terms of the transformation, the group is made of either still images or frames of a video sequence. The face identity signature is either discrete- or continuous-valued. This framework referred as *probabilistic identity characterization* integrates all the evidence of the set and handles the localization problem, illumination and pose variations through subspace identity encoding. Issues and challenges arising in this framework are addressed and efficient computational schemes are given. All instances of face recognition algorithms are be interpreted in the most general framework [210].

Part I: Face Recognition under Variations

Chapter 2

Generalized Photometric Stereo

In this chapter, we present a theory of generalized photometric stereo and its application to face recognition across illumination. We first present the generalized photometric stereo algorithm which is able to handle all appearances under different illumination of all objects in a class, in particular the human face class. In contrast, the ordinary photometric stereo algorithm handles the appearances belonging to one object under different illumination. We then evaluate this algorithm in its application to face recognition under illumination variation. Since this generalization is linear, the blending linear coefficients offer an illuminant-invariant identity signature.

Figure 2.1 motivates the proposed approach. The first row of Figure 2.1 displays one Yale object [68] under eight different illumination. Photometric stereo algorithms can recover the varying albedos and surface normals for the object, even assuming no knowledge of the illumination conditions. Here, by photometric stereo algorithm we mean any algorithm that utilizes a Lambertian reflectance model to describe the visual appearance and has the capability to recover the albedos and surface normals involved in the reflectance model. However, ordinary photomet-

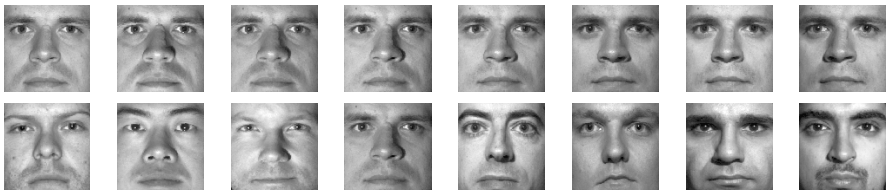


Figure 2.1: Top row: One object under eight different light sources. This can be handled by the ordinary photometric stereo algorithm. Bottom row: Eight different objects illuminated by eight different lighting sources. This cannot be handled by the ordinary photometric stereo algorithm but can be handled by the proposed generalized photometric stereo algorithm.

ric stereo algorithm cannot handle the images in the second row of Figure 2.1, where each image represents a different object under a different illumination. This motivates us to propose a generalized photometric stereo approach.

As in ordinary photometric stereo algorithm, the generalized photometric stereo algorithm utilizes a Lambertain reflectance model to depict the visual appearance. The significant difference between the ordinary and generalized photometric stereo algorithms lies in the image ensemble they analyze. The image ensemble that the ordinary photometric stereo algorithm analyzes consists of the appearances of one object under different illumination while, in general, the image ensemble that the generalized photometric stereo algorithm analyzes consists of the appearances of different objects, with each object under a different illumination. Analysis of the latter image ensemble is very difficult. To this end, we introduce a *key assumption*: These different objects belong to one class (for example, the human face class) so that they are linearly spanned by a fixed number of basis objects. Generalized photometric stereo does not assume any knowledge of the lighting sources as well as the blending coefficients. Rather, the generalized photometric stereo approach actually recovers such information. To further complicate the matter, the knowledge

of the basis objects is also unknown and needs to be recovered.

We evaluate the generalized photometric stereo algorithm for a face recognition application. The *key assumption* has two important implications. Firstly, it fits with the requirement of a recognition task that needs a generalization capability built on a training set. The idea is to learn the basis objects from the training set. Once learned, we use them to cope with arbitrary images belonging to objects other than those in the training set. Secondly, because the bases are for the object class only, the blending coefficients provide an identity encoding which is invariant to illumination. We use the blending coefficients for face recognition under illumination variation, which results in good recognition performance.

Chapter organization

Section 2.1 elaborates the generalized photometric stereo algorithm and addresses its issues and challenges. Section 2.2 details the face recognition setting and presents the experimental results using the PIE database. Appendices 2.I and 2.II give supplementary details of the algorithms proposed in the chapter.

A glossary of notations

In general, we denote a scalar by a, a vector by a, and a matrix with r rows and c columns by $\mathsf{A}_{r\times c}$. The matrix transpose is donate by A^{T}, the pseudo-inverse by $\mathsf{A}^{\dagger}$. The matrix L_2-norm is denoted by $||.||_2$.

The following notations are introduced for the sake of notational conciseness and emphasis of special structure.

- Concatenation notations: $\Rightarrow$ and $\Downarrow$.

  $\Rightarrow$ and $\Downarrow$ mean horizontal and vertical concatenations, respectively. For

example, we can represent a $n \times 1$ vector $\mathsf{a}_{n\times 1}$ by $\mathsf{a} = [a_1, a_2, ..., a_n]^{\mathsf{T}} = [\Downarrow_{i=1}^{n}\ a_i]$ and its transpose by $\mathsf{a}^{\mathsf{T}} = [a_1, a_2, ..., a_n] = [\Rightarrow_{i=1}^{n}\ a_i]$. We can use $\Rightarrow$ and $\Downarrow$ to concatenate matrices to form a new matrix. For instance, given a collection of matrices $\{\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n\}$ of size $r \times c$, we construct a $r \times cn$ matrix[1] $[\Rightarrow_{i=1}^{n}\ \mathsf{A}_i] = [\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n]$ and a $rn \times c$ matrix $[\Downarrow_{i=1}^{n}\ \mathsf{A}_i] = [\mathsf{A}_1^{\mathsf{T}}, \mathsf{A}_2^{\mathsf{T}}, \ldots, \mathsf{A}_n^{\mathsf{T}}]^{\mathsf{T}}$. In addition, we can combine $\Rightarrow$ and $\Downarrow$ to achieve a concise notation. Rather than representing a matrix $\mathsf{A}_{r\times c}$ as $[a_{ij}]$, we represent it as $\mathsf{A}_{r\times c} = [\Downarrow_{i=1}^{r}\ [\Rightarrow_{j=1}^{c}\ a_{ij}]\ ] = [\Rightarrow_{j=1}^{c}\ [\Downarrow_{i=1}^{r}\ a_{ij}]\ ]$. Also we can easily construct 'big' matrices using 'small' matrices $\{\mathsf{A}_{11}, \mathsf{A}_{12}, \ldots, \mathsf{A}_{1n}, \ldots, \mathsf{A}_{mn}\}$ of size $r \times c$. The matrix $[\Downarrow_{i=1}^{m}\ [\Rightarrow_{j=1}^{n}\ \mathsf{A}_{ij}]\ ]$ is of size $rm \times cn$, the matrix $[\Rightarrow_{i=1}^{m}\ [\Rightarrow_{j=1}^{n}\ \mathsf{A}_{ij}]\ ]$ of size $r \times cmn$.

- Kronecker (tensor) product: $\otimes$.

  It is defined as $\mathsf{A}_{m\times n} \otimes \mathsf{B}_{r\times c} = [\Downarrow_{i=1}^{m}\ [\Rightarrow_{j=1}^{n}\ a_{ij}\mathsf{B}]\ ]_{mr\times nc}$.

- Hadamard (element-wise) product: $\circ$.

  It is defined as $\mathsf{A}_{m\times n} \circ \mathsf{B}_{m\times n} = [\Downarrow_{i=1}^{m}\ [\Rightarrow_{j=1}^{n}\ a_{ij}b_{ij}]\ ]_{m\times n}$.

- Special notation: $\odot$.

  This is used for the special structure of the object-specific albedo-shape matrix T (The definitions of T, p, and N are listed below), i.e., $\mathsf{T}_{d\times 3} = [\Downarrow_{i=1}^{d}\ (p_i\mathsf{n}_i^{\mathsf{T}})] = \mathsf{p} \odot \mathsf{N}^{\mathsf{T}} = (\mathsf{p}_{d\times 1} \otimes \mathsf{1}_{1\times 3}) \circ \mathsf{N}_{3\times d}^{\mathsf{T}}$

Some special scalars, vectors, and matrices are defined as follows:

- d: number of pixels;

[1] We do not need the size of $\{\mathsf{A}_1, \mathsf{A}_2, \ldots, \mathsf{A}_n\}$ to be exactly same. We use the same matrix size for simplicity. For example, for $[\Rightarrow_{i=1}^{n}\ \mathsf{A}_i]$, we only need the number of rows of these matrices to be same.

- m: the rank used in the first rank constraint.
- i, j, i', j', l, and k: loop indices.
- $\mathsf{1}_{r\times c}$: a $r \times c$ matrix of ones.
- $\mathtt{I}_n$: an identity matrix of size $n \times n$.
- h: a pixel; $\mathsf{h}_{d\times 1}$: an image.
- p: albedo at a pixel. $\mathsf{p}_{d\times 1}$: albedo vector
- $\mathsf{n}_{3\times 1} = [\hat{a}, \hat{b}, \hat{c}]^{\mathsf{T}}$: unit surface normal vector; $\hat{a}$, $\hat{b}$, and $\hat{c}$: elements of n.
- $\mathsf{N}_{3\times d} = [\Rightarrow_{i=1}^{d} \mathsf{n}_i]$: the surface normal matrix.
- $\mathsf{t}_{3\times 1} = [a, b, c]^{\mathsf{T}}$: product of albedo and surface normal; a, b, and c: elements of t.
- $\mathsf{T}_{d\times 3} = [\Downarrow_{i=1}^{d} (p_i \mathsf{n}_i^{\mathsf{T}})]$: the object-specific albedo-shape matrix. Also, $\mathsf{T}_{d\times 3} = [\mathsf{a}, \mathsf{b}, \mathsf{c}]$ where a, b, and c are $d \times 1$ vectors.
- $\mathsf{s}_{3\times 1}$: illumination vector. $\mathsf{S}_{3\times n}$: the matrix consisting of a collection of different illumination vectors.
- $\mathsf{f}_{m\times 1}$: the vector of blending linear coefficients under the first rank constraint. $\mathsf{F}_{m\times n}$: the matrix consisting of a collection of different f's.
- $\mathsf{W}_{d\times 3m} = [\Rightarrow_{i=1}^{m} \mathsf{T}_i]$: the class-specific albedo-shape matrix. Also, $\mathsf{W}_{d\times 3m} = [\Rightarrow_{i=1}^{m} [\mathsf{a}_i, \mathsf{b}_i, \mathsf{c}_i]]$.
- A, B, C: $\mathsf{A} = [\Rightarrow_{i=1}^{m} \mathsf{a}_i]$, $\mathsf{B} = [\Rightarrow_{i=1}^{m} \mathsf{b}_i]$, and $\mathsf{C} = [\Rightarrow_{i=1}^{m} \mathsf{c}_i]$.
- W_{f}: $\mathsf{W}_{\mathsf{f}} = [\Rightarrow_{i=1}^{m} (\mathsf{T}_i \mathsf{s})]_{d\times m}$.

- $\mathsf{W_S}$: $\mathsf{W_S} = [\mathsf{Af}, \mathsf{Bf}, \mathsf{Cf}]$.
- $\mathsf{H}_{d\times n} = [\Rightarrow_{i=1}^{n} \mathsf{h}_i]$: the observation matrix consisting of a collection of images.
- $\hat{\mathsf{W}}_{d\times 3m}$: the U matrix after a rank-$3m$ SVD factorization of H.
- $\hat{\mathsf{w}}_{(\mathsf{x})}$: a $3m \times 1$ vector same as the row in $\hat{\mathsf{W}}$ associated with the pixel x
- $\mathsf{R}_{3m\times 3m}$: the ambiguity matrix in the factorization.
- r_{aj}, r_{bj}, and r_{cj}: the $(3j-2)^{th}$, $(3j-1)^{th}$, and $(3j)^{th}$ columns of the matrix R.
- τ: an indicator function.
- $\mathsf{x} = (x, y)$: pixel coordinate; $\bar{\mathsf{x}} = (-x, y)$: the symmetric point of x.
- α: the integrability constraint term.
- β: the face symmetry constraint term.

2.1 Principle of Generalized Photometric Stereo

This section describes the generalized photometric stereo algorithm. We start in Section 2.1.1 by a brief review of related literature and highlight the advantages of the proposed approach. We list in Section 2.1.2 the setting and constraints. Then we present a method to recover the albedos and surface normal for a class of objects in Sections 2.1.3 and 2.1.4. Section 2.1.3 handles the isolated task of separating the illumination (*v.i.z.* finding the illuminant vector and the blending coefficients) from an arbitrary image, which is used in the recovery algorithm presented in Section 2.1.4.

2.1.1 Literature review and proposed approach

Recovery of albedos and surface normals has been studied in the computer vision research for a long time. Usually a Lambertian reflectance model, ignoring both attached and cast shadows, is employed. Early works from the shape from shading (SFS) literature have typically assumed a constant albedo field: this assumption is not valid for many real objects and thus limits the practical applicability of the SFS algorithms. Early photometric stereo approaches require the knowledge of lighting conditions, but such knowledge is hard to gather under uncontrolled scenarios. Recent research efforts [74, 68, 94, 95, 96, 101, 103, 104] attempt to go beyond these restrictions by (i) using a varying albedo field, a more accurate model of the real world, and (ii) assuming no prior knowledge or requiring no control of the lighting sources. As a consequence, the complexity of the problem has also significantly increased.

If we fix the imaging geometry and only move the lighting source to illuminate one object, the observed images (ignoring the cast and attached shadows) lie in a subspace completely determined by three images illuminated by three independent lighting sources [101]. If an ambient component is added [103], this subspace becomes 4-D. If attached shadows are considered, the subspace dimension grows to infinity [97] but most of its energy is packed in a limited number of harmonic components, thereby leading to a low-dimensional subspace approximations in [94, 95, 100]. However, all the photometric-stereo-type approaches (except [74]) commonly restrict themselves to using *object-specific* samples and cannot perform reconstruction combining images produced by different objects.

In this chapter, we present a generalized photometric stereo algorithm that is able to handle all appearances of all objects in a class, in particular the human face

class. To this end, we impose a rank constraint (i.e. a linear generalization) on the albedos and surface normals of all human faces. We choose the human face as a working example because it naturally fits in our framework and is widely studied in the photometric stereo literature; however this does not pose any limitations in applying our algorithm to other object classes such as vehicles.

We propose a rank constraint on the product of albedo and surface normal. The rank constraint enables us to accomplish a factorization of the observation matrix that decomposes a *class-specific* ensemble into a product of two matrices: one encoding the albedos and surfaces normals for a class of objects and the other encoding blending linear coefficients and lighting conditions. A *class-specific* ensemble consists of exemplar images of different objects with each under a different illumination, which is beyond what can be analyzed using the bilinear analysis of [138]. Bilinear analysis requires exemplar images of different objects under the same set of illumination conditions. Because a factorization is always up to an invertible matrix, unique recovery of the albedos and surface normals is not possible and requires additional constraints. We use two constraints: surface integrability and face symmetry.

The surface integrability constraint [99, 137] has been used in several approaches [68, 103] to successfully recover albedo and shape. The symmetry constraint has also been employed in [102, 104] for face images. We present an approach to fusing these constraints to recover the *class-specific* albedos and surface normals, even in the presence of shadows. More importantly, this approach takes into account the effects of a varying albedo field by approximating the integrability terms using only the surface normals instead of the product of the albedos and the surface normals. Due to the nonlinearity embedded in the integrability

terms, regular algorithms such as the steepest descent are inefficient. We derive a linearized algorithm to find the solution.

2.1.2 Setting and constraints

Photometric stereo

We assume a Lambertian imaging model with a varying albedo field. A pixel h is represented as

$$h = p\ \mathsf{n}^{\mathsf{T}}\mathsf{s} = \mathsf{t}^{\mathsf{T}}\mathsf{s}, \tag{2.1}$$

where $[.]^{\mathsf{T}}$ denotes the transpose, p is the albedo at the pixel, $\mathsf{n} \equiv [\hat{a}, \hat{b}, \hat{c}]^{\mathsf{T}}$ is the unit surface normal vector at the pixel, $\mathsf{t}_{3\times 1} \equiv [a \equiv p\hat{a}, b \equiv p\hat{b}, c \equiv p\hat{c}]^{\mathsf{T}}$ is the product of albedo and surface normal, and s (a 3×1 unit vector multiplied by its intensity) specifies a distant illuminant. For time being, we consider the case without the shadow pixels and will deal with the shadow pixels later on.

An image h is a collection of d pixels $\{h_i, i = 1, ..., d\}$ [2]. By stacking all the pixels into a column vector, we have

$$\begin{aligned}
\mathsf{h}_{d\times 1} &\equiv [\Downarrow_{i=1}^{d}\ h_i] = [\Downarrow_{i=1}^{d}\ (p_i\ \mathsf{n}_i^{\mathsf{T}})]\mathsf{s} = [\Downarrow_{i=1}^{d}\ \mathsf{t}_i^{\mathsf{T}}]\mathsf{s} = [\Downarrow_{i=1}^{d}\ [a_i, b_i, c_i]]\mathsf{s} \\
&= (\mathsf{p}_{d\times 1} \odot \mathsf{N}_{3\times d}^{\mathsf{T}})\mathsf{s}_{3\times 1} = [\mathsf{a}_{d\times 1}, \mathsf{b}_{d\times 1}, \mathsf{c}_{d\times 1}]\mathsf{s}_{3\times 1} \qquad (2.2) \\
&= \mathsf{T}_{d\times 3}\ \mathsf{s}_{3\times 1}, \qquad (2.3)
\end{aligned}$$

where $\mathsf{p} \equiv [\Downarrow_{i=1}^{d}\ p_i]$ is the albedo vector, $\mathsf{N} \equiv [\Rightarrow_{i=1}^{d}\ \mathsf{n}_i]$ is the surface normal matrix, $\mathsf{a} \equiv [\Rightarrow_{i=1}^{d}\ a_i] = [\Rightarrow_{i=1}^{d}\ p_i\hat{a}_i]$, $\mathsf{b} \equiv [\Rightarrow_{i=1}^{d}\ b_i] = [\Rightarrow_{i=1}^{d}\ p_i\hat{b}_i]$, and $\mathsf{c} \equiv [\Rightarrow_{i=1}^{d}\ c_i] = [\Rightarrow_{i=1}^{d}\ p_i\hat{c}_i]$. To emphasize the structure of the T matrix which is a 'product'

[2] The index i corresponds to a spatial position $\mathsf{x} = (x, y)$. We will interchange both notations. For instance, we might also use $\mathsf{x} = 1, ..., d$.

of the albedo vector p and the surface normal N, we introduce a special notation $\odot$ to denote T by

$$\mathsf{T} \equiv \mathsf{p} \odot \mathsf{N}^{\mathsf{T}} \equiv [\Downarrow_{i=1}^{d} \mathsf{t}_i^{\mathsf{T}}] \equiv [\mathsf{a}, \mathsf{b}, \mathsf{c}]. \tag{2.4}$$

We call the T matrix as the *object-specific albedo-shape* matrix.

In the case of photometric stereo, we have n images of the *same* object, say $\{\mathsf{h}_1, \mathsf{h}_2, \ldots, \mathsf{h}_n\}$, observed at a fixed pose illuminated by n different lighting sources, forming an *object-specific* ensemble. Simple algebraic manipulation gives:

$$\mathsf{H}_{d\times n} \equiv [\Rightarrow_{i=1}^{n} \mathsf{h}_i] = \mathsf{T}[\Rightarrow_{i=1}^{n} \mathsf{s}_i] = \mathsf{T}_{d\times 3}\ \mathsf{S}_{3\times n}, \tag{2.5}$$

where H is the *observation matrix* and $\mathsf{S} \equiv [\Rightarrow_{i=1}^{n} \mathsf{s}_i]$ encodes the information on the illuminants. Hence photometric stereo is rank-3 constrained. Therefore, given at least three exemplar images for one object under three different independent illumination, we can determine the identity of a new probe image by checking if it lies in the linear span of the three exemplar images. This requires capturing at least three images for one object in the gallery set, which can be prohibitive in practical scenarios. Note that in this recognition setting, there is no need for the training set; in other words, the training set is equivalent to the gallery set.

A typical recognition setting [58], however, assumes no identity overlap between the gallery set and the training set and often stores only one exemplar image for each object in the gallery set. However, the training set can have multiple images for one object. In order to generalize from the training set to the gallery and probe sets, we note that all images in the training, gallery, and probe sets belong to the same face class, which naturally leads to the rank constraint.

The rank constraint

We impose the rank constraint on the T matrix by assuming that any T matrix is a linear combination of some basis matrices $\{\mathsf{T}_1, \mathsf{T}_2, \ldots, \mathsf{T}_m\}$ coming from some m *basis objects*. Rank constraints are often found in the literature [110, 111, 129, 116, 117, 122]. Mathematically, there exist coefficients $\{f_j;\ j = 1, \ldots, m\}$ such that

$$\mathsf{T}_{d\times 3} = \sum_{j=1}^{m} f_j \mathsf{T}_j = [\Rightarrow_{j=1}^{m} \mathsf{T}_j](\mathsf{f} \otimes \mathtt{I}_3) = \mathsf{W}_{d\times 3m}(\mathsf{f}_{m\times 1} \otimes \mathtt{I}_3), \tag{2.6}$$

where $\mathsf{f} \equiv [\Downarrow_{j=1}^{m} f_j]$, $\mathsf{W} \equiv [\Rightarrow_{j=1}^{m} \mathsf{T}_j]$, $\mathtt{I}_n$ denotes an identity matrix of dimension $n \times n$, and $\otimes$ denotes the Kronecker (tensor) product. Since the W matrix encodes all albedos and surface normals for a class of objects, we call it a *class-specific albedo-shape* matrix. Substitution of (2.6) into (2.3) yields

$$\mathsf{h}_{d\times 1} = \mathsf{T}\mathsf{s} = \mathsf{W}(\mathsf{f} \otimes \mathtt{I}_3)\mathsf{s} = \mathsf{W}(\mathsf{f} \otimes \mathsf{s}) = \mathsf{W}_{d\times 3m}\ \mathsf{k}_{3m\times 1}, \tag{2.7}$$

where $\mathsf{k} \equiv \mathsf{f} \otimes \mathsf{s}$. This leads to a two-factor bilinear analysis [138].

With the availability of n images $\{\mathsf{h}_1, \mathsf{h}_2, \ldots, \mathsf{h}_n\}$ for *different* objects, observed at a fixed pose illuminated by n different lighting sources, forming a *class-specific* ensemble, we have

$$\mathsf{H}_{d\times n} = [\Rightarrow_{i=1}^{n} \mathsf{h}_i] = \mathsf{W}[\Rightarrow_{i=1}^{n} (\mathsf{f}_i \otimes \mathsf{s}_i)] = \mathsf{W}[\Rightarrow_{i=1}^{n} \mathsf{k}_i] = \mathsf{W}_{d\times 3m}\ \mathsf{K}_{3m\times n}, \tag{2.8}$$

where $\mathsf{K} \equiv [\Rightarrow_{i=1}^{n} (\mathsf{f}_i \otimes \mathsf{s}_i)] = [\Rightarrow_{i=1}^{n} \mathsf{k}_i]$. It is a rank-$3m$ problem, which combines the rank of 3 for the illumination and the rank of m for the identity.

The rank constraint generalizes many approaches in the literature other than the photometric stereo. If the surface normal is fixed and the albedo field lies in a rank-m linear subspace, we have (2.6) satisfied. Interestingly, the 'Eigenface' approach [62] is just a special case of this approach for a fixed illumination source.

Suppose that the fixed illuminant vector is $\tilde{\mathsf{s}}$. (2.7) and (2.8) reduce to

$$\mathsf{h}_{d\times 1} = \mathsf{W}(\mathsf{f} \otimes \tilde{\mathsf{s}}) = \tilde{\mathsf{W}}_{d\times m}\mathsf{f}_{m\times 1};$$

$$\mathsf{H}_{d\times n} = [\Rightarrow_{i=1}^{n} \mathsf{h}_i] = \tilde{\mathsf{W}}[\Rightarrow_{i=1}^{n} \mathsf{f}_i] = \tilde{\mathsf{W}}_{d\times m}\mathsf{F}_{m\times n}, \tag{2.9}$$

where $\tilde{\mathsf{W}} \equiv [\Rightarrow_{i=1}^{m} \mathsf{T}_i\tilde{\mathsf{s}}]$. Therefore, our approach can also be regarded as a generalized 'Eigenface' analysis able to handle illumination variation.

Our immediate goal is to estimate W and K from the observation matrix H. The first step is to invoke an SVD factorization, $\mathsf{H} = \mathsf{U}\Lambda\mathsf{V}^{\mathsf{T}}$, and retain the top $3m$ components as $\mathsf{H} = \mathsf{U}_{3m}\Lambda_{3m}\mathsf{V}_{3m}^{\mathsf{T}}$=$\hat{\mathsf{W}}\ \hat{\mathsf{K}}$, where $\hat{\mathsf{W}} = \mathsf{U}_{3m}$ and $\hat{\mathsf{K}} = \Lambda_{3m}\mathsf{V}_{3m}^{\mathsf{T}}$. Thus, we can recover W and K up to an $3m \times 3m$ invertible matrix R with $\mathsf{W} = \hat{\mathsf{W}}\mathsf{R}$, $\mathsf{K} = \mathsf{R}^{-1}\hat{\mathsf{K}}$. Additional constraints are required to determine the R matrix. We will use the integrability and face symmetry constraints, both related to W. Moreover, K must take the special structure $\mathsf{K} = [\Rightarrow_i (\mathsf{f}_i \otimes \mathsf{s}_i)]$.

Incidentally, by noting that $\mathsf{T} = \mathsf{p} \odot \mathsf{N}^{\mathsf{T}}$, we can introduce a second rank constraint which assumes that (i) any p vector is a linear combination of some basis vectors $\{\mathsf{p}_1, \mathsf{p}_2, \ldots, \mathsf{p}_{m_1}\}$ with $m_1 < d$ and (ii) any N matrix is a linear combination of some basis matrices $\{\mathsf{N}_1, \mathsf{N}_2, \ldots, \mathsf{N}_{m_2}\}$ with $m_2 < d$. This is a common constraint used in the face recognition literature. For example, in [43, 66, 76], they all assume that shape and texture have separate bases. However, it turns out that the second rank constraint is not systematically superior to the first rank constraint in terms of recognition performance. Also, it is computationally inconvenient to use the second rank constraint.

Hence, there exist two vectors $\mathsf{f}_{m_1\times 1} \equiv [\Downarrow_i f_i]$ and $\mathsf{g}_{m_2\times 1} \equiv [\Downarrow_i g_i]$ such that

$$\mathsf{p} = [\Rightarrow_{i=1}^{m_1} \mathsf{p}_i]\mathsf{f}; \mathsf{N}^{\mathsf{T}} = [\Rightarrow_{j=1}^{m_2} \mathsf{N}_j^{\mathsf{T}}](\mathsf{g} \otimes \mathtt{I}_3), \tag{2.10}$$

and similarly the image h can be expressed as

$$\begin{aligned} \mathsf{h}_{d\times 1} &= [\Rightarrow_{i=1}^{m_1} [\Rightarrow_{j=1}^{m_2} (\mathsf{p}_i \odot \mathsf{N}_j^{\mathsf{T}})]](\mathsf{f} \otimes \mathsf{g} \otimes \mathsf{s}) \\ &= \mathsf{Y}_{d\times 3m_1m_2}(\mathsf{f}_{m_1\times 1} \otimes \mathsf{g}_{m_2\times 1} \otimes \mathsf{s}_{3\times 1}), \end{aligned} \tag{2.11}$$

where $\mathsf{Y} \equiv [\Rightarrow_{i=1}^{m_1} [\Rightarrow_{j=1}^{m_2} (\mathsf{p}_i \odot \mathsf{N}_j^{\mathsf{T}})]]$.

The integrability constraint

One common constraint used in SFS research is the integrability of the surface [68, 99, 103, 137]. Suppose that the surface function is $z = z_{(\mathsf{x})}$ with $\mathsf{x} \equiv (x, y)$, we must have $\frac{\partial}{\partial x}\frac{\partial z}{\partial y} = \frac{\partial}{\partial y}\frac{\partial z}{\partial x}$. For the given unit surface normal vector $\mathsf{n}_{(\mathsf{x})} \equiv [\hat{a}_{(\mathsf{x})}, \hat{b}_{(\mathsf{x})}, \hat{c}_{(\mathsf{x})}]^{\mathsf{T}}$ at pixel x, the integrability constraint requires that

$$\frac{\partial}{\partial x}\frac{\hat{b}_{(\mathsf{x})}}{\hat{c}_{(\mathsf{x})}} = \frac{\partial}{\partial y}\frac{\hat{a}_{(\mathsf{x})}}{\hat{c}_{(\mathsf{x})}}. \tag{2.12}$$

In other words, with $\alpha_{(\mathsf{x})}$ defined as an integrability constraint term,

$$\alpha_{(\mathsf{x})} \equiv \hat{c}_{(\mathsf{x})}\frac{\partial \hat{b}_{(\mathsf{x})}}{\partial x} - \hat{b}_{(\mathsf{x})}\frac{\partial \hat{c}_{(\mathsf{x})}}{\partial x} + \hat{a}_{(\mathsf{x})}\frac{\partial \hat{c}_{(\mathsf{x})}}{\partial y} - \hat{c}_{(\mathsf{x})}\frac{\partial \hat{a}_{(\mathsf{x})}}{\partial y} = 0. \tag{2.13}$$

If given the product of the albedo and the surface normal $\mathsf{t}_{(\mathsf{x})} \equiv [a_{(\mathsf{x})}, b_{(\mathsf{x})}, c_{(\mathsf{x})}]^{\mathsf{T}}$ with $a_{(\mathsf{x})} \equiv p_{(\mathsf{x})}\hat{a}_{(\mathsf{x})}$, $b_{(\mathsf{x})} \equiv p_{(\mathsf{x})}\hat{b}_{(\mathsf{x})}$, and $c_{(\mathsf{x})} \equiv p_{(\mathsf{x})}\hat{c}_{(\mathsf{x})}$, Eq. (2.13) still holds with $\hat{a}$, $\hat{b}$, and $\hat{c}$ replaced by a, b, and c, respectively. Practical algorithms approximate the partial derivatives by forward or backward differences or other differences with the inherent smoothness assumption. Hence, the approximations based on $\mathsf{t}_{(\mathsf{x})}$ are very rough especially at places where abrupt albedo variations exist (e.g. the boundaries of eyes, iris, eyebrow, etc.) since the smoothness assumption is seriously violated. We should by all means use $\mathsf{n}_{(\mathsf{x})}$ in order to remove this effect.

The face symmetry constraint

For a face image in a frontal view, one natural constraint is its symmetry about the central y-axis [102, 104]:

$$p_{(x,y)} = p_{(-x,y)}; \hat{a}_{(x,y)} = -\hat{a}_{(-x,y)}; \hat{b}_{(x,y)} = \hat{b}_{(-x,y)}; \hat{c}_{(x,y)} = \hat{c}_{(-x,y)}, \tag{2.14}$$

which is equivalent to, using $\mathsf{x} \equiv (x, y)$ and its symmetric point $\bar{\mathsf{x}} \equiv (-x, y)$,

$$a_{(\mathsf{x})} = -a_{(\bar{\mathsf{x}})};\ b_{(\mathsf{x})} = b_{(\bar{\mathsf{x}})}; c_{(\mathsf{x})} = c_{(\bar{\mathsf{x}})}. \tag{2.15}$$

If a face image in a non-frontal view, such a symmetry still exists but the coordinate system should be modified to take into account the view change.

2.1.3 Separating illumination

In this section, we temporarily assume that the class-specific albedo-shape matrix W is available and solve the problem of separating illumation, *v.i.z.*, for an arbitrary image h, find the illuminant vector s and the coefficient f under the first constraint (or f and g under the second constraint). For convenience in performing tasks such as recognition, we also normalize the solution f to the same range.

The first rank constraint gives rise to the basic equation $\mathsf{h} = \mathsf{W}\ (\mathsf{f} \otimes \mathsf{s})$. So, we convert the separation task to a minimization task of finding f and s to minimize the least square (LS) cost, i.e.,

$$\min_{\mathsf{f},\mathsf{s}} \mathcal{E}(\mathsf{f}, \mathsf{s}) \equiv \|\mathsf{h} - \mathsf{W}\ (\mathsf{f} \otimes \mathsf{s})\|^2, \tag{2.16}$$

Note that f and s can be recovered only up to a non-zero scalar; one can always multiply f by a non-zero scalar and divide s by the same scalar. Therefore, without loss of generality, we can simply pose an additional constraint: $\mathsf{1}^{\mathsf{T}}\mathsf{f} = 1$, where $\mathsf{1}_{m\times 1}$ is a vector of 1's.

One way to solve this is indicated in [74]. It is a two-step algorithm. First, k is approximated by $\mathsf{k} = \mathsf{W}^{\dagger}\mathsf{h}$. Then $\mathsf{k} = \mathsf{f} \otimes \mathsf{s}$ is used to solve for f and s, again using the LS approximation, i.e. finding f and s such that the cost $\|\mathsf{k} - \mathsf{f} \otimes \mathsf{s}\|^2$ is minimized. However, as pointed out in [74], the above algorithm is not robust since two approximations are involved.

Before we proceed to the actual separation algorithm, note that shadows in principle increase the rank (for the illumination only) to infinity. However, if those pixels are successfully excluded in our calculations, the rank for the illumination is still maintained to be 3 and the overall rank is $3m$.

In view of the above and considering the normalization requirement, we modify the cost function as

$$\mathcal{E}(\mathsf{f}, \mathsf{s}) \equiv \|\tau \circ (\mathsf{h} - \mathsf{W}\ (\mathsf{f} \otimes \mathsf{s}))\|^2 + (\mathsf{1}^{\mathsf{T}}\mathsf{f} - 1)^2, \tag{2.17}$$

where $\tau_{d\times 1}$ indicates the inclusion or exclusion of the pixels of the image h and $\circ$ denotes the Hadamard (or element-wise) product. Notice that (2.17) can be easily generalized to a cost function used in robust estimation if the vector norm is replaced by a robust function, and τ by an appropriate weight function.

Using the fact that Eq. (2.7) provides a series of sub-equations, which is linear in f if s is fixed and in s if f is fixed, we can design a simple iterative algorithm. Each iteration of the algorithm has three steps. In the first step, we solve for the LS estimate of f, given s and τ.

$$\mathsf{f} = \begin{bmatrix} \mathsf{W}_{\mathsf{f}} \\ \mathsf{1}^{\mathsf{T}} \end{bmatrix}^{\dagger} \begin{bmatrix} \tau \circ \mathsf{h} \\ 1 \end{bmatrix}; \quad \mathsf{W}_{\mathsf{f}} \equiv [\Rightarrow_{i=1}^{m} (\mathsf{T}_i\mathsf{s})]_{d\times m}. \tag{2.18}$$

In the second step, we solve for the LS estimate of s, given f and τ:

$$\mathsf{s} = \mathsf{W}_{\mathsf{s}}^{\dagger}(\tau \circ \mathsf{h}); \quad \mathsf{W}_{\mathsf{s}} \equiv [\ [\Rightarrow_{i=1}^{m} \mathsf{a}_i]\mathsf{f},\ [\Rightarrow_{i=1}^{m} \mathsf{b}_i]\mathsf{f},\ [\Rightarrow_{i=1}^{m} \mathsf{c}_i]\mathsf{f}\ ]_{d\times 3} \equiv [\mathsf{Af}, \mathsf{Bf}, \mathsf{Cf}], \tag{2.19}$$

where $\mathsf{A}_{d\times m} \equiv [\Rightarrow_{i=1}^{m} \mathsf{a}_i]$, $\mathsf{B}_{d\times m} \equiv [\Rightarrow_{i=1}^{m} \mathsf{b}_i]$, and $\mathsf{C}_{d\times m} \equiv [\Rightarrow_{i=1}^{m} \mathsf{c}_i]$, respectively. In the third step, given f and s we update τ as follows[3]:

$$\tau = [\ |\mathsf{h} - \mathsf{W}\ (\mathsf{f} \otimes \mathsf{s})| < \eta\ ], \tag{2.20}$$

where η is a pre-defined threshold.

Note that in (2.18) and (2.19), additional saving in computation is possible. We can form dimension-reduced matrices $\mathsf{W}_{\mathsf{f}}^{'}$ and $\mathsf{W}_{\mathsf{s}}^{'}$ and vector h' and apply the primed version in (2.18) and (2.19) The matrices $\mathsf{W}_{\mathsf{f}}^{'}$ and $\mathsf{W}_{\mathsf{s}}^{'}$ and vector h' are formed from W_{f}, W_{s}, and h, respectively, by discarding those rows corresponding to the excluded pixels.

The initial conditions can be arbitrary. But, for fast convergence, we need good initial values. In our implementation, we estimate s using the algorithm presented in [105]. To initialize τ, we employ heuristics to distinguish pixels in shadows: their intensities are close to zero. In practice, we set those pixels whose intensities are smaller than a certain threshold as missing values. In addition, we also set those pixels whose intensities are above a certain threshold as missing values to remove pixels possibly in a specular region. This is only for initialization, we update τ during iterations.

To test the stability of our algorithm, we perturb the initial conditions and find that our algorithm is very stable in the sense that it always reaches the same solution (up to the convergence error) regardless of initial conditions and generates a smaller residual than the algorithm reported in [74].

Learning f, g, and s from h using the second constraint is a straightforward generalization of the above algorithm. Appendix 2.I presents such a recovery algorithm in an even more general setting, i.e. a multilinear setting.

[3]This is a Matlab operation which performs an element-wise comparison.

2.1.4 Recovering class-specific albedos and surface normals

The recovery task is to find from the observation matrix H the *class-specific albedo-shape* matrix W (or equivalently R), which satisfies both the integrability and symmetry constraints, as well as the matrices F and S. We decompose R as $\mathsf{R}_{3m\times 3m} \equiv [\Rightarrow_{j=1}^{m} [\mathsf{r}_{aj}, \mathsf{r}_{bj}, \mathsf{r}_{cj}]]$ and treat the column vectors $\{\mathsf{r}_{aj}, \mathsf{r}_{bj}, \mathsf{r}_{cj};\ j = 1, ..., m\}$ as our computational 'units'. We also decompose $\hat{\mathsf{W}}$ as $\hat{\mathsf{W}} \equiv [\Downarrow_{\mathsf{x}=1}^{d} \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}]$ where $\hat{\mathsf{w}}_{(\mathsf{x})}$ is a $3m \times 1$ vector same as the row in $\hat{\mathsf{W}}$ corresponding to the pixel x. As $\mathsf{W} \equiv [\Downarrow_{\mathsf{x}=1}^{d} [\Rightarrow_{j=1}^{m} [a_{j(\mathsf{x})}, b_{j(\mathsf{x})}, c_{j(\mathsf{x})}]]] = \hat{\mathsf{W}}\mathsf{R}$, we have

$$a_{j(\mathsf{x})} = \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{aj},\ b_{j(\mathsf{x})} = \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{bj},\ c_{j(\mathsf{x})} = \hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}} \mathsf{r}_{cj};\ j = 1, \ldots, m. \tag{2.21}$$

As mentioned in Section 2.1.3, we must take into account attached and cast shadows. After setting them as missing values, we perform SVD with missing values [149] to find $\hat{\mathsf{W}}$. Other approaches for dealing with missing value are available in [141, 165, 169].

In view of the above, we formulate the following optimization problem: minimize over R, F, and S the cost function $\mathcal{E}$ defined as

$$\begin{aligned} \mathcal{E}(\mathsf{R}, \mathsf{F}, \mathsf{S}) &= \frac{1}{2} \sum_{i=1}^{n} \sum_{\mathsf{x}=1}^{d} \tau_i(\mathsf{x}) \{h_{i(\mathsf{x})} - \hat{\mathsf{w}}(\mathsf{x})^{\mathsf{T}} \mathsf{R}(\mathsf{f}_i \otimes \mathsf{s}_i)\}^2 \\ &\quad + \frac{\lambda_1}{2} \sum_{j=1}^{m} \sum_{\mathsf{x}=1}^{d} \{\alpha_{j(\mathsf{x})}\}^2 + \frac{\lambda_2}{2} \sum_{j=1}^{m} \sum_{\mathsf{x}=1}^{d} \{\beta_{j(\mathsf{x})}\}^2, \\ &= \mathcal{E}_0(\mathsf{R}, \mathsf{F}, \mathsf{S}) + \lambda_1 \mathcal{E}_1(\mathsf{R}) + \lambda_2 \mathcal{E}_2(\mathsf{R}), \end{aligned} \tag{2.22}$$

where $\tau_i(\mathsf{x})$ is an indicator function which takes the value one if the pixel x of the image h_i is not in shadow and zero otherwise, $\alpha_{j(\mathsf{x})}$ is the integrability constraint term based only on surface normals as defined in (2.13), and $\beta_{j(\mathsf{x})}$ is the symmetry constraint term given as

$$\beta_{j(\mathsf{x})}^2 = \{a_{j(\mathsf{x})} + a_{j(\bar{\mathsf{x}})}\}^2 + \{b_{j(\mathsf{x})} - b_{j(\bar{\mathsf{x}})}\}^2 + \{c_{j(\mathsf{x})} - c_{j(\bar{\mathsf{x}})}\}^2;\ j = 1, \ldots, m. \tag{2.23}$$

One approach could be to directly minimize the cost function over W, F, and S. This is in principle possible but numerically difficult as the number of unknowns depends on the image size, which can be quite large in practice.

As shown in [98], the recovered surface normal is up to a generalized bas-relief (GBR) ambiguity. To avoid trivial solutions such as a planar object[4], we normalize the matrix R by setting $||\mathsf{R}||_2 = 1$ where $||.||_2$ is a matrix norm. Another ambiguity between f_j and s_j is a nonzero scale, which can be removing by normalizing f to same range: $\mathsf{f}_j^{\mathsf{T}}\mathsf{1} = 1$, where $\mathsf{1}_{m\times 1}$ is a vector of 1's.

To summarize, we perform the following task:

$$min_{\mathsf{R},\mathsf{F},\mathsf{S}} \;\; \mathcal{E}(\mathsf{R},\mathsf{F},\mathsf{S}) \;\; \textit{subject to } ||\mathsf{R}||_2 = 1, \mathsf{F}^{\mathsf{T}}\mathsf{1} = \mathsf{1}. \tag{2.24}$$

An iterative algorithm can be designed to solve (2.24). While solving for F and S with R fixed is quite easy, solving for R with F and S is very difficult because the integrability constraint terms involve partial derivatives of the surface normals that are nonlinear in R. Regular algorithms such as the steepest descent are inefficient. One main contribution of this chapter is that we propose a linearized algorithm to solve for R, which is detailed in Appendix 2.II.

We now illustrate how to update $\mathsf{F} = [\Rightarrow_i \mathsf{f}_i]$, $\mathsf{S} = [\Rightarrow_i \mathsf{s}_i]$, and $\tau = [\Rightarrow_i \tau_i]$ with R fixed (or W fixed). First notice that F, S, and τ are only involved in the term $\mathcal{E}_0$. Moreover, f_i, s_i and τ_i are related to only the image h_i. This becomes the same as the illumination separation problem defined in Section 2.1.3. The proposed algorithm is also iterative in nature. After running one iterative step to obtain the updated F, S, and τ, we proceed to update R again and this process

[4]In this way, the surface normals we are recovering are versions up to a GBR ambiguity with respect to the true physical surface normals [68]. However, they are enough for tasks such as face recognition under illumination variation.

carries on until convergence.

To demonstrate how the algorithm works, we design the following scenario with $m = 2$ so that the rank of interest is 2x3=6. To defeat the photometric stereo algorithm, which requires one object illuminated by at least three sources, and the bilinear analysis, which requires two fixed objects illuminated by at least three same lighting sources, we construct eight images by taking random linear combinations of two basis objects illuminated by eight different lighting sources. Figure 2.2 displays the two basis objects under the same set of eight illumination and the synthesized images. The recovered class-specific albedo-shape matrix is also presented in Figure 2.2, which clearly shows the two basis objects. The quality of reconstruction is quite good except the nose part. The reason might be that the two basis objects have quite distinct noses so that the nose part of their linear combinations is not visually good (see the image in the last column of the third row), which propagates to the recovery results of albedos and surface normals from these combination images. Our algorithm usually converges within 100 iterations.

One notes that the special case $m = 1$ of our algorithm can be readily applied to photometric stereo (with the symmetry constraint removed) to robustly recover the albedos and surface normals for one object.

2.2 Face Recognition across Illumination

This section deals with the face recognition part, which serves as a main evaluation tool for the generalized photometric stereo algorithm. Section 2.2.1 briefly reviews the literature on face recognition across illumination. In Section 2.2.2, we relax the requirement of recovering the albedos and surface normals by utilizing sample imagery as a bootstrap set for the recognition task. We then report in Section

Figure 2.2: The first row: The first basis object under eight different illumination. The second row: The second basis object under the same set of eight different illumination. The third row: Eight images (constructed by random linear combinations of two basis objects) illuminated by eight different lighting sources. The fourth row: Recovered class-specific albedo-shape matrix W showing the product of varying albedos and surface normals of two basis objects (i.e. the three columns of T_1 and T_2) using the generalized photometric stereo algorithm.

2.2.3 face recognition results using the PIE database.

2.2.1 Literature review and proposed approach

Face recognition under illumination variation is a very challenging problem. The key is to successfully separate the illumination source from the observed appearance. Once separated, what remains is illuminant-invariant and appropriate for recognition. In addition to illumination variation, various issues embedded in the recognition setting make recognition even more difficult. We follow the recognition protocol introduced in [58]. Assuming the availability of the following three sets, namely one training set, one gallery set, and one probe set, the recognition algorithm learns from the training set the characteristic features, associates de-

scriptive features with the objects in the gallery set, and determines the identity for the objects in the probe set. Different recognition settings can be formed in terms of identity and illumination overlaps among the training, gallery, and probe sets. The most difficult setting, which is the focus of this chapter, is obviously the one in which there is no overlap at all among the three sets in terms of both identity and illumination, except the identity overlap between the gallery and probe sets. In this setting, generalizations from known illumination to unknown illumination and from known identities to unknown identities are particularly desired.

State-of-the-art research efforts can be grouped into three streams: subspace methods, reflectance-model methods, and 3D-model-based methods. (i) The first approach is very popular for the recognition problem. After removing the first three eigenvectors, principal component analysis (PCA) was reported to be more robust to illumination variation than the ordinary PCA or the 'Eigenface' approach [62]. Fisher discriminant analysis (FDA) [41, 70] has also been modified to handle illumination variations. In general, subspace learning methods are able to capture the generic face space and thus to recognize new objects not present in the training set. The disadvantage is that subspace learning is actually tuned to the lighting conditions of the training set; therefore if the illumination conditions are not similar among the training, gallery, and probe sets, recognition performance may not be acceptable. (ii) The second approach [68, 74, 101, 104] employs a Lambertian reflectance model with a varying albedo field ignoring both attached and cast shadows. The main disadvantage of this approach is the lack of generalization from known objects to unknown objects. (iii) The third approach employs 3D models. The 'Eigenhead' approach [65] assumes that the 3D geometry (or 3D depth information) of any face lies in a linear space spanned by the 3D geometry

of the training ensemble and uses a constant albedo field. The morphable model approach [66] is based on a synthesis-and-analysis strategy. Both geometry and texture are linearly spanned by those of the training ensemble. It is able to handle both illumination and pose variations with illumination directions specified. The weakness of the 3D model approaches is that they require 3D models and complicated fitting algorithms.

Compared to the above, the proposed recognition scheme possesses the following properties: (i) It is able to recognize new objects not present in the training set; (ii) It is able to handle new lighting conditions not present in the training set; and (iii) No explicit 3D model and no prior knowledge about illumination conditions are needed. In other words, we combine the advantages of subspace learning and reflectance model-based methods. Further, we can avoid the recovery burden as far as recognition is concerned by using a proper bootstrap set under the first constraint.

2.2.2 Bootstrap set

A procedure for learning the W matrix was presented in Section 2.1.4. Even though the learning algorithm is quite robust, it is possible that it gets trapped in local minima, which might subsequently yield inferior recognition results. Thus, an alternative approach without explicitly learning the W matrix is very beneficial. We now show that, as far as recognition is concerned, the W matrix under the first constraint can be replaced by a bootstrap set $\tilde{\mathsf{W}}$ consisting of sample imagery only. The bootstrap set can take various forms. In this chapter, we focus on such a bootstrap set that contains m exemplar objects captured at a fixed pose, each with three images illuminated by three independent but fixed lighting sources.

We denote $\tilde{\mathsf{h}}_{ij}$ as the image for the i^{th} exemplar object illuminated by the j^{th} exemplar lighting source. As an image can be expressed in a two-factor form using (2.7), we can write $\tilde{\mathsf{h}}_{ij}$ as

$$\tilde{\mathsf{h}}_{ij} = \mathsf{W}(\tilde{\mathsf{f}}_i \otimes \tilde{\mathsf{s}}_j);\ i = 1, ..., m; j = 1, 2, 3. \tag{2.25}$$

where $\tilde{\mathsf{f}}_i$ is the blending coefficient vector for the i^{th} exemplar object and $\tilde{\mathsf{s}}_j$ describes the j^{th} exemplar lighting source.

The bootstrap set $\tilde{\mathsf{W}}$ is then expressed as

$$\begin{aligned} \tilde{\mathsf{W}}_{d\times 3m} &= [\Rightarrow_{i=1}^{m} [\Rightarrow_{j=1}^{3} \tilde{\mathsf{h}}_{ij}] \,] = \mathsf{W}[\Rightarrow_{i=1}^{m} [\Rightarrow_{j=1}^{3} (\tilde{\mathsf{f}}_i \otimes \tilde{\mathsf{s}}_j)] \,] \\ &= \mathsf{W}_{d\times 3m}(\tilde{\mathsf{F}}_{m\times m} \otimes \tilde{\mathsf{S}}_{3\times 3}), \end{aligned} \tag{2.26}$$

where $\tilde{\mathsf{F}} \equiv [\Rightarrow_{i=1}^{m} \tilde{\mathsf{f}}_i]$ and $\tilde{\mathsf{S}} \equiv [\Rightarrow_{j=1}^{3} \tilde{\mathsf{s}}_j]$ define the (not necessarily orthogonal) bases for the identity coefficients and the light sources, respectively. Thus, any vector f lies in the linear span of $\tilde{\mathsf{F}}$, i.e., there exists a coefficient vector $\mu = [\Downarrow_{i=1}^{m} \mu_i]$ relating f with $\tilde{\mathsf{F}}$ in the following way:

$$\mathsf{f} = \sum_{i=1}^{m} \mu_i \, \tilde{\mathsf{f}}_i = \tilde{\mathsf{F}}\mu; \tag{2.27}$$

Similarly, for any vector s, there exists $\nu = [\Downarrow_{j=1}^{3} \nu_j]$ such that

$$\mathsf{s} = \sum_{j=1}^{3} \nu_j \, \tilde{\mathsf{s}}_j = \tilde{\mathsf{S}}\nu. \tag{2.28}$$

Substituting (2.27) and (2.28) into (2.7), we have

$$\begin{aligned} \mathsf{h}_{d\times 1} &= \mathsf{W}(\mathsf{f} \otimes \mathsf{s}) = \mathsf{W}((\tilde{\mathsf{F}}\ \mu) \otimes (\tilde{\mathsf{S}}\ \nu)) \\ &= \mathsf{W}(\tilde{\mathsf{F}} \otimes \tilde{\mathsf{S}})(\mu \otimes \nu) \\ &= \tilde{\mathsf{W}}_{d\times 3m}(\mu_{m\times 1} \otimes \nu_{3\times 1}) \end{aligned} \tag{2.29}$$

Therefore, if the bootstrap set $\tilde{\mathsf{W}}$ is given, finding f and s for image h is equivalent to finding μ and ν. Since (2.29) is in a bilinear form, we can compute μ and ν

via the same algorithm described in Section 2.1.3 and employ μ for subsequent recognition task.

The use of the bootstrap set yields an additional benefit. As indicated before, the rank for covering illumination variations in practice exceeds 3. Suppose that this rank is $r > 3$, we can use a bootstrap set of dimension d by rm, i.e. using images for m exemplar objects taken under r exemplar lighting conditions, to improve the recognition performance. Obviously, our separation algorithm can be generalized to handle $\mathtt{s}$ with dimension $r \times 1$. Unfortunately, no bootstrap set can be easily constructed for the second constraint using exemplar images.

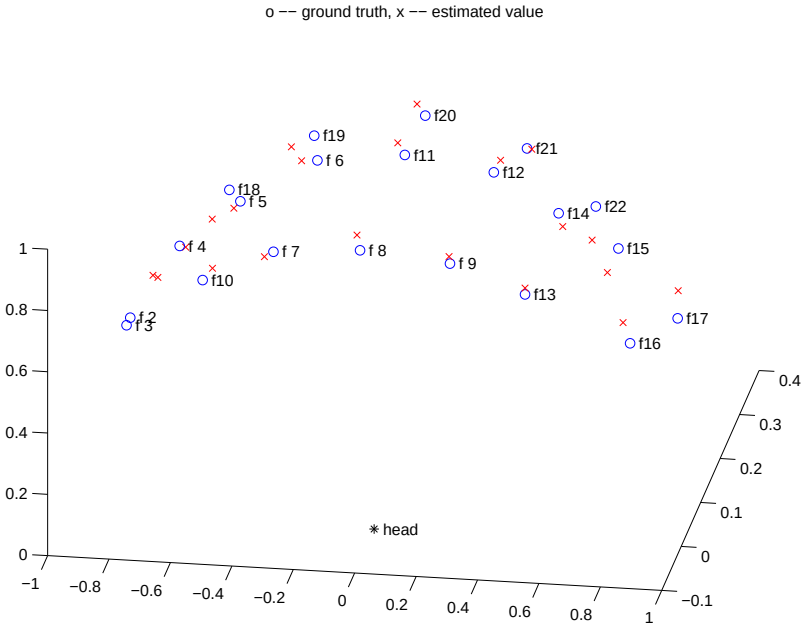


Figure 2.3: Right: Flash distribution in the PIE database. For illustrative purposes, we move their positions on a unit sphere as only the illuminant directions matter. 'o' means the ground truth and 'x' the estimated values.

2.2.3 Recognition experiments

We study an extreme recognition setting with the following features: there is no identity overlap between the training set and the gallery and probe sets; only one image per object is stored in the gallery set; the lighting conditions for the training, gallery and probe sets are completely unknown.

Our strategy is to: (i) Learn W, if needed, from the training set using the recovery algorithm described in Section 2.1.4 or construct a bootstrap set $\tilde{\mathsf{W}}$ for simplicity; (ii) With W (or $\tilde{\mathsf{W}}$) given, learn the identity signature f's (or μ's) for both the gallery and probe sets using the recovery algorithm described in Section 2.1.3, assuming no knowledge of illumination directions; and (iii) Perform recognition using the nearest correlation coefficient. Suppose that a gallery image g has its signature[5] f_g (or μ_g) and a probe image p has its signature f_p (or μ_g), their correlation coefficient is

$$\mathtt{k}(p, g) = (\mathsf{f}_p, \mathsf{f}_g)/\sqrt{(\mathsf{f}_p, \mathsf{f}_p)(\mathsf{f}_g, \mathsf{f}_g)}, \tag{2.30}$$

where (x, y) is an inner-product such as $(\mathsf{x}, \mathsf{y}) = \mathsf{x}^{\mathsf{T}}\Sigma\mathsf{y}$ with Σ learned or given. We use Σ as an identity matrix.

PIE database

We use the Pose and Illumination and Expression (PIE) database [75] in our experiment[6]. Figure 2.3 shows the distribution of all 21 flashes used in PIE and their estimated positions using our algorithm. Since the flashes are almost symmetrically distributed about the head position, we only use 12 of them distributed on

[5]In the sequel, we simply refer as $\mathsf{f} = [\mathsf{f}^{\mathsf{T}}, \mathsf{g}^{\mathsf{T}}]^{\mathsf{T}}$ for the second rank constraint

[6]We use the 'illum' part of the PIE database that is close to obeying the Lambertian model as in [70] while the 'light' part that includes an ambient light is used in [66].

the right half of the unit sphere in Figure 2.3. More specifically, the flashes we used are f08, f09, f11-f17, and f20-f22. In total, we used 68x12=816 images in a fixed view as there are 68 subjects in the PIE database. Figure 2.4 displays one PIE object under the selected 12 illuminants.

Registration is performed by aligning the eyes and mouth to desired positions. No flow computation [66] is carried on for further alignment. After the pre-processing step, the cropped out face image is of size 50 by 50, i.e. $d = 2500$. Also, we only study gray images by taking the average of the red, green, and blue channels of their color versions. We use all 68 images under one illumination to form a gallery set and under another illumination to form a probe set. The training set is taken from sources other than the PIE dataset. Thus, we have 12x11=132 tests, with each test giving rise to a recognition score.

Gallery	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Average
Probe													
f_{08}	-	96	96	87	66	60	46	29	22	85	78	53	65
f_{09}	94	-	96	96	90	87	56	40	24	84	96	68	75
f_{11}	94	91	-	97	72	72	38	28	16	100	94	51	69
f_{12}	88	94	97	-	88	93	57	41	28	94	100	76	78
f_{13}	56	87	59	85	-	100	90	71	50	54	87	100	76
f_{14}	51	85	63	93	100	-	90	66	49	59	91	99	77
f_{15}	33	40	37	49	85	88	-	93	78	32	49	97	62
f_{16}	19	26	26	32	59	44	84	-	93	26	31	63	46
f_{17}	14	28	19	26	50	41	68	94	-	19	26	44	39
f_{20}	90	85	99	97	65	69	38	26	21	-	93	53	67
f_{21}	79	94	93	100	88	94	62	49	28	91	-	76	78
f_{22}	43	65	46	75	99	99	97	76	59	43	74	-	70
Average	60	72	66	76	78	77	66	56	42	63	74	71	67

Table 2.1: Recognition rate obtained by our approach using the first rank constraint and the Yale's database as the training set.

Figure 2.4: The first and second rows display one PIE object under the selected 12 illuminants (from left to right, row 1 to row 2: f08, f09, f11-f17, and f20-f22) and the third and fourth rows one Yale object under 9 lights (most frontal lights) used in the training set.

Recognition across illumination

We first assume that all the images have been captured in a frontal view, but we do not assume that the directions and intensities of the illuminants are known.

[*Yale training set*] The training (or bootstrap) set is first taken as the Yale's illumination database [68]. There are only 10 subjects (i.e. $m = 10$) in this database and each subject has 64 images in frontal view illuminated by 64 different lights. We pick out images under 9 lights (mostly frontal) in order to cover up to second-order harmonic components [95]. Figure 2.3 shows one Yale object under $r = 9$ lights.

Table 2.1 lists the recognition rate for the PIE database using the first rank

Gallery	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Average
Probe													
f_{08}	-	100	90	66	21	9	1	9	4	60	60	1	38
f_{09}	100	-	72	94	59	31	10	24	13	51	84	13	50
f_{11}	97	91	-	100	29	24	13	15	10	100	94	19	54
f_{12}	93	97	100	-	93	90	56	59	35	96	100	69	81
f_{13}	19	62	22	68	-	97	82	100	68	13	84	81	63
f_{14}	9	15	12	62	100	-	100	84	82	12	72	100	59
f_{15}	0	3	1	4	76	100	-	74	76	1	18	100	41
f_{16}	6	25	3	31	82	65	71	-	100	3	41	57	44
f_{17}	4	12	3	31	51	56	81	100	-	3	28	59	39
f_{20}	88	76	100	99	28	28	15	12	16	-	99	19	53
f_{21}	84	97	97	100	96	88	57	74	46	96	-	71	82
f_{22}	3	4	3	13	72	100	100	50	57	3	24	-	39
Average	46	53	46	61	64	62	53	54	46	40	64	54	54

Table 2.2: Recognition rate obtained by the 'Eigenface'approach (discarding the first 3 components) using the Yale's database as the training set.

Gallery	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Average
Probe													
f_{08}	-	97	97	93	63	56	29	16	9	94	85	29	61
f_{09}	99	-	97	99	96	88	38	21	12	91	96	57	72
f_{11}	99	96	-	99	62	63	29	16	12	100	94	41	65
f_{12}	96	99	100	-	93	91	40	22	13	99	100	69	75
f_{13}	74	93	69	84	-	100	71	37	16	62	87	97	72
f_{14}	66	88	74	93	100	-	76	34	19	71	93	100	74
f_{15}	22	34	24	35	71	66	-	82	46	28	44	99	50
f_{16}	12	21	13	18	28	26	74	-	85	18	22	47	33
f_{17}	6	7	9	13	15	18	40	81	-	13	16	24	22
f_{20}	93	88	100	96	63	68	32	19	13	-	96	43	65
f_{21}	87	94	100	100	93	99	51	22	15	99	-	84	77
f_{22}	41	65	43	62	96	100	100	56	29	46	71	-	64
Average	63	71	66	72	71	70	53	37	24	65	73	63	61

Table 2.3: Recognition rate obtained by the 'Fisherface' approach using the Yale's database as the training set.

Gallery	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Average
Probe													
f_{08}	-	100	99	99	97	97	79	72	43	99	97	93	88
f_{09}	100	-	99	99	99	99	97	91	60	97	97	97	94
f_{11}	99	99	-	100	100	100	90	76	65	100	100	99	93
f_{12}	99	99	100	-	100	100	100	93	76	100	100	100	97
f_{13}	99	99	100	100	-	100	100	100	88	99	100	100	99
f_{14}	99	99	100	100	100	-	100	100	96	99	100	100	99
f_{15}	84	94	93	100	100	100	-	100	100	88	100	100	96
f_{16}	69	87	78	90	100	100	100	-	100	69	90	100	89
f_{17}	44	60	51	71	84	91	99	100	-	56	75	94	75
f_{20}	97	97	100	100	100	100	90	74	68	-	100	99	93
f_{21}	97	97	100	100	100	100	100	97	82	100	-	100	98
f_{22}	90	97	96	100	100	100	100	100	99	97	100	-	98
Average	89	93	92	96	98	99	96	91	80	91	96	98	93

Table 2.4: Recognition rate obtained by our approach with the first rank constraint and Vetter's database as the training set.

Gallery	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Average
Probe													
f_{08}	-	100	99	99	97	93	82	59	35	99	97	88	86
f_{09}	100	-	99	99	99	99	91	84	53	99	99	96	92
f_{11}	99	99	-	100	100	100	91	71	44	100	100	94	90
f_{12}	99	99	100	-	100	100	99	90	72	100	100	99	96
f_{13}	99	99	100	100	-	100	99	99	79	99	100	99	97
f_{14}	99	99	100	100	100	-	99	97	87	99	100	99	98
f_{15}	93	96	93	97	99	99	-	100	99	96	99	100	97
f_{16}	75	90	69	93	97	99	100	-	99	69	94	100	89
f_{17}	47	68	51	78	84	90	100	100	-	57	82	94	77
f_{20}	99	99	100	100	99	100	91	76	51	-	100	94	92
f_{21}	99	99	100	100	100	100	99	94	78	100	-	99	97
f_{22}	97	96	96	99	99	99	100	100	90	96	99	-	97
Average	91	94	91	97	97	98	95	88	71	92	97	96	92

Table 2.5: Recognition rate obtained by our approach with the second rank constraint and Vetter's database as the training set.

	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Average
Gly: Front f_{12}, Prb: Front	99	99	100	-	100	100	97	93	78	100	100	99	96
Gly: Front f_{12}, Prb: Side	85	89	88	94	96	96	88	81	68	86	95	91	88
Gly: Side f_{12}, Prb: Front	92	91	99	97	85	87	72	53	33	94	96	87	82
Gly: Side f_{12}, Prb: Side	100	100	100	-	100	100	99	85	63	100	100	100	95
Gly: Side f_{08}, Prb: Side	-	100	100	100	99	97	72	59	35	100	100	90	86
Gly: Side f_{17}, Prb: Side	26	41	37	57	76	84	100	100	-	43	65	91	66
Gly: Side f_{22}, Prb: Side	75	97	88	99	100	100	100	100	100	91	100	-	95

Table 2.6: Recognition rate across poses and illumination. The front view is from camera 27, and the side view from camera 05.

constraint and the Yale's database as the training set. Even with $m = 10$, we obtain quite good results, especially when the gallery and probe sets are close in terms of their flash positions. When the flashes of the gallery and probe sets become separated, the recognition rate decreases. The worst performance is with the gallery set at f08 and the probe set at f17, two most separated flashes. In general, using images under frontal or near-frontal illuminants (e.g. f09, f12, and f21) as gallery sets produces good results.

For comparison, we also implemented the 'Eigenface' approach (discarding the first 3 components) and the 'Fisherface' approach by training the subspace projection vectors from the same training set. The recognition rates are presented in Tables 2.2 and 2.3. The 'Fisherface' approach outperforms the 'Eigenface' approach, but their performances are worse than our approach. This highlights the virtue of decoupling the illumination variations.

[*Vetter training set*] Generalization capacity with $m = 10$ is rather restrictive. We now increase m from 10 to 100 by using Vetter's 3D face database [66]. As this is a 3D database, we actually have W (even p and N) available. However, we believe that using a training set of $m = 100$ from other sources, which to the best of our

knowledge is not available in the literature, can yield similar performances. Table 2.4 tabulates the recognition rates obtained by imposing the first rank constraint. Significant improvements have been achieved by increasing m. This seems to suggest that a moderate sample size of 100 is enough to span the entire face space under a fixed view.

As an interesting comparison, Blanz and Vetter [66] also reported the recognition rates across the illumination variation (with only 'f12' being the gallery set and using the 'light' part of the PIE database) and their average is 98% for color images while ours is 96% for gray images under the first rank constraint. We believe that our performances can be boosted using the color images and finer alignment. Note that our approaches look similar to [66], but there are significant differences. In [66] depths and texture maps of explicit 3D face models are used, while our image-based approach uses the concepts of albedo and surface normal and can recover the 3D models under the first constraint. Also, [66] needs a very good initialization for the lighting source.

We then experiment with the second rank constraint. Note that here we need explicit knowledge of p and N, while under the first constraint we can use a bootstrap set instead. Table 2.5 tabulates the recognition rate obtained. It seems that the use of the second rank constraint does not help much. In fact, it is slightly worse due to possible over-parameterization. In addition, it is difficult to estimate p and N using the second rank constraint. Thus, it seems beneficial to use the first rank constraint in practice.

Recognition across views and illumination

We now present our preliminary results on recognition across poses and illumination. Our approach in principle can also handle pose variation since the W matrix contains all the needed 3D information, i.e., we can recover the 3D model from it. Also as mentioned earlier, learning the W matrix can be avoided by using a bootstrap set. Here, we simply use Vetter's database to handle pose variation. Pose is roughly estimated from the geometric calibration information provided in the PIE database. We then warp the 3D model to the desired pose. The motivation is the following: suppose the pose parameter is θ, then the image h^θ at pose θ can be expressed as

$$\mathsf{h}^\theta = \mathsf{W}^\theta(\mathsf{f} \otimes \mathsf{s}). \tag{2.31}$$

In other words, the illumination-invariant signature f for image h^θ is kept the same if we have the class-specific albedo and shape matrix at pose θ. The rest just follows using the first constraint approach. Table 2.6 lists the recognition results obtained. In general, using the side view still yields quite good recognition result.

Illuminant estimation

In the above process, we achieve illuminant estimation. Figure 2.3 also shows the estimated illuminant directions. It is quite accurate for estimation of directions of flashes near frontal pose. But when the flashes are significantly off-frontal, accuracy slightly goes down.

2.3 Appendix

Appendix 2.I: Recovering multilinear coefficients from h

The algorithm presented in Sec. 2.1.4 can be generalized to recover $\{\mathsf{f}^1, ..., \mathsf{f}^n\}$ from h if the following multilinear form is satisfied:

$$\mathsf{h}_{d\times 1} = \mathsf{W}_{d\times \prod_{i=1}^{n} m_i}(\mathsf{f}^1_{m_1\times 1} \otimes \ldots \otimes \mathsf{f}^n_{m_n\times 1}), \tag{2.32}$$

where $\mathsf{W} \equiv [\Rightarrow_{j_1,\ldots,j_n} \mathsf{w}_{j_1,\ldots,j_n}]$. Again, we impose the addition constraints: $\mathsf{1}^{\mathsf{T}}\mathsf{f}^i = 1;\ i = 1, \ldots, n-1$.

In the iteration for computing f^i given all other f^j's $(j \neq i)$ fixed, we have,

$$\mathsf{h} = \mathsf{A}^i\mathsf{f}^i, \tag{2.33}$$

where $\mathsf{A}^i \equiv [\Rightarrow_{j_i=1}^{m_i} \mathsf{a}^i_{j_i}]$ and

$$\mathsf{a}^i_{j_i} = \sum_{j_1=1}^{m_1} \cdots \sum_{j_n=1}^{m_n} c^1_{j_1} \ldots c^{i-1}_{j_{i-1}} c^{i+1}_{j_{i+1}} \ldots c^n_{j_n} \mathsf{w}_{j_1,\ldots,j_n}. \tag{2.34}$$

If $\mathsf{1}^{\mathsf{T}}\mathsf{f}_i = 1$ is imposed for $i = 1, \ldots, n-1$, the LS solution to f^i is

$$\mathsf{f}^i = \begin{cases} \begin{bmatrix} \mathsf{A}^i \\ \mathsf{1}^{\mathsf{T}} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathsf{h} \\ 1 \end{bmatrix}, & i = 1, \ldots, n-1; \\ [\mathsf{A}^n]^{\dagger}\,\mathsf{h}, & i = n. \end{cases} \tag{2.35}$$

Appendix 2.II: Computing R from H

This appendix concentrates on the most difficult part of recovering the albedos and surface normals from H: updating R with F, S, and τ fixed. We will take vector derivatives of $\mathcal{E}$ with respective to $\{\mathsf{r}_{ij};\ i = a, b, c;\ j = 1, \ldots, m\}$ and treat the three terms in $\mathcal{E}$ separately.

[*About* $\mathcal{E}_0$.] With $\mathsf{f}_{j'} \equiv [\Downarrow_{j=1}^{m} f_{j'j}]$ and $\mathsf{s}_{j'} \equiv [s_{j'a}, s_{j'b}, s_{j'c}]^{\mathsf{T}}$,

$$\begin{aligned}
\frac{\partial \mathcal{E}_0}{\partial \mathsf{r}_{ij}} &= \sum_{j'=1}^{n}\sum_{\mathsf{x}=1}^{d} \tau_{j'(\mathsf{x})}\{\hat{\mathsf{w}}(\mathsf{x})^{\mathsf{T}}\mathsf{R}(\mathsf{f}_{j'} \otimes \mathsf{s}_{j'}) - h_{j'(\mathsf{x})}\}\hat{\mathsf{w}}(\mathsf{x}) f_{j'j} s_{j'i} \\
&= \sum_{j'=1}^{n}\sum_{\mathsf{x}=1}^{d} \tau_{j'(\mathsf{x})}\{\sum_{l=a,b,c}\sum_{k=1}^{m} \hat{\mathsf{w}}(\mathsf{x})^{\mathsf{T}}\mathsf{r}_{lk} f_{j'k} s_{j'l} - h_{j'(\mathsf{x})}\}\hat{\mathsf{w}}(\mathsf{x}) f_{j'j} s_{j'i} \\
&= \sum_{l=a,b,c}\sum_{k=1}^{m}\{\sum_{j'=1}^{n}\sum_{\mathsf{x}=1}^{d} \tau_{j'(\mathsf{x})} f_{j'k} s_{j'l} f_{j'j} s_{j'i} \hat{\mathsf{w}}(\mathsf{x})\hat{\mathsf{w}}(\mathsf{x})^{\mathsf{T}}\}\mathsf{r}_{lk} \\
&\quad - \sum_{j'=1}^{n}\sum_{\mathsf{x}=1}^{d} \tau_{j'(\mathsf{x})} h_{j'(\mathsf{x})} f_{j'j} s_{j'i} \hat{\mathsf{w}}(\mathsf{x}) \\
&= \sum_{l=a,b,c}\sum_{k=1}^{m} \mathsf{O}_{ij}^{lk}\mathsf{r}_{lk} - \gamma_{ij}, \qquad (2.36)
\end{aligned}$$

where $\{\mathsf{O}_{ij}^{lk}; l = a, b, c; k = 1, \ldots, m\}$ are properly defined $3m \times 3m$ matrices, and γ_{ij} is a properly defined $3m \times 1$ vector.

[*About* $\mathcal{E}_1$.] Using forward differences to approximate the partial derivatives[7],

$$\begin{aligned}
&\tfrac{\partial \hat{a}_{j(x,y)}}{\partial y} \simeq \hat{a}_{j(x,y+1)} - \hat{a}_{j(x,y)}; \quad \tfrac{\partial \hat{b}_{j(x,y)}}{\partial x} \simeq \hat{b}_{j(x+1,y)} - \hat{b}_{j(x,y)}; \\
&\tfrac{\partial \hat{c}_{j(x,y)}}{\partial x} \simeq \hat{c}_{j(x+1,y)} - \hat{c}_{j(x,y)}; \quad \tfrac{\partial \hat{c}_{j(x,y)}}{\partial_y} \simeq \hat{c}_{j(x,y+1)} - \hat{c}_{j(x,y)},
\end{aligned} \qquad (2.37)$$

we have

$$\alpha_{j(x,y)} \approx \hat{b}_{j(x+1,y)}\hat{c}_{j(x,y)} - \hat{b}_{j(x,y)}\hat{c}_{j(x+1,y)} + \hat{a}_{j(x,y)}\hat{c}_{j(x,y+1)} - \hat{a}_{j(x,y+1)}\hat{c}_{j(x,y)}. \qquad (2.38)$$

Suppose we are given the product of albedo and surface normal $[a_{j(\mathsf{x})}, b_{j(\mathsf{x})}, c_{j(\mathsf{x})}]$ as in (2.21), we can derive the albedo $p_j(\mathsf{x})$ and surface normals $\hat{a}_{j(\mathsf{x})}$, $\hat{b}_{j(\mathsf{x})}$, and $\hat{c}_{j(\mathsf{x})}$ as follows:

$$p_{j(\mathsf{x})} = \sqrt{(\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}\mathsf{r}_{aj})^2 + (\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}\mathsf{r}_{bj})^2 + (\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}\mathsf{r}_{cj})^2}, \qquad (2.39)$$

$$\hat{a}_{j(\mathsf{x})} = \frac{\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}\mathsf{r}_{aj}}{p_{j(\mathsf{x})}}, \quad \hat{b}_{j(\mathsf{x})} = \frac{\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}\mathsf{r}_{bj}}{p_{j(\mathsf{x})}}, \quad \hat{c}_{j(\mathsf{x})} = \frac{\hat{\mathsf{w}}_{(\mathsf{x})}^{\mathsf{T}}\mathsf{r}_{cj}}{p_{j(\mathsf{x})}}. \qquad (2.40)$$

[7]Partial derivatives of boundary pixels require different approximations. But, similar derivations can be derived.

So, their partial derivatives with respect to r_{aj} are

$$\frac{\partial \hat{a}_{j(\mathsf{X})}}{\partial \mathsf{r}_{aj}} = \frac{\hat{\mathsf{w}}_{(\mathsf{X})}}{p_{j(\mathsf{X})}} - \hat{\mathsf{w}}_{(\mathsf{X})}^{\mathsf{T}} \mathsf{r}_{aj} \frac{\hat{\mathsf{w}}_{(\mathsf{X})} \hat{\mathsf{w}}_{(\mathsf{X})}^{\mathsf{T}}}{p_{j(\mathsf{X})}^3} = \frac{1 - \hat{a}_{j(\mathsf{X})}^2}{p_{j(\mathsf{X})}} \hat{\mathsf{w}}_{(\mathsf{X})}, \tag{2.41}$$

$$\frac{\partial \hat{a}_{j(\mathsf{X})}}{\partial \mathsf{r}_{bj}} = -\hat{\mathsf{w}}_{(\mathsf{X})}^{\mathsf{T}} \mathsf{r}_{aj} \frac{\hat{\mathsf{w}}_{(\mathsf{X})} \hat{\mathsf{w}}_{(\mathsf{X})}^{\mathsf{T}} \mathsf{r}_{bj}}{p_{j(\mathsf{X})}^3} = \frac{-\hat{a}_{j(\mathsf{X})} \hat{b}_{j(\mathsf{X})}}{p_{j(\mathsf{X})}} \hat{\mathsf{w}}_{(\mathsf{X})}, \ \frac{\partial \hat{a}_{j(\mathsf{X})}}{\partial \mathsf{r}_{cj}} = \frac{-\hat{a}_{j(\mathsf{X})} \hat{c}_{j(\mathsf{X})}}{p_{j(\mathsf{X})}} \hat{\mathsf{w}}_{(\mathsf{X})}. \tag{2.42}$$

Similarly, we can derive their partial derivatives with respect to r_{bj} and r_{cj}, which are summarized as follows:

$$\frac{\partial \hat{k}_{j(\mathsf{X})}}{\partial \mathsf{r}_{lj}} = \frac{-\hat{k}_{j(\mathsf{X})} \hat{l}_{j(\mathsf{X})}}{p_{j(\mathsf{X})}} \hat{\mathsf{w}}_{(\mathsf{X})}, \quad \frac{\partial \hat{k}_{j(\mathsf{X})}}{\partial \mathsf{r}_{kj}} = \frac{1 - \hat{k}_{j(\mathsf{X})}^2}{p_{j(\mathsf{X})}} \hat{\mathsf{w}}_{(\mathsf{X})}, \quad k, l \in \{a, b, c\}, \ k \neq l. \tag{2.43}$$

Notice that $\frac{\partial \hat{a}_{j(\mathsf{X})}}{\partial \mathsf{r}_{bj}} = \frac{\partial \hat{b}_{j(\mathsf{X})}}{\partial \mathsf{r}_{aj}}$, $\frac{\partial \hat{a}_{j(\mathsf{X})}}{\partial \mathsf{r}_{cj}} = \frac{\partial \hat{c}_{j(\mathsf{X})}}{\partial \mathsf{r}_{aj}}$, and $\frac{\partial \hat{b}_{j(\mathsf{X})}}{\partial \mathsf{r}_{cj}} = \frac{\partial \hat{c}_{j(\mathsf{X})}}{\partial \mathsf{r}_{bj}}$, which imply saving in computations.

We now compute the partial derivative of $\alpha_{j(x,y)}$ with respect to r_{aj}:

$$\begin{aligned}
\frac{\partial \alpha_{j(x,y)}}{\partial \mathsf{r}_{aj}} &= \frac{\partial}{\partial \mathsf{r}_{aj}} \{\hat{b}_{j(x+1,y)} \hat{c}_{j(x,y)} - \hat{b}_{j(x,y)} \hat{c}_{j(x+1,y)} + \hat{a}_{j(x,y)} \hat{c}_{j(x,y+1)} - \hat{a}_{j(x,y+1)} \hat{c}_{j(x,y)}\} \\
&= \{\frac{\hat{a}_{j(x,y)} \hat{c}_{j(x,y)}}{p_{j(x,y)} p_{j(x,y+1)}} \hat{\mathsf{w}}_{(x,y)} \hat{\mathsf{w}}_{(x,y+1)}^{\mathsf{T}} - \frac{\hat{a}_{j(x,y+1)} \hat{c}_{j(x,y+1)}}{p_{j(x,y)} p_{j(x,y+1)}} \hat{\mathsf{w}}_{(x,y+1)} \hat{\mathsf{w}}_{(x,y)}^{\mathsf{T}}\} \mathsf{r}_{aj} + \\
&\quad \{\frac{\hat{a}_{j(x+1,y)} \hat{c}_{j(x+1,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathsf{w}}_{(x+1,y)} \hat{\mathsf{w}}_{(x,y)}^{\mathsf{T}} - \frac{\hat{a}_{j(x,y)} \hat{c}_{j(x,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathsf{w}}_{(x,y)} \hat{\mathsf{w}}_{(x+1,y)}^{\mathsf{T}}\} \mathsf{r}_{bj} + \\
&\quad \{\frac{\hat{a}_{j(x,y)} \hat{b}_{j(x,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathsf{w}}_{(x,y)} \hat{\mathsf{w}}_{(x+1,y)}^{\mathsf{T}} - \frac{\hat{a}_{j(x+1,y)} \hat{b}_{j(x+1,y)}}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathsf{w}}_{(x+1,y)} \hat{\mathsf{w}}_{(x,y)}^{\mathsf{T}} + \\
&\quad \frac{1 - \hat{a}_{j(x,y)}^2}{p_{j(x,y)} p_{j(x,y+1)}} \hat{\mathsf{w}}_{(x,y)} \hat{\mathsf{w}}_{(x,y+1)}^{\mathsf{T}} - \frac{1 - \hat{a}_{j(x,y+1)}^2}{p_{j(x,y)} p_{j(x+1,y)}} \hat{\mathsf{w}}_{(x,y+1)} \hat{\mathsf{w}}_{(x,y)}^{\mathsf{T}}\} \mathsf{r}_{cj} \\
&= \mathsf{P}_{aj(x,y)}^a \mathsf{r}_{aj} + \mathsf{P}_{aj(x,y)}^b \mathsf{r}_{bj} + \mathsf{P}_{aj(x,y)}^c \mathsf{r}_{cj} = \sum_{l=a,b,c} \mathsf{P}_{aj(x,y)}^l \mathsf{r}_{lj},
\end{aligned} \tag{2.44}$$

where $\mathsf{P}_{aj(x,y)}^a$, $\mathsf{P}_{aj(x,y)}^b$, and $\mathsf{P}_{aj(x,y)}^c$ are properly defined matrices of dimension $3m \times 3m$. By the same token, using properly defined $\mathsf{P}_{bj(x,y)}^a$, $\mathsf{P}_{bj(x,y)}^b$, $\mathsf{P}_{bj(x,y)}^c$, $\mathsf{P}_{cj(x,y)}^a$, $\mathsf{P}_{cj(x,y)}^b$, and $\mathsf{P}_{cj(x,y)}^c$, we can calculate

$$\frac{\partial \alpha_{j(x,y)}}{\partial \mathsf{r}_{ij}} = \sum_{l=a,b,c} \mathsf{P}_{ij(x,y)}^l \mathsf{r}_{lj}; \quad i = a, b, c, \tag{2.45}$$

and, finally,

$$\frac{\partial \mathcal{E}_1}{\partial \mathsf{r}_{ij}} = \sum_{\mathsf{x}=1}^{d} \alpha_{j(\mathsf{x})} \sum_{l=a,b,c} \mathsf{P}^l_{ij(\mathsf{x})} \mathsf{r}_{lj} = \sum_{l=a,b,c} \mathsf{P}^l_{ij} \mathsf{r}_{lj}; \quad \mathsf{P}^l_{ij} \equiv \sum_{\mathsf{x}=1}^{d} \alpha_{j(\mathsf{x})} \mathsf{P}^l_{ij(\mathsf{x})}. \tag{2.46}$$

[*About* $\mathcal{E}_2$.] The symmetry constraint term $\beta_{j(\mathsf{x})}$ defined as in (2.23) can be expressed as

$$\beta^2_{j(\mathsf{x})} = \mathsf{r}^{\mathsf{T}}_{aj} \mathsf{Q}^a_{(\mathsf{x})} \mathsf{r}_{aj} + \mathsf{r}^{\mathsf{T}}_{bj} \mathsf{Q}^b_{(\mathsf{x})} \mathsf{r}_{bj} + \mathsf{r}^{\mathsf{T}}_{cj} \mathsf{Q}^c_{(\mathsf{x})} \mathsf{r}_{cj}, \tag{2.47}$$

where $\mathsf{Q}^a_{(\mathsf{x})}$, $\mathsf{Q}^b_{(\mathsf{x})}$, and $\mathsf{Q}^c_{(\mathsf{x})}$ are symmetric matrices with size $3m \times 3m$:

$$\mathsf{Q}^a_{(\mathsf{x})} = (\hat{\mathsf{w}}_{(\mathsf{x})} + \hat{\mathsf{w}}_{(\bar{\mathsf{x}})})(\hat{\mathsf{w}}_{(\mathsf{x})} + \hat{\mathsf{w}}_{(\mathsf{x})})^{\mathsf{T}}, \mathsf{Q}^b_{(\mathsf{x})} = (\hat{\mathsf{w}}_{(\mathsf{x})} - \hat{\mathsf{w}}_{(\bar{\mathsf{x}})})(\hat{\mathsf{w}}_{(\mathsf{x})} - \hat{\mathsf{w}}_{(\mathsf{x})})^{\mathsf{T}}, \mathsf{Q}^c_{(\mathsf{x})} = \mathsf{Q}^b_{(\mathsf{x})}. \tag{2.48}$$

The derivatives of $\beta^2_{j(\mathsf{x})}/2$ and $\mathcal{E}_2$ with respective to r_{aj}, r_{bj}, and r_{cj} are

$$\frac{\partial \{\beta^2_{j(\mathsf{x})}/2\}}{\partial \mathsf{r}_{ij}} = \mathsf{Q}^i_{(\mathsf{x})} \mathsf{r}_{ij}; \quad \frac{\partial \mathcal{E}_2}{\partial \mathsf{r}_{ij}} = \sum_{\mathsf{x}=1}^{d} \mathsf{Q}^i_{(\mathsf{x})} \mathsf{r}_{ij} = \mathsf{Q}^i \mathsf{r}_{ij}; \quad \mathsf{Q}^i = \sum_{\mathsf{x}=1}^{d} \mathsf{Q}^i_{(\mathsf{x})}. \tag{2.49}$$

Combining the above derivations and using $\frac{\partial \mathcal{E}}{\partial \mathsf{r}_{ij}} = 0$, we have

$$\sum_{l=a,b,c} \sum_{k=1}^{m} \mathsf{O}^{lk}_{ij} \mathsf{r}_{lk} + \lambda_1 \sum_{l=a,b,c} \mathsf{P}^l_{ij} \mathsf{r}_{lj} + \lambda_2 \mathsf{Q}^i \mathsf{r}_{ij} = \gamma_{ij}; i = a, b, c; \ j = 1, \ldots, m. \tag{2.50}$$

We therefore arrive at a set of equations linear in $\{\mathsf{r}_{ij};\ i = a, b, c;\ j = 1, ..., m\}$ that can be solved easily. After finding the new R, we normalize it using $\mathsf{R}{=}\mathsf{R}/||\mathsf{R}||_2$.

Chapter 3

Illuminating Light Field

State-of-the-art algorithms are not able to produce satisfactory recognition performance when confronted by pose and illumination variations. In general, pose variation is slightly more difficult to handle than illumination variation. The presence of both variations further challenges the recognition algorithms.

This chapter extends the generalized photometric stereo algorithm presented in Chapter 2 to handle pose variation. The way we handle pose variation is through the 'Eigen' light approach [69]. This unified approach is image-based, in the sense that, in the training set, only 2D images are used and no explicit 3D models are needed. The unification is achieved by exploiting the fact that both approaches use a subspace model for identity. The 'Eigen' light field approach combines subspace modeling with light field and offers a pose-invariant encoding of identity. The generalized photometric stereo algorithm combines the identity subspace with the illumination model and provides an illumination-invariant description. However, the 'Eigen' light field approach assumes a fixed illumination and cannot handle illumination variations, i.e., its pose-invariant identity encoding is not invariant to variations in illumination. The generalized photometric stereo algorithm assumes a

fixed pose and cannot easily handle pose variations, i.e., its illumination-invariant identity description is not invariant to variations in pose. This motivates our integrated approach for handling both pose and illumination variations using an illumination- and pose-invariant identity signature.

Chapter organization

Section 3.1 presents the principle of the illuminating light field approach. It starts by reviewing in Section 3.1.1 the related literature, then describes Section 3.1.2 the 'Eigen' light field approach [69] that performs FR under pose variations, and finally introduces in Section 3.1.3 our integrated approach. Section 3.1.4 presents algorithms for recovering the identity signature that is invariant to illumination and pose. Section 3.2 gives our experimental results on the PIE database [75] and comparisons with other approaches.

3.1 Principle of Illuminating Light Field

3.1.1 Literature review

Identity, illumination, and pose

Three factors are involved in face recognition, namely illumination, pose, and identity. Using the human face images as examples, we now address issues involved in each of the three factors by fixing the other two.

- *Illumination.* Various illumination models are available in the literature, ranging from models for highly specular objects such as mirrors to models for matte objects. Mostly objects belong to the latter category, which is described by a Lambertian reflectance model for its simplicity. Early shape

from shading approaches [10] assumed a constant albedo field. However, this assumption is violated at locations such as eyes and mouth edges. For the human face, the Lambertian reflectance model with a varying albedo field provides a reasonable approximation [68, 74, 95, 103, 204]. The Phong illumination model also has found application [66]. This proposed method adopts the Lambertian reflectance model with a varying albedo field to model the effect of illumination.

- *Pose.* The issue of pose essentially amounts to a correspondence problem. If dense correspondences across poses are available and if a Lambertian reflectance model is further assumed, a rank-1 constraint is implied because theoretically, a 3D model can be recovered and used to render novel poses. However, recovering a 3D model from 2D images is a difficult task. There are two types of approaches: model-based and image-based. Model-based approaches [66, 139, 145, 146] require explicit knowledge of prior 3D models, while image-based approaches [125, 129, 142, 143, 144] do not use prior 3D models. In general, model-based approaches [66, 139, 145, 146] register the 2D face image to 3D models that are given beforehand. In [139, 146], a generative face model is deformed through bundle adjustment to fit 2D images. In [145], a generative face model is used to regularize the 3D model recovered using the SfM algorithm. In [66], 3D morphable models are constructed based on many prior 3D models. There are mainly three types of image-based approaches: Structure from motion (SfM) [125, 129], visual hull [142, 144], and light field rendering [143, 140] methods. The SfM approach [125] works with sparse correspondence and does not reliably recover the 3D model amenable for practical use. The visual hull methods [142, 144] assume

that the shape of the object is convex, which is not always satisfied by the human face, and also require accurate calibration information. The light field rendering methods [143, 140] relax the requirement of calibration by a fine quantization of the pose space and recover a novel view by sampling the captured data that form the so-called light field. The proposed method is image-based, so no prior 3D models are used. It handles a given set of views through an analysis analogous to the light field concept. However, no novel poses are rendered.

- *Identity.* One straightforward method to describe the identity is through discrete labels. However, using this discrete description it is impossible to establish a link between objects used in the training and testing stages in terms of the identity. An alternative way is to associate a discrete label with a continuous-valued variable, which is regarded as an identity signature. One good example is to use subspace encoding [47, 62], where linear generalization is assumed to incorporate the fact that all human faces are similar. Once the subspace basis are learned from the training set, they are used to characterize the gallery/probe set, thus enabling the required generalization capability. In this chapter, we also use the subspace method to describe the identity.

Face recognition under illumination variation

FR under illumination variation must take into account the two factors of identity and illumination. Refer to Section 2.2.1 in Chapter 2 for a review of related work.

Face recognition under pose variation

As mentioned earlier, pose variation essentially amounts to a correspondence problem. If dense correspondences across poses are available and a Lambertian reflectance is assumed, then a rank-1 constraint is implied. Unfortunately, finding correspondences is a very difficult task and, therefore there exist no subspace based on an appearance representation when confronted with pose variation. Approaches to face recognition under pose variation [68, 69, 72] avoid the correspondence problem by sampling the continuous pose space into a set of poses, *v.i.z.* storing multiple images at different poses for each person at least in the training set. In [72], view-based 'Eigenfaces' are learned from the training set and used for recognition. In [68], a denser sampling is used to cover the pose space. However, as [68] uses object-specific images, appearances belonging to a novel object (i.e. not in the training set) cannot be handled. In [69], the concept of light field [143] is used to characterize the continuous pose space. 'Eigen' light fields are learnt from the training set. However, the implementation of [69] still discretizes the pose space and recognition can be based on probe images at poses in the discretized set. One should note that the light field is not related to variation in illumination.

Face recognition under illumination and pose variations

Approaches to handling both illumination and pose variations include [66, 70, 77, 78, 202]. The approach [66] uses morphable 3D models to characterize the human faces. Both geometry and texture are linearly spanned by those of the training ensemble consisting of 3D prior models. It is able to handle both illumination and pose variations. Its only weakness is a complicated fitting algorithm. Recently, a fitting algorithm more efficient than suggested in [66] is proposed in [73]. In [70],

the Fisher light field is proposed to handle both illumination and pose variations, where the light field is used to cover the pose variation and the Fisher discriminant analysis to cover the illumination variation. Since discriminant analysis is just a statistical analysis tool which minimizes the within-class scatter while maximizing the between-class clatter and has no relationship with any physical illumination model, it is questionable that discriminant analysis is able to generalize to new lighting conditions. Instead, this generalization may be inferior because discriminant analysis tends to overly tune to the lighting conditions in the training set. The 'Tensorface' approach [77, 78] uses a multilinear analysis to handle various factors such as identity, illumination, pose, and expression. The factors of identity and illumination are suitable for linear analysis, as evidenced by the 'Eigenface' approach (assuming a fixed illumination and a fixed pose) and the subspace induced by the Lambertian model, respectively. However, the factor of expression is arguably amenable for linear analysis and the factor of pose is not amenable for linear analysis. In [202], preliminary results are reported by first warping the albedo and surface normal fields at the desired pose and then carrying on recognition as usual.

3.1.2 Pose-invariant identity signature

The light field measures the radiance in free space (free of occluders) as a 4D function of position and direction. An image is a 2D slice of the 4D light field. If the space is only 2D, the light field is then a 2D function. This is illustrated in Figure 3.1 (also see [69] for another illustration), where a camera conceptually moves along a circle, within which a square object with four differently colored sides resides. The 2D light field L is a function of θ and ϕ as properly defined

in Figure 3.1. The image of the 2D object is just a vertical line. If the camera is allowed to leave the circle, then a curve is traced out in the light field to form the image, i.e. the light field is accordingly sampled. Even though the light field for a 3D object is a 4D function, we still use the notation $\mathsf{L}(\theta, \phi)$ for the sake of simplification.

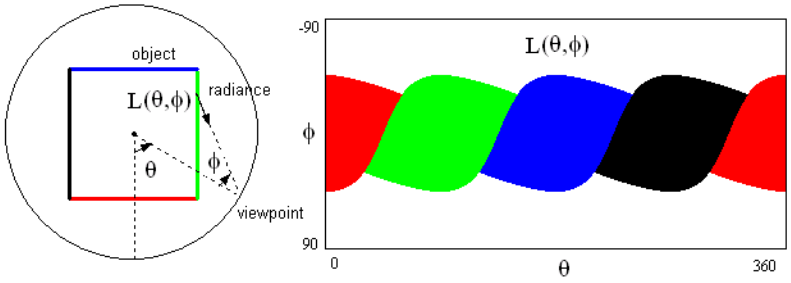


Figure 3.1: This figure illustrates the 2D light-field of a 2D object (a square with four differently colored sides), which is placed within an circle. The angles θ and ϕ are used to relate the viewpoint with the radiance from the object. The right image shows the actual light field for the square object.

Starting from the light fields $\{\mathsf{L}_n(\theta, \phi);\ n = 1, ..., N\}$ of the training samples, the 'Eigen' light field approach conducts a PCA to find the eigenvectors $\{e_i(\theta, \phi);\ i = 1, ..., m\}$ which span a rank-m subspace. The 'Eigen' light field [69] is again motivated by the similarity among the human faces. Using the fact [47, 62] that: If $\mathsf{Y}^\mathsf{T}\mathsf{Y}$ has an eigenpair (λ, v), then $\mathsf{Y}\mathsf{Y}^\mathsf{T}$ has a corresponding eigenpair $(\lambda, \mathsf{Y}\mathsf{v})$, we know that $e_i(\theta, \phi)$ is just a linear combination of the $\mathsf{L}_n(\theta, \phi)$'s, i.e., there exist a_{in}'s such that

$$e_i(\theta, \phi) = \sum_n a_{in} \mathsf{L}_n(\theta, \phi). \tag{3.1}$$

For an arbitrary subject, its light field $\mathsf{L}(\theta, \phi)$ lies in this rank-m subspace. In

other words, there exists coefficients f_i's such that, $\forall(\theta, \phi)$,

$$\mathsf{L}(\theta, \phi) = \sum_{i=1}^{m} f_i e_i(\theta, \phi) = \mathsf{e}(\theta, \phi)^{\mathsf{T}}\mathsf{f}, \tag{3.2}$$

where $\mathsf{e}(\theta, \phi) \equiv [\Downarrow_{i=1}^{m}\ e_i(\theta, \phi)]_{m\times 1}$ and $\mathsf{f} = [\Downarrow_{i=1}^{m}\ f_i]_{m\times 1}$.

As mentioned earlier, to obtain an image h^v at a particular pose v (a collection of d pixels) one should sample the light field. Suppose that one pixel h^v is the point sample of the light field associated with the coordinate (θ^v, ϕ^v), i.e.,

$$h^v = \mathsf{L}(\theta^v, \phi^v). \tag{3.3}$$

The image h^v can be expressed as

$$\mathsf{h}^v \equiv [\Downarrow_{i=1}^{d}\ h_i^v] = [\Downarrow_{i=1}^{d}\ \mathsf{L}(\theta_i^v, \phi_i^v)], \tag{3.4}$$

where (θ_i^v, ϕ_i^v) is the corresponding coordinate in the light field for the pixel h_i^v. Substituting (3.2) into (3.4) yields

$$\mathsf{h}^v = [\Downarrow_{i=1}^{d}\ \mathsf{e}(\theta_i^v, \phi_i^v)^{\mathsf{T}}]\mathsf{f} = \mathsf{E}^v\mathsf{f}, \tag{3.5}$$

where $\mathsf{E}^v \equiv [\Downarrow_{i=1}^{d}\ \mathsf{e}(\theta_i^v, \phi_i^v)^{\mathsf{T}}]_{d\times m}$.

Eq. (3.5) has an important implication: f is a pose-invariant identity signature because the pose information is encoded in E^v. This is summarized in Proposition 3.1.

Proposition 3.1: *The identity signature f as derived in (3.5) is pose-invariant.*

Constructing a light field is a practically difficult task. However, if only some specific poses are of interest with each pose sampling a subset of the light field, we can only focus on the portion of the light field that is equivalent to the union of these subsets. Suppose that the K poses are of interest are $\{v_1, ..., v_K\}$ and the corresponding images at these poses are $\{\mathsf{h}^{v_1}, \ldots, \mathsf{h}^{v_K}\}$ with h^{v_k} expressed as in

(3.4), the portion of the light field of focus is nothing but $[\Downarrow_{k=1}^{K}\ [\Downarrow_{i=1}^{d}\ \mathsf{L}(\theta_i^{v_k}, \phi_i^{v_k})]\ ]$, which is a 'long' $Kd \times 1$ vector obtained by stacking all the images at all these poses. The introduction of such a 'long' vector eases our computation: (i) If we are interested in a particular view v, we just simply take out those rows corresponding to this view. (ii) In this context, computing the 'Eigen' light field is equivalent to performing PCA on the ensemble consisting of a collection of such 'long' vectors.

The concept of light field was introduced in the computer graphics literature [143]. A strict assumption is that the scene be static. While characterizing the appearances of one object at given views using the concept of light field is legitimate, generalizing this to many objects is questionable since the lights fields belonging to different objects are not in correspondence, i.e. they are not shape-free in the terminology of [49, 76]. The mismatch in correspondence arises from differences in head sizes and locations in world coordinator system of different objects, and so on. Typically, correspondences between different objects are established using face normalization or registration is performed. Unfortunately, the normalization step ruins the static scene requirement in the light field theory. On the other hand, as argued in [49, 76], since the shape-free appearance is amenable for linear analysis, we can pursue PCA on the shape-free vector L, similar to the 'Eigen' light field approach [69]. This point is illustrated in [71]. Following [71], we also use the term light field in a loose sense.

3.1.3 Illumination- and pose-invariant identity signature

As mentioned earlier and in [143], the underlying assumption about the concept of light is one of fixed illumination. We now consider the light fields formed under varying illumination, i.e., illuminating the light field.

Clearly, the light field under a fixed illumination s, $\mathsf{L}^s(\theta, \phi)$, follows the Lambertian reflectance model:

$$\mathsf{L}^s(\theta, \phi) = \mathsf{t}(\theta, \phi)^\mathsf{T}\mathsf{s}, \tag{3.6}$$

where $\mathsf{t}(\theta, \phi)$ is the product of the albedo and the surface normal at a proper pixel and does not depend on s. Combining (3.1) and (3.6) yields the 'Eigen' light field $e_i^s(\theta, \phi)$ under the illumination s as,

$$e_i^s(\theta, \phi) = \sum_n a_{in}\mathsf{t}_n(\theta, \phi)^\mathsf{T}\mathsf{s} = \mathsf{t}_{ei}(\theta, \phi)^\mathsf{T}\mathsf{s}, \tag{3.7}$$

where $\mathsf{t}_{ei}(\theta, \phi) \equiv \sum_n a_{in}\mathsf{t}_n(\theta, \phi)$. Eq. (3.2) then becomes

$$\mathsf{L}^s(\theta, \phi) = [\Downarrow_{i=1}^m \mathsf{t}_{ei}(\theta, \phi)^\mathsf{T}\mathsf{s}]^\mathsf{T}\mathsf{f} = \mathsf{W}(\theta, \phi)(\mathsf{f} \otimes \mathsf{s}), \tag{3.8}$$

where $\mathsf{W}(\theta, \phi) \equiv [\Rightarrow_{i=1}^m \mathsf{t}_{ei}(\theta, \phi)]_{1\times 3m}$ does not depend on s. This successfully leads to a two-factor analysis [138, 187].

A pixel h^{vs} under a pose v and an illumination s is a point sample of the light field $\mathsf{L}^s(\theta, \phi)$ at coordinate (θ^v, ϕ^v), i.e.,

$$h^{vs} = \mathsf{L}^s(\theta^v, \phi^v) = \mathsf{W}(\theta^v, \phi^v)(\mathsf{f} \otimes \mathsf{s}), \tag{3.9}$$

and an image h^{vs} under the pose v and illumination s, which traces a set of d samples of the light field under illumination s, is

$$\mathsf{h}^{vs} = [\Downarrow_{i=1}^d h_i^{vs}] = [\Downarrow_{i=1}^d \mathsf{W}(\theta_i^v, \phi_i^v)](\mathsf{f} \otimes \mathsf{s}) = \mathsf{W}^v(\theta, \phi)(\mathsf{f} \otimes \mathsf{s}), \tag{3.10}$$

where $\mathsf{W}^v(\theta, \phi) \equiv [\Downarrow_{i=1}^d \mathsf{W}(\theta_i^v, \phi_i^v)]_{d\times 3m}$. Eq. (3.10) has an important implication: The coefficient vector f provides an identity signature invariant to both pose and illumination because the pose is absorbed in $\mathsf{W}^v(\theta, \phi)$ and the illumination is absorbed in s.

Proposition 3.2: *The identity signature* f *as derived in (3.10) is illumination and pose-invariant.*

The remaining questions are how to learn the basis matrix $\mathsf{W}(\theta, \phi)$ from a given training ensemble and how to compute the blending coefficient vector f as well as s for an arbitrary image h^{vs}. The next section presents the algorithms in detail.

3.1.4 Learning algorithms

Learning the basis matrix $\mathsf{W}(\theta, \phi)$

Suppose that the training ensemble is given as $\{\mathsf{L}_n^s(\theta, \phi);\ n = 1, ...N,\ s = 1, ..., S\}$, where $\mathsf{L}_n^s(\theta, \phi)$ is the light field of the n^{th} training object under illumination s (a $Kd \times 1$ vector as explained in Section 3.1.2). Learning $\mathsf{W}(\theta, \phi)$ (a $Kd \times mr$ matrix where m is the rank for the identity and r is the rank for the illumination) from the training ensemble is detailed in [138] and is further extended in [187] by imposing the integrability constraint. The main difference between [138] and [187] is the following: In [138], the recovered $\mathsf{W}(\theta, \phi)$ minimizes the approximation error in the mean square sense and not necessarily satisfies the integrability constraint. In other words, the hypothetical base objects in $\mathsf{W}(\theta, \phi)$ is not integrable. In [187], the recovered $\mathsf{W}(\theta, \phi)$ minimizes the above approximation error as well as a cost function invoked by violating the integrability constraint. As a consequence, [138] can only process the image ensemble consisting of different objects under the same set of illumination (e.g. the case considered here) while [187] can process the image ensemble consisting of different objects under completely different illumination. Here, we follow the approach in [138] to derive $\mathsf{W}(\theta, \phi)$ for simplicity. The basic underlying principle is to use a two-fold SVD algorithm that is reviewed below.

The following two matrices (A-type and B-type) are first constructed by group-

ing the 'long' vectors $\{\mathsf{L}_n^s(\theta,\phi);\ n=1,...N,\ s=1,...,S\}$ in two ways:

$$\mathsf{A}=[\Downarrow_{n=1}^{N}\ [\Rightarrow_{s=1}^{S}\ \mathsf{L}_n^s(\theta,\phi)]\ ],\quad \mathsf{B}=[\Downarrow_{s=1}^{S}\ [\Rightarrow_{n=1}^{N}\ \mathsf{L}_n^s(\theta,\phi)]\ ], \tag{3.11}$$

where A is a $KNd\times S$ matrix whose rows stack together the light fields of different identities under the same illumination and whose columns correspond to different illumination and B is a $KSd\times N$ matrix whose rows stack together the light fields under different illumination for the same identity and whose columns correspond to different identities. It is obvious that we can convert from an A-type matrix to B-type and *vice versa*.

We perform the SVD for the A matrix as $\mathsf{A}=\mathsf{U}_\mathsf{A}\mathsf{D}_\mathsf{A}\mathsf{V}_\mathsf{A}^\mathsf{T}$ and keep the top r rows of the column basis $\mathsf{V}_\mathsf{A}^\mathsf{T}$ for the illumination, denoted by S. We do a similar thing to the B matrix and keep the top m rows of the column basis $\mathsf{V}_\mathsf{B}^\mathsf{T}$ for the identities, denoted by F. Direct SVD of the A and B matrices are numerically inefficient or even prohibitive since they are extremely 'tall'. Also it is unnecessary to compute U and D as we are interested only in the V part of the SVD result. For computational savings, we observe that V_A encodes the eigenvectors of $\mathsf{A}^\mathsf{T}\mathsf{A}=\mathsf{V}_\mathsf{A}\mathsf{D}_\mathsf{A}^2\mathsf{V}_\mathsf{A}^\mathsf{T}$. Since the size of $\mathsf{A}^\mathsf{T}\mathsf{A}$ is only $S\times S$, computing its eigenvalues is numerically stable. Therefore, we simply first compute $\mathsf{A}^\mathsf{T}\mathsf{A}$ and then perform its 'Eigen' decomposition to find V_A. Similarly, we can compute V_B.

We now have the matrices S and F at our disposal. To find $\mathsf{W}(\theta,\phi)$, we first compute $\mathsf{A}'=\mathsf{A}\mathsf{S}^\mathsf{T}$, where A' is a $KNd\times r$ matrix. Notice that A' is still an A-type matrix, so we can convert A' to a B-type matrix B' following the strategy described in (3.11), where B' is a $Krd\times N$ matrix. Thirdly, we compute $\mathsf{W}'=\mathsf{B}'\mathsf{F}^\mathsf{T}$, where W' is a $Krd\times m$ matrix. The rest is to group W' to form a $Kd\times mr$ matrix W.

Recovering the blending coefficient vector f from an image

Given $\mathsf{W}(\theta,\phi) = [\Rightarrow_{i=1}^{m} [\Rightarrow_{j=1}^{r} \mathsf{W}_{ij}(\theta,\phi)]\,]_{Kd\times mr}$, where $\mathsf{W}_{ij}(\theta,\phi)$ denotes the $((i-1)*r+j)^{th}$ column of the $\mathsf{W}(\theta,\phi)$ matrix, computing f and s for an arbitrary image h^{vs} utilizes (3.10) iteratively [187]. Notice that we need only the portion of $\mathsf{W}(\theta,\phi)$ corresponding to the pose v, denoted by $\mathsf{W}^{v}(\theta,\phi) = [\Rightarrow_{i=1}^{m} [\Rightarrow_{j=1}^{r} \mathsf{W}_{ij}^{v}(\theta,\phi)]\,]_{d\times mr}$.

If f is fixed, (3.10) is linear in s and its least square (LS) solution is

$$\mathsf{s} = [\Rightarrow_{j=1}^{r} ([\Rightarrow_{i=1}^{m} \mathsf{W}_{ij}^{v}(\theta,\phi)]\mathsf{f})\,]^{\dagger}\mathsf{h}^{vs}, \tag{3.12}$$

where $[.]^{\dagger}$ is a matrix psuedo-inverse; if s is fixed, (3.10) is linear in f and its LS solution is

$$\mathsf{f} = \begin{bmatrix} [\Rightarrow_{i=1}^{m} ([\Rightarrow_{j=1}^{r} \mathsf{W}_{ij}^{v}(\theta,\phi)]\mathsf{s})\,] \\ \mathsf{1}^{\mathsf{T}} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathsf{h}^{vs} \\ 1 \end{bmatrix}, \tag{3.13}$$

where $\mathsf{1}$ is a vector of 1's. To obtain (3.13), we also impose $\mathsf{f}^{\mathsf{T}}\mathsf{1} = 1$ to normalize the solution to the same range, which facilitates the recognition task. We iterate this process until convergence. Meanwhile, we can also take into account the pixels in shadows as in [187].

Recovering the blending coefficient vector f from a group of images

This iterative algorithm can be easily modified to handle a group of Q images $\{\mathsf{h}^{v_1 s_1}, \ldots, \mathsf{h}^{v_Q s_Q}\}$ having the same f but different s's since multiple equations like (3.10) can be formulated. To be specific, we have the following iterative equations:

$$\mathsf{s}_q = [\Rightarrow_{j=1}^{r} ([\Rightarrow_{i=1}^{m} \mathsf{W}_{ij}^{v_q}(\theta,\phi)]\mathsf{f})\,]^{\dagger}\mathsf{h}^{v_q s_q}; \quad q = 1,2,\ldots,Q, \tag{3.14}$$

$$\mathsf{f} = \begin{bmatrix} [\Downarrow_{q=1}^{Q} [\Rightarrow_{i=1}^{m} ([\Rightarrow_{j=1}^{r} \mathsf{W}_{ij}^{v_q}(\theta,\phi)]\mathsf{s}_q)\,]\,] \\ \mathsf{1}^{\mathsf{T}} \end{bmatrix}^{\dagger} \begin{bmatrix} [\Downarrow_{q=1}^{Q} \mathsf{h}^{v_q s_q}] \\ 1 \end{bmatrix}. \tag{3.15}$$

In practice, using a group of images yields a robust estimate for f.

The present of shadow pixels affects the learning algorithm. Handling shadows can be performed in the same fashion as in Chapter 2.

3.2 Face Recognition across Illumination and Poses

3.2.1 PIE database and recognition setting

We use the 'illum' subset of the PIE database [75] in our experiments. This subset has 68 subjects under 21 illumination and 13 poses. Out of 21 illumination configuration, we select 12 denoted by $F = \{f_{16}, f_{15}, f_{13}, f_{21}, f_{12}, f_{11}, f_{08}, f_{06}, f_{10}, f_{18}, f_{04}, f_{02}\}$ as in [70], which typically span the set of variations. Out of the 13 poses, we select 9 denoted by $C = \{c_{22}, c_{02}, c_{37}, c_{05}, c_{27}, c_{29}, c_{11}, c_{14}, c_{34}\}$, which cover from the left profile to the right profile. In total, we have 68*12*9=7344 images. Figure 3.2 displays one PIE object under illumination and pose variations.

Registration is performed by aligning the eyes and mouth to desired positions. No flow computation is carried on for further alignment. After the pre-processing step, the used face image is of size 48 by 40, i.e. $d = 1920$. Also, we only use gray scale images by taking the average of the red, green, and blue channels of their color versions. We believe that our recognition rates can be boosted by using color images and finer registrations. Figure 3.2 shows some examples of the face images actually used in recognition.

We randomly divide the 68 subjects into two parts. The first 34 subjects are used in the training set and the remaining 34 subjects are used in the gallery and probe sets. It is guaranteed that there is no identity overlap between the training set and the gallery and probe sets. To form the light field, we use images at all available poses. Since the illumination model has generalization capability, we

Figure 3.2: Examples of the face images of one PIE object (used in the testing stage) under selected illumination and poses .

can select a minimum of 3 illumination in the training set. In our experiments, the training set includes only 9 selected illumination to cover the second-order harmonic components [95]. Notice that this is not possible in the Fisher light field approach [70] that exhausts all illumination configurations.

The images belonging to the remaining 34 subjects are used in the gallery and probe sets. The construction of the gallery and probe sets conforms to the following two scenarios: (A) We use all the 34 images under one illumination s_p and one pose v_p to form a gallery set and under the other illumination s_g and the other pose

v_g to form a probe set. There are three cases of interest: *same pose but different illumination*, *different pose but same illumination*, and *different pose and different illumination*. We mainly concentrate on the third case with $s_p \neq s_g$ and $v_p \neq v_g$. Also our approach reduces to the 'Eigen' light field approach [69] if $s_p = s_g$ and to the generalized photometric stereo approach [187] if $v_p = v_g$. Thus, we have $(9 * 12)^2 - (9 * 12) = 11,556$ tests, with each test giving rise to a recognition score. (B) We divide C into three sets: $C_1 = \{c_{22}, c_{02}, c_{37}\}$ (left-profile views), $C_2 = \{c_{05}, c_{27}, c_{29}\}$ (frontal views), and $C_3 = \{c_{11}, c_{14}, c_{34}\}$ (right-profile views) and F into 3 sets: $F_1 = \{f_{16}, f_{15}, f_{13}, f_{21}\}$ (left lights), $F_2 = \{f_{12}, f_{11}, f_{08}, f_{06}\}$ (frontal lights), and $F_3 = \{f_{10}, f_{18}, f_{04}, f_{02}\}$ (right lights). For each of the thirty four subjects, the gallery set contains all twelve images under the illumination in F_g and the poses in C_g and the probe set all twelve images under the illumination in F_p and the poses in C_g. We make sure that $(C_p, F_P) \neq (C_g, F_g)$. Thus, we have $(3 * 3)^2 - (3 * 3) = 72$ tests in this scenario that has no counterpart in the Fisher light field [70]. To make the recognition more difficult, we assume that the lighting conditions for the training, gallery and probe sets are completely unknown when recovering the identity signatures.

The testing strategy is similar to that described in Chapter 2.

1. Learn W from the training set using the bilinear learning algorithm [138, 204]. Figure 3.3 shows the W matrix obtained using the training set.

2. With W given, learn the identity signature f's (as well as s's) for all gallery and probe elements (an element is an image in Scenario A and a group of images in Scenario B) using the iterative algorithms in Section 3.1.4. Learning f and s from one single image takes about 1-2 seconds in a Matlab implementation. Figure 3.4 shows the reconstructed images using the learned f and s.

3. Perform recognition using the nearest correlation coefficient.

Gallery	f_{16}	f_{15}	f_{13}	f_{21}	f_{12}	f_{11}	f_{08}	f_{06}	f_{10}	f_{18}	f_{04}	f_{02}	Average
Probe													
c_{22}	56	41	62	68	71	71	53	65	41	44	38	21	52
c_{02}	71	76	76	91	88	94	94	94	85	71	50	32	77
c_{37}	79	82	82	94	94	97	94	94	76	65	65	50	81
c_{05}	68	85	97	100	100	97	97	97	91	82	71	44	86
c_{27}	94	100	100	100	100	–	100	100	100	97	94	76	97
c_{29}	74	82	91	100	100	100	97	97	94	91	88	65	90
c_{11}	50	53	68	79	85	97	97	88	79	82	71	62	76
c_{14}	15	24	44	71	76	82	74	82	82	74	79	56	63
c_{34}	18	18	47	50	56	65	62	56	44	44	41	38	45
Average	58	62	74	84	86	88	85	86	77	72	66	49	74

Table 3.1: Recognition rates for all the probe sets with a fixed gallery set (c_{27},f_{11}).

3.2.2 Recognition performance

Scenario A

Table 3.1 shows the recognition results for all probe sets with a fixed gallery set (c_{27},f_{11}), whose gallery images are in a frontal pose and under a frontal illumination. Using this table we compare the three cases. The case of same pose but different illumination has an average rate 97% (i.e. the average of all 11 cells on the row c_{27}), the case of different pose but same illumination has an average rate 88% (i.e. the average of all 8 cells on the column f_{11}), the case of different pose and different illumination has an average rate 70% (i.e. the average of all 88 cells excluding the row c_{27} and the column f_{11}). This shows that illumination variation is easier to handle than pose illumination and variations in both pose and illumination are the most difficult to deal with.

Gallery	f_{16}	f_{15}	f_{13}	f_{21}	f_{12}	f_{11}	f_{08}	f_{06}	f_{10}	f_{18}	f_{04}	f_{02}	Average
Probe													
c_{22}	44	44	46	45	46	49	46	49	44	32	30	14	41
c_{02}	55	58	59	62	63	62	60	60	54	48	40	22	54
c_{37}	56	59	61	64	65	62	60	58	51	47	45	34	55
c_{05}	56	63	66	67	68	65	59	58	54	51	45	36	57
c_{27}	62	66	69	70	70	70	65	69	68	67	65	54	66
c_{29}	46	53	53	61	60	63	59	62	66	68	62	60	60
c_{11}	41	43	50	53	55	61	57	58	56	61	58	51	54
c_{14}	19	24	39	49	53	58	58	61	60	61	57	48	49
c_{34}	16	21	38	44	46	51	48	51	46	45	45	42	41
Average	44	48	53	57	59	60	57	59	56	53	50	40	53

Table 3.2: Average recognition rates for all the gallery sets. For each cell, say the gallery set at ($v_g = c_{27}$,$s_g = f_{12}$), the average rate is taken over all probe sets (v_p,s_p) where $v_p \neq v_g$ and $s_p \neq s_g$. For example, the average rate for (c_{27},f_{11}) is the average of the rates in Table 3.1 excluding the row c_{27} and the column f_{11}.

We now focus on the case of different pose and different illumination. For each gallery set, we average the recognition scores of all the probe sets with both pose and illumination different from the gallery set. Table 3.2 shows the average recognition rates for all the gallery sets. As an interesting comparison, the 'grand' average is 53% (the last cell in Table 3.2) while that of the Fisher light field approach [70] is 36%. In general, when the poses and illumination of the gallery and probe sets become far apart, the recognition rates decrease. The best gallery sets for recognition are those in frontal poses and under frontal illumination and the worst gallery sets are those in profile views and off-frontal illumination. As shown in Figures 1.5 and 3.2, the worst gallery sets consist of face images almost invisible (See for example the images (c_{22}, f_{02}), (c_{34}, f_{16}), etc.), on which recognition can be hardly performed.

Figure 3.5 presents the curves of the average recognition rates (i.e. the last

columns and last rows of Tables 3.1 and 3.2) across poses and illumination. Clearly the effect of illumination variations is not as strong as due to pose variations in the sense that the curves of average recognition rates across illumination are flatter than those across poses. Figure 3.5 also shows the curves of the average recognition rates obtained based on the top 3 and top 5 matches. Using more matches increases the recognition rates significantly, which demonstrates the efficiency of our recognition scheme. For comparison, Figure 3.5 also plots the average rates obtained using the baseline PCA. These rates are well below ours. The 'grand' average is below 10% if the top 1 match is used.

Figure 3.3: The first nine columns of the learned W matrix.

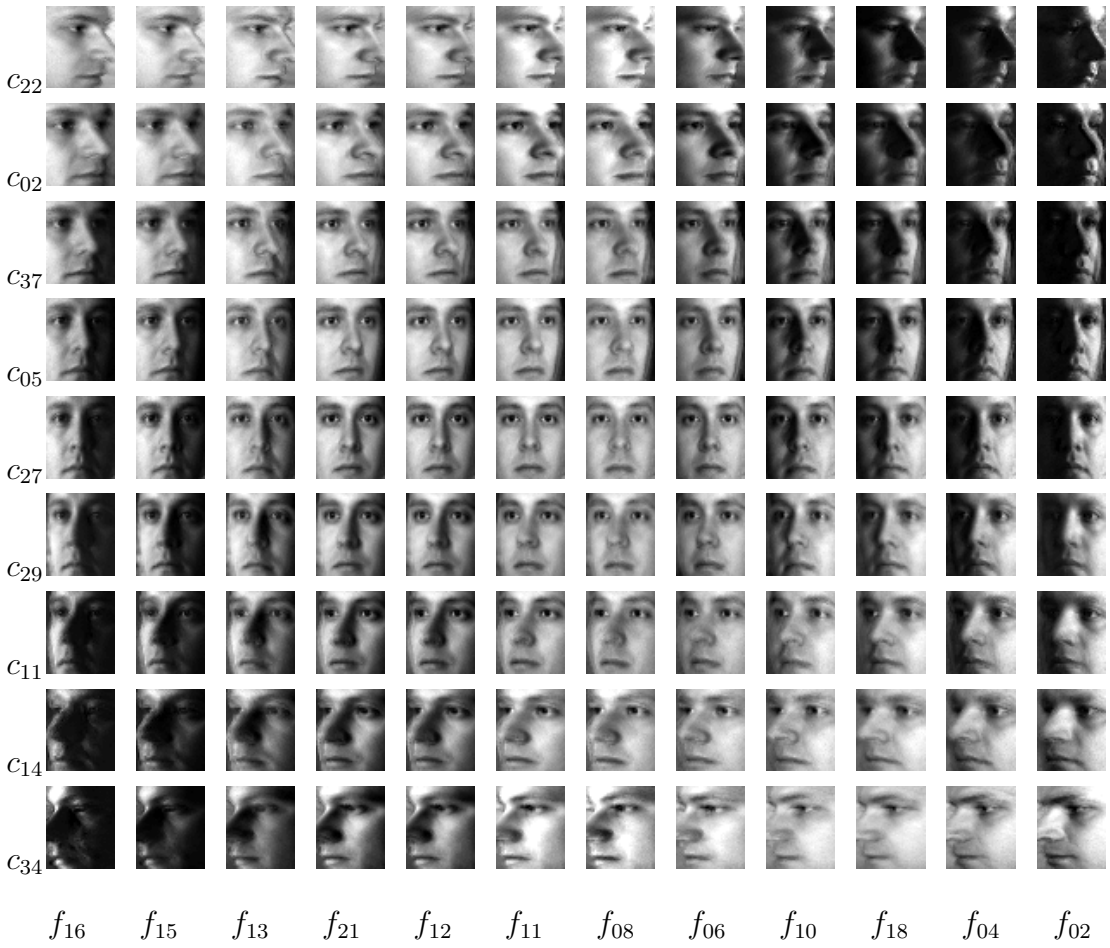
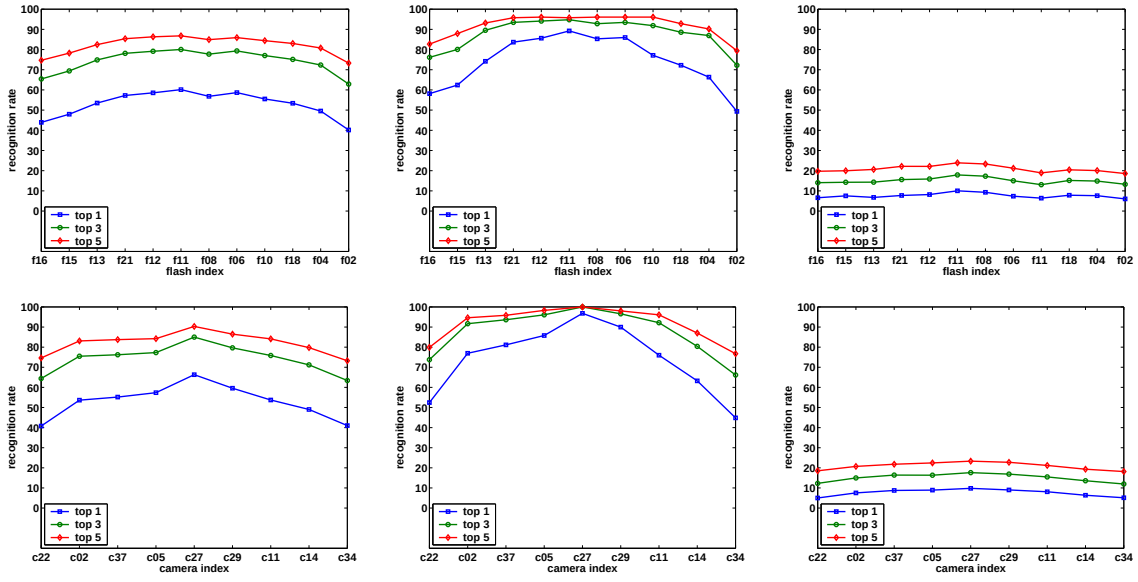


Figure 3.4: The reconstruction results of the object in Figure 3.2. Notice that only the f's and s's for the row c_{27} are used for reconstructing all the images.

Scenario B

This test scenario is designed for face recognition based on a group of images, which can be under different poses and different illumination. Table 3.3 lists the recognition rates, which are much higher than those in Tables 3.1 and 3.2. Also, similar observations can be made regarding the effects of illumination and pose variations.



(a) (b) (c)

Figure 3.5: The average recognition rates across illumination (the top row) and across poses (the bottom row) for three cases. Case (a) shows the average recognition rate (averaging over all illumination/poses and all gallery sets) obtained by the proposed algorithm using the top n matches. Case (b) shows the average recognition rate (averaging over all illumination/poses for the gallery set (c_{27}, f_{11}) only) obtained by the proposed algorithm using the top n matches. Case(c) shows the average recognition rate (averaging over all illumination/poses and all gallery sets) obtained by the 'Eigenface' algorithm using the top n matches.

3.2.3 Comparisons

Comparison with the Fisher light field

It is interesting to compare the proposed approach with the Fisher light field [70] since both of them handle pose variation in a similar fashion. The main difference lies in handling the illumination variation. Our approach uses the Lambertian model while [70] uses Fisher discriminant analysis. Therefore, our approach can

Gallery	C_1F_1	C_1F_2	C_1F_3	C_2F_1	C_2F_2	C_2F_3	C_3F_1	C_3F_2	C_3F_3	Average
Probe										
C_1F_1	–	100	85	100	94	82	62	85	94	88
C_1F_2	100	–	100	100	100	85	71	82	94	92
C_1F_3	85	97	–	88	88	91	76	62	65	82
C_2F_1	97	94	71	–	100	85	71	85	76	85
C_2F_2	97	100	85	100	–	100	76	91	85	92
C_2F_3	79	82	76	97	100	–	74	88	91	86
C_3F_2	59	59	68	85	76	71	–	100	82	75
C_3F_2	74	85	62	91	94	82	100	–	100	86
C_3F_3	88	82	62	79	79	94	85	100	–	84
Average	85	88	76	93	92	86	77	87	86	85

Table 3.3: The recognition rates for test scenario B.

generalize to novel illumination and [70] does not have such a generalization. Also, in Section 3.2 the proposed approach leads to a new recognition scenario which is not available in [70].

Comparison with the 3D morphable model

The 3D morphable model (3DMM) [66] is the state-of-the-art approach to identify faces across illumination and poses. The proposed approach differs from the 3DMM approach mainly as follows:

- *Model-based v.s. image-based.* The 3DMM approach is requires prior 3D models while the proposed approach that is image-based needs only 2D images.

  Linear assumptions are used in both approaches. The operating units in the 3DMM approach are 3D depth and texture, respectively, and two independent linear models are assumed in both units. The operating unit in the proposed approach is the product of the albedo and surface normal and a

single linear model is assumed. As in the 3DMM approach, it seems that the dimensionality of the proposed model can be 'decomposed' as the product (or the addition) of the dimensionality of the surface normals and that of the albedo field. However, empirically analysis shows [202] that such a decomposition is not necessary and might overfit the problem, thereby indicating that a subspace of rather low dimensionality can be used.

- *Handling illumination.* The Lambertian model is used in the proposed algorithm and pixels in shadows and specular reflection regions are inferred and excluded for consideration. The 3DMM approach uses the standard Phong model to directly model diffuse and specular reflection on the face surface.

  The 3DMM also takes into account inputs illuminated by colored lights using color transformation while the proposed approach only processes inputs illuminated by white lights.

- *Handling pose.* The 3DMM approach can handle images at any pose, while the current implementation of the proposed approach can handle images sampled from a given set of poses. In order to handle arbitrary pose other than those listed in the given set, the system should incorporate a tool to render novel poses using given poses, which is left for future.

  In the proposed approach, pixels at different poses might correspond to the same point in the physical 3D model. In the 3DMM approach, one point is only represented once for all the poses since the 3D model is used.

- *Experiments* Both the 3DMM and the proposed approaches conducted experiments using the PIE database. However, different portions of the PIE database are used. The 3DMM approach worked on the 'lights' part, where

an ambient light source is always present. The proposed approach worked on the 'illum' part with no ambient light source. As a consequence, some images appear almost dark (refer to Figure 3) and there is little hope to perform correct recognition based on these extreme images, explaining the relatively low recognition rates compared with those produced by the 3DMM approach.

In terms of computational complexity, the proposed algorithm is more computationally efficient than the 3DMM approach. The proposed fitting algorithm, taking 1-2 seconds to process one input image using Matlab implementation, is simply linear (rather bilinear) and has a unique minimum; while the 3DMM approach, taking 4.5 minutes to process one input image, invokes a gradient descent algorithm that does not guarantee a global minimum. Also, the proposed algorithm is able to handle face images of very small size. In the reported experiments, gray-level images are normalized to size of 48×40. The size of color images used in the 3DMM approach is unclear, but typically much larger.

Part II: Face Recognition via Kernel Learning

Chapter 4

Probabilistic Kernel Principal Component Analysis

Principal component analysis [12] is one of the most popular statistical data analysis techniques with applications in numerous areas such as data compression, image processing, computer vision, and pattern recognition, to name a few. However, the PCA has two disadvantages: (i) it lacks a probabilistic model structure which is important in many contexts such as mixture modeling and Bayesian decision (also see [167]); and (ii) it restricts itself to a linear setting, where high-order statistical information is discarded [181].

Probabilistic principal component analysis (PPCA) proposed by Tipping and Bishop [167, 168] overcomes the first disadvantage. By letting the noise component possess an isotropic structure, the PCA is implicitly embedded in a parameter learning stage for this model using the maximum likelihood estimation (MLE) method. An efficient expectation/maximization (EM) algorithm [152] is also developed to iteratively learn the parameters.

Kernel principal component analysis (KPCA) proposed by Schölkopf, Smola

and Müller [181] overcomes the second disadvantage by using the so-called 'kernel trick'. The essential idea of the KPCA is to avoid the direct evaluation of the required dot product in a high-dimensional feature space using the kernel function. The feature space is called reproducing kernel Hilbert space (RKHS). Hence, no explicit nonlinear mapping function projecting the data from the original space to the feature space is needed. Since a nonlinear function is used, albeit in an implicit fashion, high-order statistical information is captured. See [179] for a recent survey on the kernel space and application on discovering pre-image and denoised pattern in the original space.

We propose an approach to analyze kernel principal components in a probabilistic manner. It naturally unifies PPCA and KPCA in one treatment to overcome the both disadvantages of PCA. We call it the probabilistic kernel principal component analysis (PKPCA). In this chapter, we present our development of the PKPCA approach by treating the KPCA as a special case of PCA where the number of samples is smaller than the data dimension. One speciality of KPCA is the data centering issue, which is also taken into account in Section 4.2.

While the kernel part retains the nonlinear modeling power, resulting in a smaller reconstruction error, the additional probabilistic structure offers us (i) a mixture modeling capacity of PKPCA, and (ii) an efficient classification scheme.

Mixture of PKPCA is derived to model the nonlinear structure containing nonlinear substructures in a systematic way. Mixture of PKPCA nontrivially extends to the feature space induced by the kernel function, the theory of mixture of PPCA proposed by Tipping and Bishop [167, 168]. An EM algorithm [152] is also developed to iteratively but efficiently learn the parameters of interest. We also show how to compute two important quantities, namely the reconstruction error and

the Mahalanobis distance.

Our analysis can be easily incorporated for a classification task. Our performances are competitive to those produced by the mainstream kernel classifiers, such as the support vector machine (SVM) and kernel Fisher discrimination (KFD) classifier, but our analysis provides more regularized approximation to the data structure.

Chapter organization

Section 4.1 briefly reviews the essentials of RKHS. Section 4.2 presents how to compute the kernel principal components and to analyze these components in a probabilistic manner. Section 4.3 presents the mixture of PKPCA and Section 4.4 presents the classification results on synthetic data and in a face recognition application.

Two examples

Figure 4.1 shows two examples of nonlinear data structures [1] to be modeled. Figure 4.1(a) presents the first example: a C-shaped structure in the foreground. In the context of data modeling, we consider only the foreground and assume a uniform distribution within the C-shaped region and zeros outside. Figure 4.1(b) displays 200 sample points drawn from this density. In the context of pattern classification, we consider both the foreground and the background and further assume that the background class possess a uniform distribution outside the C-shaped region and zeros inside. Figure 4.1(c) shows the samples for the background class.

[1] This means that, if conventional linear modeling techniques such as linear PCA are used, the responses are badly approximated.

Figure 4.1(d) shows the second example where the foreground nonlinear data structure consists of two C-shaped substructures. Figures 4.1(e) and 4.1(f) present the drawn samples for the foreground and background classes, respectively. We mainly use this example for mixture modeling.

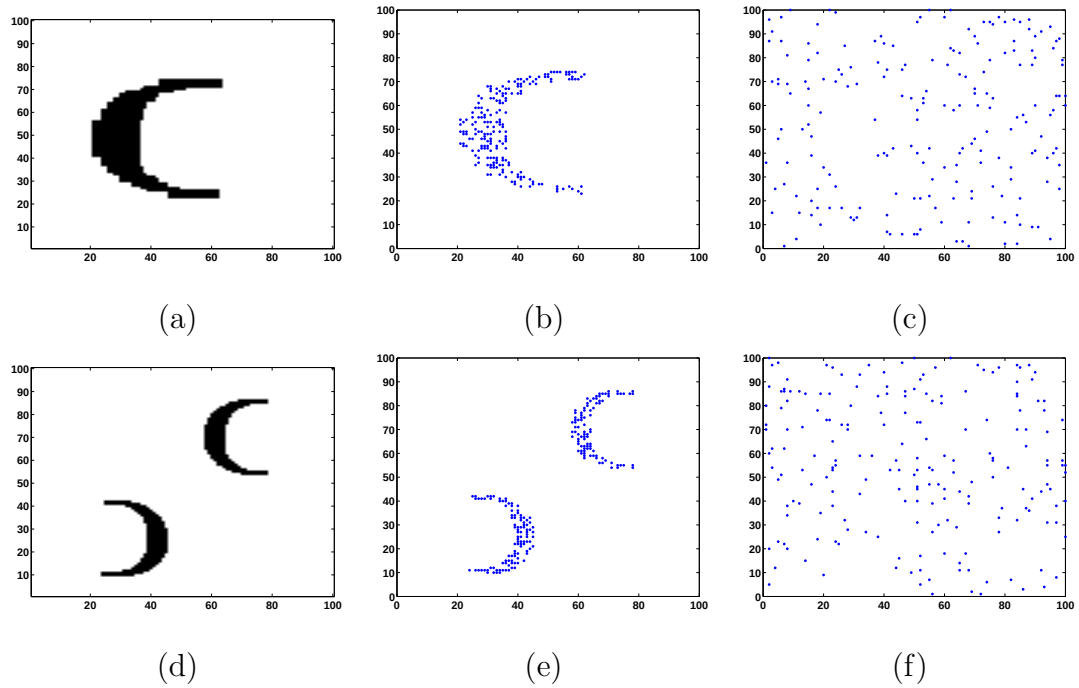


Figure 4.1: Two nonlinear data structures (a)(d) and their drawn samples (of size 200) for the foreground class (b)(e) and the background (c)(f).

4.1 Reproducing Kernel Hilbert Space (RKHS)

We illustrate the principle of the RKHS by drawing an analogy of the RKHS, a functional space, to a regular vector space $\mathcal{R}^d$. We start by a $d \times d$ positive definite matrix $\mathsf{T} = [t_i(j)]$, where $t_i(j)$ is its $(i,j)^{th}$ element. By denoting the i^{th} column by $\mathsf{t}_i = [t_i(1), \ldots, t_i(d)]^{\mathsf{T}}$, we have $\mathsf{T} = [\mathsf{t}_1, \mathsf{t}_2, \ldots, \mathsf{t}_d]$. The eigen-decomposition of

T is given as

$$\mathsf{T} = \sum_{n=1}^{d} \nu_n \xi_n \xi_n^{\mathsf{T}}; \quad \nu_n > 0,$$

where (ν_n, ξ_n)'s are eigenpairs.

We define an inner product between two elements a and b in $\mathcal{R}^d$ as

$$\begin{aligned} < \mathsf{a}, \mathsf{b} > &\equiv \mathsf{a}^{\mathsf{T}} \mathsf{T}^{-1} \mathsf{b} = \sum_{n=1}^{d} \nu_n^{-1} \mathsf{a}^{\mathsf{T}} \xi_n \xi_n^{\mathsf{T}} \mathsf{b} \\ &= \sum_{n=1}^{d} \nu_n^{-1} (\mathsf{a}, \xi_n)(\mathsf{b}, \xi_n), \end{aligned}$$

where $(\mathsf{u}, \mathsf{v}) \equiv \mathsf{u}^{\mathsf{T}} \mathsf{v}$.

Suppose that $\mathsf{g} = [g(1), g(2), \ldots, g(d)]^{\mathsf{T}} \in \mathcal{R}^d$ and the identity matrix $\mathtt{I}_d$ is written as $\mathtt{I}_d = [\mathsf{e}_1, \mathsf{e}_2, \ldots, \mathsf{e}_d]$ where e_i is the i^{th} column of the $\mathtt{I}_d$ matrix. The inner product $< ., . >$ possesses two important properties:

$$P1: \quad < \mathsf{t}_i, \mathsf{t}_j > = \mathsf{t}_i^{\mathsf{T}} \mathsf{T}^{-1} \mathsf{t}_j = \mathsf{t}_i^{\mathsf{T}} \mathsf{e}_j = t_i(j)$$

$$P2: \quad < \mathsf{t}_i, \mathsf{g} > = \mathsf{t}_i^{\mathsf{T}} \mathsf{T}^{-1} \mathsf{g} = \mathsf{e}_i^{\mathsf{T}} \mathsf{g} = g(i)$$

The RKHS, denoted by $\mathcal{H}$, can be heuristically thought of as an f-dimensional 'vector space' $\mathcal{R}^f$ (f might be finite or infinite) associated with a positive kernel function $k_{\mathsf{x}}(\mathsf{y}) = k(\mathsf{x}, \mathsf{y})$. The existence of such kernel functions is guaranteed by the Mercer's Theorem [176] and the eigensystem of $k(\mathsf{x}, \mathsf{y})$ is given as

$$k(\mathsf{x}, \mathsf{y}) = \sum_{n=1}^{f} \nu_n \xi_n(\mathsf{x}) \xi_n(\mathsf{y}); \quad \nu_n > 0; \quad \sum_{n=1}^{f} \nu_n^2 < \infty. \tag{4.1}$$

Similarly, the inner product is defined as, with $\mathsf{a}(\mathsf{x})$, $\mathsf{b}(\mathsf{x})$, and $\mathsf{g}(\mathsf{x})$ in $\mathcal{H}$,

$$< \mathsf{a}, \mathsf{b} >_{\mathcal{H}} \equiv \sum_{n=1}^{f} \nu_n^{-1} (\mathsf{a}, \xi_n)(\mathsf{b}, \xi_n),$$

where $(u, v) \equiv \int_{\mathsf{X}} u(\mathsf{x}) v(\mathsf{x}) \mathsf{dx}$. Furthermore, the two properties known as reproducing properties hold too.

$$P1: \quad < k_{\mathsf{x}}, k_{\mathsf{y}} >_{\mathcal{H}} = k_{\mathsf{x}}(\mathsf{y}),$$

$$P2: \quad < k_{\mathsf{x}}, \mathsf{g} >_{\mathcal{H}} = \mathsf{g}(\mathsf{x}).$$

An alternative perspective to view Eq. (4.1) is to consider a hypothetical nonlinear mapping $\phi : \mathcal{R}^d \rightarrow \mathcal{R}^f$ defined as

$$\phi(\mathsf{x}) = [\nu_1^{1/2}(\mathsf{x}, \xi_1), \ldots, \nu_f^{1/2}(\mathsf{x}, \xi_f)]^{\mathsf{T}}.$$

It is easy to verify that

$$\phi(\mathsf{x})^{\mathsf{T}}\phi(\mathsf{y}) = k(\mathsf{x}, \mathsf{y}) =< k_{\mathsf{x}}, k_{\mathsf{y}} >_{\mathcal{H}} .$$

Thus evaluating the dot product can be easily done by computing $k(\mathsf{x}, \mathsf{y})$ which usually takes a parametric form. This is so-called 'kernel trick', which plays an essential role in many kernel methods, such as SVM [19] and KPCA [181], kernel Fisher discriminant analysis [177, 172], and kernel independent component analysis [170]. In this chapter, we also adopt this viewpoint.

There are a lot of ways to construct a kernel function: see [17] for a list. One example of $k(\mathsf{x}, \mathsf{y})$ is the radial basis function (RBF) kernel which is widely studied in the literature and the focus of this chapter. It is defined as

$$k(\mathsf{x}, \mathsf{y}) = \exp(-\frac{1}{2\sigma^2}\|\mathsf{x} - \mathsf{y}\|^2) \quad \forall \mathsf{x}, \mathsf{y} \in \mathcal{R}^d,$$

where σ controls the kernel width. This is an infinite-dimensional RKHS, i.e., $f = \infty$.

The RBF kernel is a special example of translation-invariant kernels of the form $k(\mathsf{x}, \mathsf{y}) = k(\mathsf{x} - \mathsf{y})$ whose characteristics can be easily described using Fourier theory [173]. In particular, the functions in the RKHS exhibit smoothness since their Fourier transforms decay rapidly.

4.2 Probabilistic Analysis of Kernel Principal Components

4.2.1 Kernel principal component analysis

Suppose that $\{\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_N\}$ are the given training samples in the original data space $\mathcal{R}^d$. KPCA operates in a feature space that is in fact a RKHS $\mathcal{H}_k$ induced by a kernel function k. There exists a hypothetical nonlinear mapping function $\phi : \mathcal{R}^d \to \mathcal{R}^f$, where $f > d$ and f could even be infinite. The training samples in $\mathcal{R}^f$ are denoted by $\Phi_{f\times N} = [\phi_1, \phi_2, ..., \phi_N]$, where $\phi_n \equiv \phi(\mathsf{x}_n) \in \mathcal{R}^f$. Denote the sample mean in the feature space as

$$\bar{\phi_0} \equiv \frac{1}{N}\sum_{n=1}^{N}\phi(\mathsf{x}_n) = \Phi\mathsf{e}, \tag{4.2}$$

where $\mathsf{e}_{N\times 1} = N^{-1}\mathsf{1}$.

The $f \times f$ covariance matrix in the feature space denoted by Σ is given as

$$\Sigma \equiv \frac{1}{N}\sum_{n=1}^{N}(\phi_n - \bar{\phi_0})(\phi_n - \bar{\phi_0})^{\mathsf{T}} = \Phi\mathsf{J}\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}} = \Psi\Psi^{\mathsf{T}}, \tag{4.3}$$

where

$$\mathsf{J} \equiv N^{-1/2}(\mathtt{I}_N - \mathsf{e}\mathsf{1}^{\mathsf{T}}), \quad \Psi \equiv \Phi\mathsf{J}.$$

KPCA performs eigen-decomposition of the covariance matrix Σ in the feature space. Due to the high dimensionality of the feature space, we often have insufficient number of samples, i.e., the rank of the Σ matrix is maximally N instead of f. However, computing the eigensystem is still possible using the method presented in [47, 62].

The explicit knowledge of the nonlinear feature mapping can be avoided using the 'kernel trick' as in Section 4.1. Define

$$\bar{\mathsf{K}} \equiv \Psi^{\mathsf{T}}\Psi = \mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}}\Phi\mathsf{J} = \mathsf{J}^{\mathsf{T}}\mathsf{K}\mathsf{J}, \tag{4.4}$$

where

$$\mathsf{K} \equiv \Phi^{\mathsf{T}}\Phi$$

is the Gram matrix or the dot product matrix. The $(i,j)^{th}$ entry of the Gram matrix K can be calculated as follows:

$$\mathsf{K}^{ij} = \phi(\mathsf{x}_i)^{\mathsf{T}}\phi(\mathsf{x}_j) = k(\mathsf{x}_i, \mathsf{x}_j).$$

As in Appendix 4.I and [47, 62], the eigensystem for Σ can be derived from $\tilde{\mathsf{K}}$. Suppose that the top r eigenpairs for $\bar{\mathsf{K}}$ are $\{(\lambda_n, \mathsf{v}_n)\}_{n=1}^{q}$, where λ_n's are sorted in a non-increasing order, and the r top eigenpairs for Σ are $\{(\lambda_n, \mathsf{u}_n)\}_{n=1}^{q}$, then we can compute u_n as

$$\mathsf{u}_n = (\lambda_n)^{-1/2}\Psi\mathsf{v}_n.$$

In a matrix form (if only the top q eigenvectors are retained),

$$\mathsf{U}_q \equiv [\mathsf{u}_1, \mathsf{u}_2, ..., \mathsf{u}_q] = \Psi\mathsf{V}_q\Lambda_q^{-1/2} = \Phi\mathsf{J}\mathsf{V}_q\Lambda_q^{-1/2}, \tag{4.5}$$

where $\mathsf{V}_q \equiv [\mathsf{v}_1, \mathsf{v}_2, ..., \mathsf{v}_q]$ and $\Lambda_q \equiv \mathtt{D}[\lambda_1, \lambda_2, ..., \lambda_1]$, a diagonal matrix whose diagonal elements are $\{\lambda_1, \lambda_2, ..., \lambda_q\}$.

It is clear that we are not operating in the full feature space, but in a low-dimensional subspace of it, which is spanned by the training samples. It seems that the modeling capacity is limited by subspace dimensionality, or by the number of the samples. In reality, it however turns out that even in this subspace, the smallest eigenvalues are very close to zero, which means that the full feature space can be further captured by a subspace with an even-lower dimensionality. This motivates us to use a latent model.

4.2.2 Theory of PKPCA

Probabilistic analysis assumes that the data in the feature space follows a special factor analysis model [15] which relates an f-dimensional data $\phi(\mathsf{x})$ to a latent q-dimensional variable z as

$$\phi(\mathsf{x}) = \mu + \mathsf{W}\mathsf{z} + \epsilon,$$

where $\mathsf{z} \sim \mathtt{N}(0, \mathtt{I}_q)$, $\epsilon \sim \mathtt{N}(0, \rho\mathtt{I}_f)$, and W is a $f \times q$ *loading matrix*. Therefore, $\phi(\mathsf{x}) \sim \mathtt{N}(\mu, \mathsf{S})$, where

$$\mathsf{S} = \mathsf{W}\mathsf{W}^{\mathsf{T}} + \rho\mathtt{I}_f.$$

Typically, we have $q << N << f$.

As shown in [167, 168], the MLE's for μ and W, denoted by $\hat{\mu}$ and $\hat{\mathsf{W}}$, respectively, are given by

$$\hat{\mu} = \bar{\phi}_0 = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathsf{x}_n) = \Phi\mathsf{e}, \tag{4.6}$$

$$\hat{\mathsf{W}} = \mathsf{U}_q(\Lambda_q - \rho\mathtt{I}_q)^{1/2}\mathsf{R}, \tag{4.7}$$

where R is any $q \times q$ orthogonal matrix, i.e., $\mathsf{R}^{\mathsf{T}}\mathsf{R} = \mathsf{R}\mathsf{R}^{\mathsf{T}} = \mathtt{I}_q$, and U_q and Λ_q contain the top q eigenvectors and eigenvalues of the Σ matrix. It is in this sense that our probabilistic analysis coincides with the plain KPCA.

Substituting (4.5) into (4.7), we obtain the following:

$$\hat{\mathsf{W}} = \Psi\mathsf{V}_q\Lambda_q^{-1/2}(\Lambda_q - \rho\mathtt{I}_q)^{1/2}\mathsf{R} = \Psi\mathsf{Q} = \Phi\mathsf{J}\mathsf{Q}, \tag{4.8}$$

where the $N \times q$ matrix Q is defined as

$$\mathsf{Q} \equiv \mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R}. \tag{4.9}$$

Equation (4.8) has a very important implication: $\hat{\mathsf{W}}$ lies in a linear subspace of Φ. We name the Q matrix as *empirical loading matrix* since this relates the loading

matrix to the empirical data. Also since the matrix $(\mathtt{I}_q - \rho\Lambda_q^{-1})$ in (4.9) is diagonal, additional savings in computing its square root are realized.

The MLE for ρ, $\hat{\rho}$, is given [167, 168] as

$$\hat{\rho} = \frac{1}{f-q}\{\mathtt{tr}(\mathsf{S}) - \mathtt{tr}(\Lambda_q)\}. \tag{4.10}$$

Assuming that the remaining eigenvalues are zero, (this is a reasonable assumption supported by empirical evidences when f is finite), it is approximated as

$$\hat{\rho} \simeq \frac{1}{f-q}\{\mathtt{tr}(\mathsf{K}) - \mathtt{tr}(\Lambda_q)\}. \tag{4.11}$$

But when f is infinite, this is doubtful since this always gives $\hat{\rho} = 0$. In such a case, there is no automatic way of learning this. We temporarily set a manual choice for $\hat{\rho}$. as in [182]. However, as shown later on, we can in fact study the limiting case by letting $\hat{\rho}$ approach zero in various cases. Even when a fixed $\hat{\rho}$ is used, the optimal estimate for W (or $\hat{\mathsf{W}}$) is still the same as in (4.8). It is interesting to note that Moghaddam and Pentland [54] derived (4.10) in a different context by minimizing the Kullback-Leibler divergence distance [4, 13].

Now, the covariance matrix is estimated by

$$\hat{\mathsf{S}} = \Phi\mathsf{J}\mathsf{Q}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}} + \hat{\rho}\mathtt{I}_f = \Phi\mathsf{A}\Phi^{\mathsf{T}} + \hat{\rho}\mathtt{I}_f,$$

where

$$\mathsf{A} \equiv \mathsf{J}\mathsf{Q}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}.$$

This offers a regularized approximation to $\Sigma = \Phi\mathsf{J}\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}}$. In ridge regression [9], the form of $\mathsf{S}_1 = \Phi\mathsf{J}\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}} + \rho\mathtt{I}_f$ (with rho a pre-specified small positive number) is used to provide a regularized approximation. This has a smoothness interpretation of the regression parameters. However, the eigenvalues of S_1 always increase those of Σ by an amount of ρ but the eigenvectors of the S_1 are the same as those of Σ.

Although S is in a compact form and also regularized, inversion of the S_1 matrix involves inverting an $N \times N$ matrix, which is still prohibitive in real applications with a large N, whereas $\hat{\mathsf{S}}^{-1}$ involves inverting only a $r \times r$ M matrix (defined later). This form of S_1 is also used in [170, 171] for estimating the canonical correlation and [175] for constructing the Bhattacharyya kernel.

In [182] the covariance matrix Σ is approximated as $\mathsf{S}_2 = \Phi \mathsf{J} \mathsf{D} \mathsf{J}^{\mathsf{T}} \Phi^{\mathsf{T}} + \rho \mathtt{I}_f$, where D is a diagonal matrix whose many diagonal entries empirically shown to be zero. This is not surprising as in our computation $\mathsf{D} = \mathsf{Q}\mathsf{Q}^{\mathsf{T}}$ is rank deficient. However, we do not enforce D to be diagonal.

Inverting $\hat{\mathsf{S}}$ is also easy by invoking the Woodbury formula [8],

$$\hat{\mathsf{S}}^{-1} = (\hat{\rho}\mathtt{I}_f + \hat{\mathsf{W}}\hat{\mathsf{W}}^{\mathsf{T}})^{-1} = \hat{\rho}^{-1}(\mathtt{I}_f - \hat{\mathsf{W}}\mathsf{M}^{-1}\hat{\mathsf{W}}^{\mathsf{T}}) = \hat{\rho}^{-1}(\mathtt{I}_f - \Phi \mathsf{B} \Phi^{\mathsf{T}}),$$

where

$$\mathsf{B} \equiv \mathsf{J}\mathsf{Q}\mathsf{M}^{-1}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}},$$

and the matrix $\mathsf{M}_{r \times r}$ can be thought of as a 'reciprocal' matrix for $\hat{\mathsf{S}}$,

$$\mathsf{M} \equiv \hat{\rho}\mathtt{I}_q + \hat{\mathsf{W}}^{\mathsf{T}}\hat{\mathsf{W}} = \hat{\rho}\mathtt{I}_q + \mathsf{L}, \tag{4.12}$$

with

$$\mathsf{L} \equiv \mathsf{Q}^{\mathsf{T}}\bar{\mathsf{K}}\mathsf{Q}.$$

Using the Q matrix in 4.9, Appendix 4.II calculates various quantities in a closed form. For example,

$$\mathsf{M} = \mathsf{R}^{\mathsf{T}}\Lambda_q\mathsf{R}, \ |\mathsf{S}| = \hat{\rho}^{(f-q)}|\mathsf{M}|.$$

Refer to Appendix 4.II for details.

From now on, we will drop the $(\hat{.})$ notation that denotes the MLE estimate. Whenever we mention some parameters requiring estimates, we mean the MLE values.

Parameter learning using EM

The key for the approach developed in Section 4.2.2 is (4.8) which relates W to Φ using a linear equation and the empirical loading matrix Q. This motivates us to use the EM learning algorithm to learn the Q matrix instead of the W matrix.

We now present the EM algorithm for learning the parameters Q and ρ in PKPCA. Assume that $\mathsf{Q}^{(j)}$ and $\rho^{(j)}$ are the estimates obtained after the j^{th} iteration. The iteration proceeds as follows:

$$\mathsf{Q}^{(j+1)} = \bar{\mathsf{K}}\mathsf{Q}^{(j)}(\rho^{(j)}\mathtt{I}_q + \mathsf{M}^{-1}{\mathsf{Q}^{(j)}}^{\mathsf{T}}\bar{\mathsf{K}}^2\mathsf{Q}^{(j)})^{-1}, \tag{4.13}$$

$$\rho^{(j+1)} = \frac{1}{f}\mathtt{tr}(\bar{\mathsf{K}} - \bar{\mathsf{K}}\mathsf{Q}^{(j)}{\mathsf{M}^{(j)}}^{-1}{\mathsf{Q}^{(j+1)}}^{\mathsf{T}}\bar{\mathsf{K}}), \tag{4.14}$$

where $\mathsf{M}^{(j)}$, defined in (4.9), is evaluated using $\mathsf{Q}^{(j)}$.

As mentioned earlier, when f is infinite, using (4.14) is not appropriate and hence a manual choice of ρ is used instead. With ρ fixed, Q is nothing but the solution to (4.13) and one can check that Q given in (4.9) is the solution.

Computational efficiency

The above EM algorithm involves only inversions of $q \times q$ matrices and arrives at the same results (up to an orthogonal matrix R) as direct computation. However, in practice one may still use direct computation of complexity $O(N^3)$ since the complexity of computing $\bar{\mathsf{K}}^2$ is $O(N^3)$. If we pre-compute $\bar{\mathsf{K}}^2$, the complexity for each iteration reduces to $O(qN^2)$. Clearly, the overall computation complexity depends on the number of iterations needed for desired accuracy and the ratio of N to q. In our experiment, the EM algorithm converges to reasonable accuracy very fast, usually in less than 20 iterations.

Reconstruction error and Mahalanobis distance

Given a vector $\mathsf{y} \in \mathcal{R}^d$, we are often interested in computing the following two quantities:

1. the reconstruction error $\epsilon_\phi(\mathsf{y}) \equiv (\phi(\mathsf{y}) - \hat{\phi}(\mathsf{y}))^{\mathsf{T}}(\phi(\mathsf{y}) - \hat{\phi}(\mathsf{y}))$ where $\hat{\phi}(\mathsf{y})$ is the reconstructed version of $\phi(\mathsf{y})$;

2. the Mahalanobis distance $\delta_\phi(\mathsf{y}) \equiv (\phi(\mathsf{y}) - \bar{\phi}_0)^{\mathsf{T}}\mathsf{S}^{-1}(\phi(\mathsf{y}) - \bar{\phi}_0)$.

As shown in [167], the best predictor for $\phi(\mathsf{y})$ is $\hat{\phi}(\mathsf{y})$ given by

$$\hat{\phi}(\mathsf{y}) = \mathsf{W}(\mathsf{W}^{\mathsf{T}}\mathsf{W})^{-1}\mathsf{W}^{\mathsf{T}}(\phi(\mathsf{y}) - \bar{\phi}_0) + \bar{\phi}_0,$$

and $\phi(\mathsf{y}) - \hat{\phi}(\mathsf{y})$ is given by

$$\phi(\mathsf{y}) - \hat{\phi}(\mathsf{y}) = (\mathtt{I}_f - \mathsf{W}(\mathsf{W}^{\mathsf{T}}\mathsf{W})^{-1}\mathsf{W}^{\mathsf{T}})(\phi(\mathsf{y}) - \bar{\phi}_0) = \Pi(\phi(\mathsf{y}) - \phi_0),$$

where the $f \times f$ matrix

$$\Pi \equiv \mathtt{I}_f - \mathsf{W}(\mathsf{W}^{\mathsf{T}}\mathsf{W})^{-1}\mathsf{W}^{\mathsf{T}}$$

is symmetric and *idempotent* as

$$\Pi^2 = \Pi.$$

So, $\epsilon_\phi(\mathsf{y})$ is computed as follows:

$$\epsilon_\phi(\mathsf{y}) = (\phi(\mathsf{y}) - \bar{\phi}_0)^{\mathsf{T}}\Pi(\phi(\mathsf{y}) - \bar{\phi}_0) = a(\mathsf{y}) - \mathsf{b}(\mathsf{y})^{\mathsf{T}}\mathsf{C}\mathsf{b}(\mathsf{y}),$$

where C, a_y, and b_y are defined by:

$$\mathsf{C}_{N\times N} \equiv \mathsf{J}\mathsf{Q}(\mathsf{Q}^{\mathsf{T}}\bar{\mathsf{K}}\mathsf{Q})^{-1}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}},$$

$$a(\mathsf{y}) \equiv (\phi(\mathsf{y}) - \bar{\phi}_0)^{\mathsf{T}}(\phi(\mathsf{y}) - \bar{\phi}_0) = k(\mathsf{y},\mathsf{y}) - 2\mathsf{c}(\mathsf{y})^{\mathsf{T}}\mathsf{e} + \mathsf{e}^{\mathsf{T}}\mathsf{K}\mathsf{e},$$

$$\mathsf{b}(\mathsf{y})_{N\times 1} \equiv \Phi^{\mathsf{T}}(\phi(\mathsf{y}) - \bar{\phi}_0) = \mathsf{c}(\mathsf{y}) - \mathsf{K}\mathsf{e},$$

with

$$\mathsf{c}(\mathsf{y})_{N\times 1} \equiv \Phi^{\mathsf{T}}\phi(\mathsf{y}) = [k(\mathsf{x}_1,\mathsf{y}),...,k(\mathsf{x}_N,\mathsf{y})]^{\mathsf{T}}.$$

The Mahalanobis distance is calculated as follows:

$$\delta_\phi(\mathsf{y}) = (\phi(\mathsf{y}) - \bar{\phi}_0)^{\mathsf{T}}\mathsf{S}^{-1}(\phi(\mathsf{y}) - \bar{\phi}_0) = \rho^{-1}\{a(\mathsf{y}) - \mathsf{b}(\mathsf{y})^{\mathsf{T}}\mathsf{B}\mathsf{b}(\mathsf{y})\}. \tag{4.15}$$

Finally, an important observation is that as long as we can express $\bar{\phi}_0$ and S as in (4.2) and (4.3), i.e. there exist e and J that relate $\bar{\phi}_0$ and S to Φ, we can safely use the derivations presented in this section. This lays a solid foundation for the development of the mixture of PKPCA theory.

We can study a limiting behavior of $\delta_\phi(\mathsf{y})$ by defining

$$\hat{\delta}_\phi(\mathsf{y}) \equiv \lim_{\rho\to 0}\rho\delta_\phi(\mathsf{y}) = a(\mathsf{y}) - \mathsf{b}(\mathsf{y})^{\mathsf{T}}\hat{\mathsf{B}}\mathsf{b}(\mathsf{y}), \tag{4.16}$$

where $\hat{\mathsf{B}} \equiv \lim_{\rho\to 0}\mathsf{B}$.

Experiments on kernel modeling

This part addresses the power of kernel modeling part in PKPCA in terms of the reconstruction error. The probabilistic nature of PKPCA will be illustrated in the next sections.

We compare PPCA and PKPCA since the only difference between them is the kernel modeling part. We define the reconstruction error percentage η as follows:

$$\eta(\mathsf{y}) = \frac{\epsilon(\mathsf{y})}{\mathsf{y}^{\mathsf{T}}\mathsf{y}}, \quad \eta_\phi(\mathsf{y}) = \frac{\epsilon_\phi(\mathsf{y})}{k(\mathsf{y},\mathsf{y})},$$

where $\eta(\mathsf{y})$ is for PPCA and $\eta_\phi(\mathsf{y})$ for PKPCA.

Figure 4.2 shows the histogram of η for the famous iris data[2]. This dataset consists of 150 samples and is used in pattern classification tasks. We, however,

[2]This is available at the UCI Machine Learning Repository. The URL is http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Algorithm	PPCA $q=2$	PPCA $q=3$	PKPCA $q=9$	PKPCA $q=15$
Mean	8.23%	1.42%	3.88%	1.39%
Std. dev.	13.12%	4.52%	3.86%	1.39%

Table 4.1: PPCA and PKPCA reconstruction error percentage.

just treat it as a whole regardless of its class labels. Since it is just 4-d data, PPCA keeps at most 3 principal component, i.e. $q \leq 3$, while PKPCA has no such limit and can have $q \leq 149$. Figure 4.2 and Table 4.1 show that PKPCA with $q = 9$, i.e. using 6% percent principal components produces a small η than PPCA with $q = 2$ that uses 50% components. In addition, PKPCA with $q = 15$ that uses 10% percent principal components produces a small η than PPCA with $q = 3$, using 75% components. A larger q produces even smaller η. This improvement benefits from kernel modeling, which is able to capture the nonlinear structure of the data. However, PKPCA involves much more computation than PPCA.

4.3 Mixture Modeling of Probabilistic Kernel Principal Components

4.3.1 Theory of mixture of PKPCA

Mixture of PKPCA models the data in a high-dimensional feature space using a mixture of I densities with each mixture component $\mathsf{p}(.|i)$ being a PKPCA density associated with an empirical loading matrix Q_i that can be derived from corresponding e_i and J_i (as shown below). For ρ_i's, we assume $\rho_i \equiv \rho$ with ρ fixed.

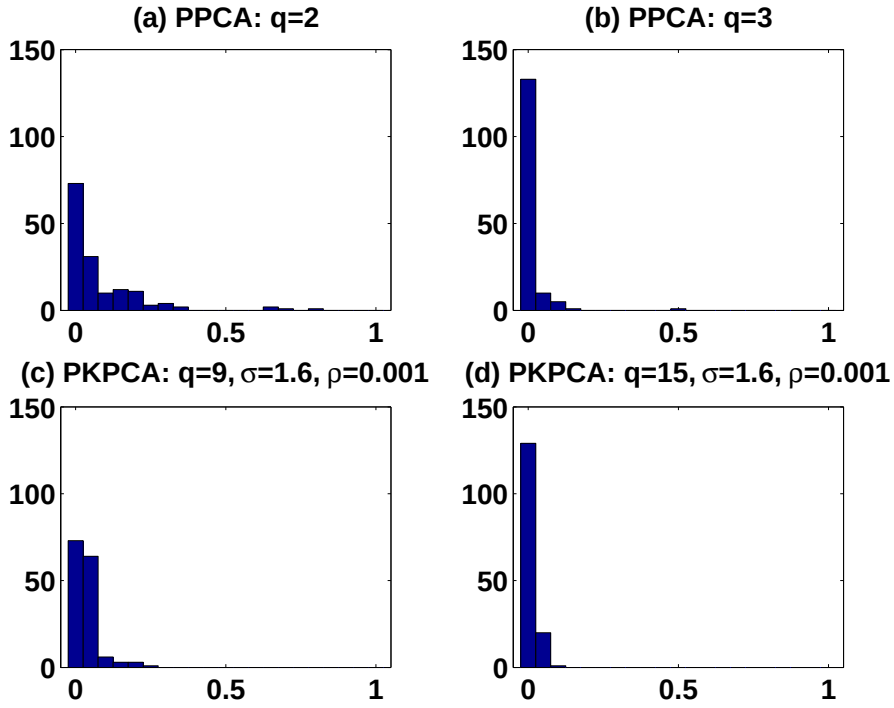


Figure 4.2: Histogram of η for iris data obtained by (a) PPCA with $q = 2$, (b) PPCA with $q = 3$, (c) PKPCA with Gaussian kernel with $q = 9$, $\sigma = 2$ and $\rho = 0.001$, and (d) PKPCA with Gaussian kernel with $q = 15$, $\sigma = 2$ and $\rho = 0.001$.

Mathematically,

$$\mathtt{p}(\phi(\mathsf{x})) = \sum_{i=1}^{I} m_i \mathtt{p}(\phi(\mathsf{x})|i) = \sum_{i=1}^{I} m_i \mathtt{N}(\bar{\phi}_i, \mathsf{S}_i),$$

where m_i's are mixing probabilities summing up to 1, and $\mathtt{p}(\phi(\mathsf{x})|i) = \mathtt{N}(\bar{\phi}_i, \mathsf{S}_i)$ is the PKPCA density for the i^{th} component defined as

$$\begin{aligned} \mathtt{N}(\bar{\phi}_i, \mathsf{S}_i) &= \frac{(2\pi)^{-f/2}}{|\mathsf{S}_i|^{1/2}} \exp\{-\frac{1}{2}\delta_{\phi,i}(\mathsf{x})\} = \frac{(2\pi)^{-f/2}}{\rho^{(f-q_i)/2}|\mathsf{M}_i|^{1/2}} \exp\{-\frac{1}{2}\delta_{\phi,i}(\mathsf{x})\} \\ &= (2\pi\rho)^{-f/2} \exp\{-\frac{1}{2}\tilde{\delta}_{\phi,i}(\mathsf{x})\} \end{aligned}$$

where $\delta_{\phi,i}(\mathsf{x})$ is the Mahalanobis distance as in (4.15) with all parameters involved coming from the i^{th} component, and

$$\tilde{\delta}_{\phi,i}(\mathsf{x}) \equiv \delta_{\phi,i}(\mathsf{x}) + \log(|\mathsf{M}_i|) + q_i \log(\rho^{-1}).$$

We call $\tilde{\delta}_{\phi}(\mathsf{x})$ as the 'generalized' Mahalanobis distance.

Parameter learning using EM

We invoke the ML principle to estimate the parameters of interest, i.e., $\{m_i, \mathsf{Q}_i\}$'s from the training data. It turns out that direct maximization is cumbersome since the log-likelihood involves summations within logarithms. The iterative EM algorithm [152, 167] is used instead.

Assume that $\{m_i^{(j)}, \mathsf{Q}_i^{(j)}\}$ are the values obtained in the j^{th} iteration. We begin by computing the posterior responsibility r_{ni}.

$$r_{ni}^{(j)} \equiv \mathsf{p}^{(j)}(i|\phi_n) = \frac{m_i \mathsf{p}^{(j)}(\phi_n|i)}{\mathsf{p}^{(j)}(\phi_n)} = \frac{m_i^{(j)} \exp\{-\frac{1}{2}\tilde{\delta}_{\phi,i}^{(j)}(\mathsf{x})\}}{\sum_{l=1}^{I} m_l^{(j)} \exp\{-\frac{1}{2}\tilde{\delta}_{\phi,l}^{(j)}(\mathsf{x})\}}. \tag{4.17}$$

There is no need to calculate r_{ni} by exactly following (4.17). One only needs to evaluate the numerator $m_i \exp\{-\frac{1}{2}\tilde{\delta}_{\phi,i}(\mathsf{x})\}$ and perform normalization to guarantee that $\sum_{i=1}^{I} r_{ni} = 1$.

The EM iterations compute the following quantities:

$$m_i^{(j+1)} = \frac{1}{N}\sum_{n=1}^{N} r_{ni}^{(j)}, \tag{4.18}$$

$$\bar{\phi}_i^{(j+1)} = \frac{\sum_{n=1}^{N} r_{ni}^{(j)} \phi_n}{\sum_{n=1}^{N} r_{ni}^{(j)}} = \sum_{n=1}^{N} e_{ni}^{(j)} \phi_n = \Phi \mathsf{e}_i^{(j)},$$

where $\mathsf{e}_i^{(j)} = [e_{1i}^{(j)}, e_{2i}^{(j)}, \ldots, e_{Ni}^{(j)}]^{\mathsf{T}}$ with

$$e_{ni}^{(j)} \equiv \frac{r_{ni}^{(j)}}{\sum_{n=1}^{N} r_{ni}^{(j)}}.$$

It is easy to show that the local responsibility-weighted covariance matrix for component i, S_i, is obtained as

$$\mathsf{S}_i^{(j+1)} \equiv \sum_{n=1}^{N} e_{ni}^{(j)} (\phi_n - \bar{\phi}_i^{(j+1)})(\phi_n - \bar{\phi}_i^{(j+1)})^{\mathsf{T}} = \Phi \mathsf{J}_i^{(j+1)} {\mathsf{J}_i^{(j+1)}}^{\mathsf{T}} \Phi^{\mathsf{T}},$$

where

$$\mathsf{J}_i^{(j+1)} \equiv (\mathtt{I}_N - \mathsf{e}_i^{(j)} \mathsf{1}^{\mathsf{T}})\ \mathtt{D}^{1/2}[e_{1i}^{(j)}, e_{2i}^{(j)}, \ldots, e_{Ni}^{(j)}].$$

Using

$$\bar{\mathsf{K}}_i^{(j+1)} = {\mathsf{J}_i^{(j+1)}}^{\mathsf{T}} \mathsf{K} \mathsf{J}_i^{(j+1)},$$

the updated $\mathsf{Q}_i^{(j+1)}$ can obtained as

$$\mathsf{Q}_i^{(j+1)} = \mathsf{V}_{q_i,i}^{(j+1)} (\mathtt{I}_{q_i} - \rho {\Lambda_{q_i,i}^{(j+1)}}^{-1})^{1/2}, \tag{4.19}$$

where $\Lambda_{q_i,i}^{(j+1)}$ and $\mathsf{V}_{q_i,i}^{(j+1)}$ are the top q_i eigenvalues and eigenvectors of $\bar{\mathsf{K}}_i^{(j+1)}$. Also, an EM algorithm for learning the Q_i matrix as shown in Section 4.2.2 can be used instead of direct computation.

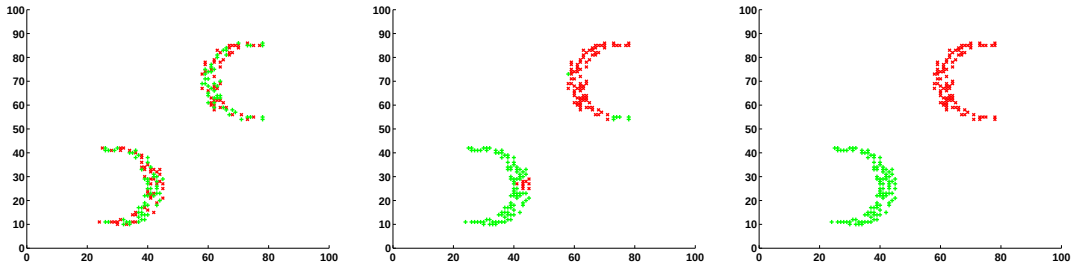
The above derivations indicate that it is not necessary to start the EM iterations from initializing the parameters e.g. $\{m_i, \mathsf{Q}_i\}$'s. Instead, we can start from assigning the posterior responsibility $\{r_{ni}\}$'s. Once assigned, we follow equations (4.18) to (4.19) to compute the updated $\{m_i, \mathsf{Q}_i\}$'s. The iterations then move on. This way we can easily incorporate any prior knowledge gained from clustering techniques such as the 'kernelized' version of the K-means algorithm [181], or other algorithms [180].

Parameter learning experiments

We now demonstrate how mixture of PKPCA performs by fitting it to the two C-shapes shown in Figure 4.4(d). We set the following parameters: $I = 2$, $q = 2$, $\rho = 1e-2$, and $\sigma = 8$. The algorithm iterations are terminated if the changes in the $\{r_{ni}\}$'s are small enough.

Figure 4.3(a) presents the initial configuration for the two C-shapes. We just generate random numbers for $\{r_{ni}\}$'s followed by a normalization step to guarantee

$\sum_{i=1}^{I} r_{ni} = 1$. Figure 4.3(b) shows the mixture assignment after the first iteration and Figure 4.3(c) the final configuration (only after 3 iterations). A final note is that the EM algorithm can still converge to a local minimum. In this case, the clustering method [180] is very helpful for initialization.



(a) (b) (c)

Figure 4.3: (a) Initial configuration. (b) After first iteration. (c) Final configuration. '+' and 'x' denote two different mixture components.

4.3.2 Why mixture of PKPCA?

It is well known [180, 181] that kernel embedding results in clustering capability. This raises the doubt whether PKPCA is sufficient to model a nonlinear structure with nonlinear substructures. We demonstrate the effectiveness of mixture of PKPCA with the following examples.

Figure 4.4(a) shows a nonlinear structure containing a single C-shape and Figures 4.4(b) and 4.4(c) the contour plots, for the 1st and 2nd kernel principal components, i.e. all points in the contour share the same principal component values. These plots capture the nonlinear shape very precisely. Now in Figure 4.4(d), a nonlinear structure containing two C-shapes is presented. Figures 4.4(e) and 4.4(f) display the contour plots corresponding to 4.4(d). Clearly, they attempt to capture both C-shapes at the same time. This is not desirable. Ideally, we want to

have two KPCAs, each modeling a different C-shape more precisely. However, the ordinary KPCA has no such capability but PKPCA does. This naturally leads us to considering a mixture of PKPCA. Section 4.4 also demonstrates this using classification results.

The successful kernel clustering algorithm [180] shows that after kernel embedding, the clusters become more separable. This further sheds light on the effectiveness of mixture of PKPCA.

One may also ask: why not use the mixture of PPCA directly? Although a mixture of PPCA is legitimate, its use is not elegant in this scenario since one may need more than 2 components for Figure 4.4(d) to capture the data structure due to the limitation of the linear setting in PCA. But mixture of PKPCA can elegantly model it using two components.

4.4 Classification

4.4.1 PKPCA or mixture of PKPCA classifier

We now demonstrate the probabilistic interpretation embedded in PKPCA using a pattern classification problem. Suppose we have N classes. For class n, a PKPCA or mixture of PKPCA density $\mathsf{p}(\phi_n(\mathsf{x})|n)$ is trained; then, the class label for a point x is determined using the Bayesian decision principle by

$$\hat{n} = \arg \max_{n=1,\ldots,N} \mathsf{p}(n)\mathsf{p}(\mathsf{x}|n) = \arg \max_{n=1,\ldots,N} \mathsf{p}(n)\mathsf{p}(\phi_n(\mathsf{x})|n)|\mathsf{J}_n(\mathsf{x})|, \tag{4.20}$$

where $\mathsf{p}(n)$ is the prior distribution, $\mathsf{p}(\mathsf{x}|n)$ is the conditional density for class n in the original space, and $\mathsf{J}_n(\mathsf{x})$ is the Jacobi matrix for class n.

To use (4.20), we are confronted by two dilemmas: (i) the Jacobi matrices, $\mathsf{J}_n(\mathsf{x})$'s, are unknown since we have no knowledge of $\phi_n(\mathsf{x})$; and (ii) the densities,

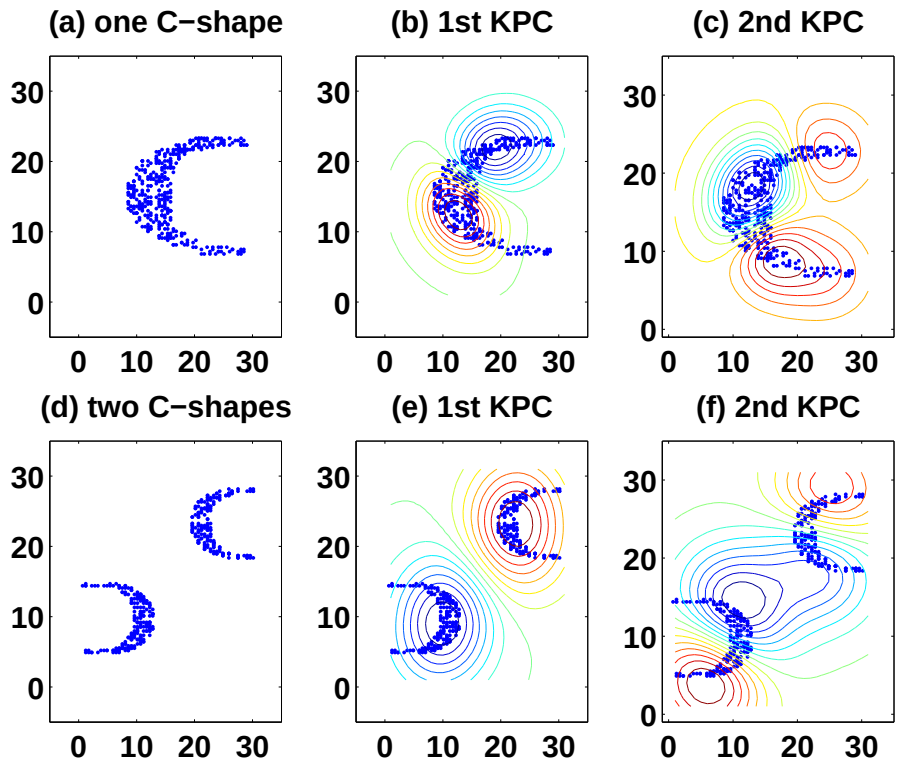


Figure 4.4: (a) One C-shape and contour plots of its (b) 1st and (c) 2nd KPCA features. (d) Two C-shapes and its contour plots of its (e) 1st and (f) 2nd KPCA features.

$\mathsf{p}(\phi_n(\mathsf{x})|n)$'s, involves infinite f. The latter is easily fixed by assuming $\rho_c \equiv \rho$ for all classes, where ρ_n is the parameter in the density $\mathsf{p}(\phi_n(\mathsf{x})|n)$ for class n.

One trick to attack the first dilemma is to use the same kernel function for all the classes with the same kernel width σ, i.e. $\sigma_n = \sigma$. However, it might not be appropriate since different classes possess different data structures. An alternative approach is that we still use different kernel functions for different classes but we approximate the Jacobi matrices. We use the following approximation:

$$|\mathsf{J}_n(\mathsf{x})| \simeq const, \quad \forall \mathsf{x}.$$

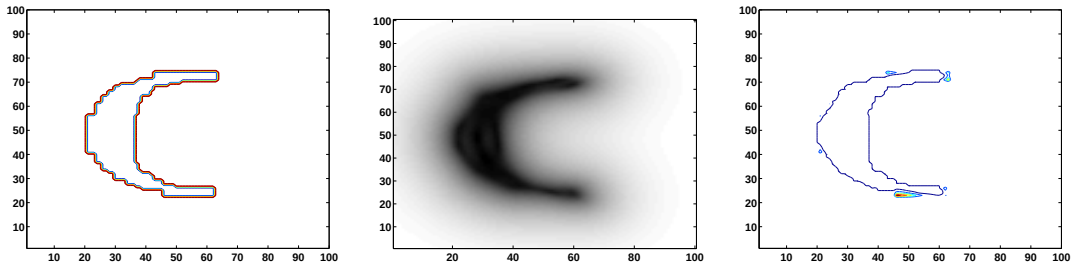
Figure 4.5 demonstrates our rationale. Figure 4.5(a) presents the contour plots

for the true density to be modeled, which is uniform inside the black C-shaped region (Figure 4.1(a)). All contour plots are located on the boundary. We fit a PKPCA density ($\sigma = 15$, $q = 20$, and $\rho = 1e-6$) based on the samples shown in Figure 4.1(b) and visualize the density using Figure 4.5(b), which displays the map of $\log(\delta_\phi(\mathtt{x}))$. To verify that the values in the C-shaped region are uniform, we show in Figure 4.5(c) the contour plots for $\tilde{\delta}_\phi(\mathtt{x})$ inside the C-shaped region. Most contours are close to the boundary, which indicates the uniformity of the density $\mathtt{p}(\phi(x))$ inside the C-shaped region and thus the Jacobi approximation which relates $\mathtt{p}(\phi(x))$ and $\mathtt{p}(\mathtt{x})$ is reasonable.

The above approximation leads to a linear decision rule. For example, in a two-class problem, the decision rule is, for some $\alpha > 0$,

$$\text{If } \ \mathtt{p}(\phi_1(x)|1) \geq \alpha\ \mathtt{p}(\phi_2(x)|2) \ \text{ then class 1; } \ \text{Else class 2}$$

In the sequel, we simply take $\alpha = 1$.



(a) (b) (c)

Figure 4.5: The approximation of the Jacobi matrix. (a) The contour plots of the true density: uniform inside the C-shaped region. (b) The map of $\log(\delta_\phi)$. (c) The contour plots of $\tilde{\delta}_\phi$ inside the C-shaped region.

Putting the above discussions together, we have the following decision rules:

- If PKPCA densities are learned for all classes, i.e., for class n, we learn

$\{\rho_n = \rho, \mathsf{Q}_n\}$, it is easy to check that the classifier performs the following:

$$\arg \min_{n=1,...,N} \tilde{\delta}^n_\phi(\mathsf{x}),$$

where $\tilde{\delta}^n$ is the 'generalized' Mahalanobis distance.

- If mixture of PKPCA densities are learned for all classes, i.e., for the class c, we learn $\{\rho_n = \rho, m_{n,1}, \mathsf{Q}_{n,1}, ..., m_{n,I_c}, \mathsf{Q}_{n,I_n}\}$ with I_n being the number of mixture components, then the classifier decides as follows:

$$\arg \max_{n=1,...,N} \sum_{j=1}^{I_n} m_{n,j} \exp\{-\frac{1}{2}\tilde{\delta}^n_{\phi,j}(\mathsf{x})\}.$$

4.4.2 Experiments

Synthetic Data

We consider a 2-class problem with foreground (class 1) and background (class 2) classes given in Figure 4.1(a), where the letter 'C' or 'O' means the foreground class. We then draw 200 samples for both classes as shown in Figures 4.1 and 4.8.

Figure 4.6 presents the classification results obtained by the PKPCA classifier with different kernel widths for different classes (PKPCA-d), the PKPCA classifier with same kernel widths for different classes (PKPCA-s), the support vector machine (SVM) [19], and the kernel Fisher discriminant analysis (KFDA) [177]. In PKPCA-s, SVM and KFD, the kernel width σ is tuned (via exhaustive search from 1 to 100) to yield the best empirical classification results and reported in Table 4.2. The PKPCA-d parameters actually used are also reported in Table 4.2, where the kernel widths for the background and foreground classes are found via the procedures described in Appendix 4.III. As shown in Figure 4.6, the classification boundary obtained by PKPCA-d is very smooth and very similar to the original

Algorithm	Single C-shape	Single O-shape	Double C-shapes
PKPCA-d	1.57% $q = 30$, $\rho = 10^{-8}$ $\sigma_1 = 15$,$\sigma_2 = 35$	3.80% $q = 20$, $\rho = 10^{-6}$ $\sigma_1 = 15$, $\sigma_2 = 35$	7.49% $q = 20$, $\rho = 10^{-6}$ $\sigma_1 = 15$, $\sigma_2 = 35$
PKPCA-s	1.95% $q = 30$, $\rho = 10^{-8}$ $\sigma = 1$	5.50% $q = 30$, $\rho = 10^{-8}$ $\sigma = 1$	1.85% $q = 20$, $\rho = 10^{-6}$ $\sigma = 1$
SVM	1.80% $\sigma = 1$	5.45% $\sigma = 1$	1.69% $\sigma = 1$
KFDA	1.84% $\sigma = 1$, 30 components	5.47% $\sigma = 1$, 20 components	1.82% $\sigma = 1$, 20 components
mix. PKPCA	NA	NA	0.70% $q = 20$, $\rho = 10^{-6}$, $I_1 = 2$, $\sigma_1 = 8$, $I_2 = 1$, $\sigma_2 = 35$

Table 4.2: Classification error on the single C-shaped, the single O-shape, and the double C-shapes.

boundary, while those of PKPCA-s, SVM and KFDA seem to only replicate the training samples, with holes and gaps. Table 4.2 indicates that our PKPCA-d classifier outperforms the SVM and KFDA classifiers by some margin. Similar observations can be made based on the experimental results on a single O-shape as shown in Figure 4.8.

The superior performance of PKPCA-d classifer mainly arises from its ability to model different classes with different kernel functions, while the PKPCA-s, SVM and KFDA employ only one kernel. This is a big advantage since as seen in our synthetic examples we clearly need different kernel widths for the foreground and

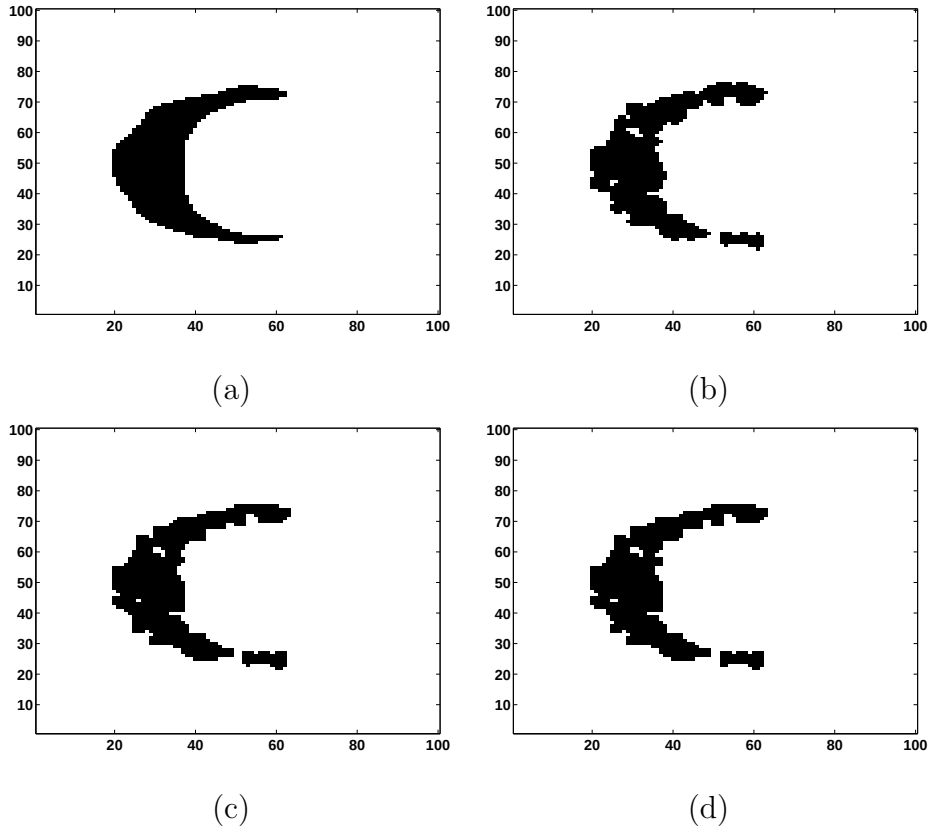


Figure 4.6: The classification results on the single C-shape obtained by (a) PKPCA-d, (b) PKPCA-s, (c) SVM, and (d) KFDA.

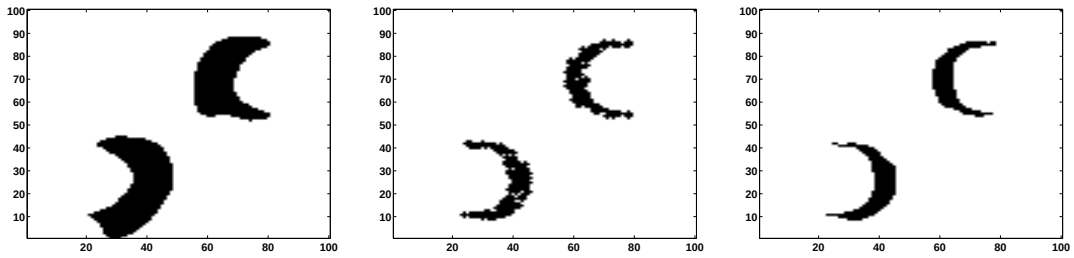


Figure 4.7: The classification results on the double C-shape obtained by (a) PKPCA-d classfier, (b) SVM, and (c) mixture of PKPCA classfier with different kernel widths.

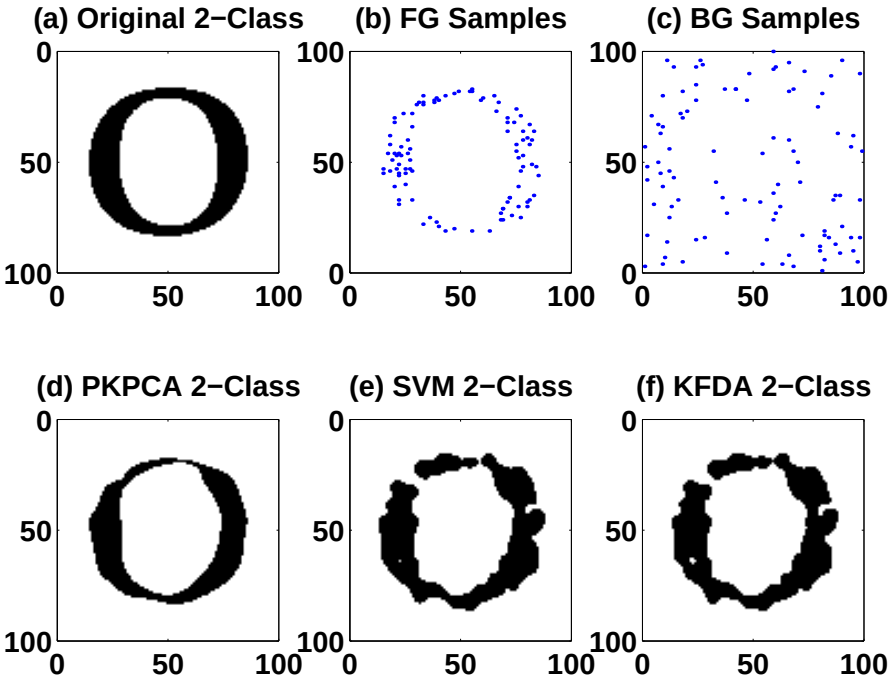


Figure 4.8: The classification results on the single O-shape.

background classes. More importantly, PKPCA provides a regularized approximation to the data structure; thus its decision boundary is very smooth. Also, the probabilistic interpretation of PKPCA enables the PKPCA classifier to deal with an N-class problem as easily as KFDA, while the SVM is basically designed for a two-class problem and extending it to an M-class is not very straightforward.

We now illustrate the mixture of PKPCA classifier by applying it to the double C-shapes shown in Figure 4.1(d). We fit the mixture of PKPCA density for the foreground class based on the samples shown in Figure 4.1(e) and the PKPCA density for the background class based on the samples shown in Figure 4.1(f). Figure 4.7 and Table 4.2 present the classification results. Clearly the mixture of PKPCA classifier produces the best performance in terms of the classification error. Also the decision boundary is very smooth.

One important observation is that the PKPCA classifier with different kernel widths performs poorly. This is because the selected kernel width attempts to cover both nonlinear substructures simultaneously, which actually over-smoothes each substructure (see Figure 4.7(a)). Hence, caution should be exercised when modeling a mixture data via PKPCA densities of different kernel widths.

IDA Benchmark

We also test our classifier on the IDA benchmark[3] repository [179]. To make our results comparable, we use the cross-validation (the same procedure as in [179]) to choose our parameters; also we invoke the PKPCA density without mixture modeling and the same kernel parameter for different classes. As tabulated in Table. 4.3, our PKPCA classifier compared favorably to those of kernel classifiers such as SVM and KFD. We believe that the classification results can be improved by using PKPCA-d or even mixture of PKPCA classifier.

A real application: face recognition

We report face recognition results using a subset of the FERET database [58] with 200 subjects only. Each subject has 3 images: (i) one taken under controlled lighting condition with a neutral expression; (ii) one taken under the same lighting condition as (i) but with different facial expressions (mostly smiling); and (iii) one taken under different lighting condition and mostly with a neutral expression. Figure 4.9 shows some face examples in this database.

Our experiment focuses on testing the generalization capability of our algorithm. It is our hope that the training stage can learn the intrinsic characteristics

[3]This is available at http://ida.first.gmd.de/~raetsch/data/benchmarks.htm.

	PKPCA-s	SVM	KFD
Banana	**10.5 ± 0.4**	11.5 ± 0.7	10.8 ± 0.5
B. Cancer	28.0 ± 4.7	26.0 ± 4.7	**25.8 ± 4.6**
Diabetes	24.8 ± 1.9	23.5 ± 1.7	**23.2 ± 1.6**
German	24.9 ± 2.2	**23.6 ± 2.1**	23.7 ± 2.2
Heart	16.8 ± 3.4	**16.0 ± 3.3**	16.1 ± 3.4
Image	**2.8 ± 0.6**	3.0 ± 0.6	3.3 ± 0.6
Ringnorm	1.6 ± 0.1	1.7 ± 0.1	**1.5 ± 0.1**
F. Solar	34.8 ± 1.9	**32.4 ± 1.8**	33.2 ± 1.7
Splice	12.2± 0.8	10.9 ± 0.7	**10.5 ± 0.6**
Thyroid	**4.0 ± 2.0**	4.8 ± 2.2	4.2 ± 2.1
Titanic	22.6 ± 1.3	**22.4 ± 1.0**	23.2 ± 2.0
Twonorm	**2.6 ± 0.2**	3.0 ± 0.2	**2.6 ± 0.2**
Waveform	11.4± 0.5	**9.9 ± 0.4**	**9.9 ± 0.4**

Table 4.3: The classification error on IDA benchmark repository. The SVM and KFD results are reported in [179].

Figure 4.9: Top row: neutral faces. Middle row: faces with facial expression. Bottom row: faces under different illumination. Image size is 24 by 21 in pixels.

of the space we are interested in. Therefore, we always keep the gallery and probe sets separate. We randomly select 300 images belonging to 100 subjects as the

gallery set for learning and the remaining 300 images as the probe set for testing. This random division is repeated 20 times and we take their averages as the final result.

General component analysis is not geared towards discrimination, thus yielding inferior recognition results in practice. To this end, Moghaddam *et al.* [55, 56] introduced the concept of intra-personal space (IPS). The IPS is constructed by collecting all the difference images between any two image pairs belonging to the same individual. The construction of the IPS is meant to capture all the possible intra-personal variations introduced during image acquisition.

Suppose that we have learned some density p_{IPS} on top of the IPS space and we are given the gallery set consisting of images $\{\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_N\}$ for N different individuals. Given a probe image y, its identity $\hat{n}$ is determined by

$$\hat{n} = \arg \max_{c=1,\ldots,N} \mathsf{p}_{IPS}(\mathsf{y} - \mathsf{x}_n) = \arg \min_{c=1,\ldots,N} \hat{\delta}_{IPS,\phi}(\mathsf{y} - \mathsf{x}_n).$$

Here we use the limiting Mahalanobis distance $\hat{\delta}$.

For comparison, we have implemented the following four methods. In PKPCA/IPS and PPCA/IPS, the IPS is constructed based on the gallery set and the PKPCA/PPCA density is fitted on top of that. In KPCA and PCA, all 300 training images are regarded lying in one face space and KPCA/PCA is then learned on that space. The classifier sets the identity of a probe image as the identity of its nearest neighbor in the gallery set.

Table 4.4 lists the recognition rate, averaging those of 20 simulations, using the top 1 match. The PKPCA/IPS algorithm attains the best performance since it combines the discriminative power of the IPS model and the merit of PKPCA. However, compared to PPCA/IPS, the improvement is not significant, indicating that second-order statistics might be enough after IPS modeling for the face recog-

nition problem. However, PKPCA may be more effective since it also takes into account high-order statistics. Another observation is that variations in illumination are easier to model than facial expression using subspace methods.

	PKPCA/IPS	PPCA/IPS	KPCA	PCA
Expression	78.55%	78.35%	63.85%	67.65%
Illumination	83.9%	81.85%	51.9%	73.1%
Average	81.23%	80.1%	57.88%	70.38%

Table 4.4: Recognition rate of various kernel and non-kernel subspace methods.

4.5 Appendix

Appendix 4.I: Two Lemmas on Matrix Computation

We introduce some related results on matrix computation using the following two lemmas. The proofs are pretty straightforward and hence skipped here.

Lemma 4.1. Suppose that A is of size $d \times q$ with $q < d$ and the matrix $\mathsf{A}^{\mathsf{T}}\mathsf{A}$ is of full rank, the matrices $\mathsf{A}^{\mathsf{T}}\mathsf{A}$ and $\mathsf{A}\mathsf{A}^{\mathsf{T}}$ have the same nonzero eigenvalues.

Lemma 4.2. Suppose that $\mathsf{B} = \rho \mathtt{I}_d + \mathsf{A}\mathsf{A}^{\mathsf{T}}+$ and $\{\tau_i;\ i = 1, 2, \ldots, q\}$ are eigenvalues of the $\mathsf{A}^{\mathsf{T}}\mathsf{A}$ matrix, the determinant $|\mathsf{B}|$ is given by

$$|\mathsf{B}| = \prod_{i=1}^{q} (\rho + \tau_i) \rho^{d-q}, \tag{4.21}$$

and the inverse matrix B^{-1} is given by

$$\mathsf{B}^{-1} = \rho^{-1}\{\mathtt{I}_f - \mathsf{A}(\rho \mathtt{I}_q + \mathsf{A}^{\mathsf{T}}\mathsf{A})^{-1}\mathsf{A}^{\mathsf{T}}\}.$$

Appendix 4.II: A List of Important Quantities

Important quantities

RKHS: $\mathcal{H} = \mathcal{R}^f$.

Original observations: $\mathsf{X}_{d\times N} = [\mathsf{x}_1, \mathsf{x}_2, ..., \mathsf{x}_N]$

Nonlinear mapping: $\phi(\mathsf{x}) : \mathcal{R}^d \to \mathcal{R}^f$

Observations in RKHS: $\Phi_{f\times N} = [\phi_1, \phi_2, ..., \phi_N]$.

Weight vector: $\mathsf{e}_{N\times 1} = N^{-1}\mathsf{1}$ (for example).

Mean: $\mu_{f\times 1} = \Phi\mathsf{e}$

Centering matrix: $\mathsf{J}_{N\times N} = N^{-1/2}(\mathtt{I}_N - \mathsf{e}\mathsf{1}^\mathsf{T})$.

Covariance matrix (c.m.): $\Sigma_{f\times f} = \Phi\mathsf{J}\mathsf{J}^\mathsf{T}\Phi^\mathsf{T}$.

Gram matrix (g.m.): $\mathsf{K}_{N\times N} = \Phi^\mathsf{T}\Phi$.

Centered g.m.: $\bar{\mathsf{K}}_{N\times N} = \mathsf{J}^\mathsf{T}\mathsf{K}\mathsf{J}$.

Eigenvalues of $\bar{\mathsf{K}}$: $\Lambda_q = \mathtt{D}[\lambda_1, \lambda_2, \ldots, \lambda_q]_{q\times q}$.

Eigenvectors of $\bar{\mathsf{K}}$: $\mathsf{V}_q = [\mathsf{v}_1, \mathsf{v}_2, \ldots, \mathsf{v}_q]_{N\times q}$.

Approximate c.m.: $\mathsf{S}_{f\times f} = \Phi\mathsf{A}\Phi^\mathsf{T} + \rho\mathtt{I}_f$.

A matrix: $\mathsf{A}_{N\times N} = \mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})\mathsf{V}_q^\mathsf{T}\mathsf{J}^\mathsf{T}$.

Inverse of S: $\mathsf{S}^{-1}_{N\times N} = \rho^{-1}(\mathtt{I}_f - \Phi\mathsf{B}\Phi^\mathsf{T})$.

B matrix: $\mathsf{B}_{N\times N} = \mathsf{J}\mathsf{V}_q(\Lambda_q^{-1} - \rho\Lambda_q^{-2})\mathsf{V}_q^\mathsf{T}\mathsf{J}^\mathsf{T}$.

C matrix: $\mathsf{C}_{N\times N} = \mathsf{J}\mathsf{Q}(\mathsf{Q}^\mathsf{T}\bar{\mathsf{K}}\mathsf{Q})^{-1}\mathsf{Q}^\mathsf{T}\mathsf{J}^\mathsf{T}$.

Q matrix: $\mathsf{Q}_{N\times q} = \mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R}$

M matrix: $\mathsf{M}_{q\times q} = \rho\mathtt{I}_q + \mathsf{Q}^\mathsf{T}\bar{\mathsf{K}}\mathsf{Q}$.

Computation related to L and M

We first compute $\mathsf{L} = \mathsf{Q}^\mathsf{T}\bar{\mathsf{K}}\mathsf{Q}$ and then M.

$$\mathsf{L} \;=\; \mathsf{R}^\mathsf{T}\mathsf{Q}^\mathsf{T}\bar{\mathsf{K}}\mathsf{Q}\mathsf{R} = \mathsf{R}^\mathsf{T}(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{V}_q^\mathsf{T}\bar{\mathsf{K}}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R}$$

$$= \mathsf{R}^{\mathsf{T}}(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\Lambda_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R} = \mathsf{R}^{\mathsf{T}}(\Lambda_q - \rho\mathtt{I}_q)\mathsf{R},$$

where the fact that $\mathsf{V}_q^{\mathsf{T}}\bar{\mathsf{K}}\mathsf{V}_q = \mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\mathsf{K}\mathsf{J}\mathsf{V}_q = \Lambda_q$ is used. Therefore,

$$\mathsf{M} = \rho\mathtt{I}_q + \mathsf{L} = \rho\mathtt{I}_q + \mathsf{R}^{\mathsf{T}}(\Lambda_q - \rho\mathtt{I}_q)\mathsf{R} = \mathsf{R}^{\mathsf{T}}\Lambda_q\mathsf{R}.$$

$$|\mathsf{M}| = |\Lambda_q| = \prod_{i=1}^{q}\lambda_i, \quad \mathsf{M}^{-1} = \mathsf{R}^{\mathsf{T}}\Lambda_q^{-1}\mathsf{R}.$$

Computation related to A, B, and C

$$\begin{aligned}\mathsf{A} &= \mathsf{J}\mathsf{Q}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}} = \mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R}\mathsf{R}^{\mathsf{T}}(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\\ &= \mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\end{aligned}$$

$$\begin{aligned}\mathsf{B} &= \mathsf{J}\mathsf{Q}\mathsf{M}^{-1}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}} = \mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R}\mathsf{R}^{\mathsf{T}}\Lambda_q^{-1}\mathsf{R}\mathsf{R}^{\mathsf{T}}(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\\ &= \mathsf{J}\mathsf{V}_q(\Lambda_q^{-1} - \rho\Lambda_q^{-2})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\end{aligned}$$

$$\begin{aligned}\mathsf{C} &= \mathsf{J}\mathsf{Q}(\mathsf{Q}^{\mathsf{T}}\bar{\mathsf{K}}\mathsf{Q})^{-1}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\\ &= \mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{R}\mathsf{R}^{\mathsf{T}}(\Lambda_q - \rho\mathtt{I}_q)^{-1}\mathsf{R}\mathsf{R}^{\mathsf{T}}(\mathtt{I}_q - \rho\Lambda_q^{-1})^{1/2}\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\\ &= \mathsf{J}\mathsf{V}_q\Lambda_1^{-1}\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\end{aligned}$$

$$\begin{aligned}\mathtt{tr}[\mathsf{A}\mathsf{K}] &= \mathtt{tr}[\mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho\Lambda_q^{-1})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\mathsf{K}] = \mathtt{tr}[(\mathtt{I}_q - \rho\Lambda_q^{-1})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\mathsf{K}\mathsf{J}\mathsf{V}_q]\\ &= \mathtt{tr}[(\mathtt{I}_q - \rho\Lambda_q^{-1})\Lambda_q] = \mathtt{tr}[\Lambda_q] - \rho q = \textstyle\sum_{i=1}^{q}\lambda_i - \rho q.\end{aligned}$$

$$\begin{aligned}\mathtt{tr}[\mathsf{B}\mathsf{K}] &= \mathtt{tr}[\mathsf{J}\mathsf{V}_q(\Lambda_q^{-1} - \rho\Lambda_q^{-2})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\mathsf{K}] = \mathtt{tr}[(\Lambda_q^{-1} - \rho\Lambda_q^{-2})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\mathsf{K}\mathsf{J}\mathsf{V}_q]\\ &= \mathtt{tr}[(\Lambda_q^{-1} - \rho\Lambda_q^{-2})\Lambda_q] = q - \rho\mathtt{tr}[\Lambda_q^{-1}] = q - \rho\textstyle\sum_{i=1}^{q}\lambda_i^{-1}.\end{aligned}$$

Computation related to S

We have shown that

$$\mathsf{S}^{-1} = \rho^{-1}(\mathtt{I}_f - \Phi\mathsf{B}\Phi^{\mathsf{T}}),$$

Also, we are often interested in computing $\mathtt{tr}(\mathsf{S}^{-1}\Sigma)$.

$$\begin{aligned}
\mathtt{tr}(\mathsf{S}^{-1}\Sigma) &= \mathtt{tr}(\mathsf{S}^{-1}\Psi\Psi^{\mathsf{T}}) = \mathtt{tr}(\Psi^{\mathsf{T}}\mathsf{S}^{-1}\Psi) = \rho^{-1}(\mathtt{tr}(\bar{\mathsf{K}}) - \mathtt{tr}(\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}}\Phi\mathsf{B}\Phi^{\mathsf{T}}\Phi\mathsf{J})) \\
&= \rho^{-1}(\mathtt{tr}(\bar{\mathsf{K}}) - \mathtt{tr}(\bar{\mathsf{K}}\mathsf{V}_q(\Lambda_q^{-1} - \rho\Lambda_q^{-2})\mathsf{V}_q^{\mathsf{T}}\bar{\mathsf{K}}) \\
&= \rho^{-1}(\mathtt{tr}(b\mathsf{K}) - \mathtt{tr}(\mathsf{V}_q\Lambda_q\Lambda_q^{-1}(\mathtt{I}_q - \rho\Lambda_q^{-1})\mathsf{V}_q^{\mathsf{T}}\Lambda_q\mathsf{V}_q^{\mathsf{T}})) \\
&= \rho^{-1}(\mathtt{tr}(\bar{\mathsf{K}}) - \mathtt{tr}(\mathsf{V}_q(\Lambda_q - \rho\mathtt{I}_q)\mathsf{V}_q^{\mathsf{T}})) = \rho^{-1}(\mathtt{tr}(\bar{\mathsf{K}}) - \mathtt{tr}(\Lambda_q - \rho\mathtt{I}_q)) \\
&= \rho^{-1}(\mathtt{tr}(\bar{\mathsf{K}}) - \textstyle\sum_{i=1}^{q}\lambda_i) + q.
\end{aligned}$$

Also, using Lemma 4.2 in Appendix 4.I, the determinant of S is given by

$$|\mathsf{S}| = \rho^{f-q}|\mathsf{M}| = \rho^{f-q}|\Lambda_q| = \rho^{f-q}\prod_{i=1}^{q}\lambda_i.$$

Appendix 4.III: Kernel selection

Only those functions satisfying the Mercer's Theorem [176] can be used as kernel functions. In general, the kernel function lies in some parameterized function family. Denote the parameter of interest by θ. For example, θ can be the polynomial degree in the polynomial kernel, or the kernel width in the Gaussian kernel. The choice of θ remains an open question with the reason being that there is no systematic criteria to judge the goodness. Again, we only focus on the Gaussian kernel case; so $\theta = \sigma$ and $f = \infty$.

It seems that PKPCA offers a systematic ML principle to follow, i.e., picking the σ which maximizes the likelihood or log-likelihood. However, it turns out that the ML principle fails as it has an inherent bias towards a large σ value. The log-likelihood $\mathcal{L}$ is given by:

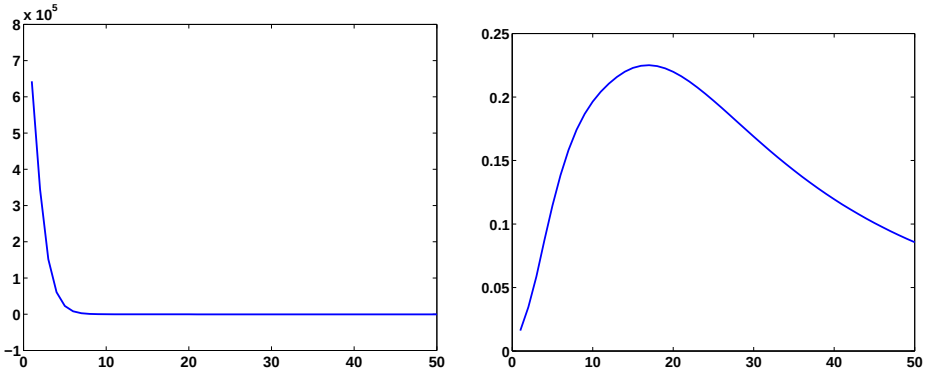
$$\begin{aligned}\mathcal{L} &= -\tfrac{Nf}{2}\log(2\pi) - \tfrac{N}{2}\log(|\mathsf{S}|) - \tfrac{1}{2}\textstyle\sum_{n=1}^{N}(\phi(\mathsf{x}_n) - \bar{\phi}_0)^{\mathsf{T}}\mathsf{S}^{-1}(\phi(\mathsf{x}) - \bar{\phi}_0) \\ &\propto -\tfrac{N}{2}\textstyle\sum_{i=1}^{q}\log(\lambda_i) - \tfrac{N}{2}\mathtt{tr}(\mathsf{S}^{-1}\Sigma) \\ &\propto -\tfrac{N}{2}\textstyle\sum_{i=1}^{q}\log(\lambda_i) - \tfrac{N}{2}\rho^{-1}(\mathtt{tr}(\bar{\mathsf{K}}) - \textstyle\sum_{i=1}^{q}\lambda_i)\end{aligned}$$

By defining the following quantity:

$$\mathcal{E}(\sigma) = -\frac{2}{N}\mathcal{L} \propto \sum_{i=1}^{q} log(\lambda_i) + \rho^{-1}(\mathtt{tr}(\bar{\mathsf{K}}) - \mathtt{tr}(\Lambda_q)), \tag{4.22}$$

the goal is to

$$\min_{\sigma} \quad \mathcal{E}(\sigma) \quad \boxed{\text{subject to}} \quad \lambda_q(\sigma) > \rho.$$



(a) (b)

Figure 4.10: (a) The curve of $\mathcal{E}(\sigma)$. (b) The curve of $\lambda_1(\sigma)$. We have set $q = 30$ and $\rho = 1e^{-6}$.

We now show how it works. Figure 4.10(a) presents the curve of $\mathcal{E}(\sigma)$ obtained using (4.22) for the C-shaped data (Figure 4.1(a)), which always has a bias toward favoring a large σ. This is not surprising since a large σ makes the matrix K_0 close to a matrix of ones; hence the matrix K becomes close to a matrix of zeros, the data variation is reduced, and therefore the likelihood is increased. If σ goes to ∞, all data essentially reduces to one point in the feature space. This is also explained

by Williams in [183]. Williams [183] has also studied the ratio of the sum of the top q eigenvalues to that of all eigenvalues, and discovered the same bias.

We propose an alternative approach by examining the first eigenvalue, which equals to the maximum variance of the projected data where the projection occurs in the feature space induced by the kernel function. Figure 4.10(b) shows the plot of the first eigenvalue $\lambda_1(\sigma)$ against σ. There is a unique maximum. We pick this as our kernel width. This choice of the kernel width seems to have a close relationship with the assumption on the Jacobi matrix in (4.4.1). Figure 4.11(a) present the map of $\log(\delta_\phi(\mathsf{x}))$ for the single C-shape (with $\sigma = 3$) and Figure 4.11(b) the contour plots of $\tilde{\delta}_\phi(\mathsf{x})$. The map is very granular and the uniformity inside the C-shaped region disappears. Figure 4.11(c) shows the map of $\log(\delta_\phi(\mathsf{x}))$ with $\sigma = 36$ and Figure 4.11(d) the contour plots of $\tilde{\delta}_\phi(\mathsf{x})$. Now, the map is over-smoothed (compare the intensity change inside and outside the C-shaped region with that of Figure 4.5(b)).

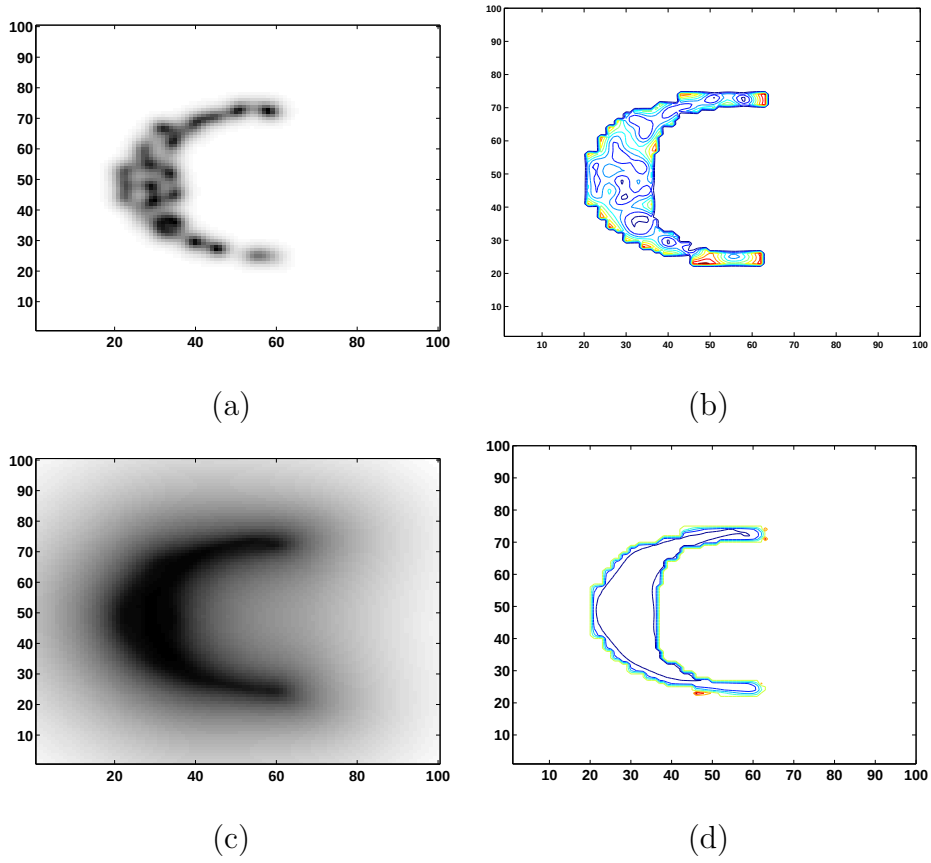


Figure 4.11: (a) The map of $\log(\delta_\phi)$ and (b) the contour plots of $\tilde{\delta}_\phi$ inside the C-shaped region, when $\sigma = 3$. (c) The map of $\log(\delta_\phi)$ and (d) the contour plots of $\tilde{\delta}_\phi$ inside the C-shaped region, when $\sigma = 36$.

Chapter 5

Probability Distances in Reproducing Kernel Hilbert Space

Probabilistic distance measures, defined as the distances between two probability distributions, are important quantities and find their uses in many research areas such as probability and statistics, pattern recognition, information theory, communication and so on. In statistics, the probabilistic distances are often used in asymptotic analysis. In pattern recognition, pattern separability is usually calibrated using probabilistic distance measures [5] like Chernoff distance and Bhattarchayya distance because they provide bounds for probability of error in a pattern classification problem. In information theory, mutual information, a special example of Kullback-Leibler divergence or relative entropy [4] is a fundamental quantity related to the channel capacity. In communication, divergence and Bhattarchayya distance measures are used for signal selection [156].

Direct evaluation of probabilistic distances is nontrivial since they involve inte-

grals. Only within certain parametric families, say the widely-used Gaussian density, we have analytic expressions for probability distances. However, the Gaussian density employs only up to second-order statistics and its modeling capacity is linear and hence rather limited when confronted with a nonlinear data structure. By nonlinear data structure, we mean that if conventional linear modeling techniques such as fitting the Gaussian density are used, the responses are badly approximated. To absorb the nonlinearity, mixture models or non-parametric densities are used in practice. For such cases, one has to resort to numerical methods for computing the probabilistic distances. Such computation is not robust in nature since two approximations are invoked: one in estimating the density and the other in evaluating the numerical integral.

In this chapter, we model the nonlinearity through a different approach: kernel methods. The essence of kernel methods is to combine a linear algorithm with a nonlinear embedding, which maps the data from the original vector space to the reproducing kernel Hilbert space (RKHS). But, we need not require any explicit knowledge of the nonlinear mapping function as long as we can cast our computations into dot product evaluations. Since a nonlinear function is used, albeit in an implicit fashion, we achieve a new paradigm to study these distances and investigate their uses in a different space.

Clearly, our computation depends on the assumption that the data is Gaussian in RKHS. This assumption has been implicitly used in many kernel methods such as [172, 181]. In [181], PCA operates on the RKHS. Even though it seems that PCA needs only the covariance matrix without the Gaussianity assumption, it is the deviation of the data from Gaussianity in the original space that drives us to search for the principal components in the nonlinear feature space. In [172],

discriminant analysis is performed on the feature space. It is well known that discriminant analysis originated as a two-class problem by assuming that each class is distributed as Gaussian with a common covariance matrix. Recently, the Gaussianity is directly adopted in the literature [170, 171, 175]. In [170, 171], it is used to compute the mutual information between two Gaussian random vectors in RKHS. In [175], it is used to construct the so-called Bhattacharyya kernel. In fact, the validity of this assumption boils down to a Gaussian process argument [175]. However, since the induced RKHS is certainly limited by the number of available samples, a regularized covariance matrix is needed in [170, 171]. We also propose a way to regularize the covariance matrix in this chapter.

Chapter organization

This chapter is organized as follows. Section 5.1 introduces several probabilistic distances often used in the literature and Section 5.2 presents a method for estimating the first- and second-order statistics for the data in RKHS. Section 5.3 elaborates the derivations of the probabilistic distances in the RKHS and their limiting behavior. Section 5.4 demonstrates the feasibility and efficiency of the proposed measures using experiments on synthetic and real examples.

5.1 Probabilistic Distances in $\mathcal{R}^d$

Consider a two-class problem and suppose that class 1 has prior probability π_1 and class-dependent density $\mathtt{p}_1(\mathtt{x})$ and class 2 has prior probability π_2 and class-dependent density $\mathtt{p}_2(\mathtt{x})$, both defined on $\mathcal{R}^d$. The following defines a list of probabilistic distance measures often found in the literature [5]:

- Chernoff distance [151]

$$J_C(\mathtt{p}_1, \mathtt{p}_2) = -\log\{\int_{\mathtt{X}} \mathtt{p}_1^{\alpha}(\mathtt{x})\mathtt{p}_2^{1-\alpha}(\mathtt{x})\mathtt{dx}\}; \tag{5.1}$$

- Bhattacharyya distance [150]

$$J_B(\mathtt{p}_1, \mathtt{p}_2) = -\log\{\int_{\mathtt{X}} [\mathtt{p}_1(\mathtt{x})\mathtt{p}_2(\mathtt{x})]^{1/2}\mathtt{dx}\}; \tag{5.2}$$

- Hellinger or Matusita distance [161]

$$J_T(\mathtt{p}_1, \mathtt{p}_2) = \{\int_{\mathtt{X}} [\sqrt{\mathtt{p}_1(\mathtt{x})} - \sqrt{\mathtt{p}_2(\mathtt{x})}]^2\mathtt{dx}\}^{1/2}; \tag{5.3}$$

- The symmetric divergence [13]

$$J_D(\mathtt{p}_1, \mathtt{p}_2) = \int_{\mathtt{X}} [\mathtt{p}_1(\mathtt{x}) - \mathtt{p}_2(\mathtt{x})] \log \frac{\mathtt{p}_1(\mathtt{x})}{\mathtt{p}_2(\mathtt{x})}\mathtt{dx}; \tag{5.4}$$

- Patrick-Fisher distance [163]

$$J_P(\mathtt{p}_1, \mathtt{p}_2) = \{\int_{\mathtt{X}} [\mathtt{p}_1(\mathtt{x})\pi_1 - \mathtt{p}_2(\mathtt{x})\pi_2]^2\mathtt{dx}\}^{1/2}; \tag{5.5}$$

- Lissack-Fu distance [158]

$$J_L(\mathtt{p}_1, \mathtt{p}_2) = \int_{\mathtt{X}} |\mathtt{p}_1(\mathtt{x})\pi_1 - \mathtt{p}_2(\mathtt{x})\pi_2|^{\alpha}\mathtt{p}^{1-\alpha}(\mathtt{x})\mathtt{dx}; \tag{5.6}$$

- Kolmogorov distance [147]

$$J_K(\mathtt{p}_1, \mathtt{p}_2) = \int_{\mathtt{X}} |\mathtt{p}_1(\mathtt{x})\pi_1 - \mathtt{p}_2(\mathtt{x})\pi_2|\mathtt{dx}; \tag{5.7}$$

where $0 < \alpha < 1$ and $\mathtt{p}(\mathtt{x}) = \mathtt{p}_1(\mathtt{x})\pi_1 + \mathtt{p}_2(\mathtt{x})\pi_2$.

It is obvious that (i) the Bhattacharyya distance is a special case of the Chernoff distance with $\alpha = 1/2$; (ii) the Hellinger distance is related to the Bhattacharyya distance as follows:

$$J_T = \{2[1 - \exp(-J_B)]\}^{1/2}; \tag{5.8}$$

and (iii) the Kolmogorov distance is a special case of the Lissack-Fu distance with $\alpha = 1$. Some interesting properties of these distances can be found in [5, 156]

In particular, the symmetric divergence is of great interest in the information theory literature [4] and has a close connection with the famous Kullback-Leibler (KL) divergence [13]. The KL divergence or relative entropy between two densities $\mathtt{p}_1(\mathtt{x})$ and $\mathtt{p}_2(\mathtt{x})$ is given by

$$J_R(\mathtt{p}_1||\mathtt{p}_2) = \int_{\mathtt{X}} \mathtt{p}_1(\mathtt{x}) \log\{\frac{\mathtt{p}_1(\mathtt{x})}{\mathtt{p}_2(\mathtt{x})}\}\mathtt{dx}. \tag{5.9}$$

However, the KL divergence is not a true metric because neither the symmetry constraint nor the triangle inequality is satisfied. The symmetric divergence, which is *symmetric*, is equal to

$$J_D(\mathtt{p}_1, \mathtt{p}_2) = J_R(\mathtt{p}_1||\mathtt{p}_2) + J_R(\mathtt{p}_2||\mathtt{p}_1). \tag{5.10}$$

As mentioned earlier, computing the above probabilistic distance measures is nontrivial. Only within certain parametric families, say the Gaussian density, we know how to analytically compute some of the above defined distance measures. Suppose that $\mathtt{N}(\mathtt{x}; \mu, \Sigma)$ is a multivariate Gaussian density defined as

$$\mathtt{N}(\mathtt{x}; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\{-\frac{1}{2}(\mathtt{x}-\mu)^{\mathtt{T}}\Sigma^{-1}(\mathtt{x}-\mu)\}, \tag{5.11}$$

where $\mathtt{x} \in \mathcal{R}^d$ and $|.|$ is the matrix determinant. With $\mathtt{p}_1(\mathtt{x}) = \mathtt{N}(\mathtt{x}; \mu_1, \Sigma_1)$ and $\mathtt{p}_2(\mathtt{x}) = \mathtt{N}(\mathtt{x}; \mu_2, \Sigma_2)$, we evaluate some of the above probabilistic distance measures as follows:

- Chernoff distance

$$J_C(\mathtt{p}_1, \mathtt{p}_2) = \frac{1}{2}\alpha(1-\alpha)(\mu_1-\mu_2)^{\mathtt{T}}[(1-\alpha)\Sigma_1+\alpha\Sigma_2]^{-1}(\mu_1-\mu_2)+\frac{1}{2}\log\frac{|(1-\alpha)\Sigma_1+\alpha\Sigma_2|}{|\Sigma_1|^{1-\alpha}|\Sigma_2|^{\alpha}}; \tag{5.12}$$

- Bhattacharyya distance

$$J_B(\mathtt{p}_1, \mathtt{p}_2) = \frac{1}{8}(\mu_1 - \mu_2)^{\mathtt{T}}[\frac{\Sigma_1 + \Sigma_2}{2}]^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{|\frac{1}{2}(\Sigma_1 + \Sigma_2)|}{|\Sigma_1|^{1/2}|\Sigma_2|^{1/2}}; \quad (5.13)$$

- Kullback-Leibler divergence or relative entropy

$$J_R(\mathtt{p}_1||\mathtt{p}_2) = \frac{1}{2}(\mu_1 - \mu_2)^{\mathtt{T}}\Sigma_2^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\log\frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2}\mathtt{tr}[\Sigma_1\Sigma_2^{-1} - \mathtt{I}_d]; \quad (5.14)$$

- The symmetric divergence

$$J_D(\mathtt{p}_1, \mathtt{p}_2) = \frac{1}{2}(\mu_1 - \mu_2)^{\mathtt{T}}({\Sigma_1}^{-1} + {\Sigma_2}^{-1})(\mu_1 - \mu_2) + \frac{1}{2}\mathtt{tr}[{\Sigma_1}^{-1}\Sigma_2 + {\Sigma_2}^{-1}\Sigma_1 - 2\mathtt{I}_d]; \quad (5.15)$$

- Patrick-Fisher distance

$$\begin{aligned} J_P(\mathtt{p}_1, \mathtt{p}_2) &= [(2\pi)^d|2\Sigma_1|]^{-1/2} + [(2\pi)^d|2\Sigma_2|]^{-1/2} \quad (5.16) \\ &- 2[(2\pi)^d|\Sigma_1 + \Sigma_2|]^{-1/2}\exp\{-\frac{1}{2}(\mu_1 - \mu_2)^{\mathtt{T}}(\Sigma_1 + \Sigma_2)^{-1}(\mu_1 - \mu_2)\}; \end{aligned}$$

where d is the dimensionality of the random vector $\mathtt{x}$ and $\mathtt{tr}[.]$ is the matrix trace. In particular, when the covariance matrices for the two densities are same, i.e., $\Sigma_1 = \Sigma_2 = \Sigma$, the Bhattacharyya distance and the symmetric divergence reduce to the Mahalanobis distance [160]:

$$J_M = J_D = 8J_B = (\mu_1 - \mu_2)^{\mathtt{T}}\Sigma^{-1}(\mu_1 - \mu_2). \quad (5.17)$$

In this chapter, we only focus on the distances defined in (5.12)-(5.15).

5.2 Mean and Covariance Marix in RKHS

5.2.1 First- and second-order statistics

Computing the probabilistic distance measures requires first- and second-order statistics in the RKHS, as shown in Section 5.1. In practice, we have to estimate

these statistics from a set of training samples. Chapter 4 presented a detailed treatment of this topic and here we recapitulate some important points.

Suppose that $\{\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_N\}$ are given observations in the original data space $\mathcal{R}^d$. We operate in the RKHS $\mathcal{R}^f$ induced by a nonlinear mapping function $\phi : \mathcal{R}^d \rightarrow \mathcal{R}^f$, where $f > d$ and f could even be infinite. The training samples in $\mathcal{R}^f$ are denoted by $\Phi_{f\times N} = [\phi_1, \phi_2, ..., \phi_N]$, where $\phi_n \equiv \phi(\mathsf{x}_n) \in \mathcal{R}^f$.

Using the maximum likelihood estimate (MLE) principle, the mean μ and the covariance matrix Σ are estimated as

$$\mu = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathsf{x}_n) = \Phi\mathsf{e}; \quad \Sigma = \frac{1}{N}\sum_{n=1}^{N}(\phi_n - \mu)(\phi_n - \mu)^{\mathsf{T}} = \Phi\mathsf{J}\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}} = \Psi\Psi^{\mathsf{T}}, \quad (5.18)$$

where the weight vector $\mathsf{e}_{N\times 1} \equiv N^{-1}\mathsf{1}$ with $\mathsf{1}$ being a vector of 1's, $\Psi \equiv \Phi\mathsf{J}$, and J is an $N \times N$ centering matrix given as

$$\mathsf{J} \equiv N^{-1/2}(\mathtt{I}_N - \mathsf{e}\mathsf{1}^{\mathsf{T}}). \quad (5.19)$$

5.2.2 Covariance matrix approximation

The covariance matrix Σ in (5.18) is rank-deficient since $f > N$. Thus, inverting such a matrix is impossible and an approximation to the covariance matrix is necessary. Later in Section 5.3 we show that this approximation can be exact by studying the limiting behavior.

Such an approximation S should possess the following features:

- It keeps the principal structure of the covariance matrix Σ. In other words, the dominant eigenvalues and eigenvectors of Σ and S should be the same.

- It is compact and regularized. The compactness is inspired by the fact that the smallest eigenvalues of the covariance matrix are very close to zero. The regularity is always desirable in the approximation theory.

- It is easy to invert.

As shown in Chapter 4, we suggested the following approximation form:

$$\mathsf{S} = \rho \mathtt{I}_f + \Phi \mathsf{J}\mathsf{Q}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}\Phi^{\mathsf{T}} = \rho \mathtt{I}_f + \Phi \mathsf{A} \Phi^{\mathsf{T}}, \tag{5.20}$$

where Q is an $N \times r$ matrix, $\mathsf{A} \equiv \mathsf{J}\mathsf{Q}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}$, and $\rho > 0$ is a pre-specified constant. Typically, $q << N << f$. Firstly, when

$$\mathsf{Q} = \mathsf{V}_q(\mathtt{I}_q - \rho \Lambda_q^{-1})^{1/2}\mathsf{R},$$

where V_q and Λ_q encode the top q eigenvectors and eigenvalues of the $\bar{\mathsf{K}}$ matrix, the top q eigenpairs of Σ are maintained. Hence, if $\rho = 0$, we exactly maintain the subspace containing the top q eigenpairs. Secondly, S is regularized and its compactness is achieved through the Q matrix. Finally, inverting S is also easy by using the Woodbury formula [8],

$$\mathsf{S}^{-1} = (\rho \mathtt{I}_f + \mathsf{W}\mathsf{W}^{\mathsf{T}})^{-1} = \rho^{-1}(\mathtt{I}_f - \mathsf{W}\mathsf{M}^{-1}\mathsf{W}^{\mathsf{T}}) = \rho^{-1}(\mathtt{I}_f - \Phi \mathsf{B} \Phi^{\mathsf{T}}), \tag{5.21}$$

where $\mathsf{B} \equiv \mathsf{J}\mathsf{Q}\mathsf{M}^{-1}\mathsf{Q}^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}$ and the matrix $\mathsf{M}_{q \times q}$ is

$$\mathsf{M} \equiv \rho \mathtt{I}_q + \mathsf{W}^{\mathsf{T}}\mathsf{W} = \rho \mathtt{I}_q + \mathsf{Q}^{\mathsf{T}}\bar{\mathsf{K}}\mathsf{Q}. \tag{5.22}$$

After obtaining Q, it is easy to check that the following equations hold:

$$\mathsf{M} = \Lambda_q, \quad |\mathsf{M}| = |\Lambda_q| = \prod_{i=1}^{q} \lambda_i, \quad \mathsf{M}^{-1} = \Lambda_q^{-1}, \quad |\mathsf{S}| = \rho^{f-q}|\Lambda_q|. \tag{5.23}$$

$$\mathsf{A} = \mathsf{J}\mathsf{V}_q(\mathtt{I}_q - \rho \Lambda_q^{-1})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}, \quad \mathsf{B} = \mathsf{J}\mathsf{V}_q(\Lambda_q^{-1} - \rho \Lambda_q^{-2})\mathsf{V}_q^{\mathsf{T}}\mathsf{J}^{\mathsf{T}}. \tag{5.24}$$

$$\mathtt{tr}[\mathsf{A}\mathsf{K}] = \mathtt{tr}[\Lambda_q] - \rho q, \quad \mathtt{tr}[\mathsf{B}\mathsf{K}] = q - \rho \mathtt{tr}[\Lambda_q^{-1}]. \tag{5.25}$$

5.3 The Probabilistic Distances in RKHS

Since the probabilistic distances involve two densities $\mathtt{p}_1$ and $\mathtt{p}_2$, we need two sets of training samples: Φ_1 for $\mathtt{p}_1$ and Φ_2 for $\mathtt{p}_2$. For each density $\mathtt{p}_i$, we can find its corresponding $\mathtt{e}_i$, J_i, μ_i, Σ_i, K_i, S_i, $\mathsf{V}_{q_i,i}$, $\Lambda_{q_i,i} = \mathsf{D}[\lambda_{1,i}, \lambda_{2,i}, \ldots, \lambda_{q_i,i}]$, A_i, B_i, etc., by keeping the top q_i principal components. In general, we can have $q_1 \neq q_2$ and $N_1 \neq N_2$ with N_i being the number of samples for the i^{th} density. In addition, we define the following dot product matrix:

$$\begin{bmatrix} \Phi_1^{\mathsf{T}} \\ \Phi_2^{\mathsf{T}} \end{bmatrix} [\Phi_1 \ \Phi_2] = \begin{bmatrix} \Phi_1^{\mathsf{T}}\Phi_1 & \Phi_1^{\mathsf{T}}\Phi_2 \\ \Phi_2^{\mathsf{T}}\Phi_1 & \Phi_2^{\mathsf{T}}\Phi_2 \end{bmatrix} \equiv \begin{bmatrix} \mathsf{K}_{11} & \mathsf{K}_{12} \\ \mathsf{K}_{21} & \mathsf{K}_{22} \end{bmatrix}, \tag{5.26}$$

where $\mathsf{K}_{ij} \equiv \Phi_i^{\mathsf{T}}\Phi_j$ and $\mathsf{K}_{21} = \mathsf{K}_{12}^{\mathsf{T}}$.

5.3.1 The Chernoff distance and the Bhattarchayya distance

As mentioned before, the Bhattarchayya distance is a special case of Chernoff distance with $\alpha = 1/2$. Hence, we focus only on the Chernoff distance.

The key quantity in computing the Chernoff distance is $\alpha_1\mathsf{S}_1 + \alpha_2\mathsf{S}_2$ with $\alpha_1 + \alpha_2 = 1$. We now analyze this quantity in detail.

$$\begin{aligned}
\alpha_1\mathsf{S}_1 + \alpha_2\mathsf{S}_2 &= \alpha_1\{\rho\mathtt{I}_f + \Phi_1\mathsf{A}_1\Phi_1^{\mathsf{T}}\} + \alpha_2\{\rho\mathtt{I}_f + \Phi_2\mathsf{A}_2\Phi_2^{\mathsf{T}}\} \\
&= \rho\mathtt{I}_f + \alpha_1\Phi_1\mathsf{A}_1\Phi_1^{\mathsf{T}} + \alpha_2\Phi_2\mathsf{A}_2\Phi_2^{\mathsf{T}} \\
&= \rho\mathtt{I}_f + [\Phi_1 \ \Phi_2] \begin{bmatrix} \alpha_1\mathsf{A}_1 & 0 \\ 0 & \alpha_2\mathsf{A}_2 \end{bmatrix} \begin{bmatrix} \Phi_1^{\mathsf{T}} \\ \Phi_2^{\mathsf{T}} \end{bmatrix} \\
&= \rho\mathtt{I}_f + [\Phi_1 \ \Phi_2] \begin{bmatrix} \alpha_1\mathsf{J}_1\mathsf{Q}_1\mathsf{Q}_1^{\mathsf{T}}\mathsf{J}_1^{\mathsf{T}} & 0 \\ 0 & \alpha_2\mathsf{J}_2\mathsf{Q}_2\mathsf{Q}_2^{\mathsf{T}}\mathsf{J}_2^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \Phi_1^{\mathsf{T}} \\ \Phi_2^{\mathsf{T}} \end{bmatrix}
\end{aligned}$$

$$= \rho \mathtt{I}_f + [\Phi_1 \ \Phi_2] \mathtt{A}_{ch} \begin{bmatrix} \Phi_1^{\mathtt{T}} \\ \Phi_2^{\mathtt{T}} \end{bmatrix}, \tag{5.27}$$

where the matrix $\mathtt{A}_{ch}$ is rank-deficient since $\mathtt{A}_{ch} = \mathtt{P}\mathtt{P}^{\mathtt{T}}$ with

$$\mathtt{P}_{(N_1+N_2)\times(q_1+q_2)} \equiv \begin{bmatrix} \sqrt{\alpha_1}\mathtt{J}_1\mathtt{Q}_1 & \mathtt{0} \\ \mathtt{0} & \sqrt{\alpha_2}\mathtt{J}_2\mathtt{Q}_2 \end{bmatrix}. \tag{5.28}$$

Therefore, the matrix $\alpha_1\mathtt{S}_1 + \alpha_2\mathtt{S}_2$ is of such a form that we can easily find its determinant and inverse.

The determinant $|\alpha_1\mathtt{S}_1 + \alpha_2\mathtt{S}_2|$ is given by

$$|\alpha_1\mathtt{S}_1 + \alpha_2\mathtt{S}_2| = \rho^{f-(q_1+q_2)}|\rho\mathtt{I}_{q_1+q_2} + \mathtt{L}| = \rho^{f-(q_1+q_2)} \prod_{i=1}^{q_1+q_2} (\tau_i + \rho), \tag{5.29}$$

where $\{\tau_i;\ i = 1, \ldots, q_1 + q_2\}$ are eigenvalues of the $\mathtt{L}$ matrix. The $\mathtt{L}$ matrix is given by

$$\begin{aligned} \mathtt{L}_{(q_1+q_2)\times(q_1+q_2)} &= \mathtt{P}^{\mathtt{T}} \begin{bmatrix} \Phi_1^{\mathtt{T}} \\ \Phi_2^{\mathtt{T}} \end{bmatrix} [\Phi_1 \ \Phi_2]\mathtt{P} = \mathtt{P}^{\mathtt{T}} \begin{bmatrix} \mathtt{K}_{11} & \mathtt{K}_{12} \\ \mathtt{K}_{21} & \mathtt{K}_{22} \end{bmatrix} \mathtt{P} \\ &= \begin{bmatrix} \alpha_1\mathtt{Q}_1^{\mathtt{T}}\mathtt{J}_1^{\mathtt{T}}\mathtt{K}_{11}\mathtt{J}_1\mathtt{Q}_1 & \sqrt{\alpha_1\alpha_2}\mathtt{Q}_1^{\mathtt{T}}\mathtt{J}_1^{\mathtt{T}}\mathtt{K}_{12}\mathtt{J}_2\mathtt{Q}_2 \\ \sqrt{\alpha_1\alpha_2}\mathtt{Q}_2^{\mathtt{T}}\mathtt{J}_2^{\mathtt{T}}\mathtt{K}_{21}\mathtt{J}_1\mathtt{Q}_1 & \alpha_2\mathtt{Q}_2^{\mathtt{T}}\mathtt{J}_2^{\mathtt{T}}\mathtt{K}_{22}\mathtt{J}_2\mathtt{Q}_2 \end{bmatrix} \\ &= \begin{bmatrix} \alpha_1\{\Lambda_{q_1,1} - \rho\mathtt{I}_{q_1}\} & \sqrt{\alpha_1\alpha_2}\mathtt{L}_{12} \\ \sqrt{\alpha_1\alpha_2}\mathtt{L}_{12}^{\mathtt{T}} & \alpha_2\{\Lambda_{q_2,2} - \rho\mathtt{I}_{q_2}\} \end{bmatrix}, \end{aligned} \tag{5.30}$$

with $\mathtt{L}_{12} \equiv \mathtt{Q}_1^{\mathtt{T}}\mathtt{J}_1^{\mathtt{T}}\mathtt{K}_{12}\mathtt{J}_2\mathtt{Q}_2$.

The inverse $\{\alpha_1\mathtt{S}_1 + \alpha_2\mathtt{S}_2\}^{-1}$ is given by

$$\{\alpha_1\mathtt{S}_1 + \alpha_2\mathtt{S}_2\}^{-1} = \rho^{-1}\{\mathtt{I}_f - [\Phi_1 \ \Phi_2]\mathtt{B}_{ch} \begin{bmatrix} \Phi_1^{\mathtt{T}} \\ \Phi_2^{\mathtt{T}} \end{bmatrix}\}, \quad \mathtt{B}_{ch} = \mathtt{P}(\rho\mathtt{I}_{q_1+q_2} + \mathtt{L})^{-1}\mathtt{P}^{\mathtt{T}}. \tag{5.31}$$

We now show how to compute the following two quantities in (5.12):

$$\begin{aligned}\mu_i^{\mathsf{T}}\{\alpha_1\mathsf{S}_1+\alpha_2\mathsf{S}_2\}^{-1}\mu_j &= \mathsf{e}_i^{\mathsf{T}}\Phi_i^{\mathsf{T}}\rho^{-1}\{\mathtt{I}_f-[\Phi_1\ \Phi_2]\mathsf{B}_{ch}\begin{bmatrix}\Phi_1^{\mathsf{T}}\\ \Phi_2^{\mathsf{T}}\end{bmatrix}\}\Phi_j\mathsf{e}_j \qquad (5.32)\\ &= \rho^{-1}\{\mathsf{e}_i^{\mathsf{T}}\mathsf{K}_{ij}\mathsf{e}_j-\mathsf{e}_i^{\mathsf{T}}[\mathsf{K}_{i1}\ \mathsf{K}_{i2}]\mathsf{B}_{ch}\begin{bmatrix}\mathsf{K}_{1j}\\ \mathsf{K}_{2j}\end{bmatrix}\mathsf{e}_j\}\equiv\rho^{-1}\xi_{ij},\end{aligned}$$

$$\begin{aligned}\log\frac{|\alpha_1\mathsf{S}_1+\alpha_2\mathsf{S}_2|}{|\mathsf{S}_1|^{\alpha_1}|\mathsf{S}_2|^{\alpha_2}} &= \sum_{i=1}^{q_1+q_2}\log(\rho+\tau_i)+(f-q_1-q_2)\log(\rho)\\ &- \alpha_1\{\sum_{i=1}^{q_1}\log(\lambda_{i,1})+(f-q_1)\log(\rho)\}\\ &- \alpha_2\{\sum_{i=1}^{q_2}\log(\lambda_{i,2})+(f-q_2)\log(\rho)\}\\ &= \alpha_1\sum_{i=1}^{q_1+q_2}\log\frac{\rho+\tau_i}{\lambda_{i,1}}+\alpha_2\sum_{i=1}^{q_1+q_2}\log\frac{\rho+\tau_i}{\lambda_{i,2}}, \qquad (5.33)\end{aligned}$$

where $\{\lambda_{i,1}; i=1,2,...,q_1\}$ and $\{\lambda_{i,2}; i=1,2,...,q_2\}$ are eigenvalues for S_1 and S_2, respectively. Notice that (i) $\{\lambda_{i,1}; i=q_1+1,...,q_1+q_2\}$ and $\{\lambda_{i,2}; i=q_2+1,...,q_1+q_2\}$, all equal to ρ's, are introduced only for notational convenience; (ii) the infinite dimensionality f in (5.32) and (5.33) disappeared as needed; and (iii) all calculations are based on the Gram matrix defined in (5.26).

Finally, we compute the Chernoff distance as follows (with $\alpha_1=1-\alpha$ and $\alpha_2=\alpha$):

$$2J_C(\mathsf{p}_1,\mathsf{p}_2)=\rho^{-1}\alpha_1\alpha_2\{\xi_{11}+\xi_{22}-2\xi_{12}\}+\alpha_1\sum_{i=1}^{q_1+q_2}\log\frac{\rho+\tau_i}{\lambda_{i,1}}+\alpha_2\sum_{i=1}^{q_1+q_2}\log\frac{\rho+\tau_i}{\lambda_{i,2}}. \qquad (5.34)$$

5.3.2 The KL divergence and the symmetric divergence

Computing the KL divergence in the RKHS is just done by collecting terms like $\mu_i^{\mathsf{T}}\mathsf{S}_j^{-1}\mu_k$ and $\mathtt{tr}\{\mathsf{S}_i\mathsf{S}_j^{-1}\}$.

$$\begin{aligned}\mu_i^{\mathsf{T}}\mathsf{S}_j^{-1}\mu_k &= \mathsf{e}_i^{\mathsf{T}}\Phi_i^{\mathsf{T}}\rho^{-1}(\mathtt{I}_f - \Phi_j\mathsf{B}_j\Phi_j^{\mathsf{T}})\Phi_k\mathsf{e}_k \\ &= \rho^{-1}(\mathsf{e}_i^{\mathsf{T}}\mathsf{K}_{ik}\mathsf{e}_k - \mathsf{e}_i^{\mathsf{T}}\mathsf{K}_{ij}\mathsf{B}_j\mathsf{K}_{jk}\mathsf{e}_k) \equiv \rho^{-1}\theta_{ijk}. \qquad (5.35)\end{aligned}$$

$$\begin{aligned}\mathtt{tr}[\mathsf{S}_i\mathsf{S}_j^{-1}] &= \mathtt{tr}[(\Phi_i\mathsf{A}_i\Phi_i^{\mathsf{T}} + \rho\mathtt{I}_f)\rho^{-1}(\mathtt{I}_f - \Phi_j\mathsf{B}_j\Phi_j^{\mathsf{T}})] \qquad (5.36)\\ &= \rho^{-1}\mathtt{tr}[\Phi_i\mathsf{A}_i\Phi_i^{\mathsf{T}}] - \rho^{-1}\mathtt{tr}[\Phi_i\mathsf{A}_i\Phi_i^{\mathsf{T}}\Phi_j\mathsf{B}_j\Phi_j^{\mathsf{T}}] + f - \mathtt{tr}[\Phi_j\mathsf{B}_j\Phi_j^{\mathsf{T}}] \\ &= \rho^{-1}\mathtt{tr}[\mathsf{A}_i\mathsf{K}_{ii}] - \rho^{-1}\mathtt{tr}[\mathsf{A}_i\mathsf{K}_{ij}\mathsf{B}_j\mathsf{K}_{ji}] + f - \mathtt{tr}[\mathsf{B}_j\mathsf{K}_{jj}] \\ &= \rho^{-1}\mathtt{tr}[\Lambda_{q_i,i}] - q_i - \rho^{-1}\mathtt{tr}[\mathsf{A}_i\mathsf{K}_{ij}\mathsf{B}_j\mathsf{K}_{ji}] + f + \rho\mathtt{tr}[\Lambda_{q_j,j}^{-1}] - q_j \\ &= \rho^{-1}\{\mathtt{tr}[\Lambda_{q_i,i}] - \eta_{ij}\} + \rho\mathtt{tr}[\Lambda_{q_j,j}^{-1}] + f - (q_i + q_j),\end{aligned}$$

where

$$\eta_{ij} \equiv \mathtt{tr}[\mathsf{A}_i\mathsf{K}_{ij}\mathsf{B}_j\mathsf{K}_{ji}].$$

Finally, we obtain the KL divergence and the symmetric divergence in the RKHS by substituting (5.35) and (5.36) into (5.14) and (5.15) with d replaced by f,

$$\begin{aligned}2J_R(\mathtt{p}_1||\mathtt{p}_2) &= \rho^{-1}\{\theta_{121} + \theta_{222} - \theta_{122} - \theta_{221}\} + \{\log|\Lambda_{q_2,2}| - \log|\Lambda_{q_1,1}|\} \qquad (5.37)\\ &+ (q_1 - q_2)\log\rho + \rho^{-1}\{\mathtt{tr}[\Lambda_{q_1,1}] - \eta_{12}\} + \rho\{\mathtt{tr}[\Lambda_{q_2,2}^{-1}]\} - (q_1 + q_2).\end{aligned}$$

$$\begin{aligned}2J_D(\mathtt{p}_1,\mathtt{p}_2) &= \rho^{-1}\{\theta_{111} + \theta_{121} + \theta_{212} + \theta_{222} - \theta_{112} - \theta_{122} - \theta_{211} - \theta_{221}\} \\ &+ \rho^{-1}\{\mathtt{tr}[\Lambda_{q_1,1}] + \mathtt{tr}[\Lambda_{q_2,2}] - \eta_{12} - \eta_{21}\} \\ &+ \rho\{\mathtt{tr}[\Lambda_{q_1,1}^{-1}] + \mathtt{tr}[\Lambda_{q_2,2}^{-1}]\} - 2(q_1 + q_2). \qquad (5.38)\end{aligned}$$

5.3.3 The Patrick-Fisher distance

Given the above derivations in Sections 5.3.1 and 5.3.2, computing the Patrick-Fisher distance $J_P(\mathtt{p}_1, \mathtt{p}_2)$ can be easily done by putting together related terms.

$$\begin{aligned} J_P(\mathtt{p}_1, \mathtt{p}_2) &= [2(2\pi)^f \rho^{f-q_1} \prod_{i=1}^{q_1} \lambda_{i,1}]^{-1/2} + [2(2\pi)^f \rho^{f-q_2} \prod_{i=1}^{q_2} \lambda_{i,2}]^{-1/2} \\ &- 2[2(2\pi)^f \rho^{f-q_1-q_2} \prod_{i=1}^{q_1+q_2} (\rho + \tau_i)]^{-1/2} \exp\{-\rho^{-1}(\xi_{11} + \xi_{22} - 2\xi_{12})\}. \end{aligned}$$

where $\{\tau_i; i = 1, 2, \ldots, q_1 + q_2\}$ are eigenvalues of the L matrix defined in (5.30) with $\alpha = 1/2$.

5.3.4 Limiting behavior

It is interesting to study the behavior of the distances when ρ approaches to zero.

First,

$$\lim_{\rho \to 0} \mathsf{A} = \hat{\mathsf{A}} \equiv \mathsf{J}\mathsf{V}_q \mathsf{V}_q^{\mathsf{T}} \mathsf{J}^{\mathsf{T}}, \quad \lim_{\rho \to 0} \mathsf{B} = \hat{\mathsf{B}} \equiv \mathsf{J}\mathsf{V}_q \Lambda_q^{-1} \mathsf{V}_q^{\mathsf{T}} \mathsf{J}^{\mathsf{T}}, \tag{5.39}$$

Then,

$$\lim_{\rho \to 0} \theta_{ijk} = \hat{\theta}_{ijk} \equiv \mathsf{e}_i^{\mathsf{T}} \mathsf{K}_{ik} \mathsf{e}_k - \mathsf{e}_i^{\mathsf{T}} \mathsf{K}_{ij} \hat{\mathsf{B}}_j \mathsf{K}_{jk} \mathsf{e}_k, \quad \lim_{\rho \to 0} \eta_{ij} = \hat{\eta}_{ij} \equiv \mathtt{tr}[\hat{\mathsf{B}}_i \mathsf{K}_{ij} \hat{\mathsf{A}}_j \mathsf{K}_{ji}]. \tag{5.40}$$

Similarly,

$$\lim_{\rho \to 0} \xi_{ij} = \hat{\xi}_{ij} \equiv = \mathsf{e}_i^{\mathsf{T}} \mathsf{K}_{ij} \mathsf{e}_j - \mathsf{e}_i^{\mathsf{T}} [\mathsf{K}_{i1} \ \mathsf{K}_{i2}] \hat{\mathsf{B}}_{ch} \begin{bmatrix} \mathsf{K}_{1j} \\ \mathsf{K}_{2j} \end{bmatrix} \mathsf{e}_j, \tag{5.41}$$

where $\hat{\mathsf{B}}_{ch} = \lim_{\rho \to 0} \mathsf{B}_{ch}$.

Finally,

$$\lim_{\rho \to 0} \rho J_C(\mathtt{p}_1, \mathtt{p}_2) = \hat{J}_C(\mathtt{p}_1, \mathtt{p}_2), \tag{5.42}$$

$$\lim_{\rho \to 0} \rho J_R(\mathtt{p}_1 || \mathtt{p}_2) = \hat{J}_R(\mathtt{p}_1 || \mathtt{p}_2), \tag{5.43}$$

$$\lim_{\rho \to 0} \rho J_D(\mathtt{p}_1, \mathtt{p}_2) = \hat{J}_D(\mathtt{p}_1, \mathtt{p}_2), \tag{5.44}$$

where

$$2\hat{J}_C(\mathtt{p}_1, \mathtt{p}_2) = \alpha(1-\alpha)\{\hat{\xi}_{11} + \hat{\xi}_{22} - 2\hat{\xi}_{12}\}, \tag{5.45}$$

$$2\hat{J}_R(\mathtt{p}_1||\mathtt{p}_2) = \hat{\theta}_{121} + \hat{\theta}_{222} - \hat{\theta}_{122} - \hat{\theta}_{221} + \mathtt{tr}[\Lambda_{q_1,1}] - \hat{\eta}_{12}, \tag{5.46}$$

$$\begin{aligned} 2\hat{J}_D(\mathtt{p}_1, \mathtt{p}_2) \;\; = \;\; & \hat{\theta}_{111} + \hat{\theta}_{121} + \hat{\theta}_{212} + \hat{\theta}_{222} - \hat{\theta}_{112} - \hat{\theta}_{122} - \hat{\theta}_{211} - \hat{\theta}_{221} \\ & + \mathtt{tr}[\Lambda_{q_1,1}] + \mathtt{tr}[\Lambda_{q_2,1}] - \hat{\eta}_{12} - \hat{\eta}_{21}. \end{aligned} \tag{5.47}$$

When $\alpha = 1/2$, we obtain the limiting distance for the Bhattacharyya distance

$$2\hat{J}_B(\mathtt{p}_1, \mathtt{p}_2) = \frac{1}{4}\{\hat{\xi}_{11} + \hat{\xi}_{22} - 2\hat{\xi}_{12}\}. \tag{5.48}$$

The limiting behavior of the Patrick-Fisher distance $J_P(\mathtt{p}_1, \mathtt{p}_2)$ is not interesting since it involves f, thus we omit its discussion.

As mentioned earlier, when $\rho = 0$ and $q_1 = q_2 = q$, we actually use the subspace of the RKHS containing the top q eigenpairs. Therefore, the derived limiting distances calibrate the pattern separability on this subspace of the RKHS and carry many optimal features their original counterparts possess, yet additionally equipped with a nonlinear embedding.

5.3.5 Kernel for set

A set here is a collection of observations. A kernel for set is a two-input kernel function that takes the two sets as inputs and satisfies the requirement of positive definiteness.

Several kernels for set have emerged in the literature. In [184], Wolf and Shashua proposed the kernel principal angle. The principal angle is defined as the angle between the principal subspaces of the two input sets and then 'kernelized'. In [174], Jebara and Kondor showed that the Bhattacharyya coefficient [156]

that operates the probability distribution defined on the original data space is a kernel. In [175], they extended the Bhattacharyya kernel to operate the probability distribution defined on the RKHS. In [178], Moreno *et. al.* proposed a kernel function based on the Kullback-Leibler divergence in the original data space.

It is obvious that our probabilistic distance measures can be adapted as kernel functions for set. First, the Bhattacharyya kernel defined in [174] differs from the Bhattacharyya distance by $-\log(.)$. Secondly, the adaptation can be in the sense of [178]. Other ways are possible by utilizing the construction rule of kernel functions.

5.4 Experimental Results

In the following experiments with both synthetic examples and a real face recognition application, we will use only the limiting distances, namely $\hat{J}_C(\mathtt{p}_1, \mathtt{p}_2)$ (or $\hat{J}_B(\mathtt{p}_1, \mathtt{p}_2)$), $\hat{J}_R(\mathtt{p}_1||\mathtt{p}_2)$, and $\hat{J}_D(\mathtt{p}_1, \mathtt{p}_2)$, since they do not depend on the choice ρ, which frees us from the burden of choosing ρ. Also, we set $q_1 = q_2 = q$.

5.4.1 Synthetic examples

To fail the KL distance between two Gaussian densities in the original space, we designed four different 2-D densities sharing the same mean (zero mean) and covariance matrix (identity matrix). As shown in Figure 5.1, the four densities are 2-D Gaussian, and 'O'-, 'D'-, and 'X'-shaped uniform densities, where say the 'O'-shaped uniform density means that it is uniform in the 'O'-shaped region and zero outside the region. Figure 5.1 actually shows 300 i.i.d. realizations sampled from these four densities. Due to the same first- and second-order statistics, the proba-

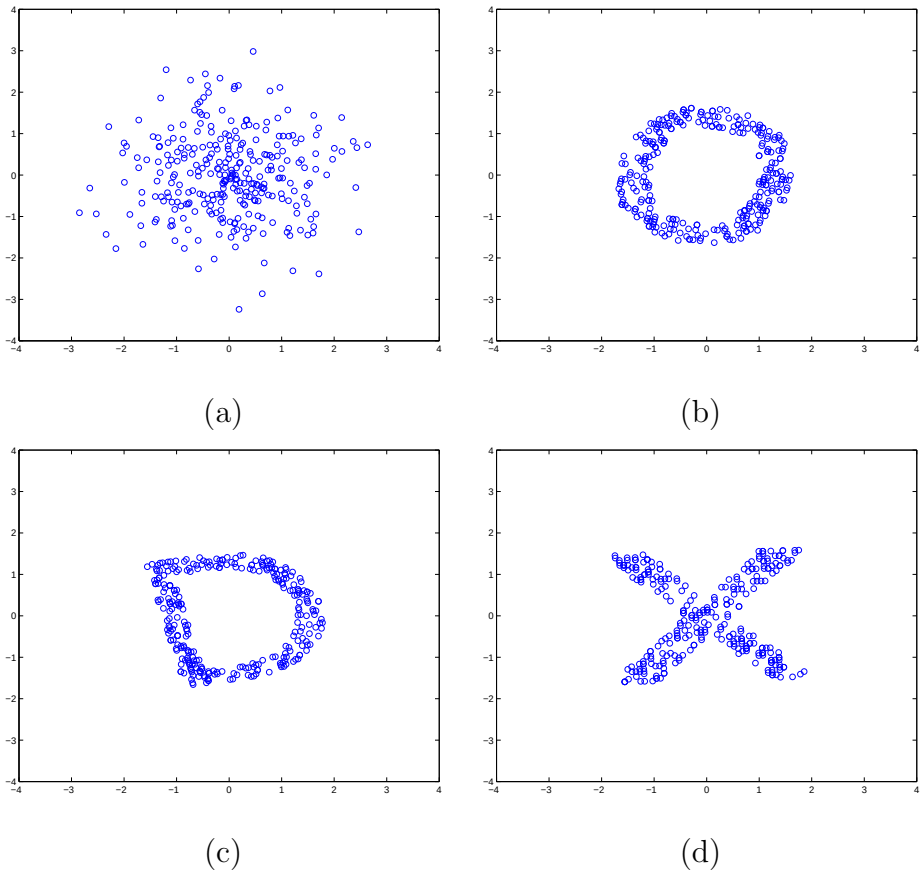
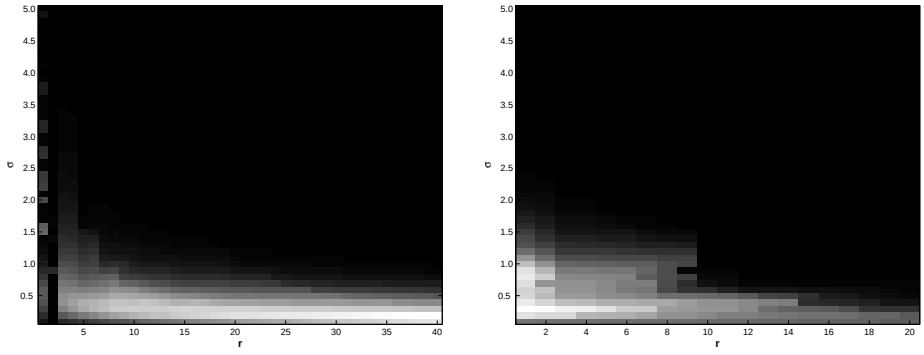


Figure 5.1: 300 i.i.d. realizations of four different densities with the same mean (zero mean) and covariance matrix (identity matrix). (a) 2-D Gaussian. (b) 'O'-shaped uniform.(c) 'D'-shaped uniform. (d) 'X'-shaped uniform.

bilistic distance between any of two densities in the original space is simply zero. This highlights the virtue of a nonlinear mapping that provides us information embedded in higher-order statistics.

Obviously, the probabilistic distances depend on q, the number of eigenpairs, and σ, the RBF kernel width. Figure 5.2 displays $\hat{J}_D$ and $\hat{J}_B$ as a function of q and σ. The effect of σ is biased: It always disfavors a large σ since a large σ tends to pool the data together. For example, when σ is infinite, all data points collapse to one single point in the RKHS and become inseparable. Generally, it



(a) (b)

Figure 5.2: (a) The symmetric divergence $\hat{J}_D(\sigma, q)$ and (b) the Bhatacharyya distance $\hat{J}_B(\sigma, q)$ between the 2-D Gaussian and the 'O'-shaped uniform as a function of σ and q.

is not necessary that a large q (or equivalently using a nonlinear subspace with a large dimension) yields a large distance. A typical subspace yielding the maximum distances is of low-dimensional.

Table 5.1 lists some computed values of the probabilistic distances. It is interesting to observe that when the shapes of two densities are close, their distance is small. For example, 'O' is closest to 'D' among all possible pairs. The closest density to the 2-D Gaussian is the 'O'-shaped uniform.

5.4.2 Face recognition from a group of images

The gallery set consists of 15 sets (one per person) while the probe set consists of 15 new sets of the same people (one per person). In these sets, the people can move their heads freely so that pose and illumination variations abound. The existence of these variations violates the Gaussianity assumption of the original data space used in [91]. Figure 5.3 shows some example faces of the in the 4^{th} gallery person, the 9^{th} gallery person, and the 4^{th} probe person (whose identity is same as the 4^{th}

$\hat{J}_R(\mathtt{p}_1 \| \mathtt{p}_2)$	Gau	'O'	'D'	'X'
Gau	-	.0740	.0782	.0808
'O'	.0584	-	.0281	.0523
'D'	.0670	.0295	-	.0436
'X'	.0944	.0505	.0417	-

(a)

$\hat{J}_B(\mathtt{p}_1, \mathtt{p}_2)$	Gau	'O'	'D'	'X'
Gau	-	.0033	.0037	.0048
'O'	.0033	-	.0021	.0099
'D'	.0037	.0021	-	.0086
'X'	.0048	.0099	.0086	-

(b)

Table 5.1: (a) The KL distances in the RKHS with $\sigma = 1$ and $q = 3$. (b) The Bhatacharyya distances in the RKHS with $\sigma = 0.5$ and $q = 1$. $\mathtt{p}_1$ is listed in the first column and $\mathtt{p}_2$ in the first row.

gallery person). The shown face images of size 32 by 32 are automatically cropped from video sequences (courtesy of [84]) using a flow tracking algorithm.

	Symmetric divergence	Bhatacharyya distance
$\hat{J}(\mathtt{p}_1, \mathtt{p}_2)$ in the RKHS	13/15	13/15
$J(\mathtt{p}_1, \mathtt{p}_2)$ in the original space $\mathcal{R}^d$	11/15	11/15

Table 5.2: The recognition score obtaining using the symmetric divergence and Bhatacharyya distance.

A generic principal component analysis is performed to reduce the dimensionality to 300. Figure 5.3 also plots the first three PCA coefficient of the 4^{th} gallery person, the 9^{th} gallery person, and the 4^{th} probe person. Clearly, the manifolds

are highly nonlinear, which indicates a need for nonlinear modeling.

Table 5.2 reports the recognition rates. The top match with the smallest distance is claimed to be the winner. For comparison, we also implemented the approaches using the symmetric divergence [91] and the Bhatacharyya distance in the original space is used for face recognition. Clearly, using the distances in RKHS yields better result. Out of 15 probe sets, we successfully classified 13 of them. In fact, Figure 5.3 shows a misclassification example in [91], where the 4^{th} probe person is misclassified as the 9^{th} gallery person, while our approach corrects this error.

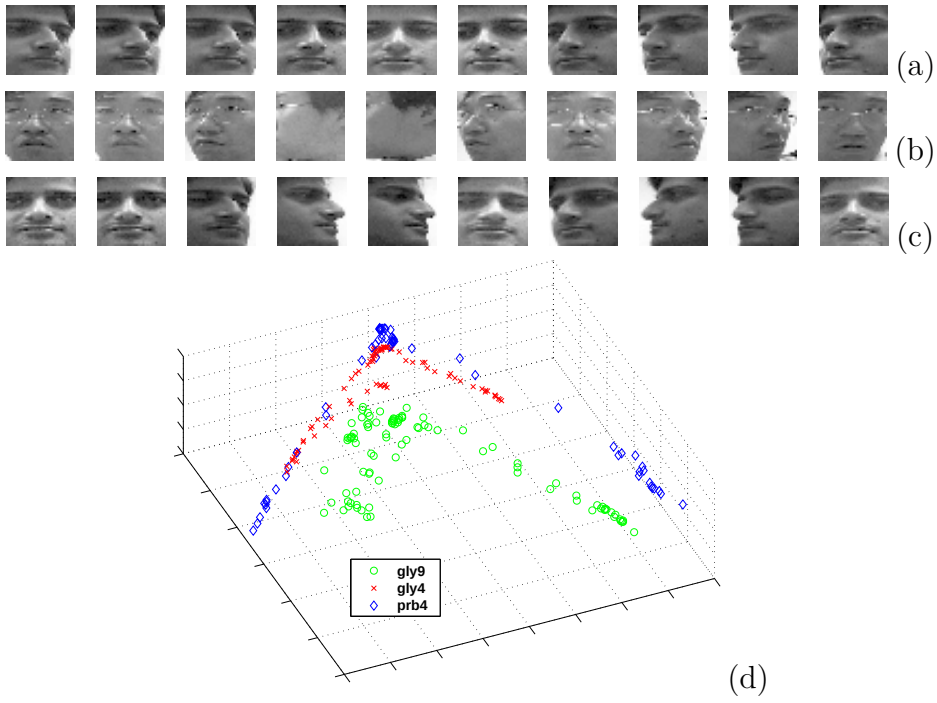


Figure 5.3: Examples of face images in the gallery and probe set. (a) The 4^{th} gallery person in 10 frames (every 8 frames) of a 80-frame sequence. (b) The 9^{th} gallery person in 10 frames (every 10 frames) of a 105-frame sequence.(a) The 4^{th} probe person in 10 frames (every 6 frames) of a 60-frame sequence. (d) The plot of first three PCA coefficients of the above three sets.

Part III: Face Tracking and Recognition from Videos

Chapter 6

Adaptive Visual Tracking

Particle filtering [114, 157, 159, 153, 6] is an inference technique [3, 18] for estimating the unknown motion state, θ_t, from a noisy collection of observations, $\mathtt{y}_{1:t} = \{\mathtt{y}_1, ..., \mathtt{y}_t\}$ 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:

$$\textit{State transition model: } \theta_t = \mathtt{f}_t(\theta_{t-1}, \mathtt{u}_t), \tag{6.1}$$

$$\textit{Observation model: } \mathtt{y}_t = \mathtt{g}_t(\theta_t, \mathtt{v}_t), \tag{6.2}$$

where $\mathtt{u}_t$ is the system noise, $\mathtt{f}_t(.,.)$ characterizes the kinematics, $\mathtt{v}_t$ is the observation noise, and $\mathtt{g}_t(.,.)$ models the observer. The particle filter approximates the posterior distribution $\mathtt{p}(\theta_t|\mathtt{y}_{1:t})$ by a set of weighted particles $\{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^J$. Then, the state estimate $\hat{\theta}_t$ can either be the minimum mean square error (MMSE) estimate,

$$\hat{\theta}_t = \theta_t^{mmse} = \mathtt{E}[\theta_t|\mathtt{y}_{1:t}] \approx J^{-1} \sum_{j=1}^{J} w_t^{(j)} \theta_t^{(j)}, \tag{6.3}$$

where $\mathtt{E}$ the expectation operator, the maximum a posteriori (MAP) estimate,

$$\hat{\theta}_t = \theta_t^{map} = \arg\max_{\theta_t} \mathsf{p}(\theta_t|\mathsf{y}_{1:t}) \approx \arg\max_{\theta_t} w_t^{(j)}, \tag{6.4}$$

or other forms based on $\mathsf{p}(\theta_t|\mathsf{y}_{1:t})$.

The state transition model characterizes the motion change between frames. In a visual tracking problem, it is ideal to have an exact motion model governing the kinematics of the object. In practice, however, approximate models are used. There are two types of approximations commonly found in the literature. (i) One is to learn a motion model directly from a training video [118, 124]. 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. (ii) Secondly, a fixed constant-velocity model with fixed noise variance is fitted as in [109, 133, 135, 185].

$$\theta_t = \theta_{t-1} + \nu_t + \mathsf{u}_t, \tag{6.5}$$

where ν_t is a constant velocity, i.e. $\nu_t = \nu_0$, and u_t has a fixed noise variance of the form $\mathsf{u}_t = r_0 * \mathsf{u}_0$ with r_0 a fixed constant measuring the extent of noise and u_0 a 'standardized' random variable/vector [1]. Since a constant ν_0 has difficulty in handling arbitrary movement, ν_0 is typically set to be $\nu_0 = 0$. If r_0 is small, it is very hard to model rapid movements; if r_0 is large, it is computationally inefficient since many more particles are needed to accommodate the large noise variance. All these factors make such a model ineffective. In this chapter, we overcome this by introducing an adaptive-velocity model.

[1] Consider the scalar case for example. If u_t is distributed as $\mathtt{N}(0, \sigma^2)$, we can write $\mathsf{u}_t = \sigma \mathsf{u}_0$ where u_0 is standard normal $\mathtt{N}(0, 1)$. This also applies to multivariate cases.

While contour is the visual cue used in many tracking algorithms [118], another class of tracking approaches [115, 127, 185] exploits an appearance model A_t. In its simplest form, we have the following observation equation[2],

$$\mathsf{z}_t = \mathcal{T}\{\mathsf{y}_t; \theta_t\} = \mathsf{A}_t + \mathsf{v}_t, \tag{6.6}$$

where z_t is the image patch of interest in the video frame y_t, parameterized by θ_t. In [115], a fixed template, $\mathsf{A}_t = \mathsf{A}_0$, is matched with observations to minimize a cost function in the form of sum of squared distance (SSD). This is equivalent to assuming that the noise v_t is a normal random vector with zero mean and a diagonal (isotropic) covariance matrix. At the other extreme, one could use a rapidly changing model [127], say, $\mathsf{A}_t = \hat{\mathsf{z}}_{t-1}$, i.e., the 'best' patch of interest in the previous frame. However, a fixed template cannot handle appearance changes in the video, while a rapidly changing model is susceptible to drift. Thus, it is necessary to have a model which is a compromise between these two cases. In [120], Jepson *et. al.* proposed an online appearance model (OAM) for a robust visual tracker, which is a mixture of three components. Two EM algorithms are used, one for updating the appearance model and the other for deriving the tracking parameters.

Our approach to visual tracking is to make both observation and state transition models adaptive in the framework of a particle filter, with provisions for handling occlusion. The main features of our tracking approach are as follows:

- Appearance-based. The only visual cue used in our tracker is the 2-D appearance; i.e., we employ only image intensities, though in general features

[2]For the sake of simplicity, we denote: $\mathsf{z}_t \equiv \mathcal{T}\{\mathsf{y}_t; \theta_t\}$, $\mathsf{z}_t^{(j)} \equiv \mathcal{T}\{\mathsf{y}_t; \theta_t^{(j)}\}$, $\hat{\mathsf{z}}_t \equiv \mathcal{T}\{\mathsf{y}_t; \hat{\theta}_t\}$. Also, we can always vectorize the 2-D image by a lexicographical scanning of all pixels and denote the number of pixels by d.

derived from image intensities, such as the phase information of the filter responses [120] or the Gabor feature graph presentation [85], are also applicable. No prior object models are invoked. In addition, we only use gray scale images.

- Adaptive observation model. We adopt an appearance-based approach. The original OAM is modified and then embedded in our particle filter. Therefore, the observation model is adaptive as the appearance A_t involved in (6.6) is adaptive.

- Adaptive state transition model. Instead of using a fixed model, we use an adaptive-velocity model, where the adaptive motion velocity ν_t is predicted using a first-order linear approximation based on the appearance difference between the incoming observation and the previous particle configuration. We also use an adaptive noise component, i.e, $\mathsf{u}_t = r_t * \mathsf{u}_0$, whose magnitude r_t is a function of the prediction error. It is natural to vary the number of particles based on the degree of uncertainty r_t in the noise component.

- Handling occlusion. Occlusion is handled using robust statistics [11, 115, 108]. We robustify the likelihood measurement and the adaptive velocity estimate by downweighting the 'outlier' pixels. If occlusion is declared, we stop updating the appearance model and estimating the motion velocity.

Chapter organization

This chapter is organized as follows. We briefly review the related literature on visual tracking and particle filters in Section 6.1. We examine the details of an adaptive observation model in Section 6.2.1, with a special focus on the adaptive

appearance model, and of an adaptive state transition model in Section 6.2.2 with a special focus on how to calculate the motion velocity. Handling occlusion is discussed in Section 6.2.3, and experimental results on tracking vehicles and human faces in Section 6.3.

6.1 Related Literature

6.1.1 Visual tracking

Roughly speaking, previous work on visual tracking can be divided into two groups: deterministic tracking and stochastic tracking. Our approach combines the merits of both stochastic and deterministic tracking approaches in a unified framework using a particle filter. We give below a brief review of both approaches.

Deterministic approaches usually reduce to an optimization problem, e.g., minimizing an appropriate cost function. The definition of the cost function is a key issue. A common choice in the literature is the SSD used in many optical flow approaches [115].[3] A gradient descent algorithm is most commonly used to find the minimum. Very often, only a local minimum can be reached. In [115], the cost function is defined as the SSD between the observation and a fixed template, and the motion is parameterized as affine. Hence the task is to find the affine parameter minimizing the cost function. Using a Taylor series expansion and keeping only the first-order terms, a linear prediction equation is obtained. It has been shown that for the affine case, the system matrix can be computed efficiently since a fixed template is used. Mean shift [113] is an alternative deterministic approach

[3]We note that using SSD is equivalent to using a model where the noise obeys an iid Gaussian distribution; therefore this case can also be viewed as stochastic tracking.

to visual tracking, where the cost function is derived from the color histogram.

Stochastic tracking approaches often reduce to an estimation problem, e.g., estimating the state for a time series state space model. Early works [106, 112] used the Kalman filter or its variants [1] to provide solutions. However, this restricts the type of model that can be used. Recently sequential Monte Carlo (SMC) algorithms [6, 114, 157, 159], which can model nonlinear/non-Gaussian cases, have gained prevalence in the tracking literature due in part to the CONDENSATION algorithm [118]. Stochastic tracking improves robustness over its deterministic counterpart by its capability for escaping the local minimum since the searching directions are for the most part random even though they are governed by a deterministic state transition model. Toyama and Blake [130] proposed a probabilistic paradigm for tracking with the following properties: Exemplars are learned from the raw training data and embedded in a mixture density; The kinematics is also learned; The likelihood measurement is constructed on a metric space. Other approaches are also discussed in Section 6.1.2. However, as far as computational load is concerned, stochastic algorithms in general are more intense. Note that the stochastic approaches can often be formulated as optimization problems.

6.1.2 Particle filter

General particle filter algorithm

Given the state transition model in (6.1) characterized by the state transition probability $\mathsf{p}(\theta_t|\theta_{t-1})$ and the observation model in (6.2) characterized by the likelihood function $\mathsf{p}(\mathsf{y}_t|\theta_t)$, the problem is reduced to computing the posterior probability $\mathsf{p}(\theta_t|\mathsf{y}_{1:t})$. The nonlinearity/nonnormality in (6.1) and (6.2) result in Kalman filter [1] being ineffective. The particle filter is a means to approximate the poste-

rior distribution $\mathtt{p}(\theta_t|\mathtt{y}_{1:t})$ by a set of weighted particles $\mathcal{S}_t = \{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^J$ with $\sum_{j=1}^J w_t^{(j)} = 1$. It can be shown [159] that $\mathcal{S}_t$ is *properly weighted* with respect to $\mathtt{p}(\theta_t|\mathtt{y}_{1:t})$ in the sense that, for every bounded function $\mathtt{h}(.)$,

$$\lim_{J\to\infty} \sum_{j=1}^{J} w_t^{(j)} \mathtt{h}(\theta_t^{(j)}) = \mathtt{E}_{\mathtt{p}}[\mathtt{h}(\theta_t)]. \tag{6.7}$$

Given $\mathcal{S}_{t-1} = \{\theta_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^J$ which is properly weighted with respect to $\mathtt{p}(\theta_{t-1}|\mathtt{y}_{1:t-1})$, we first resample $\mathcal{S}_{t-1}$ to reach a new set of samples with equal weights $\{\theta_{t-1}^{'(j)}, 1\}_{j=1}^J$. We then draw samples $\{\mathtt{u}_t^{(j)}\}_{j=1}^J$ for $\mathtt{u}_t$ and propagate $\theta_{t-1}^{'(j)}$ to $\theta_t^{'(j)}$ by (6.1). The new weight is updated as

$$w_t \propto \mathtt{p}(\mathtt{y}_t|\theta_t) \tag{6.8}$$

The complete algorithm is summarized in Figure 6.1.

Initialize *a sample set* $\mathcal{S}_0 = \{\theta_0^{(j)}, 1)\}_{j=1}^J$ *according to prior distribution* $\mathtt{p}(\theta_0)$.

For $t = 1, 2, \ldots$

  For $j = 1, 2, \ldots, J$

    Resample $\mathcal{S}_{t-1} = \{\theta_{t-1}^{(j)}, w_{t-1}^{(j)}\}$ *to obtain a new sample* $(\theta_{t-1}^{'(j)}, 1)$.

    Predict *the sample by drawing* $\mathtt{u}_t^{(j)}$ *for* $\mathtt{u}_t$ *and computing* $\theta_t^{(j)} = \mathtt{f}_t(\theta_{t-1}^{'(j)}, \mathtt{u}_t^{(j)})$.

    Compute *the transformed image* $\mathtt{z}_t^{(j)} = \mathcal{T}\{\mathtt{y}_t; \theta_t\}$.

    Update *the weight using* $w_t^{(j)} = \mathtt{p}(\mathtt{y}_t|\theta_t^{(j)}) = \mathtt{p}(\mathtt{z}_t^{(j)}|\theta_t^{(j)})$.

  End

  Normalize *the weight using* $w_t^{(j)} = w_t^{(j)} / \sum_{j=1}^J w_t^{(j)}$.

End

Figure 6.1: The general particle filter algorithm.

Variations of Particle Filters

Sequential Importance Sampling (SIS) [153, 159] draws particles from a *proposal distribution* $\mathtt{q}(\theta_t|\theta_{t-1}, \mathtt{y}_{1:t})$ and then for each particle a proper weight is assigned as

follows:

$$w_t \propto \mathtt{p}(\mathtt{y}_t|\theta_t)\mathtt{p}(\theta_t|\theta_{t-1})/\mathtt{q}(\theta_t|\theta_{t-1},\mathtt{y}_{1:t}). \tag{6.9}$$

Selection of the proposal distribution $\mathtt{q}(\theta_t|\theta_{t-1},\mathtt{y}_{1:t})$ is usually dependent on the application. For example, in the ICONDENSATION algorithm [119] which fuses low-level and high-level visual cues in the conventional CONDENSATION algorithm [118], the proposal distribution, a fixed Gaussian distribution for low-level color cue, is used to predict the particle configurations, then the posterior distribution of the high-level shape cue is approximated using SIS. It is interesting to note that two different cues can be even combined together into one state vector to yield a robust tracker, using the co-inference algorithm [133] and the approach proposed in [131]. We also use a prediction scheme but our prediction is based on the same visual cue i.e. the appearance in the image, and it is directly used in the state transition model rather than used as a proposal distribution. Additional visual cues are not used.

6.2 Appearance-Adaptive Models

6.2.1 Adaptive observation model

The adaptive observation model arises from the adaptive appearance model A_t. We use a modified version of OAM as developed in [120]. The differences between our appearance model and the original OAM are highlighted below.

Mixture appearance model

The original OAM assumes that the observations are explained by different causes, thereby indicating the use of a mixture density of components. In the original OAM

presented in [120], three components are used, namely the W-component characterizing the two-frame variations, the S-component depicting the stable structure within all past observations (though it is slowly-varying), and the L-component accounting for outliers such as occluded pixels.

We modify the OAM to accommodate our appearance analysis in the following aspects. (i) We directly use the image intensities while they use phase information derived from image intensities. Direct use of image intensities is computationally more efficient than using the phase information that requires filtering and visually more interpretable. (ii) As an option, in order to further stabilize the tracker one could use an F-component which is a fixed template that one is expecting to observe most often. For example, in face tracking this could be just the facial image as seen from a frontal view. In the sequel, we derive the equations as if there is an F-component. However, the effect of this component can be ignored by setting its initial mixing probability to zero. (iii) We embed the appearance model in a particle filter to perform tracking while they use the EM algorithm. (iv) In our implementation, we do not incorporate the L-component because we model the occlusion in a different manner (using robust statistics) as discussed in Section 6.2.3.

We now describe the mixture appearance model. The appearance model at time t,

$$\mathsf{A}_t = \{\mathsf{W}_t, \mathsf{S}_t, \mathsf{F}_t\},$$

is a time-varying one that models the appearances present in all observations up to time $t-1$. It obeys a mixture of Gaussians, with $\mathsf{W}_t, \mathsf{S}_t, \mathsf{F}_t$ as mixture centers $\{\mu_{i,t};\ i = w, s, f\}$ and their corresponding variances $\{\sigma^2_{i,t};\ i = w, s, f\}$ and mixing

probabilities $\{m_{i,t};\ i = w, s, f\}$. Notice that

$$\{m_{i,t}, \mu_{i,t}, \sigma_{i,t}^2;\ i = w, s, f\}$$

are 'images' consisting of d pixels that are assumed to be independent of each other.

In summary, the observation likelihood is written as

$$\mathtt{p}(\mathsf{y}_t|\theta_t) = \mathtt{p}(\mathsf{z}_t|\theta_t) = \prod_{j=1}^{d}\{\sum_{i=w,s,f} m_{i,t}(j)\mathtt{N}(\mathsf{z}_t(j); \mu_{i,t}(j), \sigma_{i,t}^2(j))\}, \tag{6.10}$$

where $\mathtt{N}(x; \mu, \sigma^2)$ is a normal density

$$\mathtt{N}(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2}\exp\{-\rho(\frac{x-\mu}{\sigma})\}, \quad \rho(x) = \frac{1}{2}x^2. \tag{6.11}$$

Model update

To keep the chapter self-contained, we show how to update the current appearance model A_t to A_{t+1} after $\hat{\mathsf{z}}_t$ becomes available, i.e., we want to compute the new mixing probabilities, mixture centers, and variances for time $t+1$,

$$\{m_{i,t+1}, \mu_{i,t+1}, \sigma_{i,t+1}^2;\ i = w, s, f\}.$$

It is assumed that the past observations are exponentially 'forgotten' with respect to their contributions to the current appearance model. Denote the exponential envelop by $\alpha\exp(-\tau^{-1}(t-k))$ for $k \leq t$, where $\tau = n_h/\log 2$, n_h is the half-life of the envelope in frames, and $\alpha = 1 - \exp(-\tau^{-1})$ to guarantee that the area under the envelope is 1. We just sketch the updating equations as follows and refer the interested readers to [120] for technical details and justifications.

The EM algorithm [152] is invoked. Since we assume that the pixels are independent of each other, we can deal with each pixel separately. The following

computation is valid for $j = 1, 2, \ldots, d$ where d is the number of pixels in the appearance model.

First, the posterior responsibility probabilities are computed as

$$o_{i,t}(j) \propto m_{i,t}(j)\mathtt{N}(\hat{\mathsf{z}}_t(j); \mu_{i,t}(j), \sigma^2_{i,t}(j)); \quad i = w, s, f, \quad \& \sum_{i=w,s,f} o_{i,t}(j) = 1. \tag{6.12}$$

Then, the mixing probabilities are updated as

$$m_{i,t+1}(j) = \alpha\ o_{i,t}(j) + (1 - \alpha)\ m_{i,t}(j); \quad i = w, s, f, \tag{6.13}$$

and the first- and second-moment images $\{\mathsf{M}_{p,t+1};\ p = 1, 2\}$ are evaluated as

$$\mathsf{M}_{p,t+1}(j) = \alpha\ \hat{\mathsf{z}}^p_t(j) o_{s,t}(j) + (1 - \alpha)\ \mathsf{M}_{p,t}(j); \quad p = 1, 2. \tag{6.14}$$

Finally, the mixture centers and the variances are updated as:

$$\mathsf{S}_{t+1}(j) = \mu_{s,t+1}(j) = \frac{\mathsf{M}_{1,t+1}(j)}{m_{s,t+1}(j)}, \quad \sigma^2_{s,t+1}(j) = \frac{\mathsf{M}_{2,t+1}(j)}{m_{s,t+1}(j)} - \mu^2_{s,t+1}(j). \tag{6.15}$$

$$\mathsf{W}_{t+1}(j) = \mu_{w,t+1}(j) = \hat{\mathsf{z}}_t(j), \quad \sigma^2_{w,t+1}(j) = \sigma^2_{w,1}(j), \tag{6.16}$$

$$\mathsf{F}_{t+1}(j) = \mu_{f,t+1}(j) = \mathsf{F}_1(j), \quad \sigma^2_{f,t+1}(j) = \sigma^2_{f,1}(j). \tag{6.17}$$

Model initialization

To initialize A_1, we set $\mathsf{W}_1 = \mathsf{S}_1 = \mathsf{F}_1 = \mathsf{T}_0$ (with T_0 supplied by a detection algorithm or manually), $\{m_{i,1}, \sigma^2_{i,1};\ i = w, s, f\}$, and $\mathsf{M}_{1,1} = m_{s,1}\mathsf{z}_0$ and $\mathsf{M}_{2,1} = m_{s,1}\sigma^2_{s,1} + \mathsf{T}^2_0$.

6.2.2 Adaptive state transition model

The state transition model we use incorporates a term for modeling adaptive velocity. The adaptive velocity is calculated using a first-order linear prediction method

based on the appearance differences between two successive frames. The previous particle configuration is incorporated in the prediction scheme.

Construction of the particle configuration involves the costly computation of image warping (in the experiments reported here, it usually accounts for about half of the computations). In a conventional particle filtering algorithm, the particle configuration is used only to update the weight, i.e., computing weight for each particle by comparing the warped image with the online appearance model using the observation equation. But, our approach in addition uses the particle configuration in the state transition equation. In some sense, we 'maximally' utilize the information contained in the particles (without wasting the costly computation of image warping) since we use it in both state and observation models.

In [128], random samples are guided by deterministic search. Momentum for each particle is computed as the sum of absolute difference between two frames. If the momentum is below a threshold, a deterministic search is first performed using a gradient descent method and a small number of offsprings is then generated using stochastic diffusion; otherwise, stochastic diffusion is performed to generate a large number of offsprings. The stochastic diffusion is based on a second-order autoregressive process. But, the gradient descent method does not utilize the previous particle configuration in its entirety. Also, the generated particle configuration could severely deviate from the second-order autoregressive model, which clearly implies the need for an adaptive model.

Adaptive velocity

With the availability of the sample set $\Theta_{t-1} = \{\theta_{t-1}^{(j)}\}_{j=1}^{J}$ and the image patches of interest $\mathcal{Z}_{t-1} = \{\mathsf{z}_{t-1}^{(j)}\}_{j=1}^{J}$, for a new observation y_t, we can predict the shift in

the motion vector (or adaptive velocity) $\nu_t = \theta_t - \hat{\theta}_{t-1}$ using a first-order linear approximation [107, 115, 121, 123], which essentially comes from the constant brightness constraint, i.e., there exists a θ_t such that

$$\mathcal{T}\{\mathsf{y}_t; \theta_t\} \simeq \hat{\mathsf{z}}_{t-1}. \tag{6.18}$$

Approximating $\mathcal{T}\{\mathsf{y}_t; \theta_t\}$ using a first-order Taylor series expansion around $\tilde{\theta}_t$ (we set $\tilde{\theta}_t = \hat{\theta}_{t-1}$) yields

$$\mathcal{T}\{\mathsf{y}_t; \theta_t\} \simeq \mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t\} + \mathsf{C}_t(\theta_t - \tilde{\theta}_t) = \mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t\} + \mathsf{C}_t\nu_t, \tag{6.19}$$

where C_t is the Jacobian matrix.

Combining (6.18) and (6.19) gives

$$\hat{\mathsf{z}}_{t-1} \simeq \mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t\} + \mathsf{C}_t\nu_t, \tag{6.20}$$

i.e.,

$$\nu_t = \theta_t - \tilde{\theta}_t \simeq -\mathsf{B}_t(\mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t\} - \hat{\mathsf{z}}_{t-1}), \tag{6.21}$$

where B_t is the pseudo-inverse of the C_t matrix, which can be efficiently estimated from the available data Θ_{t-1} and $\mathcal{Z}_{t-1}$.

Specifically, to estimate B_t we stack into matrices the differences in motion vectors and image patches, using $\hat{\theta}_{t-1}$ and $\hat{\mathsf{z}}_{t-1}$ as pivotal points:

$$\delta\Theta_{t-1} = [\theta_{t-1}^{(1)} - \hat{\theta}_{t-1},\ \ldots,\ \theta_{t-1}^{(J)} - \hat{\theta}_{t-1}], \tag{6.22}$$

$$\delta\mathcal{Z}_{t-1} = [\mathsf{z}_{t-1}^{(1)} - \hat{\mathsf{z}}_{t-1},\ \ldots,\ \mathsf{z}_{t-1}^{(J)} - \hat{\mathsf{z}}_{t-1}]. \tag{6.23}$$

The least square (LS) solution for B_t is

$$\mathsf{B}_t = (\delta\Theta_{t-1}\delta\mathcal{Z}_{t-1}^{\mathsf{T}})(\delta\mathcal{Z}_{t-1}\delta\mathcal{Z}_{t-1}^{\mathsf{T}})^{-1}. \tag{6.24}$$

However, it turns out that the matrix $\delta\mathcal{Z}_{t-1}\delta\mathcal{Z}_{t-1}^{\mathsf{T}}$ is very often rank-deficient due to the high dimensionality of the data (unless the number of the particles at least exceeds the data dimension). To overcome this, we use the SVD as

$$\delta\mathcal{Z}_{t-1} = \mathsf{USV}^{\mathsf{T}} \tag{6.25}$$

It can be easily shown that

$$\mathsf{B}_t = \delta\Theta_{t-1}\mathsf{VS}^{-1}\mathsf{U}^{\mathsf{T}}. \tag{6.26}$$

To gain some computational efficiency, we can further approximate

$$\mathsf{B}_t = \delta\Theta_{t-1}\mathsf{V}_q\mathsf{S}_q^{-1}\mathsf{U}_q^{\mathsf{T}}, \tag{6.27}$$

by retaining the top q components. Notice that if only a fixed template is used [121], the B matrix is fixed and pre-computable. But, in our case, the appearance is changing so that we have to compute the B_t matrix in each time step.

In practice, one may run several iterations till $\tilde{\mathsf{z}}_t = \mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t + \nu_t\}$ stabilizes, i.e., the error ϵ_t defined below is small enough.

$$\epsilon_t = \phi(\tilde{\mathsf{z}}_t, \mathsf{A}_t) = \frac{2}{d}\sum_{j=1}^{d}\{\sum_{i=w,s,f} m_{i,t}(j)\rho(\frac{\tilde{\mathsf{z}}_t(j) - \mu_{i,t}(j)}{\sigma_{i,t}(j)})\}. \tag{6.28}$$

In (6.28), ϵ_t measures the distance between $\mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t + \nu_t\}$ and the updated appearance model A_t. The iterations proceed as follows: We initially set $\tilde{\theta}_t^1 = \hat{\theta}_{t-1}$. For the first iteration, we compute ν_t^1 as usual. For the k^{th} iteration, we use the predicted $\tilde{\theta}_t^k = \tilde{\theta}_t^{k-1} + \nu_t^{k-1}$ as a pivotal point for the Taylor expansion in (6.19) and the rest of the calculation then follows. It is rather beneficial to run several iterations especially when the object moves very fast in two successive frames since $\hat{\theta}_{t-1}$ might cover the target in y_t in a small portion. After one iteration, the computed ν_t might be not accurate, but indicates a good minimization direction. Using several iterations helps to find ν_t (compared to $\hat{\theta}_{t-1}$) more accurately.

We use the following adaptive state transition model

$$\theta_t = \hat{\theta}_{t-1} + \nu_t + \mathsf{u}_t, \tag{6.29}$$

where ν_t is the predicted shift in the motion vector. The choice of u_t is discussed below. One should note that we are not using (6.29) as a proposal function to draw particles, which requires using (6.9) to compute the particle weight. Instead we directly use it as the state transition model and hence use (6.8) to compute the particle weight. Our model can be easily interpreted as a time-varying state model.

It is interesting to note that the approach proposed in [131] also uses motion cues as well as color parameter adaptation. Our approach is different from [131] in that: (i) We use the motion cue in the state transition model while they use it as part of observations; (ii) We only use the gray images without using the color cue which is used in [131]; and (iii) We use an adaptive appearance model which is updated by the EM algorithm while they use an adaptive color model which is updated by a stochastic version of the EM algorithm.

Adaptive noise

The value of ϵ_t determines the quality of prediction. Therefore, if ϵ_t is small, which implies a good prediction, we only need noise with small variance to absorb the residual motion; if ϵ_t is large, which implies a poor prediction, we then need noise with large variance to model the potentially large jumps in the motion state.

To this end, we use u_t of the form $\mathsf{u}_t = r_t * \mathsf{u}_0$, where r_t is a function of ϵ_t. Since ϵ_t defined in (6.28) is a 'variance'-type measure, we use

$$r_t = \max(\min(r_0\sqrt{\epsilon_t}, r_{max}), r_{min}), \tag{6.30}$$

where r_{min} is the lower bound to maintain a reasonable sample coverage and r_{max} is the upper bound to constrain the computational load.

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) [3], we should adjust the particle number J_t in a similar fashion, i.e.,

$$J_t = J_0 r_t / r_0. \tag{6.31}$$

Fox [154] also presents an approach to improve the efficiency of particle filters by adapting the particle numbers on-the-fly. His approach is to divide the state space into bins and approximate the posterior distribution by a multinomial distribution. A small number of particles is used if the density is focused on a small part of the state space and a large number of particles if the uncertainty in the state space is high. In this way, the error between the empirical distribution and the true distribution (approximated as a multinomial in his analysis) measured by Kullback-Leilber distance is bounded. However, in his approach, since the state space (only 2D) is exhaustively divided, the number of particles is at least several thousand, while our approach uses at most a few hundred. Our attempt is not to explore the state space (6-D affine space) exhaustively, but only regions that have high potential for the object to be present.

Comparison between the adaptive velocity model and the zero velocity model

We demonstrate the necessity of the adaptive velocity model by comparing it with the zero velocity model. Figure 6.2 shows the particle configurations created from the adaptive velocity model (with $J_t < J_0$ and $r_t < r_0$ computed as above) and the zero velocity model (with $J_t = J_0$ and $r_t = r_0$). Clearly, the adaptive-velocity model generates particles very efficiently, i.e, they are tightly centered around the object of interest so that we can easily track the object at time t; while the zero-velocity model generates more particles widely spread to explore larger regions, leading to unsuccessful tracking as widespread particles often lead to a local minimum.

Tracking result at $t-1$ Particle configuration at t Tracking result at t

Figure 6.2: Particle configurations from (top row) the adaptive velocity model and (bottom row) the zero-velocity model.

6.2.3 Handling occlusion

Occlusion is usually handled in two ways. One way is to use joint probabilistic data associative filter (JPDAF) [2, 126]; and the other one is to use robust statistics

[11]. We use robust statistics here.

Robust statistics

We assume that occlusion produces large image differences which can be treated as 'outliers'. Outlier pixels cannot be explained by the underlying process and their influences on the estimation process should be reduced. Robust statistics provide such mechanisms.

We use the $\hat{\rho}$ function defined as follows:

$$\hat{\rho}(x) = \begin{cases} \frac{1}{2}x^2 & \text{if} \quad |x| \leq c \\ cx - \frac{1}{2}c^2 & \text{if} \quad |x| > c \end{cases}, \tag{6.32}$$

where x is normalized to have unit variance and the constant c controls the outlier rate. In our experiment, we take $c = 1.435$ based on experimental experience. If $|x| > c$ is satisfied, we declare the corresponding pixel as an outlier.

Robust likelihood measure and adaptive velocity estimate

The likelihood measure defined in Eq. (6.10) involves a multi-dimensional normal density. Since we assume that each pixel is independent, we consider the one-dimensional normal density. To make the likelihood measure robust, we replace the one-dimensional normal density $\mathtt{N}(x; \mu, \sigma^2)$ by

$$\hat{\mathtt{N}}(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} \exp(-\hat{\rho}(\frac{x-\mu}{\sigma})). \tag{6.33}$$

Note that this is not a density function any more, but since we are dealing with discrete approximation in the particle filter, normalization makes it a probability mass function.

Existence of outlier pixels severely violates the constant brightness constraint and hence affects our estimate of the adaptive velocity. To downweight the influ-

ence of the outlier pixels in estimating the adaptive velocity, we introduce a $d \times d$ diagonal matrix L_t with its i^{th} diagonal element being $\mathsf{L}_t(i) = \eta(x_i)$ where x_i is the pixel intensity of the difference image $(\mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t\} - \hat{\mathsf{z}}_{t-1})$ normalized by the variance of the OAM stable component and

$$\eta(x) = \frac{1}{x}\frac{d\hat{\rho}(x)}{dx} = \begin{cases} 1 & \text{if} \quad |x| \leq c \\ c/|x| & \text{if} \quad |x| > c \end{cases}, \tag{6.34}$$

Eq. (6.21) becomes

$$\nu_t \simeq -\mathsf{B}_t\mathsf{L}_t(\mathcal{T}\{\mathsf{y}_t; \hat{\theta}_{t-1}\} - \hat{\mathsf{z}}_{t-1}). \tag{6.35}$$

This is similar in principle to the weighted least square algorithm.

Occlusion declaration

If the number of the outlier pixels in $\hat{\mathsf{z}}_t$ (compared with the OAM), say d_{out}, exceeds a certain threshold, i.e., $d_{out} > \lambda d$ where $0 < \lambda < 1$ (we take $\lambda = 0.15$), we declare occlusion. Since the OAM has more than one component, we count the number of outlier pixels with respect to every component and take the maximum.

If occlusion is declared, we stop updating the appearance model and estimating the motion velocity. Instead, we (i) keep the current appearance model, i.e., $\mathsf{A}_{t+1} = \mathsf{A}_t$ and (ii) set the motion velocity to zero, i.e., $\nu_t = 0$ and use the maximum number of particles sampled from the diffusion process with largest variance, i.e., $r_t = r_{max}$, and $J_t = J_{max}$.

The adaptive particle filtering algorithm with occlusion analysis is summarized in Figure 6.3.

Initialize *a sample set* $\mathcal{S}_0 = \{\theta_0^{(j)}, 1/J_0)\}_{j=1}^{J_0}$ *according to prior distribution* $\mathbf{p}(\theta_0)$.

Initialize *the appearance model* $\mathbf{A}_1$.

Set $OCC_{FLAG} = 0$ *to indicate no occlusion.*

For $t = 1, 2, \ldots$

If *(*$OCC_{FLAG} == 0$*)*

Calculate *the state estimate* $\hat{\theta}_{t-1}$ *by Eq. (6.3) or (6.4), the adaptive velocity* ν_t *by Eq. (6.21), the noise variance* r_t *by Eq. (6.30), and the particle number* J_t *by Eq. (6.31).*

Else

$r_t = r_{max}$, $J_t = J_{max}$, $\nu_t = 0$.

End

For $j = 1, 2, \ldots, J_t$

Draw *the sample* $\mathbf{u}_t^{(j)}$ *for* $\mathbf{u}_t$ *with variance* r_t.

Construct *the sample* $\theta_t^{(j)} = \hat{\theta}_{t-1} + \nu_t + \mathbf{u}_t^{(j)}$ *by Eq. (6.29).*

Compute *the transformed image* $\mathbf{z}_t^{(j)}$.

Update *the weight using* $w_t^{(j)} = \mathbf{p}(\mathbf{y}_t|\theta_t^{(j)}) = p(\mathbf{z}_t^{(j)}|\theta_t^{(j)})$.

End

Normalize *the weight using* $w_t^{(j)} = w_t^{(j)} / \sum_{j=1}^{J} w_t^{(j)}$.

Set OCC_{FLAG} *according to the number of the outlier pixels in* $\hat{\mathbf{z}}_t$.

If *(*$OCC_{FLAG} == 0$*)*

Update *the appearance model* $\mathbf{A}_{t+1}$ *using* $\hat{\mathbf{z}}_t$.

End

End

Figure 6.3: The proposed visual tracking algorithm with occlusion handling.

6.3 Experimental results on visual tracking

In our implementation, we used the following choices. We consider affine transformation only. Specifically, the motion is characterized by $\theta = (a_1, a_2, a_3, a_4, t_x, t_y)$ where $\{a_1, a_2, a_3, a_4\}$ are deformation parameters and $\{t_x, t_y\}$ denote the 2-D trans-

lation parameters. Even though significant pose/illumincation changes are present in the video, we believe that our adaptive appearance model can easily absorb them and therefore for our purposes the affine transformation is a reasonable approximation. Regarding photometric transformations, only a zero-mean-unit-variance normalization is used to partially compensate for contrast variations. The complete image transformation $\mathcal{T}\{\mathsf{y};\theta\}$ is implemented as follows: affine transform y using $\{a_1, a_2, a_3, a_4\}$, crop out the region of interest at position $\{t_x, t_y\}$ with the same size as the still template in the appearance model, and perform zero-mean-unit-variance normalization.

We demonstrate our algorithm by tracking a disappearing car, a moving tank acquired by a camera mounted on a micro air vehicle, and a moving face under occlusion. Table 6.1 summarizes some statistics about the video sequences and the appearance model size used.

We initialize the particle filter and the appearance model with a detector algorithm (we actually used the face detector described in [132] for the face sequence) or a manually specified image patch in the first frame. r_0 and J_0 are also manually set, depending on the sequence.

6.3.1 Car tracking

We first test our algorithm to track a vehicle with the F-component but without occlusion analysis. The result of tracking a fast moving car is shown in Figure 6.4 (column 1)[4]. The tracking result is shown with a bounding box. We also show the stable and wandering components separately (in a double-zoomed size) at the corner of each frame. The video is captured by a camera mounted on the

[4]Accompanying videos are available at http://www.cfar.umd.edu/~shaohua/research/.

Video	Car	Tank	Face
# of frames	500	300	800
Frame size	576x768	240x360	240x360
A_t size	24x30	24x30	30x26
Occlusion	No	No	Yes (twice)
'adp'	o	o	x
'fa'	o	o	x
'fm'	x	x	x
'fb'	x	x	x
'adp & occ'	o	o	o

Table 6.1: Comparison of tracking results obtained by particle filters with different configurations. 'A_t size' means pixel size in the component(s) of the appearance model. 'o' means success in tracking. 'x' means failure in tracking.

car. In this footage the relative velocity of the car with respect to the camera platform is very large, and the target rapidly decreases in size. Our algorithm's adaptive particle filter successfully tracks this rapid change in scale. Figure 6.5(a) plots the scale estimate (calculated as $\sqrt{(a_1^2 + a_2^2 + a_3^2 + a_4^2)/2}$) recovered by our algorithm. It is clear that the scale follows a decreasing trend as time proceeds. The pixels located on the car in the final frame are about 12 by 15 in size, which makes the vehicle almost invisible. In this sequence we set $J_0 = 50$ and $r_0 = 0.25$. The algorithm implemented in a standard Matlab environment processes about 1.2 frames per second (with $J_0 = 50$) running on a PC with a PIII 650 CPU and 512M memory.

Figure 6.4: The car sequence. Notice the fast scale change present in the video. Column 1: the tracking results obtained with an adaptive motion model and an adaptive appearance model ('adp'). Column 2: the tracking results obtained with an adaptive motion model but a fixed appearance model ('fa'). In this case, the corner shows the tracked region. Column 3: the tracking results obtained with an adaptive appearance model but a fixed motion model ('fm').

6.3.2 Tank tracking in an aerial video

Figure 6.6 shows our results on tracking a tank in an aerial video with degraded image quality due to motion blur. Also, the movement of the tank is very jerky and arbitrary because of platform motion, as seen in Figure 6.5(b) which plots the

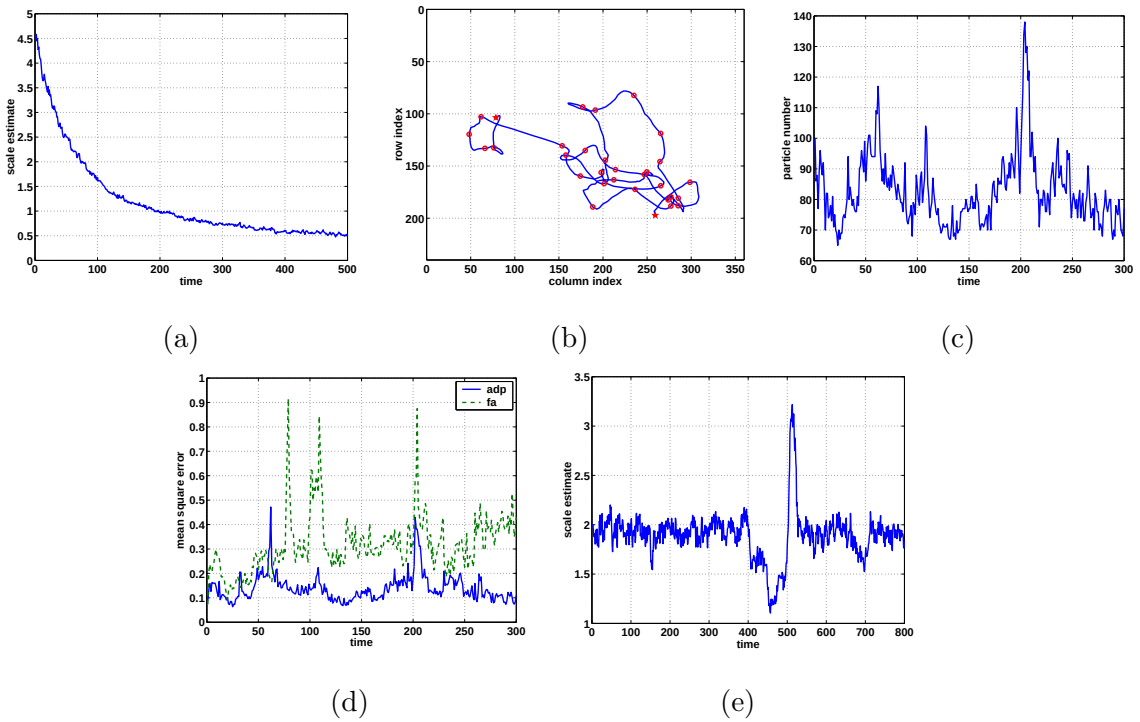


Figure 6.5: (a) The scale estimate for the car. (b) The 2-D trajectory of the centroid of the tracked tank. '*' means the starting and ending points and '.' points are marked along the trajectory every 10 frames. (c) The particle number J_t *vs.* t obtained when tracking the tank. (d) The MSE invoked by the 'adp' and 'fa' algorithms. (e) The scale estimate for the face sequence.

2-D trajectory of the centroid of the tracked tank every 10 frames, covering from the left to the right in 300 frames. Although the tank moved about 100 pixels in column index in a certain period of 10 frames, the tracking is still successful.

Figure 6.5(c) displays the plot of actual number of particles J_t as a function of time t. The average number of particle is about 83, where we set J_0 to be 100, which means that in this case we actually saved about 20% in computation by using an adaptive J_t instead of a fixed number of particles.

To further illustrate the importance of the adaptive appearance model, we

Frame 1 Frame 31 Frame 49

Frame 116 Frame 228 Frame 300

Figure 6.6: Tracking a moving tank in a video acquired by an airborne camera.

computed the mean square error (MSE) invoked by two particle filter algorithms, one (referred as 'adp' in Section 6.3.4) using the adaptive appearance model and the other (referred as 'fa' in Section 6.3.4) using a fixed appearance model. Computing the MSE for the 'fa' algorithm is straightforward, with T_0 denoting the fixed template,

$$MSE_{fa}(t) = d^{-1} \sum_{j=1}^{d} (\hat{\mathsf{z}}_t(j) - \mathsf{T}_0(j))^2. \tag{6.36}$$

Computing the MSE for the 'adp' algorithm is as follows:

$$MSE_{adp}(t) = d^{-1} \sum_{j=1}^{d} \{ \sum_{i=w,s,f} m_{i,t} (\hat{\mathsf{z}}_t(j) - \mu_{i,t}(j))^2 \}. \tag{6.37}$$

Figure 6.5(d) plots the functions of $MSE_{fa}(t)$ and $MSE_{adp}(t)$. Clearly, using the adaptive appearance model invokes smaller MSE for almost all 300 frames. The average MSE for the 'adp' algorithm is 0.1394 [5] while that for the 'fa' algorithm is 0.3169!

[5]The range of MSE is very reasonable since we are using image patches after the zero-mean-unit-variance normalization not the raw image intensities.

6.3.3 Face tracking

We present one example of successful tracking of a human face using a hand-held video camera in an office environment, where both camera and object motion are present.

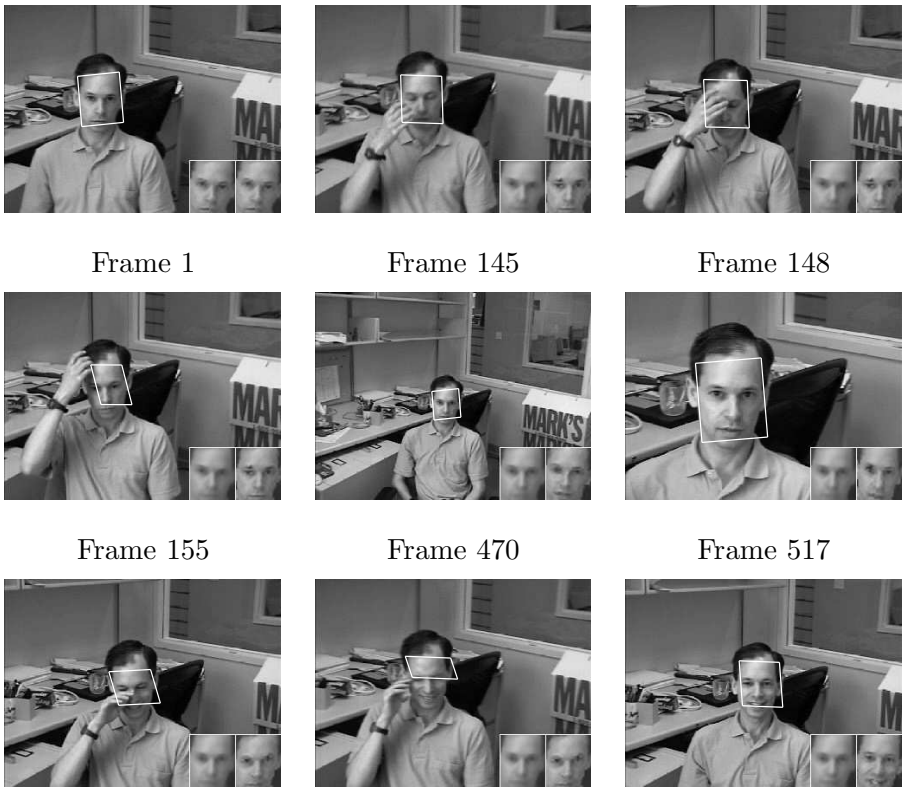
Figure 6.7 presents the tracking results on the video sequence featuring the following variations: moderate lighting variations, quick scale changes (back and forth) in the middle of the sequence, and occlusion (twice). The results are obtained by incorporating the occlusion analysis in the particle filter, but we did not use the F-component. Notice that the adaptive appearance model remains fixed during occlusion.

Figure 6.8 presents the tracking results obtained using the particle filter without occlusion analysis. We have found that the predicted velocity actually accounts for the motion of the occluding hand since the outlier pixels (mainly on the hand) dominate the image difference ($\mathcal{T}\{\mathsf{y}_t; \tilde{\theta}_t\} - \hat{\mathsf{z}}_{t-1}$). Updating the appearance model deteriorates the situation.

Figure 6.5(e) plots the scale estimate against time t. We clearly observe a rapid scale change (a sudden increase followed by a decrease within about 50 frames) in the middle of the sequence (though hard to display the recovered scale estimates are in perfect synchrony with the video data).

6.3.4 Comparison

We illustrate the effectiveness of our adaptive approach ('adp') by comparing the particle filter either with (a) an adaptive motion model but a fixed appearance model ('fa'), or with (b) a fixed motion model but an adaptive appearance model ('fm'); or with (c) a fixed motion model and a fixed appearance model ('fb'). Table



Frame 1 Frame 145 Frame 148

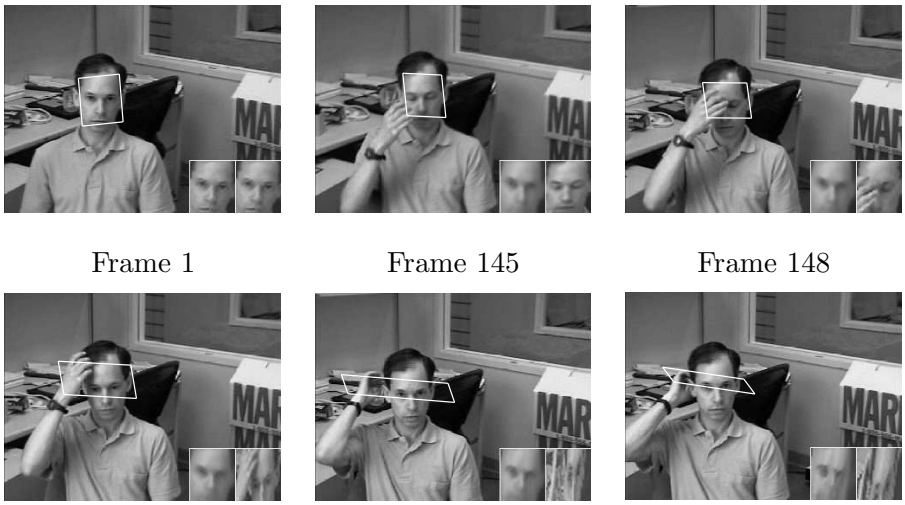
Frame 155 Frame 470 Frame 517

Frame 685 Frame 695 Frame 800

Figure 6.7: The face sequence. Frames 145, 148, and 155 show the first occlusion. Frames 470 and 517 show the smallest and largest face observed. Frames 685, 690, and 710 show the second occlusion.

6.1 lists the tracking results obtained using particle filters under the above situations, where 'adp & occ' refers to the adaptive approach with occlusion handling. Figure 6.4 also shows the tracking results on the car sequence when the 'fa' and 'fm' options are used.

Table 6.1 seems to suggest that the adaptive motion model plays a more important role than the adaptive appearance model since 'fa' always yields successful tracking while 'fm' fails, the reasons being that (i) the fixed motion model is unable to adapt to quick motion present in the video sequences, and (ii) the appearance



Frame 1 Frame 145 Frame 148

Frame 155 Frame 170 Frame 200

Figure 6.8: Tracking results on the face sequence using the adaptive particle filter without occlusion analysis.

changes in the video sequences, though significant in some cases, are still within the range of the fixed appearance model. However, as seen in the videos, 'adp' produces much smoother tracking results than 'fa', demonstrating the power of the adaptive appearance model.

Chapter 7

Simultaneous Tracking and Recognition

Following [58], we define a *still-to-video* scenario: the gallery consists of still facial templates and the probe set consists of video sequences containing the facial region. Denote the gallery as $\mathcal{I} = \{\mathsf{I}_1, \mathsf{I}_2, \ldots, \mathsf{I}_N\}$, indexed by the identity variable n, which lies in a finite sample space $\mathcal{N} = \{1, 2, \ldots, N\}$. Though significant research has been conducted on the still-to-still face recognition problem, research efforts on still-to-video recognition, are relatively fewer due to the following challenges [27] in typical surveillance applications: poor video quality, significant illumination and pose variations, and low image resolution. Most existing video-based recognition systems [79] attempt the following: the face is first detected and then tracked over time. Only when a frame satisfying certain criteria (size, pose) is acquired, recognition is performed using still-to-still recognition technique. For this, the face part is cropped from the frame and transformed or registered using appropriate transformations. This *tracking-then-recognition* approach attempts to resolve uncertainties in tracking and recognition *sequentially and separately.*

There are several unresolved issues in the *tracking-then-recognition* approach: criteria for selecting good frames and estimation of parameters for registration. Also, still-to-still recognition does not effectively exploit temporal information. A common strategy that selects several good frames, performs recognition on each frame and then votes on these recognition results for a final solution is rather *ad hoc*.

To overcome these difficulties, we propose a *tracking-and-recognition* approach, which attempts to resolve uncertainties in tracking and recognition *simultaneously* in a unified probabilistic framework. To fuse temporal information, the time series state space model is adopted to characterize the evolving kinematics and identity in the probe video. Three basic components of the model are:

- a *motion equation* governing the kinematic behavior of the tracking motion vector,
- an *identity equation* governing the temporal evolution of the identity variable,
- an *observation equation* establishing a link between the motion vector and the identity variable.

Using the SIS [114, 118, 153, 157, 159] technique, the joint posterior distribution of the motion vector and the identity variable, i.e., $\mathtt{p}(n_t, \theta_t|\mathtt{y}_{0:t})$ [1] is estimated at each time instant and then propagated to the next time instant governed by motion and identity equations. The marginal distribution of the identity variable, i.e., $\mathtt{p}(n_t|\mathtt{y}_{0:t})$, is estimated to provide a recognition result. An SIS algorithm is developed to approximate the distribution $\mathtt{p}(n_t|\mathtt{y}_{0:t})$ in the still-to-video scenario.

[1] For notational convenience, e.g. in (7.5) and (7.6), we introduce in this chapter a dummy variable $\mathtt{y}_0$.

It achieves computational efficiency over its CONDENSATION counterpart by considering the discrete nature of the identity variable.

It is worth emphasizing that (i) our model can take advantage of any still-to-still recognition algorithm [41, 44, 48, 62] by embedding distance measures used therein in our likelihood measurement; and (ii) it allows a variety of image representations and transformations. Section 7.3.4 presents an enhancement technique by incorporating the sophisticated appearance-based models in Chapter 6. The appearance models are used for tracking (modeling inter-frame appearance changes) and recognition (modeling appearance changes between video frames and gallery images), respectively. Table 7.1 summarizes the proposed approach and others, in term of using temporal information.

Process	Operation	Temporal information
Visual tracking	Modeling the inter-frame differences	Used in tracking
Visual recognition	Modeling the difference between probe and gallery images	Not applicable
Tracking-then-recognition	Combining tracking and recognition sequentially	Used only in tracking
Tracking-and-recognition	Unifying tracking and recognition	Used in both tracking and recognition

Table 7.1: Use of temporal information in various tracking/recognition processes.

Chapter organization

The organization of the chapter is as follows: Section 7.1 reviews some related studies on (i) face modeling and recognition and (ii) video-based tracking and recog-

nition in the literature. Section 7.2 introduces the time series state space model for recognition and establishes the time-evolving behavior of $\mathtt{p}(n_t|\mathtt{y}_{0:t})$. Section 7.2.3 briefly reviews the SIS principles from the viewpoint of a general state space model and develops a SIS algorithm to solve the still-to-video recognition problem, with special emphasis on its computational efficiency. Section 7.3 describes the experimental scenarios for still-to-video recognition and presents results using data collected at UMD, NIST/USF, and CMU (MoBo database) as part of the DARPA HumanID effort.

7.1 Related Literature

7.1.1 Face modeling and recognition

Statistical approaches to face modeling have been very popular since Turk and Pentland's work on eigenface [62]. In the statistical approach, the two-dimensional appearance of face image is treated as a vector by scanning the image in lexicographical order, with the vector dimension being the number of pixels in the image. In the eigenface approach [62], all face images consists of a distinctive face subspace. This subspace is linear and spanned by the eigenvectors of the covariance matrix found using PCA. Typically we keep the number of eigenvectors much less than the true dimension of the vector space. The task of face recognition is then to find the closest matches in this face subspace. However, PCA might not be efficient in terms of recognition accuracy since the construction of the face subspace does not capture discrimination between humans. This motivates the use of LDA [41, 44] and its variants. In LDA, the linear subspace is constructed [7] in such a manner that the within-class scatter is minimized and the between-class scatter is

maximized. This idea is further generalized in the approach called Bayesian face recognition [55], where intra-personal space (IPS) and extra-personal space (EPS) are used in lieu of within-class scatter and between-class scatter measures. The IPS models the variations in the appearance of the same individual and the EPS models the variations in appearances due to differences in the identity. Probabilistic subspace density is then fitted on each space. A Bayesian decision is taken using a *maximum a posteriori* (MAP) rule to determine the identity.

In the famous EGM [48] algorithm, the face is represented as a labeled graph. The nodes of the graph are located at facial landmarks, e.g., the pupils, the tip of nose, etc. Also, each node is labeled with jets derived from responses obtained by convolving the image with a family of Gabor functions. The edge characterizes the geometric distance between two nodes. Face recognition is then formalized as a graph matching problem.

All the above approaches are based on 2-D appearance and perform poorly when significant pose and illumination variations are present [58]. To completely resolve such challenges, 3-D face modeling [66, 83] is necessary. However, building a 3-D face model is a very difficult and complicated task in the literature even though structure from motion has been studied for several decades.

7.1.2 Video-based tracking and recognition

Nearly all video-based recognition systems apply still-image-based recognition to selected good frames. The face images are warped into frontal views whenever pose and depth information about the faces is available [79].

In [82, 90, 93], RBF (Radial Basis Function) networks are used for tracking and recognition purposes. In [82], the system uses an RBF (Radial Basis Function)

network for recognition. Since no warping is done, the RBF network has to learn the individual variations as well as possible transformations. The performance appears to vary widely, depending on the size of the training data. [93] presents a fully automatic person authentication system. The system uses video break, face detection, and authentication modules and cycles over successive video images until a high recognition confidence is reached. This system was tested on three image sequences; the first was taken indoors with one subject present, the second was taken outdoors with two subjects, and the third was taken outdoors with one subject in stormy conditions. Perfect results were reported on all three sequences, when verified against a database of 20 still face images.

In [92], a system called *PersonSpotter* is described. This system is able to capture, track and recognize a person walking toward or passing a stereo CCD camera. It has several modules, including a head tracker, and a landmark finder. The landmark finder uses a dense graph consisting of 48 nodes learned from 25 example images to find landmarks such as eyes and nose tip. An elastic graph matching scheme is employed to identify the face.

A multimodal based person recognition system is described in [79]. This system consists of a face recognition module, a speaker identification module, and a classifier fusion module. The most reliable video frames and audio clips are selected for recognition. The 3D head information is used to detect the presence of an actual person as opposed to an image of that person. Recognition and verification rates of 100% were achieved for 26 registered clients.

In [87, 88], recognition of face over time is implemented by constructing a face identity surface. The face is first warped to a frontal view, and its Kernel Discriminant Analysis (KDA) features over time form a trajectory. It is shown

that the trajectory distances accumulate recognition evidence over time.

In [86], a generic approach to simultaneous object tracking and verification is proposed. The approach is based on posterior probability density estimation using sequential Monte Carlo methods [118, 153, 157, 159]. Tracking is formulated as a probability density propagation problem and the algorithm also provides verification results. However, no systematic evaluation of recognition was done. Our approach looks similar to this algorithm; however, there are significant differences from the algorithm described in [86]. (i) In [86], basically only the tracking motion vector is parameterized in the state-space model. The identity is involved only in the initialization step to rectify the template onto the first frame of the sequence. However, in our approach both tracking motion vector and identity variables are parameterized in the state-space model, which offers us one more degree of freedom and leads to a different approach for deriving the solution. (ii) The SIS technique is used in both approaches to numerically approximate the posterior probability given the observation. Again in [86], it is the posterior probability of motion vector and the verification probability is estimated by marginalizing over a proper region of state space redefined at each time instant. However, we always compute the joint density, i.e., the posterior probability of motion vector and identity variable and the posterior probability of identity variable is just a free estimate obtained by marginalizing over the motion vector. Note that there is no time propagation of verification probability in [86] while we always propagate the joint density. One consequence is that we guarantee that $\sum_{n_t \in \mathcal{N}} \mathtt{p}(n_t|\mathtt{y}_{0:t}) = 1$, but there is no such guarantee in [86].

7.2 Stochastic Models and Algorithms for Recognition from Video

In this section, we present the details on the propagation model for recognition and discuss its impact on the posterior distribution of identity variable.

7.2.1 Time series state space model

Motion equation

In its most general form, the motion model can be written as

$$\theta_t = \mathsf{g}(\theta_{t-1}, \mathsf{u}_t); \quad t \geq 1, \tag{7.1}$$

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

Choice of θ_t is application dependent. Affine motion parameters are often used when there is no significant pose variation available in the video sequence. However, if a 3-D face model is used, then the 3-D motion parameters should be used accordingly.

Identity equation

$$n_t = n_{t-1}; \quad t \geq 1, \tag{7.2}$$

assuming that the identity does not change as time proceeds.

Observation equation

By assuming that the transformed observation is a noise-corrupted version of some still template in the gallery, the observation equation can be written as

$$\mathcal{T}\{\mathsf{y}_t;\theta_t\} = \mathsf{I}_{n_t} + \mathsf{v}_t; \quad t \geq 1, \tag{7.3}$$

where v_t is *observation noise* at time t, whose distribution determines the observation likelihood $\mathsf{p}(\mathsf{y}_t|n_t,\theta_t)$, and $\mathcal{T}\{\mathsf{y}_t;\theta_t\}$ is a transformed version of the observation y_t. This transformation could be either geometric or photometric or both. However, when confronting sophisticated scenarios, this model is far from sufficient. One should use the complicated likelihood measurement as shown in Section 7.3.2.

We assume statistical independence between all noise variables and prior knowledge on the distributions $\mathsf{p}(\theta_0|\mathsf{y}_0)$ and $\mathsf{p}(n_0|\mathsf{y}_0)$. Using the overall state vector $\mathsf{x}_t = (n_t,\theta_t)$, Eq. (7.1) and (7.2) can be combined into one state equation (in a normal sense) which is completely described by the overall state transition probability

$$\mathsf{p}(\mathsf{x}_t|\mathsf{x}_{t-1}) = \mathsf{p}(n_t|n_{t-1})\mathsf{p}(\theta_t|\theta_{t-1}) \ . \tag{7.4}$$

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

7.2.2 Posterior probability of identity variable

The evolution of the posterior probability $\mathsf{p}(n_t|\mathsf{y}_{0:t})$ as time proceeds is very interesting to study as the identity variable does not change by assumption, i.e.,

$\mathtt{p}(n_t|n_{t-1}) = \delta(n_t - n_{t-1})$, where $\delta(.)$ is a discrete impulse function at zero.

Using time recursion, Markov properties, and statistical independence embedded in the model, we can easily derive:

$$\begin{aligned}
\mathtt{p}(n_{0:t}, \theta_{0:t}|\mathtt{y}_{0:t}) &= \mathtt{p}(n_{0:t-1}, \theta_{0:t-1}|\mathtt{y}_{0:t-1}) \frac{\mathtt{p}(\mathtt{y}_t|n_t, \theta_t)\mathtt{p}(n_t|n_{t-1})\mathtt{p}(\theta_t|\theta_{t-1})}{\mathtt{p}(\mathtt{y}_t|\mathtt{y}_{0:t-1})} \\
&= \mathtt{p}(n_0, \theta_0|\mathtt{y}_0) \prod_{s=1}^{t} \frac{\mathtt{p}(\mathtt{y}_s|n_s, \theta_s)\mathtt{p}(n_s|n_{s-1})\mathtt{p}(\theta_s|\theta_{s-1})}{\mathtt{p}(\mathtt{y}_s|\mathtt{y}_{0:s-1})} \\
&= \mathtt{p}(n_0|\mathtt{y}_0)\mathtt{p}(\theta_0|\mathtt{y}_0) \prod_{s=1}^{t} \frac{\mathtt{p}(\mathtt{y}_s|n_s, \theta_s)\delta(n_s - n_{s-1})\mathtt{p}(\theta_s|\theta_{s-1})}{\mathtt{p}(\mathtt{y}_s|\mathtt{y}_{0:s-1})}. \quad (7.5)
\end{aligned}$$

Therefore, by marginalizing over $\theta_{0:t}$ and $n_{0:t-1}$, we obtain

$$\mathtt{p}(n_t = l|\mathtt{y}_{0:t}) = \mathtt{p}(l|\mathtt{y}_0) \int_{\theta_0} \dots \int_{\theta_t} \mathtt{p}(\theta_0|\mathtt{y}_0) \prod_{s=1}^{t} \frac{\mathtt{p}(\mathtt{y}_s|l, \theta_s)\mathtt{p}(\theta_s|\theta_{s-1})}{\mathtt{p}(\mathtt{y}_s|\mathtt{y}_{0:s-1})} \mathtt{d}\theta_t \dots \mathtt{d}\theta_0. \quad (7.6)$$

Thus $\mathtt{p}(n_t = l|\mathtt{y}_{0:t})$ is determined by the prior distribution $\mathtt{p}(n_0 = l|\mathtt{y}_0)$ and the product of the likelihood functions, $\prod_{s=1}^{t} \mathtt{p}(\mathtt{y}_s|l, \theta_s)$. If a uniform prior is assumed, then $\prod_{s=1}^{t} \mathtt{p}(\mathtt{y}_s|l, \theta_s)$ is the only determining factor.

In the appendix, we show that, under some minor assumptions, the posterior probability for the correct identity l, $\mathtt{p}(n_t = l|\mathtt{y}_{0:t})$, is lower-bounded by an increasing curve which converges to 1.

To measure the evolving uncertainty remaining in the identity variable as observations accumulate, we use the notion of entropy [4]. In the context of this problem, conditional entropy $\mathtt{H}(n_t|\mathtt{y}_{0:t})$ is used. However, the knowledge of $\mathtt{p}(\mathtt{y}_{0:t})$ is needed to compute $\mathtt{H}(n_t|\mathtt{y}_{0:t})$. We assume that it degenerates to an impulse at the actual observations $\tilde{\mathtt{y}}_{0:t}$ since we observe only this particular sequence, i.e., $\mathtt{p}(\mathtt{y}_{0:t}) = \delta(\mathtt{y}_{0:t} - \tilde{\mathtt{y}}_{0:t})$. Thus,

$$\mathtt{H}(n_t|\mathtt{y}_{0:t}) = -\sum_{n_t=1}^{N} \mathtt{p}(n_t|\tilde{\mathtt{y}}_{0:t}) \log_2 \mathtt{p}(n_t|\tilde{\mathtt{y}}_{0:t}). \quad (7.7)$$

Under the assumptions listed in the appendix, we expect that $\mathtt{H}(n_t|\mathtt{y}_{0:t})$ decreases

as time proceeds since we start from an equi-probable distribution to a degenerate one.

7.2.3 SIS algorithms and computational efficiency

Consider a general time series state space model fully determined by (i) the overall state transition probability $\mathsf{p}(\mathsf{x}_t|\mathsf{x}_{t-1})$, (ii) the observation likelihood $\mathsf{p}(\mathsf{y}_t|\mathsf{x}_t)$, and (iii) prior probability $\mathsf{p}(\mathsf{x}_0)$ and statistical independence among all the noise variables. We wish to compute the posterior probability $\mathsf{p}(\mathsf{x}_t|\mathsf{y}_{0:t})$.

If the model is linear with Gaussian noise, it is analytically solvable by a Kalman filter which essentially propagates the mean and variance of a Gaussian distribution over time. For nonlinear and non-Gaussian cases, an extended Kalman filter (EKF) and its variants have been used to arrive at an approximate analytic solution [1]. Recently, the SIS technique or particle filter algorithm, a special case of Monte Carlo method, [118, 153, 157, 159] has been used to provide a numerical solution and propagate an arbitrary distribution over time. However, since we are dealing with a mixed-type distribution, additional properties are available to be exploited when developing the SIS algorithms.

First, two following two propositions are useful.

Proposition 7.1 When $\pi(x)$ is a PMF defined on a finite sample space, the proper sample set should exactly include all samples in the sample space.

Proposition 7.2 If a set of weighted random samples $\{(\mathsf{x}^{(m)}, \mathsf{y}^{(m)}, w^{(m)})\}_{m=1}^{M}$ is proper with respect to $\pi(\mathsf{x}, \mathsf{y})$, then a new set of weighted random samples $\{(\mathsf{y}'^{(k)}, w'^{(k)})\}_{k=1}^{K}$, which is proper with respect to $\pi(\mathsf{y})$, the marginal of $\pi(\mathsf{x}, \mathsf{y})$, can be constructed as follows:

1) Remove the repetitive samples from $\{\mathsf{y}^{(m)}\}_{m=1}^{M}$ to obtain $\{\mathsf{y}'^{(k)}\}_{k=1}^{K}$, where all

$\mathsf{y}^{'(k)}$'s are distinct;

2) Sum the weight $w^{(m)}$ belonging to the same sample $\mathsf{y}^{'(k)}$ to obtain the weight $w^{'(k)}$, i.e.,

$$w^{'(k)} = \sum_{m=1}^{M} w^{(m)} \delta(\mathsf{y}^{(m)} - \mathsf{y}^{'(k)}) \tag{7.8}$$

In the context of this framework, the posterior probability $\mathtt{p}(n_t, \theta_t | \mathsf{y}_{0:t})$ is represented by a set of *indexed and weighted* samples

$$\mathcal{S}_t = \{(n_t^{(m)}, \theta_t^{(m)}, w_t^{(m)})\}_{m=1}^{M} \tag{7.9}$$

with n_t as the above index. By Proposition 7.2, we can sum the weights of the samples belonging to the same index n_t to obtain a proper sample set $\{n_t, \beta_{n_t}\}_{n_t=1}^{N}$ with respect to the posterior PMF $\mathtt{p}(n_t | \mathsf{y}_{0:t})$.

A straightforward implementation of the particle filter algorithm (Figure 7.1) for simultaneous tracking and recognition is not efficient in terms of its computational load. Since $\mathcal{N} = \{1, 2, \ldots, N\}$ is a countable sample space, we need N samples for the identity variable n_t according to Proposition 7.1. Assume that, for each identity variable n_t, J samples are needed to represent θ_t. Hence, we need $M = J * N$ samples in total. Further assume that one resampling step takes T_r seconds (s), one predicting step T_p s, computing one transformed image T_t s, evaluating likelihood once T_l s, one updating step T_u s. Obviously, the bulk of computation is $J*N*(T_r+T_p+T_t+T_l)$ s to deal with one video frame as the computational time for the normalizing step and the marginalizing step is negligible. It is well known that computing the transformed image is much more expensive than other operations, i.e., $T_t >> \max(T_r, T_p, T_l)$. Therefore, as the number of templates N grows, the computational load increases dramatically.

There are various approaches in the literature to reduce the computational cost of the conventional particle filter algorithm. In [128], random particles are guided

Initialize *a sample set* $\mathcal{S}_0 = \{(n_0^{(m)}, \theta_0^{(m)}, 1)\}_{m=1}^{M}$ *according to prior distributions* $\mathbf{p}(n_0|\mathbf{y}_0)$ *and* $\mathbf{p}(\theta_0|\mathbf{y}_0)$.

For $t = 1, 2, \ldots$

For $m = 1, 2, \ldots, M$

Resample $\mathcal{S}_{t-1} = \{(n_{t-1}^{(m)}, \theta_{t-1}^{(m)}, w_{t-1}^{(m)})\}_{m=1}^{M}$ *to obtain a new sample* $(n_{t-1}^{'(m)}, \theta_{t-1}^{'(m)}, 1)$.

Predict *a sample by drawing* $(n_t^{(m)}, \theta_t^{(m)})$ *from* $\mathbf{p}(n_t|n_{t-1}^{'(m)})$ *and* $\mathbf{p}(\theta_t|\theta_{t-1}^{'(m)})$.

Compute *the transformed image* $\mathbf{z}_t^{(m)} = \mathcal{T}\{\mathbf{y}_t; \theta_t^{(m)}\}$.

Update *the weight using* $\alpha_t^{(m)} = \mathbf{p}(\mathbf{y}_t|n_t^{(m)}, \theta_t^{(m)})$.

End

Normalize *each weight using* $w_t^{(m)} = \alpha_t^{(m)} / \sum_{m=1}^{M} \alpha_t^{(m)}$.

Marginalize *over* θ_t *to obtain the weight* β_{n_t} *for* n_t.

End

Figure 7.1: The conventional particle filter algorithm for simultaneous tracking and recognition.

by deterministic search. Assumed density filtering approach [148], different from particle filter, is even more efficient. Those approaches are general and do not explicitly exploit the special structure of the distribution in this setting: a mixed distribution of continuous and discrete variables. To this end, we propose the following algorithm.

As the sample space $\mathcal{N}$ is countably finite, an exhaustive search of sample space $\mathcal{N}$ is possible. Mathematically, we release the random sampling in the identity variable n_t by constructing samples as follows: for each $\theta_t^{(j)}$,

$$(1, \theta_t^{(j)}, w_{t,1}^{(j)}), (2, \theta_t^{(j)}, w_{t,2}^{(j)}), \ldots, (N, \theta_t^{(j)}, w_{t,N}^{(j)}).$$

We in fact use the following notation for the sample set,

$$\mathcal{S}_t = \{(\theta_t^{(j)}, w_t^{(j)}, w_{t,1}^{(j)}, w_{t,2}^{(j)}, \ldots, w_{t,N}^{(j)})\}_{j=1}^{J}, \tag{7.10}$$

with $w_t^{(j)} = \sum_{n=1}^{N} w_{t,n}^{(j)}$. The proposed algorithm is summarized in Figure 7.2.

Initialize *a sample set* $\mathcal{S}_0 = \{(\theta_0^{(j)}, N, 1, ..., 1)\}_{j=1}^{J}$ *according to prior distribution* $p(\theta_0|z_0)$.

For $t = 1, 2, \ldots$

For $j = 1, 2, \ldots, J$

Resample $\mathcal{S}_{t-1} = \{(\theta_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^{J}$ *to obtain a new sample* $(\theta_{t-1}^{'(j)}, 1, w_{t-1,1}^{'(j)}, \ldots, w_{t-1,N}^{'(j)})$, *where* $w_{t-1,n}^{'(j)} = w_{t-1,n}^{(j)} / w_{t-1}^{(j)}$ *for* $n = 1, 2, \ldots, N$.

Predict *a sample by drawing* $(\theta_t^{(j)})$ *from* $p(\theta_t|\theta_{t-1}^{'(j)})$.

Compute *the transformed image* $z_t^{(m)} = \mathcal{T}\{y_t; \theta_t^{(m)}\}$.

For $n = 1, \ldots, N$

Update *the weight using* $\alpha_{t,n}^{(j)} = w_{t-1,n}^{'(j)} * p(y_t|n, \theta_t^{(j)})$.

End

End

Normalize *each weight using* $w_{t,n}^{(j)} = \alpha_{t,n}^{(j)} / \sum_{n=1}^{N} \sum_{j=1}^{J} \alpha_{t,n}^{(j)}$ *and* $w_t^{(j)} = \sum_{n=1}^{N} w_{t,n}^{(j)}$.

Marginalize *over* θ_t *to obtain the weight* β_{n_t} *for* n_t.

End

Figure 7.2: The computationally efficient particle filter algorithm for simultaneous tracking and recognition.

The crux of this algorithm lies in the fact that, instead of propagating random samples on both motion vector and identity variable, we can keep the samples on the identity variable fixed and let those on the motion vector be random. Although

we propagate only the marginal distribution for motion tracking, we still propagate the joint distribution for recognition purposes.

The bulk of computation of the proposed algorithm is $J*(T_r+T_p+T_t)+J*N*T_l$ s, a tremendous improvement over the conventional particle filter when dealing with a large database since the majority computational time $J*T_t$ does not depend on N.

7.3 Still-to-Video Face Recognition Experiments

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

Database-0 was collected outside a building. Subjects walked straight towards a video camera in order to simulate typical scenarios in visual surveillance. Database-0 includes one face gallery, and one probe set. The images in the gallery are listed in Figure 7.3. The probe contains 12 videos, one for each individual. Figure 7.3 gives some frames in a probe video.

In Database-1, we have video sequences with subjects walking in a slant path towards the camera. There are 30 subjects, each having one face template. There are one face gallery and one probe set. The face gallery is shown in Figure 7.4. The probe contains 30 video sequences, one for each subject. Figure 7.4 gives some example frames extracted from one probe video. As far as imaging conditions are concerned, the gallery is very different from the probe, especially in lighting. This is similar to the 'FC' test protocol of the FERET test [58]. These images/videos were collected, as part of the HumanID project, by National Institute of Standards and Technology and University of South Florida researchers.

Database-2, Motion of Body (MoBo) database, was collected at the Carnegie Mellon University [81] under the HumanID project. There are 25 different individuals in total. The video sequences show the individuals walking on a tread-mill so that they move their heads naturally. Different walking styles have been simulated to assure a variety of conditions that are likely to appear in real life: *walking slowly*, *walking fast*, *inclining* and *carrying an object*. Therefore, four videos per person and 99 videos in total (with one *carrying* video missing) are available. However, the probe set we use in this section includes only 25 *slowWalk* videos. Some example images of the videos (*slowWalk*) are shown in Figure 7.5. Figure 7.5 also shows the face gallery in Database-2 with face images in almost frontal view cropped from probe videos and then normalized using their eye positions.

Table 7.2 summaries the features of the three databases.

Database	Database-0	Database-1	Database-2
No. of subjects	12	30	25
Gallery	Frontal face	Frontal face	Frontal face
Motion in probe	Walking straight towards the camera	Walking in an angle towards the camera	Walking on tread-mill
Illumination variation	No	Large	No
Pose variation	No	Slight	Large

Table 7.2: Summary of three databases experimented.

7.3.1 Results for Database-0

We consider an affine transformation. Specifically, the motion is characterized by $\theta = (a_1, a_2, a_3, a_4, t_x, t_y)$ where $\{a_1, a_2, a_3, a_4\}$ are deformation parameters and $\{t_x, t_y\}$ are 2-D translation parameters. It is a reasonable approximation since there

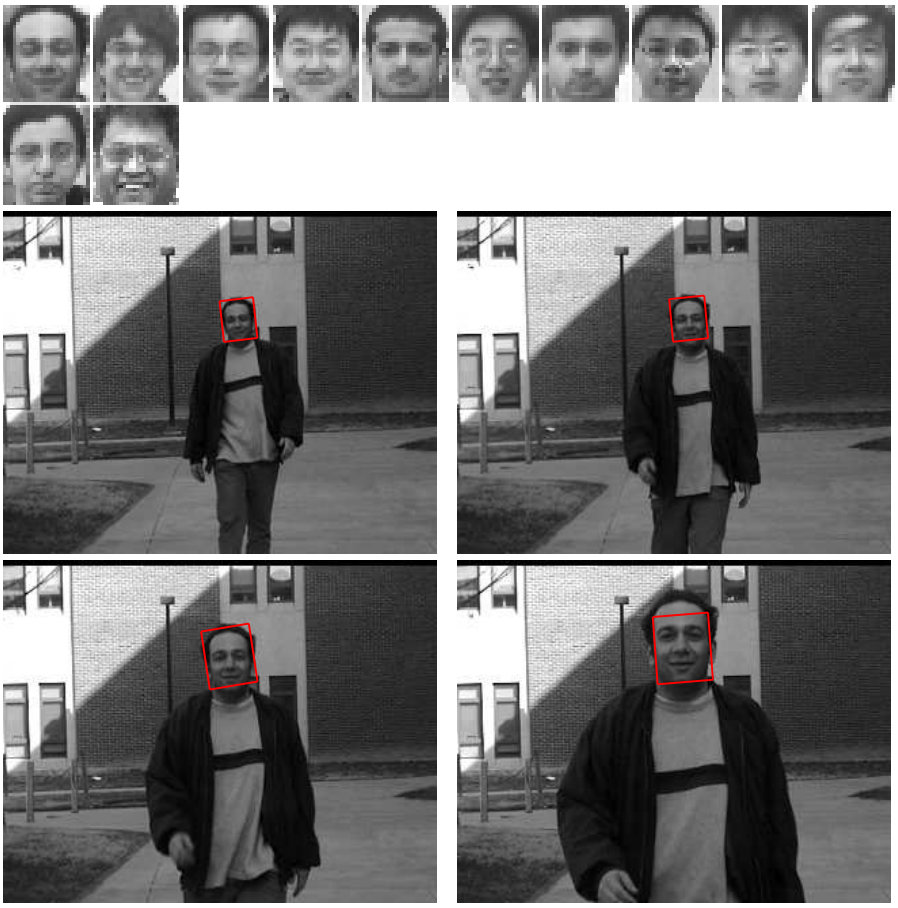


Figure 7.3: Database-0. The 1st row: the face gallery with image size being 30×26. The 2nd and 3rd rows: 4 example frames in one probe video with image size being 320×240 while the actual face size ranges approximately from 30×30 in the first frame to 50×50 in the last frame. Notice that the sequence is taken under a well-controlled condition so that there are no illumination or pose variations between the gallery and the probe.

is no significant out-of-plane motion as the subjects walk towards the camera. Regarding the photometric transformation, only zero-mean-unit-variance operator is performed to partially compensate for contrast variations. The complete transformation $\mathcal{T}\{\mathsf{y};\theta\}$ is processed as follows: affine transform y using $\{a_1, a_2, a_3, a_4\}$,

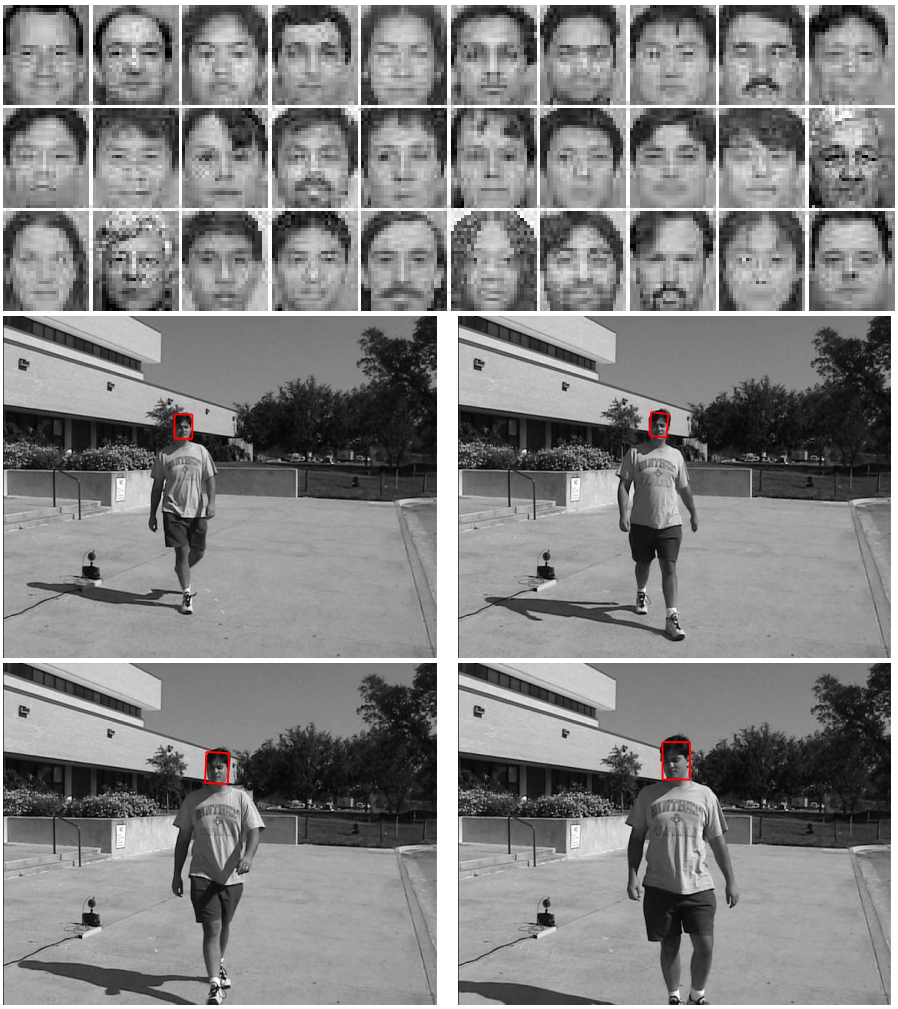


Figure 7.4: Database-1. The 1st row: the face gallery with image size being 30×26. The 2nd and 3rd rows: 4 example frames in one probe video with image size being 720×480 while the actual face size ranges approximately from 20×20 in the first frame to 60×60 in the last frame. Notice the significant illumination variations between the probe and the gallery.

crop out the interested region at position $\{t_x, t_y\}$ with the same size as the still template in the gallery, and perform zero-mean-unit-variance operation.

Prior distribution $\mathtt{p}(\theta_0|\mathtt{y}_0)$ is assumed to be Gaussian, whose mean comes from

Figure 7.5: Database-2. The 1st row: the face gallery with image size being 30×26. The 2nd and 3rd rows: some example frames in one probe video (*slowWalk*). Each video consists of 300 frames (480×640 pixels per frame) captured at 30 Hz. The inner face regions in these videos contain between 30×30 and 40×40 pixels. Notice the significant pose variation available in the video.

the initial detector and whose covariance matrix is manually specified.

A time-invariant first-order Markov Gaussian model with constant velocity is used for modeling motion transition. Given the scenario that the subject is walking

towards the camera, the scale increases with time. However, under perspective projection, this increase is no longer linear, causing the constant-velocity model to be not optimal. However, experimental results show that as long as the samples of θ can cover the motion, this model is sufficient.

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

$$\mathsf{p}_1(\mathsf{y}_t|n_t,\theta_t) = \mathtt{LAP}(\|\mathcal{T}\{\mathsf{y}_t;\theta_t\} - \mathsf{I}_{n_t}\|;\sigma_1,\tau_1) \tag{7.11}$$

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

$$\mathtt{LAP}(x;\sigma,\tau) = \begin{cases} \sigma^{-1}\exp(-x/\sigma) & \text{if } \ x \leq \tau\sigma \\ \sigma^{-1}\exp(-\tau) & \text{otherwise} \end{cases} \tag{7.12}$$

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

Figure 7.6 presents the plot of the posterior probability $\mathsf{p}(n_t|\mathsf{y}_{0:t})$, the conditional entropy $\mathsf{H}(n_t|\mathsf{y}_{0:t})$ and the minimum mean square error (MMSE) estimate of the scale parameter $sc = \sqrt{(a_1^2 + a_2^2 + a_3^2 + a_4^2)/2}$, all against t. In Figure 7.3, the tracked face is superimposed on the image using a bounding box.

Suppose the correct identity for Figure 7.3 is l. From Figure 7.6, we can easily observe that the posterior probability $\mathsf{p}(n_t = l|\mathsf{y}_{0:t})$ increases as time proceeds and eventually approaches 1, and all others $\mathsf{p}(n_t = j|\mathsf{y}_{0:t})$ for $j \neq l$ go to 0. Figure 7.6 also plots the decrease in conditional entropy $\mathsf{H}(n_t|\mathsf{y}_{0:t})$ and the increase in scale parameter, which matches with the scenario of a subject walking towards a camera.

Table 7.3 summarizes the average recognition performance and computational time of the conventional and the proposed particle filter algorithm when applied to Database-0. Both algorithms achieved 100% recognition rate with top match. The proposed algorithm is much more efficient than the conventional one. It is more than 10 times faster as shown in Table I. This experiment was implemented in C++ on a PC with P-III 1G CPU and 512M RAM with the number of motion samples J chosen to be 200, the number of templates in the gallery N to be 12.

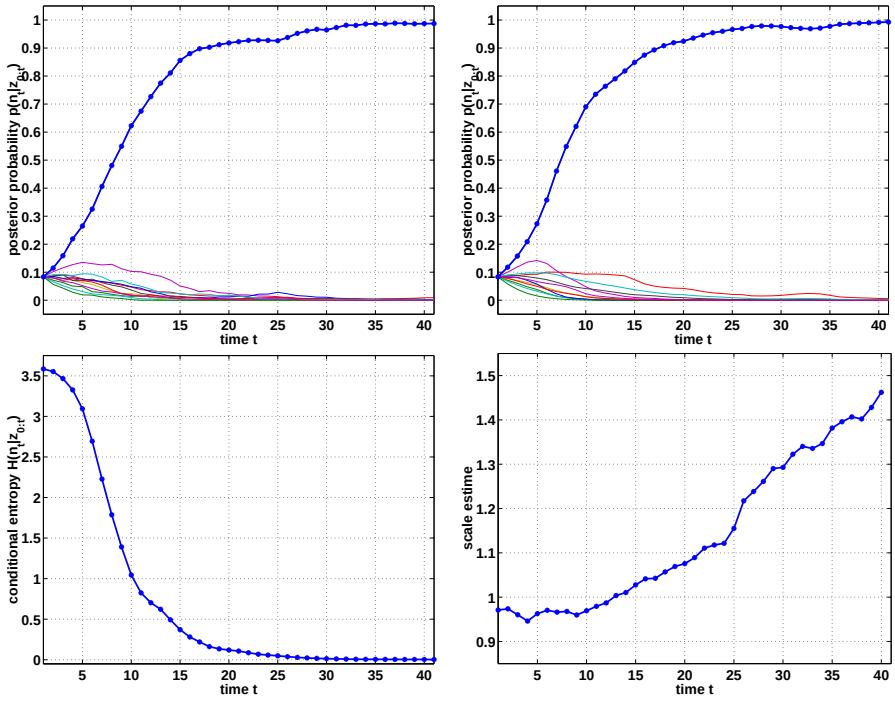


Figure 7.6: Posterior probability $\mathsf{p}(n_t|\mathsf{y}_{0:t})$ against time t, obtained by the CONDENSATION algorithm (top left) and the proposed algorithm (top right). Conditional entropy $\mathsf{H}(n_t|\mathsf{y}_{0:t})$ (bottom left) and MMSE estimate of scale parameter sc (bottom right) against time t. The conditional entropy and the MMSE estimate are obtained using the proposed algorithm.

Algorithm	Conventional algorithm	Efficient algorithm
Recognition rate within top 1 match	100%	100%
Time per frame	7s	0.5s

Table 7.3: Recognition performance of algorithms when applied to Database-0.

7.3.2 Results for Database-1

Case 1: Tracking and Recognition using Laplacian Density

We first investigate the performance using the same setting as described in Section 7.3.1. In other words, we still use the affine transformation, first-order Markov Gaussian state transition model, 'truncated' Laplacian observation likelihood, etc.

Table 7.4 shows that the recognition rate is very poor, only 13% are correctly identified using top match. The main reason is that the 'truncated' Laplacian density is far from sufficient to capture the appearance difference between the probe and the gallery, thereby indicating a need for a different appearance modeling. Nevertheless, the tracking accuracy [2] is reasonable with 83% successfully tracked because we are using multiple face templates in the gallery to track the specific face in the probe video. After all, faces in both the gallery and the probe belong to the same class of human face and it seems that the appearance change is within the class range.

[2]We manually inspect the tracking results by imposing the MMSE motion estimate on the final frame as shown in Figs. 7.3 and 7.4 and determine if tracking is successful or not for this sequence. This is done for all sequences and tracking accuracy is defined as the ratio of the number of sequences successfully tracked to the total number of all sequences.

Case 2: Pure Tracking using Laplacian Density

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

$$\mathsf{p}_2(\mathsf{y}_t|\theta_t) = LAP(\|\mathcal{T}\{\mathsf{y}_t;\theta_t\} - \mathsf{T}_0\|;\sigma_2,\tau_2), \tag{7.13}$$

where σ_2 and τ_2 are set manually. The other setting, such as motion parameter and model, is the same as in Case 1. We still can run the CONDENSATION algorithm to perform pure tracking.

Table 7.4 shows that 87% are successfully tracked by this simple tracking model, which implies that the appearance within the video remains similar.

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

Table 7.4: Performances of algorithms when applied to Database-1.

Case 3: Tracking and Recognition using Probabilistic Subspace Density

As mentioned in Case 1, we need a new appearance model to improve the recognition accuracy. As reviewed in Section 7.1.1, there are various approaches in the literature. We decided to use the approach suggested by Moghaddam et al. [55] due to its computational efficiency and high recognition accuracy. However, in our implementation, we model only intra-personal variations instead of both intra/extra-personal variations for simplicity.

We need at least two facial images for one identity to construct the intra-personal space (IPS). Apart from the available gallery, we crop out the second image from the video ensuring no overlap with the frames actually used in probe videos. Figure 7.7 (top row) shows a list of such images. Compare with Figure 7.4 to see how the illumination varies between the gallery and the probe.

We then fit a probabilistic subspace density [56] on top of the IPS. It proceeds as follows: a regular PCA is performed for the IPS. Suppose the eigensystem for the IPS is $\{(\lambda_i, \mathsf{e}_i)\}_{i=1}^{d}$, where d is the number of pixels and $\lambda_1 \geq ... \geq \lambda_d$. Only top s principal components corresponding to top s eigenvalues are then kept while the residual components are considered as isotropic. We refer the reader to the original paper [56] for the full details. Figure 7.7 (middle row) shows the eigenvectors for the IPS. The density is written as follows:

$$\mathtt{Q}_{IPS}(\mathsf{x}) = \{\frac{exp(-\frac{1}{2}\sum_{i=1}^{s}\frac{y_i^2}{\lambda_i})}{(2\pi)^{s/2}\prod_{i=1}^{s}\lambda_i^{1/2}}\}\{\frac{exp(-\frac{\epsilon^2}{2\rho})}{(2\pi\rho)^{(d-s)/2}}\}, \tag{7.14}$$

where $y_i = \mathsf{e}_i^T\mathsf{x}$ for $i = 1, ..., s$ is the i^{th} principal component of x, $\epsilon^2 = \|\mathsf{x}\|^2 - \sum_{i=1}^{s} y_i^2$ is the reconstruction error, and $\rho = (\sum_{i=s+1}^{d}\lambda_i)/(d-q)$. It is easy to write the likelihood as follows:

$$\mathtt{p}_3(\mathsf{y}_t|n_t, \theta_t) = \mathtt{Q}_{IPS}(\mathcal{T}\{\mathsf{y}_t; \theta_t\} - \mathsf{I}_{n_t}). \tag{7.15}$$

Table 7.4 lists the performance by using this new likelihood measurement. It turns out that the performance is significantly better that in Case 1, with 93% tracked successfully and 83% recognized within top 1 match. If we consider the top 3 matches, 97% are correctly identified.

Case 4: Tracking and Recognition using Combined Density

In Case 2, we have studied appearance changes within a video sequence. In Case 3, we have studied the appearance change between the gallery and the probe. In

Figure 7.7: Database-1. Top row: the second facial images for estimating probabilistic density. Middle row: top 10 eigenvectors for the IPS. Bottom row: the facial images cropped out from the largest frontal view.

Case 4, we attempt to take advantage of both cases by introducing a combined likelihood defined as follows:

$$\mathtt{p}_4(\mathtt{y}_t|n_t,\theta_t) = \mathtt{p}_3(\mathtt{y}_t|n_t,\theta_t)\mathtt{p}_2(\mathtt{y}_t|\theta_t) \tag{7.16}$$

Again, all other setting is the same as in Case 1. We now obtain the best performance so far: no tracking error, 93% are correctly recognized as the first match, and no error in recognition when top 3 matches are considered.

Case 5: Still-to-still Face Recognition

To make a comparison, we also performed an experiment on still-to-still face recognition. We selected the probe video frames with the best frontal face view (i.e. biggest frontal view) and cropped out the facial region by normalizing with respect to the eye coordinates manually specified. This collection of images is shown in Figure 7.7 (bottom row) and it is fed as probes into a still-to-still face recognition system with the learned probabilistic subspace as in Case 3. It turns out that the recognition result is 57% correct for the top one match, and 83% for the top 3 matches. The cumulative match curves for Case 1 and Cases 3-5 are presented in Figure 7.8. Clearly, Case 4 is the best among all. We also implemented the original algorithm by Moghaddam et al. [56], i.e., both intra/extra-personal variations are considered, the recognition rate is similar to that obtained in Case 5.

7.3.3 Results for Database-2

The recognition result for Database-2 is presented in Figure 7.8, using the cumulative match curve. We still use the same setting as in Case 1 of section 7.3.2. However, due to the pose variations present in the database, using one frontal view is not sufficient to represent all the appearances under different poses and the recognition rate is hence not so high, 56% when only the top match is considered and 88% when top 3 matches are considered. We do not use probabilistic subspace modeling for this database because such modeling requires manually cropping out multiple templates for each individual. Also, pre-selecting video frames from the same probe video and ensuring that they do not overlap with the probe frames is time-consuming. What is desirable is to automatically select such templates from different sources other than the probe video. Since we have multiple videos

available for one individual in Database-2, this motivates us to obtain more representative views for one face class, leading to the discussions in [194].

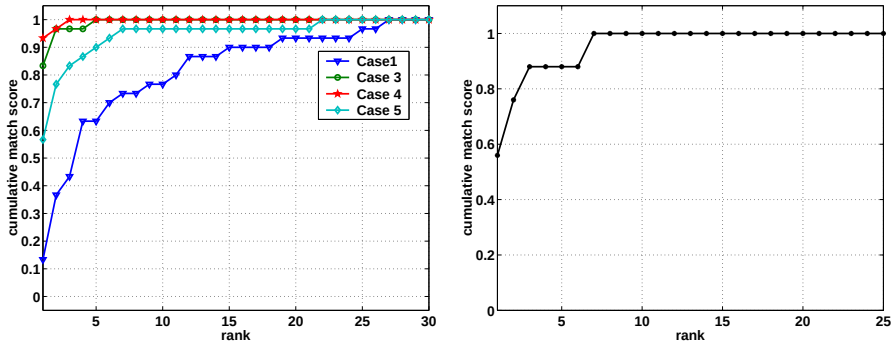


Figure 7.8: Cumulative match curves for Database-1 (left) and Database-2 (right).

7.3.4 Enhanced results

Visual tracking models the inter-frame appearance differences and visual recognition models the appearance differences between video frames and gallery images. Simultaneous tracking and recognition provides a mechanism of jointly modeling inter-frame appearance differences and the appearance differences between video frames and gallery images. As in Section 7.3.2, this joint modeling of appearance differences in both tracking and recognition in one framework actually improves both tracking and recognition accuracies over approaches that separate tracking and recognition as two tasks. The more effective the model choices are, improved performance in tracking and recognition is expected. We explore this avenue by incorporating the models used in Chapter 6.

We use the same adaptive-velocity motion model (6.29) and the same identity equation (7.2). The observation likelihood is modified to combine contributions (or scores) from both tracking and recognition in the likelihood yields the best performance in both tracking and recognition.

To compute the tracking score $\mathtt{p}_a(\mathtt{y}_t|\theta_t)$ which measures the inter-frame appearance changes, we use the appearance model introduced in Section 6.2.1 and the quantity defined in (6.10) as $\mathtt{p}_a(\mathtt{y}_t|\theta_t)$.

To compute the recognition score which measures the appearance changes between probe videos and gallery images, we assume the same model as in (7.3), i.e., the transformed observation is a noise-corrupted version of some still template in the gallery, and the noise distribution determines the recognition score $\mathtt{p}_n(\mathtt{y}_t|n_t, \theta_t)$. We will physically define this quantity below.

To fully exploit the fact that all gallery images are in frontal view, we also compute below how likely the patch $\mathtt{z}_t$ is in frontal view and denote this score by $\mathtt{p}_f(\mathtt{y}_t|\theta_t)$. If the patch is in frontal view, we accept a recognition score; otherwise, we simply set the recognition score as equiprobable among all identities, i.e., $1/N$. The complete likelihood $\mathtt{p}(\mathtt{y}_t|n_t, \theta_t)$ is now defined as

$$\mathtt{p}(\mathtt{y}_t|n_t, \theta_t) \propto \mathtt{p}_a \ \{\mathtt{p}_f \ \mathtt{p}_n + (1 - \mathtt{p}_f) \ N^{-1}\}. \tag{7.17}$$

Model components in detail

- *A. Modeling inter-frame appearance changes*

  Inter-frame appearance changes are related to the motion transition model and the appearance model for tracking, which were explained in Sections 6.2.1 and 6.2.2.

- *B. Being in frontal view*

  Since all gallery images are in frontal view, we simply measure the extent of being frontal by fitting a probabilistic subspace (PS) density on the top of the gallery images [54, 56], assuming that they are i.i.d. samples from the

frontal face space (FFS). $\mathtt{p}_f(\mathtt{y}_t|\theta_t)$ is written as follows:

$$\mathtt{p}_f(\mathtt{y}_t|\theta_t) = \mathtt{Q}_{FFS}(\mathtt{z}_t), \tag{7.18}$$

where the density $\mathtt{Q}(.)$ is defined same as that in (7.14).

- *C. Modeling appearance changes between probe video frames and gallery images*

  We adopt the MAP rule developed in [56] for the recognition score $\mathtt{p}_n(\mathtt{y}_t|n_t, \theta_t)$. Two subspaces are constructed to model appearance variations. The IPS is meant to cover all the variations in appearances belonging to the same person while the EPS is used to cover all the variations in appearances belonging to different people. More than one facial image per person is needed to construct the IPS. Apart from the available gallery, we crop out four images from the video ensuring no overlap with frames used in probe videos. The above PS density estimation method is applied separately to the IPS and the EPS, yielding two different eigensystems. The recognition score $\mathtt{p}_n(\mathtt{y}_t|n_t, \theta_t)$ is finally computed as, assuming equal priors on the IPS and the EPS,

$$\mathtt{p}_n(\mathtt{y}_t|n_t, \theta_t) = \frac{\mathtt{Q}_{IPS}(\mathtt{z}_t - \mathtt{I}_{n_t})}{\mathtt{Q}_{IPS}(\mathtt{z}_t - \mathtt{I}_{n_t}) + \mathtt{Q}_{EPS}(\mathtt{z}_t - \mathtt{I}_{n_t})}. \tag{7.19}$$

D. Proposed algorithm

We adjust the particle number J_t based on the following considerations. (i) The first issue is same as (6.31) based on the prediction error. (ii) As shown above, the uncertainty in the identity variable n_t is characterized by an entropy measure $\mathtt{H}_t$ for $\mathtt{p}(n_t|\mathtt{y}_{1:t})$ and $\mathtt{H}_t$ is a non-increasing function (under one weak assumption). Accordingly, we increase the number of particles by a fixed amount J_{fix} if H_t

Initialize *a sample set* $\mathcal{S}_0 = \{\theta_0^{(j)}, w_0^{(j)} = 1/J_0)\}_{j=1}^{J_0}$ *according to prior distribution* $p(\theta_0)$. *Set* $\beta_{0,l} = 1/N$. *Initialize the appearance mode* A_1.

For $t = 1, 2, \ldots$

Calculate *the MAP estimate* $\hat{\theta}_{t-1}$, *the adaptive motion shift* ν_t *by Eq. (6.21), the noise variance* r_t *by Eq. (6.30), and particle number* J_t *by Eq. (7.20).*

For $j = 1, 2, \ldots, J_t$

Draw *the sample* $\mathsf{u}_t^{(j)}$ *for* u_t *with variance* R_t.

Construct *the sample* $\theta_t^{(j)}$ *by Eq. (6.29).*

Compute *the transformed image* $\mathsf{z}_t^{(j)}$.

For $l = 1, 2, ..., N$

Update *the weight using* $\alpha_{t,l}^{(j)} = \beta_{t-1,l} p(\mathsf{y}_t | l, \theta_t^{(j)}) = \beta_{t-1,l} p(\mathsf{z}_t^{(j)} | l, \theta_t^{(j)})$ *by Eq. (7.17).*

End

End

Normalize *the weight using* $w_{t,l}^{(j)} = \alpha_{t,l}^{(j)} / \sum_{j,l} \alpha_{t,l}^{(j)}$ *and compute* $w_t^{(j)} = \sum_j w_{t,l}^{(j)}$ *and* $\beta_{t,l} = \sum_j w_{t,l}^{(j)}$.

Update *the appearance model* A_{t+1} *using* $\hat{\mathsf{z}}_t$.

End

Figure 7.9: The visual tracking and recognition algorithm.

increases; otherwise we deduct J_{fix} from J_t. Combining these two, we have

$$J_t = J_0 \frac{r_t}{r_0} + J_{fix} * (-1)^{i[\mathsf{H}_{t-1} < \mathsf{H}_{t-2}]}\}, \tag{7.20}$$

where $i[.]$ is an indication function.

The proposed particle filtering algorithm for simultaneous tracking and recognition is summarized in Figure 7.9, where $w_{t,l}^{(j)}$ is the weight of the particle $(n_t = l, \theta_t = \theta_t^{(j)})$ for the posterior density $p(n_t, \theta_t | \mathsf{y}_{1:t})$; $w_t^{(j)}$ is the weight of the particle $\theta_t = \theta_t^{(j)}$ for the posterior density $p(\theta_t | \mathsf{y}_{1:t})$; and $\beta_{t,l}$ is the weight of the particle $n_t = l$ for the posterior density $p(n_t | \mathsf{y}_{1:t})$. Occlusion analysis can also be included

in Figure 7.9.

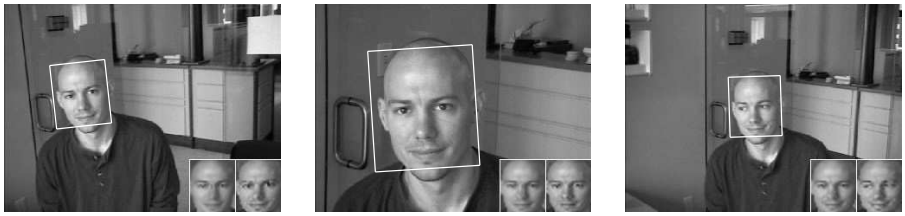
Figure 7.10: Row 1-3: the gallery set with 29 subjects in frontal view. Rows 4, 5, and 6: the top 10 eigenvectors for FFS, IPS, and EPS, respectively.

Experimental results on visual tracking and recognition

We have applied our algorithm for tracking and recognition of human faces captured by a hand-held video camera in office environments. There are 29 subjects in the database. Figure 7.10 lists all the images in the galley set and the top 10 eigenvectors for FFS, IPS, and EPS, respectively. Figure 7.11 presents some frames (with tracking results) in the video sequence for 'Subject-2' featuring quite large pose variations, moderate illumination variations, and quick scale changes (back and forth toward the end of the sequence).

Tracking is successful for all video sequences and 100% recognition rate is achieved, while early approaches fail to track in several video sequences due to its inability to handle significant appearance changes caused by pose and illumination variations. The posterior probabilities $\mathtt{p}(n_t|\mathtt{y}_{1:t})$ with $n_t = 1, 2, ...N$ obtained

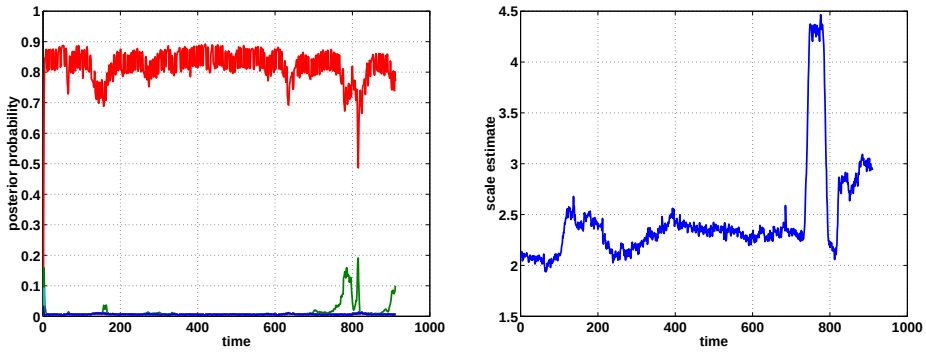
Frame 1 Frame 160 Frame 290



Frame 690 Frame 750 Frame 800

Figure 7.11: Example images in 'Subject-2' probe video sequence and the tracking results.

for the 'Subject-2' sequence are plotted in Figure 7.12(a). We start from a uniform prior for the identity variable, i.e., $\mathtt{p}(n_0) = N^{-1}$ for $n_0 = 1, 2, ...N$. It is very fast, taking about less than 10 frames, to reach above 0.9 level for the posterior probability corresponding to 'Subject-2', while all other posterior probabilities corresponding to other identities approach zero. This is mainly attributed to the discriminative power of the MAP recognition score induced by IPS and EPS modeling. The previous approach [185] usually takes about 30 frames to reach 0.9 level since only intra-personal modeling is adopted. Figure 7.12(b) captures the scale change in the 'Subject-2' sequence.



(a) (b)

Figure 7.12: Results on the 'Subject-2' sequence. (a) Posterior probabilities against time t for all identities $\mathtt{p}(n_t|\mathtt{y}_{1:t})$, $n_t = 1, 2, ..., N$. The line close to 1 is for the true identity. (b) Scale estimate against time t.

7.4 Appendix

Appendix 7.I: Derivation of the lower bound for the posterior probability of identity

Suppose that the following two assumptions hold:

- (A) The prior probability for each identity is same,

$$\mathtt{p}(n_0 = j|\mathtt{y}_0) = 1/N; \quad j \in \mathcal{N}, \tag{7.21}$$

- (B) for the correct identity $l \in \mathcal{N}$, there exists a constant $\eta > 1$ such that,

$$\mathtt{p}(\mathtt{y}_t|n_t = l, \theta_t) \geq \eta \mathtt{p}(\mathtt{y}_t|n_t = j, \theta_t); \quad t \geq 1, j \in \mathcal{N}, j \neq l. \tag{7.22}$$

Substitution of Eq. (7.21) and (7.22) into Eq. (7.6) gives rise to

$$\begin{aligned} \mathtt{p}(n_t = l|\mathtt{y}_{0:t}) &= \frac{1}{N} \int_{\theta_0} \cdots \int_{\theta_t} \mathtt{p}(\theta_0|\mathtt{y}_0) \prod_{s=1}^{t} \frac{\mathtt{p}(\mathtt{y}_s|n_s = l, \theta_s)\mathtt{p}(\theta_s|\theta_{s-1})}{\mathtt{p}(\mathtt{y}_s|\mathtt{y}_{0:s-1})} \mathtt{d}\theta_t \ldots \mathtt{d}\theta_0 \\ &\geq \frac{1}{N} \int_{\theta_0} \cdots \int_{\theta_t} \mathtt{p}(\theta_0|\mathtt{y}_0) \prod_{s=1}^{t} \frac{\eta \mathtt{p}(\mathtt{y}_s|n_s = j, \theta_s)\mathtt{p}(\theta_s|\theta_{s-1})}{\mathtt{p}(\mathtt{y}_s|\mathtt{y}_{0:s-1})} \mathtt{d}\theta_t \ldots \mathtt{d}\theta_0 \end{aligned}$$

$$
\begin{aligned}
&= \frac{\eta^t}{N} \int_{\theta_0} \dots \int_{\theta_t} \mathtt{p}(\theta_0|z_0) \prod_{s=1}^{t} \frac{\mathtt{p}(\mathtt{y}_s|n_s = j, \theta_s)\mathtt{p}(\theta_s|\theta_{s-1})}{\mathtt{p}(\mathtt{y}_s|\mathtt{y}_{0:s-1})} \mathtt{d}\theta_t \dots \mathtt{d}\theta_0 \\
&= \eta^t \mathtt{p}(n_t = j|\mathtt{y}_{0:t}); \quad j \in \mathcal{N}, j \neq l, \qquad (7.23)
\end{aligned}
$$

where $\eta^t = \prod_{s=1}^{t} \eta$.

More interestingly, from Eq. (7.23), we have

$$
(N-1)\mathtt{p}(n_t = l|\mathtt{y}_{0:t}) \geq \eta^t \sum_{j=1, j\neq l}^{N} \mathtt{p}(n_t = j|\mathtt{y}_{0:t}) = \eta^t(1 - \mathtt{p}(n_t = l|\mathtt{y}_{0:t})), \qquad (7.24)
$$

i.e.,

$$
\mathtt{p}(n_t = l|\mathtt{y}_{0:t}) \geq h(\eta, t), \qquad (7.25)
$$

where

$$
h(\eta, t) = \frac{\eta^t}{\eta^t + N - 1}. \qquad (7.26)
$$

Eq. (7.25) has two implications.

1. Since the function $h(\eta, t)$ which provides a lower bound for $\mathtt{p}(n_t = l|\mathtt{y}_{0:t})$ is monotonically increasing against time t, $\mathtt{p}(n_t = l|\mathtt{y}_{0:t})$ has a probable trend of increase over t, even though not in a monotonic manner.

2. Since $\eta > 1$ and $\mathtt{p}(n_t = l|\mathtt{y}_{0:t}) \leq 1$,

$$
\lim_{t\to\infty} \mathtt{p}(n_t = l|\mathtt{y}_{0:t}) = 1, \qquad (7.27)
$$

implying that $\mathtt{p}(n_t = l|\mathtt{y}_{0:t})$ degenerates in the identity l for some sufficiently large t.

However, all these derivations are based on assumptions (A) and (B). Though it is easy to satisfy (A), difficulty arises in practice in order to satisfy (B) for all the frames in the sequence. Fortunately, as we have seen in the experiment in Section 7.3, numerically this degeneracy is still reached even if (B) is satisfied only for most but not all frames in the sequence.

Appendix 7.II: More on assumption (B)

A trivial choice for η is the lower bound on the likelihood ratio, i.e.,

$$\eta = \inf_{t\geq 1, j\neq l, \theta_t \in \Theta} \frac{\mathtt{p}(\mathtt{y}_t | n_t = l, \theta_t)}{\mathtt{p}(\mathtt{y}_t | n_t = j, \theta_t)}. \tag{7.28}$$

This choice is of theoretical interest. In practice, how good is the assumption (B) satisfied? Figure 7.13 plots against the logarithm of the scale parameter, the 'average' likelihood of the correct identity,

$$\frac{1}{N} \sum_{n \in \mathcal{N}} \mathtt{p}(\mathtt{I}_n | n, \theta),$$

and that of the incorrect identities,

$$\frac{1}{N(N-1)} \sum_{m \in \mathcal{N}, n \in \mathcal{N}, m \neq n} \mathtt{p}(\mathtt{I}_m | n, \theta),$$

of the face gallery as well as the 'average' likelihood ratio, i.e., the ratio between the above two quantities. The observation is that only within a narrow 'band' the condition (B) is well satisfied. Therefore, the success of SIS algorithm depends on how good the samples lie in a similar 'band' in the high-dimensional affine space. Also, the lower bound η in assumption (B) is too strict. If we take the mean of the 'average' likelihood ratio shown in Figure 7.13 as an estimate of η (roughly 1.5), Eq. (7.25) tells that, after 20 frames, the probability $\mathtt{p}(l|\mathtt{y}_{0:t})$ reaches 0.99! However, this is not reached in the experiments due to noise in the observations and incomplete parameterization of transformations.

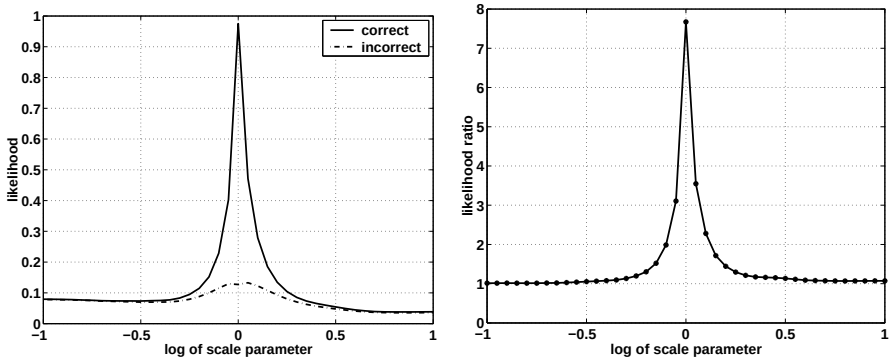


Figure 7.13: Left: The 'average' likelihood of the correct hypothesis and incorrect hypotheses against the log of scale parameter. Right: The 'average' likelihood ratio against the log of scale parameter.

Chapter 8

Probabilistic Identity Characterization

Visual face recognition is an important task. Even though a lot of research has been carried out, state-of-the-art recognizers still yield unsatisfactory results especially when confronted with pose and illumination variations. In addition, the recognizers are further complicated by the registration requirement as the images that the recognizers process contain transformed appearances of the object. Below, we simply use the term 'transformation' to model all the variations involved, be it registration, or pose and illumination variations.

While most recognizers process a single image, there is a growing interest in using a group of images [80, 84, 88, 89, 91, 184, 185]. In terms of the transformations embedded in the group or the temporal continuity between the transformations, the group can be either independent or not. Examples of the independent group (I-group) are face databases that store multiple appearances for one object. Examples of the dependent group are video sequences. If the temporal information is stripped, video sequences reduce to I-groups. In this chapter, whenever we mention

video sequences, we mean dependent groups of images.

Approaches that use the I-groups can be roughly divided into two categories. The first category is based on manifold matching. In [88], hypothetical identity surfaces are constructed by computing the linear coefficients of view space. Illumination variations are not accounted for. Discriminant features are then extracted to overcome other variations. In [80], manifolds are formed for every I-group. Recognition is performed by computing the shortest distance between two manifolds. The manifold takes a certain parameterized form and the parameters are directly learned from the visual appearances. Robustness to pose and illumination variations are not reported. The second category is based on statistical learning. In [91], a multi-variate Gaussian density is fitted for every I-group. Recognition is achieved by computing the Kullback-Leibler distance [4] between two Gaussian densities. However, the Gaussian assumption is easily violated if pose and illumination variations exist. In [184], principal subspaces are learned for each I-group and principal angle between the two principal subspace are used for recognition. The computation of principal angle is also carried on the feature space embedded by kernel functions. One common disadvantage of the above approaches is that they also assume that the face regions have already been cropped beforehand, using either a detector or a tracker.

Approaches using video sequences utilize temporal information for recognition as well. In [185], simultaneous tracking and recognition is implemented in a probabilistic framework. The joint posterior probability of the tracking parameter and the identity variable is approximated using the SIS algorithm and the marginal posterior probability of the identity variable is used for recognition. However, only an affine localization parameter is used for tracking and pose and illumination

variations are not considered. In addition, exemplars are learned from the gallery videos to cover pose and illumination variations. In [89], hidden Markov models are used to learn the dynamics before successive appearances. In [84], pose variations are handled by learning the view-discretized appearance manifolds from the training ensemble. Transition probabilities from one view to another view are used to regularize the search space. However, in [84, 89], the cropped images are used for testing.

In this chapter, we propose a general framework which possesses the following features:

- It processes either a single image or a group of images (including the I-group and video sequence).
- It handles the localization problem, illumination and pose variations.
- The identity description could be either discrete or continuous. The continuous identity encoding typically arises from subspace modeling.
- It is probabilistic and integrates all the available evidence.

Chapter organization

In Section 8.1 we introduce the generic framework which provides a probabilistic characterization of the object identity. In Section 8.2 we address issues and challenges arising in this framework. In Section 8.3 we focus on how to achieve an identity encoding which is invariant to localization, illumination and pose variations. In Section 8.3.2, we present some efficient computational methods. In Section 8.3.3, we present experimental results.

8.1 Principle of Probabilistic Identity Characterization

Suppose α is the identity signature, which represents the identity in an abstract manner. It can be either discrete- or continuous- valued. If we have an N-class problem, α is discrete taking value in $\{1, 2, ..., N\}$. If we associate the identity with image intensity or feature vectors derived from say subspace projections, α is continuous-valued. Given a group of images $\mathtt{y}_{1:T} \doteq \{\mathtt{y}_1, \mathtt{y}_2, ..., \mathtt{y}_T\}$ containing the appearances of the same but unknown identity, *probabilistic identity characterization is equivalent to finding the posterior probability* $\mathtt{p}(\alpha|\mathtt{y}_{1:T})$.

As the image only contains a transformed version of the object, we also need to associate it a transformation parameter θ, which lies in a transformation space Θ. The transformation space Θ is usually application dependent. Affine transformation is often used to compensate for the localization problem. To handle illumination variation, the lighting direction is used. If pose variation is involved, 3D transformation is needed or a discrete set is used if we quantize the continuous view space.

We assume that the prior probability of α is $\pi(\alpha)$, which is assumed to be, in practice, a *non-informative* prior. A non-informative prior is uniform in the discrete case and treated as a constant, say 1, in the continuous case.

The key to our probabilistic identity characterization is as follows:

$$\begin{aligned} & \mathtt{p}(\alpha|\mathtt{y}_{1:T}) \propto \pi(\alpha)\mathtt{p}(\mathtt{y}_{1:T}|\alpha) \\ = \; & \pi(\alpha) \int_{\theta_{1:T}} \mathtt{p}(\mathtt{y}_{1:T}|\theta_{1:T}, \alpha)\mathtt{p}(\theta_{1:T})\mathtt{d}\theta_{1:T} \\ = \; & \pi(\alpha) \int_{\theta_{1:T}} \prod_{t=1}^{T} \mathtt{p}(\mathtt{y}_t|\theta_t, \alpha)\mathtt{p}(\theta_t|\theta_{1:t-1})\mathtt{d}\theta_{1:T}, \end{aligned} \tag{8.1}$$

where the following rules, namely (a) *observational conditional independence* and (b) *chain rule*, are applied:

$$(a)\ \mathtt{p}(\mathtt{y}_{1:T}|\theta_{1:T},\alpha) = \prod_{t=1}^{T} \mathtt{p}(\mathtt{y}_t|\theta_t,\alpha); \tag{8.2}$$

$$(b)\ \mathtt{p}(\theta_{1:T}) = \prod_{t=1}^{T} \mathtt{p}(\theta_t|\theta_{1:t-1});\ \mathtt{p}(\theta_1|\theta_0) \doteq \mathtt{p}(\theta_1). \tag{8.3}$$

Equation (8.1) involves two key quantities: the *observation likelihood* $\mathtt{p}(\mathtt{y}_t|\theta_t,\alpha)$ and the *state transition probability* $\mathtt{p}(\theta_t|\theta_{1:t-1})$. The former is essential to a recognition task, the ideal case being that it possesses a discriminative power in the sense that it always favors the correct identity and disfavors the others; the latter is also very helpful especially when processing video sequences, which constrains the search space.

We now study two special cases of $\mathtt{p}(\theta_t|\theta_{1:t-1})$.

8.1.1 Independent group (I-group)

In this case, the transformations $\{\theta_t;\ t = 1,\ldots,T\}$ are independent of each other, i.e.

$$\mathtt{p}(\theta_t|\theta_{1:t-1}) = \mathtt{p}(\theta_t). \tag{8.4}$$

Eq. (8.1) becomes

$$\mathtt{p}(\alpha|\mathtt{y}_{1:T}) \propto \pi(\alpha) \prod_{t=1}^{T} \int_{\theta_t} \mathtt{p}(\mathtt{y}_t|\theta_t,\alpha)\mathtt{p}(\theta_t)\mathtt{d}\theta_t. \tag{8.5}$$

In this context, the probability $\mathtt{p}(\theta_t)$ can be regarded as a prior for θ_t, which is often assumed to be Gaussian with mean $\hat{\theta}$ or non-informative.

The most widely studied case in the literature is $T = 1$, i.e. there is only a single image in the group. Due to its importance, sometimes we will distinguish

it from the I-group (with $T > 1$) depending on the context. We will present in Section 8.2 the shortcomings of many contemporary approaches.

It all boils down to how to compute the integral in (8.5) in real applications. In the sequel, we show how to efficiently approximate it.

8.1.2 Video sequence

In the case of video sequence, temporal continuity between successive video frames implies that the transformations $\{\theta_t;\ t = 1, \ldots, T\}$ follow a Markov chain. Without loss of generality, we assume a first-order Markov chain, i.e.

$$\mathtt{p}(\theta_t|\theta_{1:t-1}) = \mathtt{p}(\theta_t|\theta_{t-1}). \tag{8.6}$$

Eq. (8.1) becomes

$$\mathtt{p}(\alpha|\mathtt{y}_{1:T}) \propto \pi(\alpha) \int_{\theta_{1:T}} \prod_{t=1}^{T} \mathtt{p}(\mathtt{y}_t|\theta_t, \alpha)\mathtt{p}(\theta_t|\theta_{t-1})\mathtt{d}\theta_{1:T}. \tag{8.7}$$

The difference between (8.5) and (8.7) is whether the product lies inside or outside the integral. In (8.5), the product lies outside the integral, which divides the quantity of interest into 'small' integrals that can be computed efficiently; while (8.7) does not have such a decomposition, causing computational difficulty.

8.1.3 Difference from Bayesian estimation

Our framework is very different from the traditional Bayesian parameter estimation setting, where a certain parameter β should be estimated from the i.i.d. observations $\{\mathtt{x}_1, \mathtt{x}_2, ..., \mathtt{x}_T\}$ generated from a parametric density $\mathtt{p}(\mathtt{x}|\beta)$. If we assume that β has a prior probability $\pi(\beta)$, then the posterior probability $\mathtt{p}(\beta|\mathtt{x}_{1:T})$ is computed as

$$\mathtt{p}(\beta|\mathtt{x}_{1:T}) \propto \pi(\beta)\mathtt{p}(\mathtt{x}_{1:T}|\beta) = \pi(\beta) \prod_{t=1}^{T} \mathtt{p}(\mathtt{x}_t|\beta) \tag{8.8}$$

and used to derive the parameter estimate $\hat{\beta}$. One should not confuse our transformation parameter θ with the parameter β. Notice that β is fixed in $\mathtt{p}(\mathtt{x}_t|\beta)$ for different t's. However, each $\mathtt{y}_t$ is associates with a θ_t. Also, α is different from β in the sense that α describes the identity and β helps to describe the parametric density.

To make our framework more general, we can also incorporate the β parameter by letting the observation likelihood be $\mathtt{p}(\mathtt{y}|\theta, \alpha, \beta)$. Equation (8.1) then becomes

$$\begin{aligned} & \mathtt{p}(\alpha|\mathtt{y}_{1:T}) \propto \pi(\alpha)\mathtt{p}(\mathtt{y}_{1:T}|\alpha) \qquad (8.9) \\ = \; & \pi(\alpha) \int_{\beta,\theta_{1:T}} \mathtt{p}(\mathtt{y}_{1:T}|\theta_{1:T}, \alpha, \beta)\mathtt{p}(\theta_{1:T})\pi(\beta)\mathtt{d}\theta_{1:T}\mathtt{d}\beta \\ = \; & \pi(\alpha) \int \prod_{t=1}^{T} \mathtt{p}(\mathtt{y}_t|\theta_t, \alpha, \beta)\mathtt{p}(\theta_t|\theta_{1:t-1})\pi(\beta)\mathtt{d}\theta_{1:T}\mathtt{d}\beta, \end{aligned}$$

where $\theta_{1:T}$ and β are assumed to be statistically independent. In this chapter, we will focus only on (8.1) as if we already know the true parameter β in (8.9). This greatly simplifies our computation.

8.2 Recognition Setting and Issues

Equation (8.1) lays a theoretical foundation, which is universal for all recognition settings: (i) recognition is based on a single image (an I-group with $T = 1$), an I-group with $T \geq 2$, or a video sequence; (ii) the identity signature is either discrete- or continuous-valued; and (iii) the transformation space takes into account all available variations, such as localization and variations in illumination and pose.

8.2.1 Discrete identity signature

In a typical pattern recognition scenario, say an N-class problem, the identity signature for $\mathtt{y}_{1:T}$, $\hat{\alpha}$, is determined by the Bayesian decision rule:

$$\hat{\alpha} = \arg \max_{\{1,2,...,N\}} \mathtt{p}(\alpha|\mathtt{y}_{1:T}). \tag{8.10}$$

Usually $\mathtt{p}(\mathtt{y}|\theta, \alpha)$ is a class-dependent density, either pre-specified or learned. This is a well studied problem and we will not focus on this.

8.2.2 Continuous identity signature

If the identity signature is continuous-valued, two recognition schemes are possible. The first is to derive a point estimate $\hat{\alpha}$ (e.g. conditional mean, mode) from $\mathtt{p}(\alpha|\mathtt{y}_{1:T})$ to represent the identity of image group $\mathtt{y}_{1:T}$. Recognition is performed by matching $\hat{\alpha}$'s belonging to different groups of images using a metric $\mathtt{k}(.,.)$. Say, $\hat{\alpha}_1$ is for group 1 and $\hat{\alpha}_2$ for group 2, the point distance

$$\hat{\mathtt{k}}_{1,2} \doteq \mathtt{k}(\hat{\alpha}_1, \hat{\alpha}_2)$$

is computed to characterize the difference between groups 1 and 2.

Instead of comparing the point estimates, the second scheme directly compares different distributions that characterize the identities for different groups of images. Therefore, for two groups 1 and 2 with the corresponding posterior probabilities $\mathtt{p}(\alpha_1)$ and $\mathtt{p}(\alpha_2)$, we use the following expected distance [134]

$$\bar{\mathtt{k}}_{1,2} \doteq \int_{\alpha_1} \int_{\alpha_2} \mathtt{k}(\alpha_1, \alpha_2)\mathtt{p}(\alpha_1)\mathtt{p}(\alpha_2)\mathtt{d}\alpha_1\mathtt{d}\alpha_2.$$

Ideally, we wish to compare the two probability distributions using quantities such as the Kullback-Leibler distance [4]. However, computing such quantities is numerically prohibitive when α is of high dimensionality.

The second scheme is preferred as it utilizes the complete statistical information, while in the first one, point estimates use partial information. For examples, if only the conditional mean is used, the covariance structure or higher-order statistics is thrown away. However, there are circumstances when the first scheme makes sense: the posterior distribution $\mathtt{p}(\alpha|\mathtt{y}_{1:T})$ is highly peaked or even degenerate at $\hat{\alpha}$. This might occur when (i) the variance parameters are taken to be very small; or (ii) we let T go to ∞, i.e. keep observing the same object for a long time.

8.2.3 The effects of the transformation

Even though recognition based on single images has been studied for a long time, most efforts assume only one alignment parameter $\hat{\theta}$ and compute the probability $\mathtt{p}(\mathtt{y}|\hat{\theta},\alpha)$. Any recognition algorithm computing some distance measures can be thought of as using a properly defined Gibbs distribution. The underlying assumption is that

$$\mathtt{p}(\theta) = \delta(\theta - \hat{\theta}), \tag{8.11}$$

where $\delta(.)$ is an impulse function. Using (8.11), (8.5) becomes

$$\mathtt{p}(\alpha|\mathtt{y}) \propto \pi(\alpha) \int_{\theta} \mathtt{p}(\mathtt{y}|\theta,\alpha)\delta(\theta-\hat{\theta})\mathtt{d}\theta = \pi(\alpha)\mathtt{p}(\mathtt{y}|\hat{\theta},\alpha). \tag{8.12}$$

Incidentally, if the Laplace's method is used to approximate the integral (refer to the Appendix 8.I for details) and the maximizer $\hat{\theta}_\alpha = \arg\max_\theta \mathtt{p}(\mathtt{y}|\theta,\alpha)\mathtt{p}(\theta)$ does not depend on α, say $\hat{\theta}_\alpha = \hat{\theta}$, then

$$\begin{aligned} \mathtt{p}(\alpha|\mathtt{y}) &\propto \pi(\alpha) \int_{\theta} \mathtt{p}(\mathtt{y}|\theta,\alpha)\mathtt{p}(\theta)\mathtt{d}\theta \\ &\simeq \pi(\alpha)\mathtt{p}(\mathtt{y}|\hat{\theta},\alpha)\mathtt{p}(\hat{\theta})\sqrt{(2\pi)^r/|\mathtt{I}(\hat{\theta})|}. \end{aligned} \tag{8.13}$$

This gives rise to the same decision rule as implied by (8.12) and also partly explains why the simple assumption (8.11) can work in practice.

The alignment parameter is therefore very crucial for a good recognition performance. Even a slightly erroneous $\hat{\theta}$ may affect the recognition system significantly. It is very beneficial to have a continuous density $\mathtt{p}(\theta)$ such as a Gaussian or even a non-informative since marginalization of $\mathtt{p}(\theta, \alpha|\mathtt{y})$ over θ yields a robust estimate of $\mathtt{p}(\alpha|\mathtt{y})$.

In addition, our Bayesian framework also provides a way to estimate the best alignment parameter through the posterior probability:

$$\mathtt{p}(\theta|\mathtt{y}) \propto \int_{\alpha} \mathtt{p}(\mathtt{y}|\theta, \alpha)\pi(\alpha)\mathtt{d}\alpha. \tag{8.14}$$

8.2.4 Asymptotic behaviors

When we have an I-group or a video sequence, we are often interested in discovering the asymptotic (or large-sample) behaviors of the posterior distribution $\mathtt{p}(\alpha|\mathtt{y}_{1:T})$ when T is large. In [185], the discrete case of α in a video sequence is studied. However it is very challenging to extend this study to a continuous case. Experimentally (refer to Section 8.3.3), we find that $\mathtt{p}(\alpha|\mathtt{y}_{1:T})$ becomes more and more peaked as N increase, which seems to suggest a degenerancy in the true value α_{true}.

8.3 Subspace Identity Encoding

The main challenge is to specify the likelihood $\mathtt{p}(\mathtt{y}|\theta, \alpha)$. Practical considerations require that (i) the identity encoding coefficient α is compact so that our target space where α resides is of low dimensional; and (ii) α should be invariant to transformations and tightly clustered so that we can safely focus on a small portion of the spaces.

Inspired by the popularity of subspace analysis, we assume that the observation y can be well explained by a subspace, whose basis vectors are encoded in a matrix denoted by B, i.e. there exists linear coefficients α such that $\mathsf{y} \approx \mathsf{B}\alpha$. Clearly, α naturally encodes the identity. However, the observation under the transformation condition (parameterized by θ) deviates from the canonical condition (parameterized by say $\bar{\theta}$) under which the B matrix is defined. To achieve an identity encoding that is invariant to the transformation, there are two possible ways. One way is to inverse-warp the observation y from the transformation condition θ to the canonical condition $\bar{\theta}$ and the other way is to warp the basis matrix B from the canonical condition $\bar{\theta}$ to the transformation condition θ. In practice, inverse-warping is typically difficult. For example, we cannot easily warp an off-frontal view to a frontal view without explicit 3D depth information that is unavailable. Hence, we follow the second approach, which is also known as *analysis-by-synthesis* approach. We denote the basis matrix under the transformation condition θ by B_θ.

8.3.1 Invariant to localization, illumination, and pose

Localization parameter, denoted by ε, includes the face location, scale and in-plane rotation. Typically, an affine transformation is used. We absorb the localization parameter ε in the observation using $\mathcal{T}\{\mathsf{y}; \varepsilon\}$, where the $\mathcal{T}\{.; \varepsilon\}$ is a localization operator, extracting the region of interest and normalizing it to match with the size of the basis.

The illumination parameter, denoted by λ, is a vector specifying the illuminant direction (and intensity if required). The pose parameter, denoted by υ, is a continuous-valued random variable. However, practical systems [67, 69] often discretize this due to the difficulty in handling 3D to 2D projection. Suppose the

quantized pose set is $\{1, \ldots, V\}$. To achieve pose invariance, we concatenate all the images [69] $\{\mathsf{y}^1, \ldots, \mathsf{y}^V\}$ under all the views and a fixed illumination λ to form a high-dimensional vector $\mathsf{Y}^\lambda = [\mathsf{y}^{1,\lambda}, ..., \mathsf{y}^{V,\lambda}]^\mathsf{T}$. To further achieve invariance to illuminations, we invoke the Lambertian reflectance model, ignoring shadow pixels. Now, λ is actually a 3-D vector describing the illuminant. We now follow Chapter 3 to derive a bilinear analysis summarized below.

Since all y^υ's are illuminated by the same λ, the Lambertian model gives,

$$\mathsf{Y}^\lambda = \mathsf{W}\lambda. \tag{8.15}$$

Following [204], we assume that

$$\mathsf{W} = \sum_{i=1}^{m} \alpha_i \mathsf{W}_i, \tag{8.16}$$

and we have

$$\mathsf{Y}^\lambda = \sum_{i=1}^{m} \alpha_i \mathsf{W}_i \lambda, \tag{8.17}$$

where W_i's are illumination-invariant bilinear basis and $\alpha = [\alpha_1, \ldots, \alpha_m]^\mathsf{T}$ provides an illuminant-invariant identity signature. Those bilinear basis can be easily learned as shown in [138, 202]. Thus α is also pose-invariant because, for a given view υ, we take the part in Y corresponding to this view and still have

$$\mathsf{y}^{\lambda,\upsilon} = \sum_{i=1}^{m} \alpha_i \mathsf{W}_i^\upsilon \lambda. \tag{8.18}$$

In summary, the basis matrix B_θ for $\theta = (\varepsilon, \lambda, \upsilon)$ with ε absorbed in y is expressed as $\mathsf{B}_{\lambda,\upsilon} = [\mathsf{W}_1^\upsilon \lambda, \ldots, \mathsf{W}_m^\upsilon \lambda]$.

We focus on the following likelihood:

$$\begin{aligned} \mathtt{p}(\mathsf{y}|\theta) &= \mathtt{p}(\mathsf{y}|\varepsilon, \lambda, \upsilon, \alpha) \\ &= \mathtt{Z}_{\lambda,\upsilon,\alpha}^{-1} \exp\{-\mathtt{D}(\mathcal{T}\{\mathsf{y}; \varepsilon\}, \mathsf{B}_{\lambda,\upsilon}\alpha)\}, \end{aligned} \tag{8.19}$$

where $\mathtt{D}(\mathsf{y}, \mathsf{B}_\theta\alpha)$ is some distance measure and $\mathtt{Z}_{\lambda,v,\alpha}$ is the so-called partition function which plays a normalization role. In particular, if we take $\mathtt{D}$ as

$$\mathtt{D}(\mathcal{T}\{\mathsf{y};\varepsilon\}, \mathsf{B}_{\lambda,v}\alpha) = (\mathcal{T}\{\mathsf{y};\varepsilon\} - \mathsf{B}_{\lambda,v}\alpha)^{\mathsf{T}}\Sigma^{-1}(\mathcal{T}\{\mathsf{y};\varepsilon\} - \mathsf{B}_{\lambda,v}\alpha)/2, \tag{8.20}$$

with a given Σ (say $\Sigma = \sigma^2\mathtt{I}$ where $\mathtt{I}$ is an identity matrix), then (8.19) becomes a multivariate Gaussian and the partition function $\mathtt{Z}_{\lambda,v,\alpha}$ does not depend on the parameters any more. However, even though (8.19) is a multivariate Gaussian, the posterior distribution $\mathtt{p}(\alpha|\mathsf{y}_{1:T})$ is no longer Gaussian.

8.3.2 Computational issues

The integral

If the transformation space Θ is discrete, it is easy to evaluate the integral[1] $\int_\theta \mathtt{p}(\mathsf{y}|\theta,\alpha)\mathtt{p}(\theta)\mathtt{d}\theta$, which becomes a sum. If Θ is continuous, in general, computing integral $\int_\theta \mathtt{p}(\mathsf{y}|\theta,\alpha)\mathtt{p}(\theta)\mathtt{d}\theta$ is a difficult task. Many techniques are available in the literature. Here we mainly focus on two techniques: Monte Carlo simulation [14, 16] and Laplace's method [16, 136].

Monte Carlo simulation. The underlying principle is the law of large number (LLN). If $\{\mathsf{x}^{(1)}, \mathsf{x}^{(2)}, \ldots, \mathsf{x}^{(K)}\}$ are K i.i.d. samples of the density $\mathtt{p}(\mathsf{x})$, for any bounded function $\mathtt{h}(\mathsf{x})$,

$$\lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathtt{h}(\mathsf{x}^{(k)}) = \int_{\mathsf{x}} \mathtt{h}(\mathsf{x})\mathtt{p}(\mathsf{x})\mathtt{d}\mathsf{x} = \mathtt{E}_{\mathtt{p}}[\mathtt{h}]. \tag{8.21}$$

Alternatively, when drawing i.i.d. samples from $\mathtt{p}(\mathsf{x})$ is difficult, we can use importance sampling [14, 16]. Suppose that the *importance function* $\mathtt{q}(\mathsf{x})$ has i.i.d. realizations $\{\mathsf{x}^{(1)}, \mathsf{x}^{(2)}, \ldots, \mathsf{x}^{(K)}\}$. The pdf $\mathtt{p}(\mathsf{x})$ can be represented by a weighted

[1] We drop the subscript $[.]_t$ notation as this is a general treatment.

sample set $\{(\mathtt{x}^{(k)}, w_{\mathtt{p}}^{(k)})\}_{k=1}^{K}$, where the weight for the sample $\mathtt{x}^{(k)}$ is

$$w_{\mathtt{p}}^{(k)} = \mathtt{p}(\mathtt{x}^{(k)})/\mathtt{q}(\mathtt{x}^{(k)}), \tag{8.22}$$

in the sense that for any bounded function $\mathtt{h}(\mathtt{x})$,

$$\lim_{K\to\infty} \sum_{k=1}^{K} w_{\mathtt{p}}^{(k)} \mathtt{h}(\mathtt{x}^{(k)}) = \sum_{k=1}^{K} \frac{\mathtt{p}(\mathtt{x}^{(k)})}{\mathtt{q}(\mathtt{x}^{(k)})} \mathtt{h}(\mathtt{x}^{(k)}) = \mathtt{E}_{\mathtt{p}}[\mathtt{h}]. \tag{8.23}$$

Laplace's method [16, 136]. The general approach of this method is presented in Appendix 8.I. This is a good approximation to the integral only if the integrand is uniquely peaked and reasonably mimics the Gaussian function.

In our context, we use importance sampling (or i.i.d sampling if possible) for ε and the Laplace's method for λ and enumerate υ. We draw i.i.d. samples $\{\varepsilon^{(1)}, \varepsilon^{(2)}, \ldots, \varepsilon^{(K)}\}$ from $\mathtt{q}(\varepsilon)$ and, for each sample $\varepsilon^{(k)}$, compute the weight $w_{\varepsilon^{(k)}} = \mathtt{p}(\varepsilon^{(k)})/\mathtt{q}(\varepsilon^{(k)})$. If the i.i.d. sampling is used, the weights are always ones. Putting things together, we have (assuming $\pi(\alpha)$ is a non-informative prior)

$$\begin{aligned} \mathtt{p}(\alpha|\mathtt{y}) &\propto \int_{\varepsilon,\lambda,\upsilon} \mathtt{p}(\mathtt{y}|\varepsilon,\lambda,\upsilon,\alpha)\mathtt{p}(\varepsilon)\mathtt{p}(\lambda)\mathtt{p}(\upsilon)\mathtt{d}\varepsilon\mathtt{d}\lambda\mathtt{d}\upsilon \\ &\simeq \frac{1}{K}\sum_{k=1}^{K} w_{\varepsilon^{(k)}} \frac{1}{V}\sum_{\upsilon=1}^{V} \mathtt{p}(\mathtt{y}|\varepsilon^{(k)}, \hat{\lambda}_{\varepsilon^{(k)},\upsilon,\alpha}, \upsilon, \alpha) \times \\ &\qquad \mathtt{p}(\hat{\lambda}_{\varepsilon^{(k)},\upsilon,\alpha})\sqrt{(2\pi)^r/|\mathtt{I}(\hat{\lambda}_{\varepsilon^{(k)},\upsilon,\alpha})|}, \end{aligned} \tag{8.24}$$

where $\hat{\lambda}_{\varepsilon^k,\upsilon,\alpha}$ is the maximizer

$$\hat{\lambda}_{\varepsilon^{(k)},\upsilon,\alpha} = \arg\min_{\lambda} \mathtt{p}(\mathtt{y}|\varepsilon^{(k)}, \lambda, \upsilon, \alpha)\mathtt{p}(\lambda), \tag{8.25}$$

r is the dimensionality of λ, and $\mathtt{I}(\hat{\lambda}_{\varepsilon,\upsilon,\alpha})$ is a properly defined matrix. Refer to Appendix 8.II for computing $\hat{\lambda}_{\varepsilon,\upsilon,\alpha}$ and $\mathtt{I}(\hat{\lambda}_{\varepsilon,\upsilon,\alpha})$ if the likelihood is given as (8.19) and (8.20) and a non-informative prior $\mathtt{p}(\lambda)$ is assumed. Similar derivations can be conducted for an I-group of observations $\mathtt{y}_{1:T}$.

The distances $\bar{\mathtt{k}}$ and $\hat{\mathtt{k}}$

To evaluate the expected distance $\bar{\mathtt{k}}$, we use the Monte Carlo method. In our context, the target distribution is $\mathtt{p}(\alpha|\mathtt{y}_{1:T})$. Based on the above derivations, we know how to evaluate the target distribution, but not to draw sample from it. Therefore, we use importance sampling. Other sampling techniques such as Monte Carlo Markov chain [14, 16] can also be applied.

Suppose that, say for group 1, the importance function is $\mathtt{q}_1(\alpha_1)$, and weighted sample set is $\{\alpha_1^{(i)}, w_1^{(i)}\}_{i=1}^I$, the expected distance is approximated as

$$\bar{\mathtt{k}}_{1,2} \simeq \frac{\sum_{i=1}^I \sum_{j=1}^J w_1^{(i)} w_2^{(j)} \mathtt{k}(\alpha_1^{(i)}, \alpha_2^{(j)})}{\sum_{i=1}^I w_1^{(i)} \sum_{j=1}^J w_2^{(j)}}. \tag{8.26}$$

The point distance is approximated as

$$\hat{\mathtt{k}}_{1,2} \simeq \mathtt{k}(\frac{\sum_{i=1}^I w_1^{(i)} \alpha_1^{(i)}}{\sum_{i=1}^I w_1^{(i)}}, \frac{\sum_{j=1}^J w_2^{(j)} \alpha_2^{(j)}}{\sum_{j=1}^J w_2^{(j)}}). \tag{8.27}$$

8.3.3 Experimental results

We use the 'illum' subset of the PIE database [75] in our experiments. This subset has 68 subjects under 21 illumination configurations and 13 poses. Out of the 21 illumination configurations, we select 12 of them denoted by F,

$$F = \{f_{16}, f_{15}, f_{13}, f_{21}, f_{12}, f_{11}, f_{08}, f_{06}, f_{10}, f_{18}, f_{04}, f_{02}\},$$

which typically span the set of variations. Out of the 13 poses, we select 9 of them denoted by C,

$$C = \{c_{22}, c_{02}, c_{37}, c_{05}, c_{27}, c_{29}, c_{11}, c_{14}, c_{34}\},$$

which cover from the left profile to the frontal to the right profile. In total, we have $68 * 12 * 9 = 7344$ images. Fig 3.2 displays one PIE object under the illumination and pose variations.

We randomly divide the 68 subjects into two parts. The first 34 subjects are used in the training set and the remaining 34 subjects are used in the gallery and probe sets. It is guaranteed that there is no identity overlap between the training set and the gallery set.

During training, the images are pre-preprocessed by aligning the eyes and mouth to desired positions. No flow computation is carried on for further alignment. After the pre-processing step, the used face image is of size 48 by 40, i.e. $d = 48 * 40 = 1920$. Also, we only study gray images by taking the average of the red, green, and blue channels of their color versions.

The training set is used to learn the basis matrix B_θ or the bilinear basis W_i's. As mentioned before, θ includes the illumination direction λ and the view pose υ, where λ is a continuous-valued random vector and υ is a discrete random variable taking values in $\{1, \ldots, V\}$ with $p = 9$ (corresponding to C).

The images belonging to the remaining 34 subjects are used in the gallery and probe sets. The construction of the gallery and probe sets conforms the following: To form a gallery set of the 34 subjects, for each subject, we use an I-group of 12 images under all the illuminations under one pose υ_p; to form a probe set, we use I-groups under the other pose υ_g. We mainly concentrate on the case with $\upsilon_p \neq \upsilon_g$. Thus, we have $9 * 8 = 72$ tests, with each test giving rise to a recognition score. The 1-NN (nearest neighbor) rule is applied to find the identity for a probe I-group.

During testing, we no longer use the pre-processed images and therefore the unknown transformation parameter includes the affine localization parameter, the light direction, and the discrete view pose. The prior distribution $\mathsf{p}(\varepsilon_t)$ is assumed to be a Gaussian, whose mean is found by a background subtraction algorithm

and whose covariance matrix is manually specified. We use i.i.d. sampling from $\mathtt{p}(\varepsilon_t)$ since it is Gaussian. The metric $\mathtt{k}(.,.)$ actually used in our experiments is the correlation coefficient:

$$\mathtt{k}(\mathtt{x},\mathtt{y}) = \{(\mathtt{x}^{\mathtt{T}}\mathtt{y})^2\}/\{(\mathtt{x}^{\mathtt{T}}\mathtt{x})(\mathtt{y}^{\mathtt{T}}\mathtt{y})\}.$$

Figure 8.1 shows the marginal posterior distribution of the first element α^1 of the identity variable α, i.e., $\mathtt{p}(\alpha^1|\mathtt{y}_{1:T})$, with different N's. From Figure 8.1, we notice that (i) the posterior probability $\mathtt{p}(\alpha^1|\mathtt{y}_{1:T})$ has two modes, which might fail those algorithms using the point estimate, and (ii) it becomes more peaked and tightly-supported as T increases, which empirically supports the asymptotic behavior mentioned in Section 8.2.

Figure 8.2 shows the recognition rates for all the 72 tests. In general, when the poses of the gallery and probe sets are far apart, the recognition rates decrease. The best gallery sets for recognition are those in frontal poses and the worst gallery sets are those in profile views. These observation are similar to those made in Chapter 3.

For comparison, Table 8.1 shows the average recognition rates for four different methods: our two probabilistic approaches using $\bar{\mathtt{k}}$ and $\hat{\mathtt{k}}$, respectively, the PCA approach [62], and the statistical approach [91] using the KL distance. When implementing the PCA approach, we learned a generic face subspace from all the training images, stripping their illumination and pose conditions; while implementing the KL approach, we fit a Gaussian density on every I-group and the learning set is not used. Our approaches outperform the other two approaches significantly due to the transformation-invariant subspace modeling. The KL approach [91] performs even worse than the PCA approach simply because no illumination and pose learning is used in the KL approach while the PCA approach has a learning

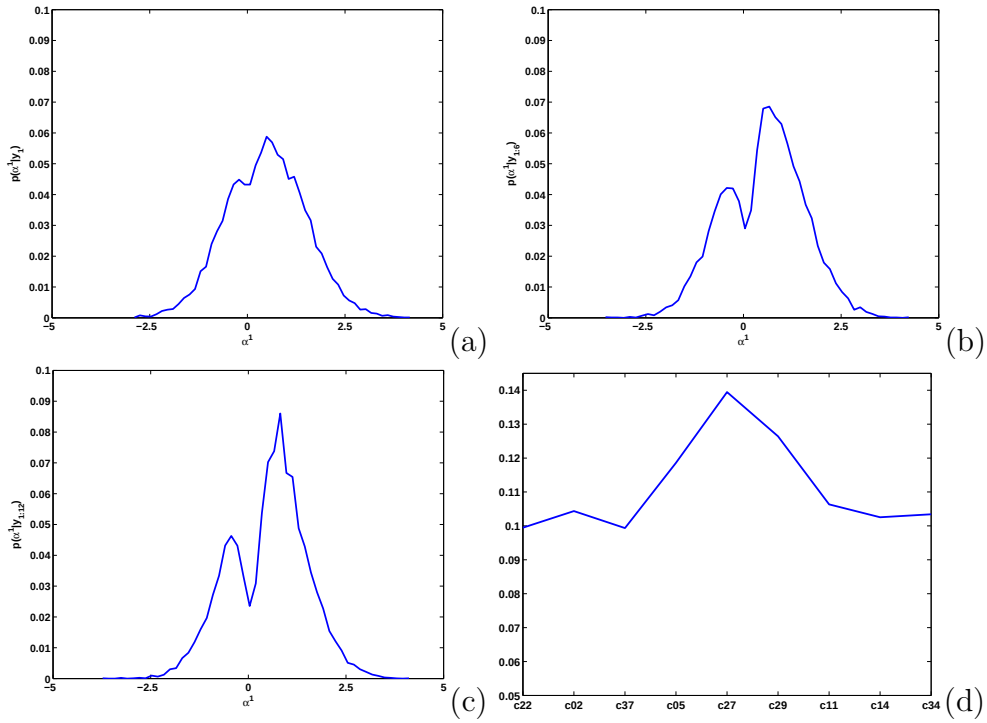


Figure 8.1: The posterior distributions $\mathtt{p}(\alpha^1|\mathtt{y}_{1:T})$ with different T's: (a) $\mathtt{p}(\alpha^1|\mathtt{y}_1)$; (b) $\mathtt{p}(\alpha^1|\mathtt{y}_{1:6})$; and (c) $\mathtt{p}(\alpha^1|\mathtt{y}_{1:12})$, and (d) the posterior distribution $\mathtt{p}(\upsilon|\mathtt{y}_{1:12})$. Notice that $\mathtt{p}(\alpha^1|\mathtt{y}_{1:T})$ has two modes and becomes more peaked as T increases.

algorithm based on image ensembles taken under different illuminations and poses (though this specific information is stripped).

Method	$\bar{\mathtt{k}}$	$\hat{\mathtt{k}}$	PCA	KL [91]
Rec. Rate (top 1)	82%	76%	36%	6%
Rec. Rate (top 3)	94%	91%	56%	15%

Table 8.1: Recognition rates of different methods.

As earlier mentioned in Section 8.2.3, we can infer the transformation parameters using the posterior probability $\mathtt{p}(\theta|\mathtt{y}_{1:T})$. Figure 8.1 also shows the obtained $\mathtt{p}(\upsilon|\mathtt{y}_{1:12})$ for one probe I-group. In this case, the actual pose is $\upsilon = 5$ (i.e. cam-

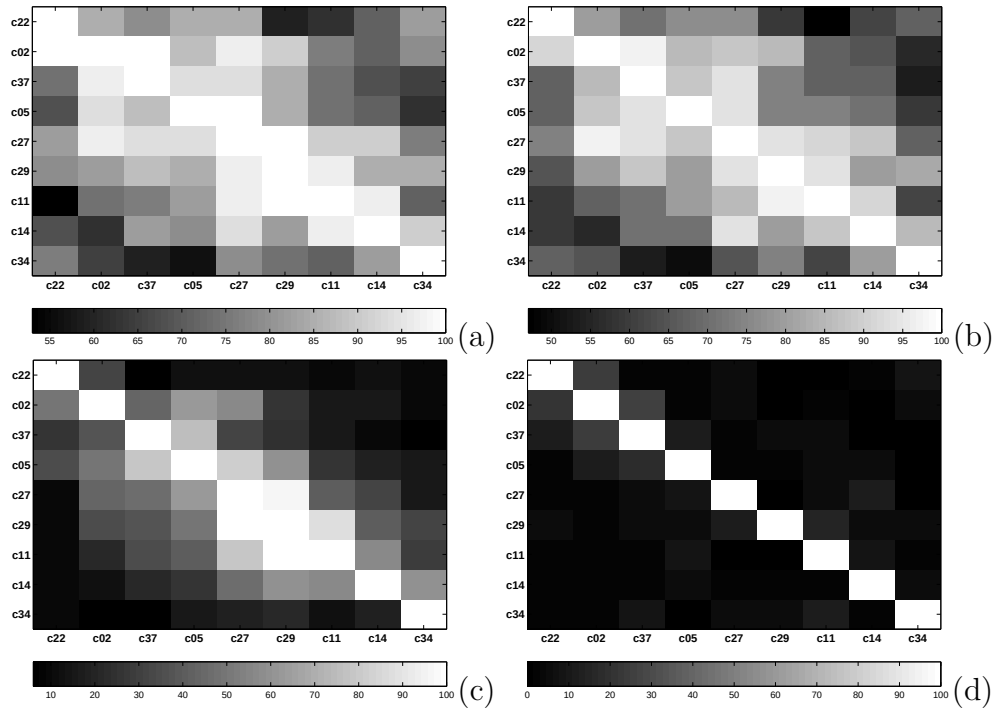


Figure 8.2: The recognition rates of all tests. (a) Our method based on $\bar{\mathtt{k}}$. (b) Our method based on $\hat{\mathtt{k}}$. (c) The PCA approach [62]. (d) The KL approach. Notice the different ranges of values for different methods and the diagonal entries should be ignored.

era c_{27}), which has the maximum probability in Figure 8.1(d). Similarly, we can find an estimation for ε, which is quite accurate as the back ground subtraction algorithm already provides a clean position.

8.4 Appendix

Appendix 8.I – Laplace's method

We are interested in computing the following quantity, for $\theta = [\theta_1, \theta_2, \ldots, \theta_r]^{\mathtt{T}} \in \mathcal{R}^r$, $\mathtt{J} = \int \mathtt{p}(\theta)\mathtt{d}\theta$. Suppose that $\hat{\theta}$ is the maximizer of $\mathtt{p}(\theta)$ or equivalently $\log \mathtt{p}(\theta)$

which satisfies

$$\frac{\partial \mathtt{p}(\theta)}{\partial \theta}|_{\hat{\theta}} = 0 \; or \; \frac{\partial \log \mathtt{p}(\theta)}{\partial \theta}|_{\hat{\theta}} = 0. \tag{8.28}$$

We expand $\log \mathtt{p}(\theta)$ around $\hat{\theta}$ using a Taylor series:

$$\log \mathtt{p}(\theta) \simeq \log \mathtt{p}(\hat{\theta}) - \frac{1}{2}(\theta - \theta^*)^{\mathtt{T}} \mathtt{I}(\hat{\theta})(\theta - \hat{\theta}), \tag{8.29}$$

where $\mathtt{I}(\theta)$ is an $r \times r$ matrix whose ij^{th} element is

$$\mathtt{I}_{ij}(\theta) = -\frac{\partial^2 \log \mathtt{p}(\theta)}{\partial \theta_i \partial \theta_j}. \tag{8.30}$$

Note that the first-order term in (8.29) is zero by virtue of (8.28). If $\mathtt{p}(\theta)$ is a pdf function with parameter θ, then $\mathtt{I}(\theta)$ is the famous Fisher information matrix [16]. Substituting (8.29) into $\mathtt{J}$ gives

$$\begin{aligned} \mathtt{J} &\simeq \mathtt{p}(\hat{\theta}) \int \exp\{-\frac{1}{2}(\theta - \hat{\theta})^{\mathtt{T}} \mathtt{I}(\hat{\theta})(\theta - \hat{\theta})\} \mathtt{d}\theta \\ &= \mathtt{p}(\hat{\theta}) \sqrt{(2\pi)^r / |\mathtt{I}(\hat{\theta})|}. \end{aligned} \tag{8.31}$$

Appendix 8.II – About $\hat{\lambda}_{\varepsilon,\upsilon,\alpha}$

If a non-information prior $\mathtt{p}(\lambda)$ is assumed[2], the maximizer $\hat{\lambda}_{\varepsilon,\upsilon,\alpha}$ satisfies

$$\begin{aligned} \hat{\lambda}_{\varepsilon,\upsilon,\alpha} &= \arg\max_{\lambda} \mathtt{p}(\mathsf{y}|\varepsilon, \lambda, \upsilon, \alpha) \\ &= \arg\min_{\lambda} (\mathcal{T}\{\mathsf{y}; \varepsilon\} - \mathsf{B}_{\lambda,\upsilon}\alpha)^{\mathtt{T}} (\mathcal{T}\{\mathsf{y}; \varepsilon\} - \mathsf{B}_{\lambda,\upsilon}\alpha) \\ &= \arg\min_{\lambda} \mathtt{L}(\varepsilon, \upsilon, \lambda, \alpha) \end{aligned} \tag{8.32}$$

where $\mathtt{L}(\varepsilon, \upsilon, \lambda, \alpha) \doteq (\mathcal{T}\{\mathsf{y}; \varepsilon\} - \mathsf{B}_{\lambda,\upsilon}\alpha)^{\mathtt{T}} (\mathcal{T}\{\mathsf{y}; \varepsilon\} - \mathsf{B}_{\lambda,\upsilon}\alpha)$.

Using the fact that

$$\mathsf{B}_{\lambda,\upsilon}\alpha = [\mathsf{W}_1^{\upsilon}\lambda, \ldots, \mathsf{W}_m^{\upsilon}\lambda]\alpha = \mathsf{B}_{\alpha,\upsilon}\lambda; \; \mathsf{B}_{\alpha,\upsilon} \doteq \sum_{i=1}^{m} \alpha_i \mathsf{W}_i^{\upsilon}, \tag{8.33}$$

[2]If a Gaussian prior is assumed, a similar derivation can be carried.

The term $\mathtt{L}(\varepsilon, \upsilon, \lambda, \alpha)$ becomes

$$\mathtt{L}(\varepsilon, \upsilon, \lambda, \alpha) = (\mathcal{T}\{\mathsf{y}; \varepsilon\} - \mathsf{B}_{\alpha,\upsilon}\lambda)^{\mathsf{T}}(\mathcal{T}\{\mathsf{y}; \varepsilon\} - \mathsf{B}_{\alpha,\upsilon}\lambda), \tag{8.34}$$

which is quadratic in λ. The optimum $\hat{\lambda}_{\varepsilon,\upsilon,\alpha}$ is unique and its value is

$$\hat{\lambda}_{\varepsilon,\upsilon,\alpha} = (\mathsf{B}_{\alpha,\upsilon}{}^{\mathsf{T}}\mathsf{B}_{\alpha,\upsilon})^{-1}\mathsf{B}_{\alpha,\upsilon}{}^{\mathsf{T}}\mathsf{y} = \mathsf{B}_{\alpha,\upsilon}{}^{\dagger}\mathcal{T}\{\mathsf{y}; \varepsilon\}. \tag{8.35}$$

where $[.]^{\dagger}$ is the pseudo-inverse. Substituting (8.35) into $\mathtt{L}(\varepsilon, \upsilon, \lambda, \alpha)$ yields

$$\mathtt{L}(\varepsilon, \upsilon, \hat{\lambda}_{\varepsilon,\upsilon,\alpha}, \alpha) = \mathcal{T}\{\mathsf{y}; \varepsilon\}^{\mathsf{T}}(\mathtt{I}_d - \mathsf{B}_{\alpha,\upsilon}\mathsf{B}_{\alpha,\upsilon}{}^{\dagger})\mathcal{T}_{\varepsilon}\{\mathsf{y}\}. \tag{8.36}$$

It is easy to show that $\mathtt{I}(\lambda)$ is no longer a function of λ and equals to

$$\mathtt{I} = \sigma^{-2}\mathsf{B}_{\alpha,\upsilon}{}^{\mathsf{T}}\mathsf{B}_{\alpha,\upsilon}. \tag{8.37}$$

Chapter 9

Conclusions

9.1 Summary

This doctoral dissertation addressed several approaches for unconstrained face recognition from three aspects. The first aspect is to directly model illumination and pose variations. The second aspect is to use nonlinear kernel learning to characterize the face appearance manifold. The third aspect is to perform recognition using video sequences.

Here are some of the key contributions made in the thesis:

- In the generalized photometric stereo approach in Chapter 2, we proposed a rank constraint on the product of albedo and surface normal that provides a very compact yet efficient encoding of the identity. In the literature, usually two separate linear subspaces [43, 66] are constructed for shape and texture, respectively, assuming the independence between them. This assumption might result in an overfit for the problem [202].

  By using the integrability and symmetry constraints, we then achieve a lin-

earized algorithm that recovers the *class-specific* albedos and surface normals under the most general and hence most difficult setting, i.e., the observation matrix consists of different objects under different illuminations. In particular, this algorithm takes into account the effect of varying albedo field in the integrability term.

- The proposed illuminating light field approach in Chapter 3 is image-based and requires no explicit 3D model. It is computationally efficient and able to deal with images of small size. In contrast, the 3D model-based approach [66] is computationally intense and needs image of large size.

- Probabilistic analysis of kernel principal components in Chapter 4 provides a tool for modeling nonlinear manifold in an interpretable manner. This also implicitly characterizes the high order statistical information. The probabilistic nature enables a mixture modeling of kernel principal component analysis and an effective classification scheme.

- Computing the probabilistic distance measures (e.g. the Chernoff distance, the Bhatacharyya distance, the KL distance, and the divergence distance) between two Gaussian densities in the RKHS is presented in Chapter 5. Since the RKHS might be infinite-dimensional, we derive a limiting distance which can be easily computed. This leads to a novel paradigm for studying pattern separability, especially for visual pattern lying in a nonlinear manifold.

- Presented in Chapter 6 is an adaptive method for visual tracking which stabilizes the tracker by embedding deterministic linear prediction into stochastic diffusion. Numerical solutions have been provided using particle filters with the adaptive observation model arising from the adaptive appearance model,

adaptive state transition model, and adaptive number of particles. Occlusion analysis is also embedded in the particle filter.

- A systematic method for face recognition from a probe video, compared with a gallery of still templates is introduced in Chapter 7. A time series state space model is used to accommodate the video and SIS algorithms provide the numerical solutions to the model. This probabilistic framework, which overcomes many difficulties arising in conventional recognition approaches using video, is registration-free and poses no need for selecting good frames. It turns out that an immediate recognition decision can be made in our framework due to the degeneracy of the posterior probability of the identity variable. The conditional entropy can also serve as a good indication for the convergence.

- We present in Chapter 8 a generic framework of modeling human identity for a single image, a group of images, or a video sequence . This framework provides a complete statistic description of the identity. Various current recognition schemes are just instances of this generic framework.

9.2 Future works

Unconstrained face recognition can be expanded in a multitude of ways. The following just lists some potential avenues to explore in the context of the proposed approaches:

- In Chapters 2 and 3, we utilize a Lambertian reflectance model to describe illumination phenomenon. However, the Lambertian reflectance model is a rather simple model and unable to handle cast shadows and specular regions.

Although we employ a simple technique to exclude pixels in cast shadow and specular regions, it turns out when the light comes from extreme directions (e.g. highly off-frontal ones), the recognition performance drops quickly. We need to investigate these lighting conditions. Alternatively, a complex illumination model providing a better illumination description can be used.

- In the illuminating light field approach of Chapter 3, we need an image-based rendering technique to handle novel poses. Some promising works along this line are [67, 110, 111].

- On probabilistic analysis of kernel principal components and probability distances on RKHS, possible future works include (i) how to design or select the kernel function for a given task, be it classification or modeling; (ii) evaluating the kernels for set based on the derived probabilistic distances (as argued in Section 5.3.5) in a classification device such as Support Vector Machine for various applications; (iii) utilizing probabilistic distances for an independent component analysis (ICA) as in [170].

- The visual tracking algorithm of Chapter 6 can be extended in many ways [206, 212]. (i) Combining shape information into appearance. Appearance and shape are two very important visual cues arguably presented in a complementary fashion [133]. (ii) Utilizing appearance from multiple views. Using multiple views can overcome some difficulty in a single view. For example, an object might be occluded in one view but not the other one. Using the multi-view geometry, we can infer the movement of the object in the occluded view [207]. (iii) Here we mostly model the movement of the foreground object. Joint modeling of foreground and background movements is very promising

[212, 213] since the stabilization obtained by background modeling significantly reduces the clutter in the background that confuses the foreground tracking algorithm.

- In simultaneous tracking and recognition of Chapter 7, various issues exist. (i) Robustness. Generally speaking, our approach is more robust than still-image-based approach since we essentially compute the recognition score based on all video frames and, in each frame, all kinds of transformed versions of the face part corresponding to the sample configurations that are considered. However, since we take no explicit measure when handling frames with outlier or other unexpected factors, recognition scores based on those frames might be low. But, this is a problem for other approaches too. The assumption that the identity does not change as time proceeds, i.e., $\mathtt{p}(n_t|n_{t-1}) = \delta(n_t - n_{t-1})$, could be relaxed by having nonzero transition probabilities between different identity variables. Using nonzero transition probabilities will enable us an easier transition to the correct choice in case that the initial choice is incorrectly chosen, making the algorithm more robust.

  (ii) Resampling. In the recognition algorithm, the marginal distribution $\{(\theta_{t-1}^{(j)}, w_{t-1}^{\prime(j)})\}_{j=1}^{J}$ is sampled to obtain the sample set $\{(\theta_t^{(j)}, 1)\}_{j=1}^{J}$. This may cause problems in principle since there is no conditional independence between θ_t and n_t given $\mathtt{y}_{0:t}$. However, in a practical sense, this is not a big disadvantage because the purpose of resampling is to 'provide chances for the good streams (samples) to amplify themselves and hence rejuvenate the sampler to produce better results for future states as the system evolves' [159]. The resampling scheme can either be simple random sampling with

weights (like in CONDENSATION), residual sampling, or local Monte Carlo methods.

- Further, in the experimental part of Chapter 8, we can extend our approach to perform recognition from video sequences with localization, illumination, and pose variations. Again, Sequential Monte Carlo methods can be used to accommodate temporal continuity. This leads to a very high-dimensional state space to explore. Efficient simulation techniques are desired. In fact, the issue of computation load also exist for the efficient algorithm in Chapter 7. There, two important numbers affecting the computation are J, the number of motion samples, and N, the size of the database. (i) The choice of J is an open question in the statistics literature. In general, larger J produces more accurate results. (ii) The choice of N depends on application. Since a small database is used in this experiment, it is not a big issue here. However, the computational burden may be excessive if N is large. One possibility is to use a continuous parameterized representation, say α as in Chapter 8, instead of discrete identity variable n. Now the task reduces to computing $\mathtt{p}(\alpha_t, \theta_t | \mathtt{y}_{0:t})$.

The approaches taken in this thesis by no means cover the whole spectrum of the unconstrained face recognition problem and address only a small portion of all available issues. Some possible important issues, other than those addressed in the thesis, include the following:

- *Aging.* Aging is a very important topic in unconstrained face recognition. Often the stored gallery images are taken well before the probe images. For example, passengers hold passports with photos taken when the passport was issued years ago. While one solution is to maintain the gallery images

up-to-date, a systematic solution is theoretical modeling of the generic affect of aging. This modeling is very difficult due to the individualized variation. Presented in [50] is just one attempt with limited success. More research efforts are certainly worthwhile.

- *Expression.* Facial expression analysis and modeling attracts a lot of attention [42, 60, 61] and some approaches [60] focus on expression recognition, i.e., identifying different modalities of facial expression such as happy, angry, disgust, etc. Face recognition under expression variation has not been fully explored. Clearly expression recognition and face recognition under expression variation are two different topics. However, expression recognition and modeling is a crucial component for accurate face recognition under expression variation.

  Further, facial expressions manifest themselves in a temporal dimension. The manner that an individual poses expressions (in natural contexts) presents certain behavioral aspect of the face biometric. Utilizing temporal information embedded in facial expression for face recognition under expression variation is an interesting research topic.

- *Distorted imagery.*

  Images as one main digital media are to be compressed, stored, transmitted and so on. Compression schemes sacrifice image quality for fewer bits to encode the image, storage devices are susceptible to various damages, transmission channels are often noisy. All these results in distorted images. How to perform face recognition accounting for sources of distortions [199] is a very practical research topic that needs to be explored.

BIBLIOGRAPHY

[**Books on general topics**]

[1] B. Anderson and J. Moore, *Optimal Filtering.* New Jersey: Prentice Hall, Engle-wood Cliffs, 1979.

[2] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.

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

[4] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* Wiley, 1991.

[5] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Prentice Hall International, 1982.

[6] A. Doucet, N. d. Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, New York, 2001.

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

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations.* The Johns Hopkins University Press, 1996.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, New York, 2001.

[10] B. Horn and M. Brooks (Eds.) *Shape from Shading.* MIT Press, 1989.

[11] P.J. Huber, *Robust statistics.* Wiley, 1981.

[12] I. T. Jolliffe, *Principal Component Analysis.* New York: Springer-Verlag, 2002.

[13] Kullback, *Information Theory and Statistics.* Wiley, New York, 1959.

[14] J.S. Liu, *Monte Carlo Strtegies in Scientific Computing.* Springer, 2001.

[15] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis.* Academic Press, 1979.

[16] C. Robert and G. Casella, *Monte Carlo Statistical Methods.* Springer, 1999.

[17] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[18] M.A. Tanner, *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.* Springer, 1996.

[19] V. N. Vapnik, *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, ISBN 0-387-94559-8, 1995.

[**Books and Review Papers on face recognition**]

[20] M.S.Bartlett, *Face Image Analysis by Unsupervised Learning.* Kluwer Academic Publishers, 2001.

[21] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings of IEEE*, vol. 83, pp. 705–740, 1995.

[22] S. Gong, S.J. McKenna, *Dynamic Vision: From Images to Face Recognition.* Imperial College Press, 2000.

[23] P.W. Hallinan, G. Gordon, A. Yuille, P. Giblin, and D. Mumford, *Two- and Three-Dimensional Patterns of the Face.* A. K. Peters, Ltd., 1999.

[24] T. Kanade, *Computer Recognition of Human Faces.* Birhauser, Basel, Switzerland, and Stuggart, Germany, 1973.

[25] S.Z. Li, A.K. Jain (Eds.), *Handbook of Face Recognition.* Springer-Verlag, 2004.

[26] H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang (Eds.), *Face Recognition: From Theory to Applications.* Springer-Verlag, 1998.

[27] W. Zhao, R. Chellappa, A. Rosenfeld, and J. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 12, 2003.

[**Biometrics**]

[28] Biometric Catalog. http://www.biomtricscatalog.org.

[29] Biometric Consortium. http://www.biometrics.org.

[30] Deparment of Homeland Security (DHS), US-VISIT Program. http://www.dhs.goc/dhspublic/interapp/editorial/editorial_0333.xml.

[31] National Institue of Standards and Technologies (NIST), Biometrics Web Site. http://www.nist.gov/biometrics.

[32] D.M. Blackburn, "Biometrics 101 (version 3.1)" http://www.biometricscatalog.org/biometrics/Introduction.asp, March 2004.

[33] R. Hietmeyer, "Biometric identification promises fast and secure processings of airline passengers," *The Internationl Civil Aviation Organization Journal*, vol. 55, no. 9, pp. 10-11, 2000.

[34] P.J. Phillips, R.M. McCabe, and R. Chellappa, "Biometric image processing and recognition," *Proceedings of European Signal Processing Conference*, 1998.

[**Psychophysical and neural aspects**]

[35] I. Biederman and P. Kalocsai, "Neural and psychophysical analysis of object and face recognition," In *Face Recognition: From Theory to Applications*, H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang (Eds.), Springer-Verlag, 1998.

[36] V. Bruce, *Recognizing Faces.* Lawrence Erlbaum Associates, London, U.K., 1988.

[37] V. Bruce, P.J.B. Hancock, and A.M. Burton, "Human face perception and identification," In *Face Recognition: From Theory to Applications*, H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie, and T.S. Huang (Eds.), Springer-Verlag, 1998.

[38] A.J. O'Toole, "Psychological and neural perspectives on human faces recognition," In *Handbook of Face Recognition*, S.Z. Li and A.K. Jain (Eds.), Springer, 2004.

[39] B. Knight and A. Johnston, "The role of movement in face recognition," *Visual Cognition*, vol. 4, pp. 265-274, 1997.

[**Face recognition from still images**]

[40] M.S. Barlett, H.M. Ladesand, and T.J. Sejnowski, "Independent component representations for face recognition," *Proceedings of SPIE 3299*, pp. 528-539, 1998.

[41] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[42] M.J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local paramterized models of image motion," *International Journal of Computer Vision*, vol. 25, pp. 23-48, 1997.

[43] T. Cootes, G. Edwards, and C. Taylor, "Active appearance model," *European Conference on Computer Vision*, 1998.

[44] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," *Journal of Optical Society of America A*, pp. 1724–1733, 1997.

[45] T. Huang, Z. Xiong, and Z. Zhang, "Face recognition applications," *Handbook of Face Recognition*, S. Li and A. K. Jain (Eds.), Springer, 2004.

[46] M.D. Kelly, "Visual identification of people by computer," *Tech. rep. AI-130*, Stanford AI project, Stanform, CA, 1970.

[47] M. Kirby and L. Sirovich, "Application of Karhunen-Loéve procedure of the characterization of human faces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 103–108, 1990.

[48] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. v. d. Malsburg, R.P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. Computers*, vol. 42, no. 3, pp. 300–311, 1993.

[49] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 442-455, 1997.

[50] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Toward automatic simulation of aging affects on face images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 442-455, 2002.

[51] S.H. Lin, S.Y. Kung, and J.J. Lin, "Face recognition/detection by probabilistic decision based neural network," *IEEE Trans. Neural Networks*, vol. 9, pp. 114-132, 1997.

[52] C. Liu and H. Wechsler, "Evolutionary pursuit and its applications to face recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 570-582, 2000.

[53] M.J. Lyons, J. Biudynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357-1362, 1999.

[54] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-19, no. 7, pp. 696–710, 1997.

[55] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian modeling of facial similarity," *Advances in Neural Information Processing Systems*, vol. 11, pp. 910–916, 1999.

[56] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 780–788, 2002.

[57] P.J. Phillips, "Support vector machines applied to face recognition," *Advances in Neural Information Processing Systems*, vol. 11, pp. 803-809, 1998.

[58] P.J. Phillips, H. Moon, S. Rizvi, and P.J. Rauss, "The FERET evaluation methodology fro face-recognition algorithms," *IEEE Trans. attern Analysis and Machine Intelligence*, vol. 22, pp. 1090–1104, 2000.

[59] P.J. Phillips, P. Grother, R.J. Micheals, D.M. Blackburn, E. Tabbssi, and M. Bone, "Face recognition vendor test 2002: evaluation report" *NISTIR 6965*, http://www.frvt.org, 2003.

[60] Y. Tian, T. Kanade, and J. Cohn, "Recognizing action units of facial expression analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1-19, 2001.

[61] Y. Tian, T. Kanade, and J. Cohn, "Recognizing action units of facial expression analysis," In *Handbook of Face Recognition*, S.Z. Li and A.K. Jain (Eds.), Springer, 2004.

[62] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 72–86, 1991.

[63] M.-H. Yang, "Kernel eigenfaces vs. kernel Fisherfaces: Face recognition using kernel methods," *Proceedings of International Conference on Automatic Face and Gesture Recognition*, 2002.

[64] W. Zhao, R. Chellappa, and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 361-341, Nara, Japan, 1998.

[**Face recognition across illumination and poses**]

[65] J. Atick, P. Griffin, and A. Redlich, "Statistical approach to shape from shading: Reconstrunction of 3-dimensional face surfaces from single 2-dimentional images," *Neural Computation*, vol. 8, pp. 1321–1340, 1996.

[66] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1063–1074, 2003.

[67] T. Cootes, K. Walker, and C. Taylor, "View-based Active appearance models," *Proceedings of International Conference on Automatic Face and Gesture Recognition*, 2000.

[68] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 643 –660, 2001.

[69] R. Gross, I. Matthews, and S. Baker, "Eigen light-fields and face recognition across pose," *Proceedings of Intenational Conference on Automatic Face and Gesture Recognition*, Washington D.C., 2002.

[70] R. Gross, I. Matthews, and S. Baker, "Fisher light-fields for face recognition across pose and illumination," *Proceedings of the German Symposium on Pattern Recognition*, Washington D.C., 2002.

[71] R. Gross, I. Matthews, and S. Baker, "Appearance-based face recognition and light-fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 4, pp. 449 - 465, April, 2004.

[72] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 1994.

[73] S. Romdhani and T. Vetter, "Efficient, robust and accurate fitting of a 3D morphable model," *Proceedings of IEEE Internationl Conference on Computer Vision*, pp. 59-66, Nice, France, 2003.

[74] A. Shashua and T. R. Raviv, "The quotient image: Class based re-rendering and recognition with varying illuminations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 129–139, 2001.

[75] T. Sim, S. Baker, and M. Bast, "The CMU pose, illuminatin, and expression (PIE) database," *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 53–58, Washington D.C., 2002.

[76] T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 733–742, 1997.

[77] M.A.O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," *European Conference on Computer Vision*, vol. 2350, pp. 447-460, Copenhagen, Denmark, May 2002.

[78] M. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," *Proceedings of International Conference on Pattern Recognition*, Quebec City, Canada, 2002.

[**Face recognition from video sequences**]

[79] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland, "Multimodal person recognition using unconstrained audio and video," *Proceedings of International Conference on Audio- and Video-Based Person Authentication*, pp. 176–181, Washington D.C., 1999.

[80] A. Fitzgibbon and A. Zisserman, "Joint manifold distance: a new approach to appearance based clustering," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003.

[81] R. Gross and J. Shi, "The CMU Motion of Body (MoBo) Database," *CMU-RI-TR-01-18*, 2001.

[82] A. Howell and H. Buxton, "Face recognition using radial basis function neural networks," *Proceedings of British Machine Vision Conference*, pp. 455–464, 1996.

[83] T. Jebara and A. Pentland, "Parameterized structure from motion for 3D adaptive feedback tracking of faces," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 144 –150, Puerto Rico, 1997.

[84] K. Lee, M. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003.

[85] B. Li and R. Chellappa, "Face verification through tracking facial features," *Journal of Optical Society of America A*, vol. 18, no. 12, pp. 2969–2981, 2001.

[86] B. Li and R. Chellappa, "A generic approach to simultaneous tracking and verification in video," *IEEE Transaction on Image Processing*, vol. 11, no. 5, pp. 530–554, 2002.

[87] Y. Li, S. Gong, and H. Liddell, "Modelling faces dynamically across views and over time," *Proceedings of International Conference on Computer Vision*, pp. 554 –559, Hawaii, 2001.

[88] Y. Li, S. Gong, and H. Liddell, "Constructing facial identity surfaces in a nonlinear discriminant space," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.

[89] X. Liu and T. Chen, "Video-based face recognition using adaptive hidden markov models," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003.

[90] S. McKenna and S. Gong, "Non-intrusive person authentication for access control by visual tracking and face recognition," *Proceedings of International Conference on Audio- and Video-based Biometric Person Authentication*, pp. 177–183, Crans-Montana, Switzerland, 1997.

[91] G. Shakhnarovich, J. Fisher, and T. Darrell, "Face recognition from long-term observations," *Proc. European Conference on Computer Vision*, Copenhagen, Denmark, 2002.

[92] J. Steffens, E. Elagin, and H. Neven, "Personspotter - fast and robust system for human detection, tracking, and recognition," *Proceedings of Internationl Conference on Automatic Face and Gesture Recognition*, pp. 516–521, Nara, Japan, 1998.

[93] H. Wechsler, V. Kakkad, J. Huang, S. Gutta, and V. Chen, "Automatic video-based person authentication using the RBF network," *Proceedings of International Conference on Audio- and Video-based Biometric Person Authentication*, pp. 85–92, Crans-Montana, Switzerland, 1997.

[**Lighting and illumination**]

[94] R. Basri and D. Jacobs, "Photometric stereo with general, unknown lighting," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, pp. 374–381, Hawaii, 2001.

[95] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 218–233, 2003.

[96] H. Hayakawa, "Photometric stereo under a light source with arbitrary motion," *Journal of Optical Society of America, A*, vol. 11, 1994.

[97] P.N. Belhumeur and D.J. Kriegman, "What is the set of images of an object under all possible illumination conditions?" *International Journal of Computer Vision*, vol. 28, pp. 245–260, 1998.

[98] P. Belhumeur, D. Kriegman, and A. Yuille, "The bas-relief ambiguity," *International Journal of Computer Vision*, vol. 35, pp. 33–44, 1999.

[99] R. T. Frankot and R. Chellappa, "A method for enforcing integrability in shape from shaging problem," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 439–451, 1987.

[100] R. Ramamoothi and P. Hanrahan, "On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object," *Journal of the Optical Society of America (JOSA A)*, vol. 18, pp. 2448–2459, 2001.

[101] A. Shashua, "On photometric issues in 3d visual recognition from a single 2D image," *International Journal of Computer Vision*, vol. 21, pp. 99–122, 1997.

[102] I. Shimshoni, Y. Moses, and M. Lindenbaum., "Shape reconstruction of 3D bilaterally symmetric surfaces," *International Journal of Computer Vision*, vol. 39, pp. 97–100, 2000.

[103] A.L. Yuille, D. Snow, R. Epstein, and P.N. Belhumeur, "Determining generative models of objects under varying illumination: Shape and albedo from

multiple images using svd and integrability," *Internationl Journal of Computer Vision*, vol. 35, pp. 203–222, 1999.

[104] W. Zhao and R. Chellappa, "Symmetric shape from shading using self-ratio image," *International Journal of Computer Vision*, vol. 45, pp. 55–752, 2001.

[105] Q. F. Zheng and R. Chellappa, "Estimation of illuminant direction, albedo and shape from shading," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 680–702, 1991.

[**Tracking, detection, and registration**]

[106] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 562–575, 1995.

[107] A. Bergen, P. Anadan, K. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," *European Conference on Computer Vision*, pp. 237–252, Stockholms, Sweden, 1992.

[108] M.J. Black and A.D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *European Conference on Computer Vision*, vol. 1, pp. 329–342, Cambridge, UK, 1996.

[109] M.J. Black and D.J. Fleet, "Probabilistic detection and tracking of motion discontinuities," *Proceedings of International Conference on Computer Vision*, vol. 2, pp. 551–558, Greece, 1999.

[110] M.E. Brand, "Morphable 3D Models from Video," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.

[111] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering nonrigid 3D shape fomr image streams," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, 2000.

[112] 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. Aerospace and Electronic Systems*, vol. AES-26, pp. 639–656, 1990.

[113] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149, Hilton Head, SC, 2000.

[114] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings on Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[115] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1025–1039, 1998.

[116] M. Irani, "Multi-frame optical flow estimation using subspace constraints," *Proceedings of International Conference on Computer Vision*, pp. 626-633, Greece, 1999.

[117] M. Irani and P. Anandan, "Factorization with Uncertainty," *European Conference on Computer Vision*, pp. 539-553, Dublin, Ireland, 2000.

[118] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," *European Conference on Computer Vision*, pp. 343–356, Cambridge, UK, 1996.

[119] M. Isard and A. Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," *Euporean Conference on Computer Vision*, vol. 1, pp. 767–781, Freiburg, Germany, 1998.

[120] A. D. Jepson, D. J. Fleet, and T. El-Maraghi, "Robust online appearance model for visual tracking," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 415–422, Hawaii, 2001.

[121] F. Jurie and M. Dhome, "A simple and efficient template matching algorithm," *Proceedings of International Conference on Computer Vision*, vol. 2, pp. 544–549, Vancouver, BC, 2001.

[122] Q. Ke and T. Kanade, "A subspace approach to layer extraction," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.

[123] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *International Joint Conference on Artifical Intelligence*, 1981.

[124] B. North, A. Blake, M. Isard, and J. Rittscher, "Learning and classification of complex dynamics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1016–1034, 2000.

[125] G. Qian and R. Chellappa, "Structure from motion using sequential monte carlo methods," *Proceedings of International Conference on Computer Vision*, pp. 614 –621, Vancouver, BC, 2001.

[126] C. Rasmussen and G. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 560–576, 2001.

[127] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," *European Conference on Computer Vision*, vol. 2, pp. 702–718, Copenhagen, Denmark, 2002.

[128] J. Sullivan and J. Rittscher, "Guiding random particle by deterministic search," *Proceedings of International Conference on Computer Vision*, vol. 1, pp. 323 –330, Vancouver, BC, 2001.

[129] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[130] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," *Proceedings of International Conference on Computer Vision*, pp. 50–59, Vancouver, BC, 2001.

[131] J. Vermaak, P. Peraz, M. Gangnet, and A. Blake, "Towards improved obsevation models for visual tracking: selective adaption," *European Conference on Computer Vision*, pp. 645–660, Copenhagen, Denmark, 2002.

[132] P. Voila and M. Jones, "Robust real-time object detection," *Second Intl. Workshop on Stat. and Comp. Theories of Vision*, Vancouver, BC, 2001.

[133] Y. Wu and T. S. Huang, "A co-inference approach to robust visual tracking," *Proceedings of International Confererence on Computer Vision*, vol. 2, pp. 26–33, Vancouver, BC, 2001.

[134] C. Yang, R. Duraiswami, A. Elgammal, and L. Davis, "Real-time kernel-based tracking in joint feature-spatial spaces," *Tech. Report CS-TR-4567, Univ. of Maryland*, 2004.

[**Others in computer vision and graphics**]

[135] M. J. Black and A. D. Jepson, "A probabilistic framework for matching temporal trajectories," *Proceedings of International Conference on Computer Vision*, pp. 176–181, Greece, 1999.

[136] R. Bolle and D. Cooper, "On optimally combining pieces of information with application to estimating 3-d complex-object position from range data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 619–638, 1986.

[137] D. Forsyth, "Shape from texture and integrability," *Proc. International Conference on Computer Vision*, pp. 447–453, Vancouver, BC, 2001.

[138] W. T. Freeman and J. B. Tenenbaum, "Learning bilinear models for two-factor problems in vision," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997.

[139] P. Fua, "Regularized bundle adjustment to model heads from image sequences without calibrated data," *Internationl Journal of Computer Vision*, vol. 38, pp. 153-157, 2000.

[140] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," *Proceedings of SIGGRAPH*, pp. 43-54, New Orleans, LA, USA, 1996.

[141] D. Jacobs, "Linear fitting with missing data for structure-from-motion," *Computer Vision and Image Understanding*, vol. 82, pp. 57–81, 2001.

[142] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. Pattern Analysis and Machine Intelligences*, vol. 16, no. 2, pp. 150-162, 1994.

[143] M. Levoy and P. Hanrahan, "Light field rendering," *Proceedings of ACM SIGGRAPH*, New Orleans, LA, USA, 1996.

[144] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan, "Image-based visual hulls," *Proceedings of SIGGRAPH*, pp. 369 - 374, New Orleans, LA, USA, 2000.

[145] A. Roy Chowdhury and R. Chellappa, "Face reconstruction from video using uncertainty analysis and a generic model," *Computer Vision and Image Understanding*, vol. 91, pp. 188-213, 2003.

[146] Y. Shan, Z. Liu, and Z. Zhang "Model-based bundle adjustment with applicaiton to face modeling," *Proceedings of Internationl Conference on Computer Vision*, pp. 645–651, Vancouver, BC, 2001.

[**Statistical analysis and computing**]

[147] B. Adhikara and D. Joshi, "Distance discrimination et resume exhaustif," *Publs. Inst. Statis.*, vol. 5, pp. 57–74, 1956.

[148] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," *Proceedings of the 14th Annual Conference on Uncertainty in AI (UAI)*, pp. 33 – 42, Madison, Wisconsin, 1998.

[149] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," *European Conference on Computer Vision*, pp. 707–720, Copenhagen, Denmark, 2002.

[150] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[151] H. Chernoff, "A measure of asymptotic efficiency of tests for a hypothesis based on a sum of observations," *Annals of Math. Stat.*, vol. 23, pp. 493–507, 1952.

[152] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm." *J. Roy. Statist. Soc. B*, vol. 39, 1977.

[153] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–209, 2000.

[154] D. Fox, "KLD-sampling: Adaptive particle filters and mobile robot localization," *Neural Information Processing Systems (NIPS)*, 2001.

[155] A. Hyvarinen, "Survey on Independent Component Analysis," *Neural Computing Surveys*, vol. 2, pp. 94-128, 1999.

[156] T. Kailath, "The divergance and Bhattacharyya distance measures in signal selection," *IEEE Trans. on Comm. Tech.*, vol. COM-15, pp. 52–60, 1967.

[157] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *J. Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[158] T. Lissack and K. Fu, "Error estimation in pattern recognition via L-distance between posterior density functions," *IEEE Trans. Information Theory*, vol. 22, pp. 34–45, 1976.

[159] J. S. Liu and R. Chen, "Sequential monte carlo for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1031–1041, 1998.

[160] P. Mahalanobis, "On the generalized distance in statistics," *Proc. National Inst. Sci. (India)*, vol. 12, pp. 49–55, 1936.

[161] K. Matusita, "Decision rules based on the distance for problems of fit, two samples and estimation," *Ann. Math. Stat.*, vol. 26, pp. 631–640, 1955.

[162] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dececember 2000.

[163] E. Patrick and F. Fisher, "Nonparametric feature selection," *IEEE Trans. Information Theory*, vol. 15, pp. 577–584, 1969.

[164] P. Penev and J. Atick, "Local feature analysis: A general statistical theory for object representation," *Networks: Computations in Neural Systems*, vol. 7, pp. 477-500, 1996.

[165] H. Shum, K. Ikeuchi, and R. Reddy, "Principal component analysis with missing data and its applications to polyhedral object modeling," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 854–867, 1995.

[166] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319-2323, December 2000.

[167] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analysers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[168] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 61, pp. 611–622, 1999.

[169] T. Wiberg, "Computation of principal components when data are missing," *Proc. Second Symp. Computational Statistics*, pp. 229–236, 1976.

[**Machine learning and kernel methods**]

[170] F. Bach and M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

[171] F. Bach and M. I. Jordan, "Learning graphical models with Mercer kernels," *Advances in Neural Information Processing Systems*, 2002.

[172] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, pp. 2385–2404, 2000.

[173] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neutral networks architectures," *Neural Computation*, vol. 7, pp. 219–269, 1995.

[174] T. Jebara and R. Kondor, "Bhattarcharyya and expected likelihood kernels," *Conference on Learning Theory (COLT)*, 2003.

[175] R. Kondor and T. Jebara, "A kernel between sets of vectors," *Intenational Conference on Machine Learning (ICML)*, 2003.

[176] J. Mercer, "Functions of positive and negative type and their connection with the thoery of integral equations," *Philos. Trans. Roy. Soc. London*, vol. A 209, pp. 415–446, 1909.

[177] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX*, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. IEEE, 1999, pp. 41–48.

[178] P. Moreno, P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for svm classfication in multimedia applications," *Neural Information Processing Systems*, 2003.

[179] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introducation to kernel-based learning algorithms," *IEEE Trans. Neutral Networks*, vol. 12, pp. 181–202, 2001.

[180] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," *Neural Information Processing Systems*, 2002.

[181] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[182] M. Tipping, "Sparse kernel prinicipal component analysis," *Neural Information Processing Systems*, 2001.

[183] C. K. I. Williams, "On a connection between kernel PCA and metric multidimensional scaling," *Neural Information Processing Systems*, 2001.

[184] L. Wolf and A. Shashua, "Kernel principal angles for classification machines with applications to image sequence interpretation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003.

[**Shaohua Zhou's publications**]

[185] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," *Computer Vision and Image Understanding*, vol. 91, pp. 214–245, 2003.

[186] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Processing (to appear)*, 2004.

[187] S. Zhou, R. Chellappa, and D. Jacobs, "Generalized photometric stereo and its applicaitons to face recognition," *International Journal of Computer Vision (submitted)*.

[188] S. Zhou and R. Chellappa, "Image-based face recognition under illumination and pose variantons," *Journal of the Optical Society of America (submitted)*.

[189] S. Zhou and R. Chellappa, "Probabilisitic distances in reproducing kernel Hilbert space," *IEEE Trans. on Information Theory (under preparation)*.

[190] R. Chellappa and S. Zhou, "Face tracking and recognition from video," *Handbook of Face Recognition*, S. Li and A. K. Jain (Eds.), Springer, 2004.

[191] S. Zhou and R. Chellappa, "Face recognition from still images and videos," *Handbook of Image and Video Processing*, A. Bovik (Ed.), Academic Press, 2004.

[192] S. Zhou, V. Krueger, and R. Chellappa, "Face recgnition from video: A condensation approach," *Proceedings of International Conference on Automatic Face and Gesture Recognition*, Washington, D.C., USA, May 2002.

[193] S. Zhou and R. Chellappa, "Probabilistic human Recognition from video," *European Conference on Computer Vision*, vol. 3, pp. 681-697, Copenhagen, Denmark, May 2002.

[194] V. Krueger and S. Zhou, "Exemplar-based face recgnition from video," *European Conference on Computer Vision*, Copenhagen, Denmark, 2002.

[195] R. Chellappa, S. Zhou, and B. Li, "Bayseain methods for probabilistic human recgnition from video," *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Orlando, Florida, USA, 2002.

[196] S. Zhou and R. Chellappa, "A robust algorithm for probabilistic human recognition from video," *Proceedings of International Conference on Pattern Recognition*, Quebec City, Canada, 2002.

[197] R. Chellappa, V. Krueger, and S. Zhou, "Probabilistic recgnition of human faces from video," *Proceedings of IEEE International Conference on Image Processing*, Rochester, NY, 2002.

[198] S. Zhou, "Probabilistic analysis of kernel principal components: classification and mixture modeling," *CfAR Technial Report, CAR-TR-993*, 2003.

[199] S. Zhou and R. Chellappa, "Simultaneous tracking and recognition of human faces from video," *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing*, 2003.

[200] J. Li, S. Zhou, and C. Shekhar, "A comparison of subspace analysis for face recognition," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[201] S. Zhou, R. Chellappa, and B. Moghaddam, "Adaptive visual tracking and recognition using particle filters," *Proceedings of IEEE International Conference on Multimedia & Expo*, Baltimore, USA, 2003.

[202] S. Zhou and R. Chellappa, "Rank constrained recognition under unknown illuminations," *IEEE Intl. Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, 2003.

[203] S. Zhou, R. Chellappa, and B. Moghaddam, "Appearance tracking using adaptive models in a particle filter," *Proceedings of Asian Conference on Computer Vision*, Korea, January 2004.

[204] S. Zhou, R. Chellappa, and D. Jacobs, "Characterization of human faces under illumination variations using rank, integrability, and symmetry constraints," *European Conference on Computer Vision*, Prague, Czech, May 2004.

[205] J. Li and S. Zhou, "Probabilistic face recognition with compressed imagery," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.

[206] J. Shao, S. Zhou, and R. Chellappa, "Appearance-based visual tracking and recognition with trilinear tensor," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.

[207] Z. Yue, S. Zhou, and R. Chellappa, "Robust two-camera visual tracking with homography," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.

[208] S. Zhou and R. Chellappa, "Illuminating light field: Image-based face recognition across illuminations and poses," *Proceedings of International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 2004.

[209] S. Zhou, R. Chellappa, and B. Moghaddam, "Intra-personal kernel space for face recognition," *Proceedings of International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 2004.

[210] S. Zhou and R. Chellappa, "Probabilistic identity characterization for face recognition," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington D.C., USA, June 2004.

[211] S. Zhou and R. Chellappa, "Multiple-exemplar discriminant analysis for face recognition," *Proceedings of International Conference on Pattern Recognition*, Cambridge, UK, August 2004.

[212] J. Shao, S. Zhou, and Q. Zheng, "Robust appearance-based tracking of moving object from moving platform," *Proceedings of International Conference on Pattern Recognition*, Cambridge, UK, August 2004.

[213] J. Shao, S. Zhou, and R. Chellappa, "Simultaneous background and foreground modeling for tracking in surveillance video," *Proceedings of IEEE International Conference on Image Processing*, Singapore, October 2004.