UNIVERSITY OF CALIFORNIA
RIVERSIDE

A Theoretical Analysis of Image Appearance Models with Applications in Face Recognition

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Yilei Xu

December 2008

Dissertation Committee:

Professor Amit K. Roy-Chowdhury, Chairperson
Professor Bir Bhanu
Professor Christian Shelton

Acknowledgments

First of all, I would like to express my deep gratitude to my advisor Professor Amit K. Roy-Chowdhury. He introduced me into this field and guided me very patiently and devotedly throughout my whole study. I benefited a lot from his deep insight, extensive knowledge, and the talent of seeing the big picture. Without his help, I would not have been here. I will cherish this experience; his professional attitude and integrity will always be an ideal to me.

I am also grateful to my committee members and other faculty, including Professor Bir Bhanu, Professor Christian Sheldon, Professor Ertem Tuncel, Professor Jay A. Farrell, Professor Ilya Dumer, Professor Jie Chen, Professor Sheldon Tan, Professor Ping Liang, and Professor Daniel Xu. Their inspiring questions, helpful discussions, and patient teaching led to the fruition of this thesis.

I am happy to have had some very excellent colleagues at Riverside, first of whom that I would like to thank being Bi Song. Through many nice conversations, I got lots of help and support from her, not only in research work, but also in many other aspects. Thanks to Ozgun Bursalioglu, although she was in our lab for only two years, she offered me very kind help when I needed. Also, my sincere thanks go to Dr. Rong Wang, Dr. Jayanth Nayak, Dr. Jiangang Yu, and Dr. Ju Han.

The experiences of working with Dr. Xiaoming Liu, Dr. Bernd Wachmann and Dr. John Weiss during my internships are very valuable to me. They kindly advised me through my internships, broadened my horizons, and introduced me the techniques the industry was using. During the internships, I got not only professional improvement, but,

more importantly, a chance to view the things from another perspective.

I had a very nice time at Riverside, largely due to my great friends here. My first thank goes to Chin-hung Lin, who will always be there and help me when I need him. Others I would like to express my gratitude would be Dr. Jin Tang, Weihua Zhu, Dr. Kezhu Hong, Meng Cao, Min Liu, and Xiaofei Song among others. All their kind help is appreciated.

Not only the friends in Riverside, but also my friends in China and other parts of the world gave me a lot of support and help. Here I should mention Wenjun Liu, Yao Shen, Xi Yin, Chao Wang, Zhiguo Li, Wenjun Jin, Bin Liu, Heng Zhang, Yi Pei, and many others too numerous to mention.

Thanks my girlfriend Jing Chen, without whose help and encouragement I would not have been here. During the period of my PhD, I went through lots of confusion, depression, and impatience. Although she didn't know about the field, every time she tried everything she could to help me and get me back on the ground. Thanks to her for her understanding and constant support throughout my study.

Thanks my parents for bringing me up, educating me, and preparing me for this achievement. Far away in another hemisphere, I can feel their love at any time, any place. This love can never be paid back.

Finally, the work in this thesis was largely funded by the National Science Foundation under grant IIS-0712253 and the support is gratefully acknowledged.

Dedicated

to my parents

and Jing Chen

ABSTRACT OF THE DISSERTATION

A Theoretical Analysis of Image Appearance Models with Applications in Face Recognition

by

Yilei Xu

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2008
Professor Amit K. Roy-Chowdhury, Chairperson

Image appearance modeling is considered to be one of the fundamental problems in computer vision. Its successful solution has numerous applications in object tracking, recognition, surveillance, image and video processing, etc. However, the fact that the image appearance is determined by a large number of factors, including object shape, texture, pose, illumination and camera models, makes it to be a very challenging problem. In this thesis, we present a theory of analytical image appearance modeling, which is derived from fundamental physical laws, and show some applications of this theory in tracking and recognition. We rigorously prove that the image appearance space can be closely approximated to be multilinear, with the illumination and texture subspaces being trilinearly combined with the direct sum of the motion and deformation subspaces. This result allows us to understand theoretically many of the successes and limitations of the linear and multi-linear approaches existing in the computer vision literature (Principle Components Analysis, 3D Morphable Model, Active Appearance Model/Active Shape Model, Multilinear Model), and

also identifies some of the conditions under which they are valid.

Starting from this theory, we show that it is possible to estimate low-dimensional manifolds that describe object appearance while retaining the geometrical information about the 3D structure of the object, termed as Geometry-Integrated Appearance Manifold (GAM). By using a combination of analytically derived geometrical models and statistical learning methods, this can be achieved using a much smaller training set than most of the existing approaches. We also show how to estimate, accurately and efficiently, the parameters of the GAM model through an inverse compositional (IC) tracking framework. We prove the theoretical convergence of this method and show that it leads to significant reduction in computational burden.

One of the most important applications of image appearance models is in object recognition. In this thesis, we present an analysis-by-synthesis framework for face recognition from video sequences that is robust to large changes in facial pose and lighting conditions. This method is based on the analytical image appearance model and the IC tracking framework. The method can handle situations where the pose and lighting conditions in the training and testing data are completely disjoint. We evaluate the algorithm on a face video dataset, and compare against image-based recognition algorithms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The appearance of an image is determined by a large number of factors, including object shape, texture, pose, illumination and camera models. Modeling the appearance of an image is a fundamental problem in computer vision. Although physical laws have revealed clear analytical relationships between image intensity and all the above factors, they are combined together in a highly nonlinear fashion. Knowing only the images, it will be very difficult to use such a complex model to estimate the physical parameters and will require high computational cost. Thus, moderately complex but accurate enough models will be more attractive to computer vision problems.

A number of models, like Active Appearance/Shape Models (AAM/ASM) [46, 14], 3D Morphable Models (3DMM) [11], Multilinear Models (MLM) [81, 85], or non-linear manifolds [40] have been used to construct and parameterize the image appearance manifold

in terms of these factors. To resolve questions about the effectiveness and accuracy of these methods, experimental evaluations have been carried out on larger and larger datasets. While these experiments are very valuable contributions, it is also important to analyze the accuracy of these models from the fundamental physical laws of image formation. In this thesis, we rigorously prove that *the image space of a moving and deforming object under varying illumination can be closely approximated to be multi-linear, with the illumination subspace and the texture subspace being trilinearly combined with the direct sum of the motion and deformation subspaces.* This result allows us to understand the conditions under which each of them is valid. It provides a concise analytical representation of the image space in terms of different physical factors that affect the image formation process.

Starting from this theory, we show how to combine this analytical model with statistical models for robustness in describing non-regular shape variations due to different identity and non-rigid deformations. Retaining the geometrical information about the 3D structure of the object, this model is termed as Geometry-Integrated Appearance Manifold. Due to the same reason, GAM can be constructed using a much smaller training set than most of the existing approaches. We also show how to estimate, accurately and efficiently, the parameters of the GAM model through an inverse compositional (IC) tracking framework. We prove the theoretical convergence of this method and show that it leads to significant reduction in computational burden.

One of the most important applications of image appearance models is in object recognition. In this thesis, we present an *analysis-by-synthesis* framework for face recognition from video sequences that is robust to large changes in facial pose and lighting

conditions. This method is based on the analytical image appearance model and the IC tracking framework. The method can handle situations where the pose and lighting conditions in the training and testing data are completely disjoint. We collect a dataset of 57 face videos on which we perform an experimental evaluation and compare against existing methods.

1.2 Literature Review

Traditionally, motion and illumination (i.e., the geometric and photometric issues) have been studied separately. One of the classical methods for 2D motion estimation on the image plane is optical flow [29]. It assumes that the intensity of a particular point does not change over time. Estimation of 3D motion and structure, usually referred to as the Structure from Motion (SfM) [13, 4, 83, 78, 77] problem, is another classical research area in computer vision. While largely constrained to the analysis of rigid objects, it has been recently extended to non-rigid objects under orthographic projection [84]. For reconstructing 3D structure from discrete views obtained over a wide baseline, stereo reconstruction algorithms (and multi-camera generalizations) have been proposed [19, 26]. However, most SfM and stereo reconstructions algorithms do not take illumination variation into consideration. To understand the inaccuracies that arise in the solution of the 3D reconstruction problems, a number of strategies for statistical analysis of the errors and robust statistical algorithms have been developed [15, 100, 97, 20, 65, 67, 66, 56, 49]. A method for shape reconstruction of a moving object under arbitrary, unknown illumination, assuming motion is known, was presented in [76]. The authors in [98] proposed to model the change of

illumination in optical flow and combine it with structure from motion, photometric stereo, and multi-view stereo in an optimization framework. In [33], the authors proposed a multi-view stereo algorithm that can estimate the three-dimensional shape and non-Lambertian reflectance parameters under fixed illumination. However, none of the above methods provide an explicit expression relating the image, and the motion, structure and illumination variables for video sequences.

In the study of illumination, Shape from Shading (SfS) [22, 28, 50] is one of the earliest and most widely known methods. It is based on the Lambertian reflectance law, and relies on the illumination information in a *single* image to estimate the 3D structure in a scene. Shashua [71] and Moses [47] proposed that, ignoring the effect of shadows, the set of images under varying illumination lies in a 3D linear subspace, and derived the representation of the space. Using this fact and under the condition that the object and camera are fixed, they showed that three images obtained under three independent lighting conditions is sufficient to reconstruct the image set without prior knowledge of illumination conditions. This is known as Photometric Stereo. When an uniform ambient illumination component is considered, the subspace of the image becomes 4D. Belhumeur and Kriegman [9] showed that the set of images of an object under arbitrary illumination forms a convex cone in the space of all possible images. Furthermore, they also proved that, when attached shadow is considered, the subspace dimension grows to infinity. However, most of the energy is packed in a limited number of lower order harmonics, thereby leading to a low-dimensional subspace approximation. In [8] and [60], the authors independently derived that it is possible to use low order spherical harmonics to accurately approximate the reflectance images. Specifi-

cally, they analytically derived a 9D spherical harmonics based linear representation of the images produced by a Lambertian object with attached shadows. An overall framework for modeling reflected light as a convolution of incident illumination with the bidirectional reflectance distribution functions, along with applications, was presented in [61]. For the specular objects, higher orders of the spherical harmonics functions with non-negativity constraints were used for describing the image space [73].

Partial differential equations have been used for representing shape deformations [35] with a lot of success in tracking problems. Another common approach for modeling a deforming object is to use a linear combination of bases. 3DMM methods [11] decompose the 3D shape and texture of a face along the principle component directions, and is well known in applications of face image synthesis and face recognition. AAM [46, 14] is similar to the 3DMM, but is applied in 2D shape and texture. Shape analysis has also been used to study deforming shapes, especially in human activities [86]; however, it focuses on 2D shapes and thus is not well designed for modeling pose and illumination variations.

Only recently, these factors were studied together. To combine the effects of these various factors, linear, multi-linear, and non-linear models of object shape/appearance have been popularly used for modeling the image appearance. Principal Components Analysis (PCA) is one of the early attempts at modeling the image appearance variation due to the change of identity in face images, and later applied to model the variations due to the changes of illumination. Active Appearance Model (AAM) / Active Shape Model(ASM) [46, 14] tried to model the appearance variation due to the changes of shape and texture. 3D Morphable Model (3DMM) [11] is similar to AAM in that it uses linear models for

approximating the 3D shape and texture. However, the image appearance manifold is a highly nonlinear function of the parameters and becomes computationally expensive. A recent work [36] studied the face shape reconstruction problem from two-tone images with a top-down approach. Multilinear Models (MLM) assume the image space to be multilinear in the identity, pose, and illumination, and apply multi-linear SVD to learn the bases. Locally linear models have been another approach for representing the image appearance space [41, 63, 80]. Non-linear manifolds [40] have also been proposed for modeling the facial expression variations across different people. The authors in [30] proposed a dynamic shape and appearance model, which is able to capture occlusions, scene deformations, arbitrary viewpoint variations and changes in its radiance. However, none of the above methods provide an analytical analysis of the validity of these models, which is the focus of this thesis.

1.3 Contributions of the Thesis

In this thesis, we consider a general image formation process - an imaged object undergoing a rigid motion (i.e., pose change) while deforming (non-rigid motion) and the illumination changing randomly. The theoretical derivation is based on a few weak assumptions that are usually applicable - a finite dimensional vector space representation of illumination, small motion between two consecutive frames, and a smooth 3D surface (shape and texture) of the object that is differentiable. Starting from this theory, we show that it is possible to learn complex manifolds of object appearance, which we term as "Geometry-Integrated Appearance Manifolds" (GAMs). Applications of the theory are provided in

illumination invariant tracking and video-based face recognition. The following lists the main contributions of the thesis.

• Starting from fundamental physics-based models governing rigid object motion, deformations, the interaction of light with the object and perspective projection, we derive a description of the mathematical space in which an image lies. Specifically, we prove that the image space can be closely approximated to be multilinear, with the illumination and texture variation subspace being trilinearly combined with the direct sum of the motion, deformation subspaces.

• This result allows us to analyze theoretically the validity of many of the linear, locally linear and multi-linear approaches existing in the computer vision literature, while also identifying some of the physical constraints under which they are valid. In fact, as explained in Section 2.2, we can now understand theoretically why some methods have worked well in some situations, but not so well in others.

• We propose a Geometry-Integrated Appearance Manifold (GAM), which is a *quadrilinear* manifold of object appearance that is able to represent the combined effects of illumination, pose, identity and deformation. The basis vectors of the tangent space to this manifold depend upon the 3D surface normals of the object. Such a representation is not possible through methods that rely purely on learning-based approaches using 2D images.

• The GAM is computed using a combination of analytically derived geometrical models and statistical learning methods. Thus, construction of the GAM requires significantly less data than AAM/ASM [14, 46], 3DMM [11], Probabilistic Appearance Manifold (PAM) [41], and Multilinear Models (MLM) [85] (see Table 3.1 in Section 3.4). This also makes

the learned manifold less dependent upon the actual examples that were used.

• We propose a novel Inverse Compositional (IC) algorithm to efficiently and accurately track objects on this manifold through changes of pose, lighting and deformations. We are able to extend two-frame IC tracking methods to multiple frames without any significant sacrifice in accuracy. We show that IC approaches can be used not only for estimating 3D motion, but also the time-varying lighting conditions in the scene, including the effects of attached shadows.

• We rigorously prove the convergence of the motion and lighting estimates on the GAM from first principles, analyze the computational savings, and provide results on the numerical correctness of the estimates.

• We present a novel framework for *pose and illumination invariant video-based face recognition* that is based on (i) learning joint illumination and motion models from video, (ii) synthesizing novel views based on the learned parameters, and (iii) designing measurements that can compare two time sequences while being robust to outliers. The pose and illumination conditions in the gallery and probe can be *completely disjoint*. With this novel video-based face recognition algorithm, we can handle a variety of lighting conditions, including the presence of multiple point and extended light sources, as well as gradual and sudden changes of lighting patterns over time.

1.4 Organization of the Thesis

The thesis is organized as follows: In Chapter 2, we derive the theory of the analytical image appearance model and provide an outline of the proof. Details of the derivation are given in the Appendices. Some accuracy analysis results of the model are also given. Chapter 3 explains how to combine the analytical model and statistical model to obtain the GAM. A novel inverse compositional 3D pose and lighting estimation algorithm on the GAM is given in Chapter 4. A proof of the convergence and an analysis of the computational savings of the algorithm are also provided. Chapter 5 describes an application of the above methods in illumination and pose invariant video-based face recognition. We conclude the thesis and highlight future work in Chapter 6.

Chapter 2

Analytical Image Appearance Model: Theoretical Derivation

2.1 Introduction

A number of methods have been proposed in the last few years to describe the appearance space of the image of an object, including Active Appearance/Shape Models (AAM/ASM) [46, 14], 3D Morphable Models (3DMM) [11], Multilinear Models (MLM) [81, 85], or non-linear manifolds [40]. However, most of these methods works by assuming a-priori the form of the image space and then applying statistical techniques to learn the models. How physically accurate are these models? How reliable are they in describing the appearance of a particular image? In addition, due to the fact that these methods rely on statistical learning approaches, they need to collect a huge amount of training data under different conditions, and the performance of the trained model depends heavily upon the

quality of this training data.

We start from the first principles, and rigorously prove that *the image space of a moving and deforming object under varying illumination can be closely approximated to be multi-linear, with the illumination subspace and the texture subspace being trilinearly combined with the direct sum of the motion and deformation subspaces.* This result allows us to understand the conditions under which each of them is valid. It provides an analytical representation of the image space in terms of different physical factors that affect the image formation process. The main results in this chapter were presented in [95, 90, 91].

In the following parts of this chapter, we will first start with the most simple case where only illumination varies. Then rigid motion is considered, and then the deformation and texture variation. The detailed derivation can be found in Appendices. We also discuss the implications of the result with respect to many of the existing heuristic models in section 2.4.1. Thorough accuracy analysis and visualization of the bases of the analytical appearance model are provided in section 2.5.

2.2 Theoretical Derivation of the Image Appearance Space

2.2.1 Problem formulation

Consider an object whose images are being captured by a perspective camera. We attach the world reference frame to the camera. Let the 3D surface of the object be described by $\mathcal{C}(u, v) \in \mathbb{R}^3$ in the object reference frame, where $\mathcal{C}$ is parameterized using u and v. Consider two time instances t_1 and $t_2 = t_1 + \Delta t$, between which the object can move

rigidly and deform (see Fig. 2.1).

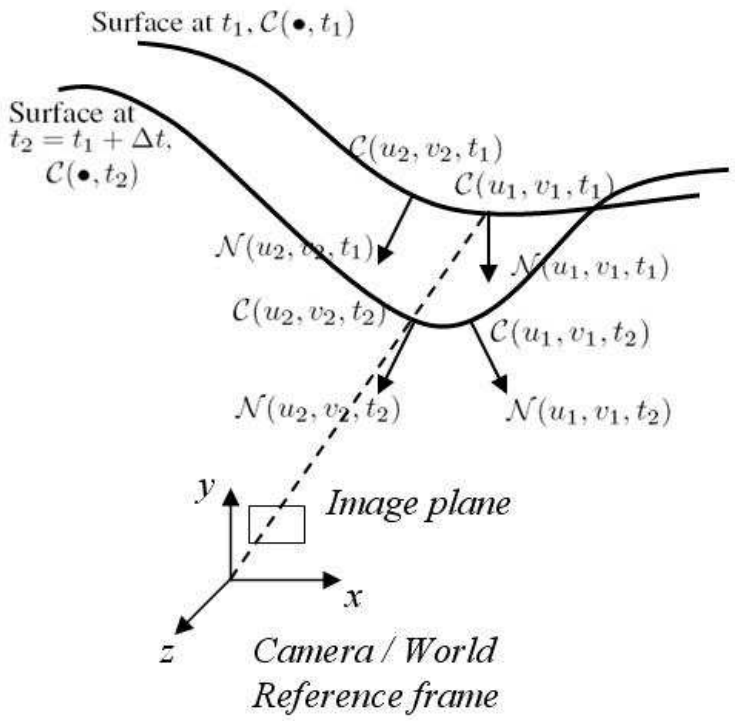


Figure 2.1: Pictorial representation depicting imaging framework.

Let the pose of the object with respect to the camera reference frame before the motion be defined as the translation $\mathbf{T}$ and rotation matrix $\mathbf{R}$. The rigid motion of the object is defined as the translation $\Delta\mathbf{T} = \mathbf{V}\Delta t$ of the centroid and the rotation $\Delta\boldsymbol{\Omega} = \omega\Delta t$ about the centroid of the object during the time interval Δt. $\Delta\mathbf{R} = e^{\hat{\omega}\Delta t}$ is the rotation matrix due to $\Delta\boldsymbol{\Omega}$, and $\hat{\omega} \in \mathbb{SO}(3)$ is the skew-symmetric matrix corresponding to $\omega \in \mathbb{R}^3$. Deformation is defined in the object reference frame. While the object is deforming, its texture may also change and the illumination may be different at t_1 and t_2. Our goal is to express the image $\mathcal{I}_{t_2}$ mathematically as a function of $\mathcal{I}_{t_1}$, motion $\Delta\mathbf{T}$ and $\Delta\boldsymbol{\Omega}$, deformation,

illumination, and texture change.

We make the following assumptions.

A1) Δt is small, which implies that the rigid motion and deformation between t_1 and t_2 are small.

A2) Illumination is represented by a set of finite dimension linear orthogonal bases.

A3) $\mathcal{C}(u, v)$ is smooth and the deformation is smooth, allowing $\frac{\partial^2 \mathcal{C}}{\partial u \partial t} = \frac{\partial^2 \mathcal{C}}{\partial t \partial u}$ and $\frac{\partial^2 \mathcal{C}}{\partial v \partial t} = \frac{\partial^2 \mathcal{C}}{\partial t \partial v}$, and texture ρ is spatially smooth.

Assumption (A1) is made since we are describing the local image appearance space. Assumptions (A2) and (A3) are valid in most practical situations.

For ease of explanation, we start from a fixed rigid object under varying illumination. Then we consider the problems of a moving rigid object under varying illumination (**Theorem 1**) and a fixed deforming object under varying illumination (**Theorem 2**). Next we consider a moving and deforming object under fixed illumination (**Theorem 3**), and a moving and deforming object under varying illumination (**Theorem 4**). We prove that the image space of a moving and deforming object under varying illumination is a locally multilinear. As we show in Appendix F, a locally multilinear subspace can be embedded in a higher-dimensional globally multilinear space. Thus, we show that the image appearance space is multilinear and provide an exact physical parametrization of this space. When we relax Assumption (A1), the image space become nonlinear.

2.2.2 Fixed Rigid Object under Varying Illumination

In [8, 60], the authors showed that, when a rigid object is fixed with respect to the camera, the reflectance image $\mathcal{I}$ of size $P \times Q$ can be represented as

$$\mathcal{I} = \mathbf{l}^{\mathbf{T}} \mathcal{B}_l(\mathbf{n}) = \mathcal{B}_l(\mathbf{n}) \times_l \mathbf{l}, \tag{2.1}$$

where the 2D tensor $\mathcal{I} \in \mathbb{R}^{1 \times P \times Q}$ is the reflectance image, $\mathbf{l} \in \mathbb{R}^{N_l \times 1}$ is the illumination coefficient vector determined by the illumination conditions, $\mathcal{B}_l \in \mathbb{R}^{N_l \times P \times Q}$ is the tensor version of a set of basis images, $\mathbf{n}$ is the unit norm vector at the reflection point, and $\times_l$ is the *mode-n product* (see Appendix A) along the illumination dimension. For a Lambertian object with attached shadows, $N_l \approx 9$. The bases for each pixel can be expressed as [8]

$$b_i(\mathbf{n}_j) = \rho_j r_i Y_i(\mathbf{n}_j), i = 0, 1, \ldots, \tag{2.2}$$

where ρ encrypts the surface reflectance property at the reflection point, Y_i is the spherical harmonics function, and r_i is a constant for each spherical harmonics order. For each pixel, b_i is a vector. Arranging the b_i for all the pixels together will give the tensor $\mathcal{B}_l$. When the Lambertian reflectance property is not satisfied, higher orders of the spherical harmonics functions will be needed [73].

2.2.3 Moving Rigid Object under Varying Illumination

Under this scenario, we need to consider the relative motion between the camera and the object. When the object moves with respect to the camera, the reflection point on the surface $\mathcal{C}(\bullet, t_1)$ corresponding to each pixel will be different from the one of $\mathcal{C}(\bullet, t_2)$. Using the dynamics we can express the change of the surface normal on the reflection point

in terms of the motion parameters $\mathbf{m} = (\Delta\mathbf{T}^{\mathbf{T}}, \Delta\boldsymbol{\Omega}^{\mathbf{T}})^{\mathbf{T}}$, where $\Delta\mathbf{T}$ is the translation of the centroid of the object and $\Delta\boldsymbol{\Omega}$ is the rotation about the centroid of the object. Integrating the dynamics into the framework in Section 2.2.2, we have

Theorem 1 *Under Assumptions (A1), (A2), the image space of a moving rigid object under varying illumination is bilinear in the illumination and motion parameters.*

The detailed derivation of Theorem 1 is shown in the Appendix B. Using tensor notation (see Appendix A), the equation (B.15) in Appendix B can be expressed succinctly as:

$$\mathcal{I} = (\mathcal{B}_l + \mathcal{B}_{ml} \times_m \mathbf{m}) \times_l \mathbf{l}, \tag{2.3}$$

where $\mathcal{B}_m \in \mathbb{R}^{N_l \times 6 \times P \times Q}$ is the tensor version of the motion bases (please refer to the Appendix B for the exact forms of $\mathcal{B}_l$, and $\mathcal{B}_m$).

This bilinear space result integrates the effects of illumination and motion in generating an image from a 3D object using a perspective camera. When the object does not move, motion $\mathbf{m}$ is zero, and thus the result is the same as the one in [8], a linear subspace for a rigid object under varying illumination. When the illumination remains the same, the reflectance image spans a linear subspace of motion variables. When the illumination and motion variables all change, the image space is "close to" bilinear. Thus the joint illumination and motion space for a sequence of images is bilinear with N_l illumination variables and six motion variables. The shape of the object is encoded in the tensors $\mathcal{B}_l$ and $\mathcal{B}_{ml}$. Although this theory incorporates motion into the framework, it only models the motion of the rigid object, restricting the applicability of Theorem 1 to deforming objects.

2.2.4 Deforming Object at Fixed Pose under Varying Illumination

Consider that the pose of the object is fixed with respect to the camera but is deforming. The surface of the object is a function of time, i.e. $\mathcal{C}(u,v,t) : \mathbb{R}^2 \times [0,T) \rightarrow \mathbb{R}^3$. Assume that the evolution of the surface obeys the following PDE:

$$\frac{\partial \mathcal{C}(u,v,t)}{\partial t} = \beta(u,v,t)\mathcal{N}(u,v,t). \tag{2.4}$$

The derivation of this model can be found in Section 2.1 of [68]. Thus, given the parameterization (u,v), the deformation of the object is defined by the function $\beta(u,v,t)$, where $\mathcal{N}(u,v,t)$ is the surface normal at $\mathcal{C}(u,v,t)$. At the time instance t, $\beta(u,v,t)$ is a 2D function and can be decomposed using most of the 2D transformation techniques, including 2D unitary transforms, wavelet transforms, and B-spline basis among others. Assuming the deformation of an object to be smooth over (u,v), most of the energy of $\beta(u,v,t)$ at time instance t would be concentrated in the low frequency components. Decomposing $\beta(u,v,t)$ using the top N_D bases, we have

$$\beta(u,v,t) = \Phi_d(u,v) \times_d \mathbf{b}_d(t), \tag{2.5}$$

where $\Phi_d \in \mathbb{R}^{N_D \times 1}$ is the vector of the top N_D basis at (u,v), $\mathbf{b}_d \in \mathbb{R}^{N_D \times 1}$ encrypts the deformation at (u,v) as a function of t, and $\times_d$ indicates the tensor product along the deformation dimension.

Using the same parameterization, the texture function on the surface can be decomposed using top $\mathcal{N}_\rho$ bases as

$$\rho(u,v,t) = \Phi_\rho(u,v) \times_\rho \mathbf{b}_\rho(t), \tag{2.6}$$

where $\mathbf{b}_\rho \in \mathbb{R}^{N_\rho \times 1}$ and $\Phi_\rho \in \mathbb{R}^{N_\rho \times 1}$. Similarly, $\times_\rho$ indicates the tensor product along the texture dimension. Then we have the following theorem:

Theorem 2 *Under Assumptions (A1), (A2) and (A3), the image space of a fixed deforming object under varying illumination is trilinear in the illumination, deformation and texture parameters.*

Outline of the proof: We first define some notation required for our derivation. Let $\mathcal{C}(\bullet, t_1)$ and $\mathcal{C}(\bullet, t_2)$ represent the same object before and after deformation respectively, as shown in Fig. 2.1. The ray from the optical center to a particular pixel (x, y) intersects with the surface of the object at some point. Before the object's deformation, the ray intersects with the surface at $\mathcal{C}(u_1, v_1, t_1)$, and after deformation, it intersects at $\mathcal{C}(u_2, v_2, t_2)$. During the deformation, $\mathcal{C}(u_2, v_2, t_1)$ evolves to $\mathcal{C}(u_2, v_2, t_2)$. Note that $\mathcal{C}(u_2, v_2, t_2)$ may not overlap with $\mathcal{C}(u_1, v_1, t_1)$ - they are just on the same projection ray.

From (2.1), we see that when the illumination coefficient, $\mathbf{l}$, is known, only the norm and the reflectance of the surface point of interest affect the reflection intensity at a particular pixel. The difference between $\mathcal{N}(u_1, v_1, t_1)$ and $\mathcal{N}(u_2, v_2, t_2)$ consists of two parts. The first part is the change from $\mathcal{N}(u_1, v_1, t_1)$ to $\mathcal{N}(u_2, v_2, t_1)$, which can be approximated using a first order Taylor expansion at $\mathcal{C}(u_1, v_1, t_1)$, while the second part is due to the deformation from $\mathcal{N}(u_2, v_2, t_1)$ to $\mathcal{N}(u_2, v_2, t_2)$. Thus we can express the change in norm as

$$\Delta\mathcal{N} = \mathcal{N}(u_2, v_2, t_2) - \mathcal{N}(u_1, v_1, t_1) = \mathbf{J}_{\mathcal{N}|u_1,v_1,t_1}\boldsymbol{\Delta} + \frac{\partial \mathcal{N}(u_2, v_2, t)}{\partial t}|_{t_1}\Delta t, \tag{2.7}$$

where $\mathbf{J}_{\mathcal{N}}|_{u_1,v_1,t_1}$ is the Jacobian matrix of the norm, $\mathcal{N}(u, v, t)$, with respect to the param-

eters (u, v) at point $\mathcal{C}(u_1, v_1, t_1)$, and $\mathbf{\Delta}$ is the difference between the surface parameters (u_2, v_2) and (u_1, v_1). The term $\frac{\partial \mathcal{N}(u_2,v_2,t)}{\partial t}\Delta t$ is due to the deformation.

For the texture change, using (2.6) we have

$$\begin{aligned} \rho(u_2, v_2, t_2) &= \Phi_\rho(u_2, v_2) \times_\rho \mathbf{b}_\rho^{\mathbf{T}}(t_2) \\ &= (\Phi_\rho(u_1, v_1) + \nabla\Phi_\rho|_{u_1,v_1}\Delta) \times_\rho \mathbf{b}_\rho^{\mathbf{T}}(t_2), \end{aligned} \tag{2.8}$$

Thus, $\Delta\ \mathcal{N}$ and $\rho(u_2, v_2, t_2)$ can be substituted into the expression for the basis images in (2.2), which can be rewritten as

$$\begin{aligned} b_i(u_2, v_2, t_2) &= ((\Phi_\rho(u_1, v_1) + \nabla\Phi_\rho|_{u_1,v_1}\Delta) \times_\rho \mathbf{b}_\rho(t_2))r_i Y_i(\mathcal{N}(u_1, v_1, t_1) + \Delta\mathcal{N}) \\ &= ((\Phi_\rho(u_1, v_1) + \nabla\Phi_\rho|_{u_1,v_1}\Delta)r_i Y_i(\mathcal{N}(u_1, v_1, t_1)) \\ &\quad +\Phi_\rho(u_1, v_1)r_i\nabla Y_i|_{\mathcal{N}(u_1,v_1,t_1)}\Delta\mathcal{N}) \times_\rho \mathbf{b}_\rho(t_2) + O(\Delta^2). \end{aligned} \tag{2.9}$$

The last term is a higher order term, which we will ignore for now.

Let us now introduce a subscript $_w$ to denote the variables in the world reference frame. Since $\mathcal{C}_w(u_1, v_1, t_1)$ and $\mathcal{C}_w(u_2, v_2, t_2)$ are on the same ray (see Fig. 2.1), we can represent the difference between them using a unit vector $\mathbf{r}$ under the perspective camera model as

$$\mathcal{C}_w(u_2, v_2, t_2) - \mathcal{C}_w(u_1, v_1, t_1) = k\mathbf{r}. \tag{2.10}$$

The transformation between the world frame and the object frame can be written as

$$\begin{aligned} \mathcal{C}_w(u_1, v_1, t_1) &= \mathbf{R}\mathcal{C}(u_1, v_1, t_1) + \mathbf{T}, \\ \mathcal{C}_w(u_2, v_2, t_2) &= \mathbf{R}\mathcal{C}(u_2, v_2, t_2) + \mathbf{T}. \end{aligned} \tag{2.11}$$

Note that the pose of the object is fixed during the deformation. Using (2.4),(2.5) and (2.6), the evolution of the object surface can be rewritten in a discrete format as

$$\mathcal{C}(u_2, v_2, t_2) \quad = \quad \mathcal{C}(u_2, v_2, t_1) + \mathbf{b}_d^{\mathbf{T}}(t_1)\Phi_d(u_2, v_2)\mathcal{N}(u_2, v_2, t_1)\Delta t. \tag{2.12}$$

Under Assumption (A2), which implies that the deformation between the two consecutive frames is small, the point $\mathcal{C}(u_2, v_2, t_1)$ should be close to the point $\mathcal{C}(u_1, v_1, t_1)$. Thus, we may alternatively consider that the new point $\mathcal{C}(u_2, v_2, t_1)$ is on the tangent plane that passes through the point $\mathcal{C}(u_1, v_1, t_1)$, i.e.,

$$\mathcal{C}(u_2, v_2, t_1) = \mathcal{C}(u_1, v_1, t_1) + \alpha_u \mathcal{T}_u|_{u_1,v_1,t_1} + \alpha_v \mathcal{T}_v|_{u_1,v_1,t_1}, \tag{2.13}$$

where $\mathcal{T}_u|_{u_1,v_1,t_1}$ represents the tangent $\mathcal{T}_u$ at (u_1, v_1, t_1). After a series of manipulations (see Appendix C), we have

$$\mathbf{A} \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} = -\mathbf{b}_d^{\mathbf{T}}(t_1)\Phi_d \left(\mathbf{I} - \frac{\mathbf{R}^{-1}\mathbf{r}\mathcal{N}^{\mathbf{T}}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}} \right) \Delta t, \text{where}$$

$$\mathbf{A} = (\mathbf{I} - \frac{\mathbf{R}^{-1}\mathbf{r}\mathcal{N}^{\mathbf{T}}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}})(\mathbf{b}_d^{\mathbf{T}}(t_1)\Phi_d\mathbf{J}_{\mathcal{N}}\Delta t + \mathcal{N}\mathbf{b}_d^{\mathbf{T}}(t_1)\nabla\Phi_d\Delta t) + (\mathcal{T}_u|_{t_1}, \mathcal{T}_v|_{t_1}). \tag{2.14}$$

Note that in (2.14), $\mathcal{T}_u, \mathcal{T}_v, \mathcal{N}, \mathbf{J}_{\mathcal{N}}, \mathbf{R}, \mathbf{r}$ are computed at t_1, and $\Phi_d, \nabla\Phi_d$ are constants in time. The first term $(\mathbf{I} - \frac{\mathbf{R}^{-1}\mathbf{r}\mathcal{N}^{\mathbf{T}}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}})(\mathbf{b}_d^{\mathbf{T}}\Phi_d\mathbf{J}_{\mathcal{N}}\Delta t + \mathcal{N}\mathbf{b}_d^{\mathbf{T}}\nabla\Phi_d\Delta t) \sim O(\Delta t)$, while the second term $(\mathcal{T}_u|_{t_1}, \mathcal{T}_v|_{t_1}) \sim O(1)$. Thus, using Assumption (A2) that Δt is small, the first term in the right hand side of the expression of $\mathbf{A}$ in (2.14) can be ignored with respect to the second term. Consequently, the solution of (α_u, α_v) can be written as

$$\begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} = \mathbf{B}\mathbf{b}_d(t_1)\Delta t, \text{where}$$

$$\mathbf{B} = -(\mathcal{T}_u, \mathcal{T}_v)^{+}(\mathbf{I} - \frac{\mathbf{R}^{-1}\mathbf{r}\mathcal{N}^{\mathbf{T}}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}})\mathcal{N}\Phi_d^{\mathbf{T}}, \tag{2.15}$$

and $(\mathcal{T}_u, \mathcal{T}_v)^+$ indicates the pseudo inverse of the non-square matrix $(\mathcal{T}_u, \mathcal{T}_v)$.

In (2.7), using Assumption (A2) to neglect the terms $O(\Delta t^2)$ with respect to $O(\Delta t)$ and Assumption (A3) for smooth deformation (see Appendix D), we have

$$\frac{\partial \mathcal{N}}{\partial t}|_{u_2,v_2,t_1} \boldsymbol{\Delta} t \approx -\left(\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u_1,v_1,t_1))\mathbf{J}_{\mathcal{N}}(\Phi_d|(u_1,v_1))\right)^{\mathbf{T}} \mathbf{b}_d(t_1)\Delta t. \tag{2.16}$$

Thus, substituting (2.15) and (2.16) back into (2.7), the change of the norm can be expressed as

$$\Delta\mathcal{N} = (\mathbf{J}_{\mathcal{N}|u_1,v_1,t_1}\mathbf{B} - \nabla\mathcal{C}|_{u_1,v_1,t_1}\nabla\Phi_d|^{\mathbf{T}}_{u_1,v_1,t_1})\mathbf{b}_d(t_1)\Delta t. \tag{2.17}$$

Thus, both $\Delta\mathcal{N}$ and Δ are linear functions of $\mathbf{b}_d$. Substituting back into (2.9), and using tensor notation, we will have

$$\mathcal{I} = (\mathcal{B}_{\rho l} + \mathcal{B}_{d\rho l} \times_d \mathbf{b}_d \Delta t) \times_\rho \mathbf{b}_\rho \times_l \mathbf{l}, \tag{2.18}$$

where $\mathcal{B}_{d\rho l} \in \mathbb{R}^{N_D \times N_\rho \times N_l \times P \times Q}$ is the tensor version of the deformation and texture change basis, and $\mathcal{B}_{\rho l} \in \mathbb{R}^{1 \times N_\rho \times N_l \times P \times Q}$. Thus, the image space is a locally trilinear function of the illumination, deformation, and texture change parameters. The locality property comes because this description is for a small deformation from a specific shape. This locally trilinear space can be embedded into a globally trilinear one as shown in Appendix F.□

2.2.5 Moving and Deforming Object under Fixed Illumination

Theorem 3 *Under Assumptions (A1), (A2) and (A3), the image space of a rigidly moving and deforming object under fixed illumination is a bilinear, with the texture subspace being bilinearly combined with the direct sum of the motion and deformation subspaces.*

Outline of the Proof: Reconsider Figure 2.1. We still have

$$\mathcal{C}_w(u_2, v_2, t_2) - \mathcal{C}_w(u_1, v_1, t_1) = k\mathbf{r}, \tag{2.19}$$

$$\mathcal{C}_w(u_1, v_1, t_1) = \mathbf{R}\mathcal{C}(u_1, v_1, t_1) + \mathbf{T},$$

$$\mathcal{C}_w(u_2, v_2, t_2) = \mathbf{\Delta R R}\mathcal{C}(u_2, v_2, t_2) + \mathbf{\Delta T} + \mathbf{T}, \tag{2.20}$$

where $\mathbf{r}$ is the unit vector along the projection ray. Similarly, the deformation of the object can still be described using (2.12). Because the time interval between the two consecutive frames is small, the motion and deformation are small. Using similar reasoning as used for deriving (2.13), we again have

$$\mathcal{C}(u_2, v_2, t_1) = \mathcal{C}(u_1, v_1, t_1) + \alpha_u \mathcal{T}_u|_{u_1,v_1,t_1} + \alpha_v \mathcal{T}_v|_{u_1,v_1,t_1}. \tag{2.21}$$

From Appendix E, we have

$$\mathbf{A}\begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} = (\mathbf{I} - \frac{\mathbf{R}^{-1}\mathbf{r}\mathcal{N}^{\mathbf{T}}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}})(\hat{\mathbf{C}}_1 \Delta\mathbf{\Omega} - \mathbf{R}^{-1}\Delta\mathbf{T} - \mathcal{N}\Phi_d^{\mathbf{T}}\mathbf{b}_d(t_1)\Delta t)),$$

where

$$\mathbf{A} \;=\; (\mathcal{T}_u, \mathcal{T}_v) + (\mathbf{I} - \frac{\mathbf{R}^{-1}\mathbf{r}\mathcal{N}^{\mathbf{T}}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}})(\mathbf{b}_d^{\mathbf{T}}(t_1)\Phi_d\mathbf{J}_{\mathcal{N}} + \mathcal{N}\mathbf{b}_d^{\mathbf{T}}(t_1)\nabla\Phi_d)\Delta t. \tag{2.22}$$

Under similar reasoning used in deriving (2.15), we can again neglect the second term in the expression of $\mathbf{A}$ in (2.22), and the solution to $(\alpha_u, \alpha_v)^{\mathbf{T}}$ can be obtained as

$$\begin{aligned}\begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} \;&=\; -(\mathcal{T}_u, \mathcal{T}_v)^{+}(\mathbf{I} - \frac{\mathbf{R}^{-1}\mathbf{r}\mathcal{N}^{\mathbf{T}}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}})(\hat{\mathbf{C}}_1 \Delta\mathbf{\Omega} - \mathbf{R}^{-1}\Delta\mathbf{T} - \mathcal{N}\Phi_d^{\mathbf{T}}\mathbf{b_d(t_1)}\Delta t) \\ &\triangleq\; \mathbf{D}\Delta\mathbf{\Omega} + \mathbf{E}\Delta\mathbf{T} + \mathbf{F}\mathbf{b}_d(t_1)\Delta t. \end{aligned} \tag{2.23}$$

However, when there exists both rigid motion and deformation, the temporal change of $\mathcal{N}(u_2, v_2)$ from t_1 to t_2 consists of two parts: one due to the deformation, and one due to the rotation. In Appendix D, we derived the temporal change of norm from (2.4), which is purely due to deformation, i.e.,

$$\frac{\partial \mathcal{N}}{\partial t}|_{u_2,v_2,t_1} \mathbf{\Delta} t|_{\Delta \mathbf{\Omega}=0} \approx -\left(\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u_1, v_1, t_1))\mathbf{J}_{\mathcal{N}}(\Phi_d|(u_1, v_1))\right)^{\mathbf{T}} \mathbf{b}_d(t_1)\Delta t. \tag{2.24}$$

The temporal change of normal due to the rigid rotation by $\Delta\mathbf{\Omega}$ is

$$\frac{\partial \mathcal{N}}{\partial t}|_{u_2,v_2,t_1} \Delta t|_{\mathbf{b}_d=0} \approx -\hat{\mathcal{N}}|_{u_1,v_1,t_1} \Delta\mathbf{\Omega}. \tag{2.25}$$

Thus, substituting (2.23), (2.24) and (2.25) back into (2.7) and (2.8), the change of the norm can be expressed as

$$\begin{aligned} \Delta\mathcal{N} &= (\mathbf{J}_{\mathcal{N}}|_{u_1,v_1,t_1}\mathbf{D} - \hat{\mathcal{N}}|_{u_1,v_1,t_1})\Delta\mathbf{\Omega} + \mathbf{J}_{\mathcal{N}}|_{u_1,v_1,t_1}\mathbf{E}\Delta\mathbf{T} \\ &\quad +(\mathbf{J}_{\mathcal{N}}|_{u_1,v_1,t_1}\mathbf{F} - \nabla\mathcal{C}|_{u_1,v_1,t_1}\nabla\Phi_d|^{\mathbf{T}}_{u_1,v_1,t_1})\mathbf{b}_d\Delta t. \end{aligned} \tag{2.26}$$

Thus, both $\Delta\mathcal{N}$ and Δ are linear functions of $\Delta\mathbf{T}$, $\Delta\mathbf{\Omega}$ and $\mathbf{b}_d$. Substituting back into (2.9), and using tensor notation, we will have

$$\mathcal{I} = (\mathcal{G}_\rho + \mathcal{G}_{md\rho} \times_m \begin{pmatrix} \mathbf{V} \\ \omega \\ \mathbf{b}_d \end{pmatrix} \Delta t) \times_\rho \mathbf{b}_\rho, \tag{2.27}$$

where $\mathcal{G}_\rho = \mathcal{B}_{\rho l} \times_l \mathbf{l}$ and $\mathcal{G}_{md\rho} = \mathcal{B}_{md\rho l} \times_l \mathbf{l}$ are the joint deformation, rigid motion and texture basis obtained by substituting (2.26) into (2.9). □

2.2.6 Moving and Deforming Object under Varying Illumination - Main Result

Theorem 4 *The image space of a rigidly moving and deforming object under varying illumination is multi-linear, with the illumination subspace and the texture subspace being trilinearly combined with the direct sum of the motion and deformation subspaces.*

Outline of the Proof: When both texture and illumination are represented as functions of t as $\mathbf{b}_\rho(t)$ and $\mathbf{l}(t)$, using augmented variables we can have the following directly from (2.27):

$$\mathcal{I}_t = \mathcal{B}_{l\rho md} \times_l \mathbf{l}(t) \times_\rho \mathbf{b}_\rho(t) \times_m \begin{pmatrix} \mathbf{V}\Delta t \\ \omega\Delta t \\ \mathbf{b}_d\Delta t \\ 1 \end{pmatrix}, \tag{2.28}$$

where $\mathcal{B}_{l\rho dm} \in \mathbb{R}^{N_l \times N_\rho \times (6+N_D+1) \times P \times Q}$ is the tensor version of the joint illumination, texture, rigid motion and deformation bases. The result is valid in a local region around pose $(\mathbf{T}, \mathbf{R})$ and the shape used for computing $\mathcal{B}_{l\rho dm}$. As shown in Appendix F, the locality result can be extended to a global one. □

2.3 Discussion of the Theoretical Results

2.3.1 Implications of the Results

The result in (2.28) implies that the illumination and texture subspaces are trilinearly combined with the union of the rigid motion and deformation subspaces. The first

two factors, illumination and texture, describe the photometric effects while the last two factors, rigid motion and deformation, describe the geometric effects. Equation (2.28) models the illumination and texture variations globally while the rigid motion and deformation are modeled locally on the manifold of the image appearance. To construct the image space representing all possible pose and deformations, we divide the whole space into a set of clusters, each cluster being identified with a cardinal point in pose and deformation space.

The effects of 3D translation can be removed by centering and scale normalization, while in-plane rotation to a pre-defined pose can mitigate the effects of rotation about the z-axis. Thus the image of object under arbitrary pose, $\mathbf{p}$, can always be described by the multilinear object representation at a predefined $(\mathbf{T}_x^{pd}, \mathbf{T}_y^{pd}, \mathbf{T}_z^{pd}, \mathbf{\Omega}_z^{pd})$, with only $\mathbf{\Omega}_x$ and $\mathbf{\Omega}_y$ depending upon the particular pose. Thus, the image manifold under any pose can be approximated by the collection of a few tangent planes on distinct $\mathbf{\Omega}_x^j$ and $\mathbf{\Omega}_y^j$, denoted as $\mathbf{p}_j$.

Concerning the deformation, in many applications, the total deformation is small and the theory may be adequate to capture the global effects. For our example, if we consider the applications on face images, the energy of the variation due to expression usually is small when compared with the shape of the face. Thus, we consider the deformations due to the expression to be "local" around the neutral face, and use the multi-linear bases computed at the neutral face to model them.

2.3.2 Implications of the Assumptions

We used three assumptions for deriving Theorems 1, 2, 3 and 4. Assumption (A2) essentially says that we use a finite-dimensional basis illumination model. This is widely used. For Lambertian surfaces, the dimension number is small, while non-Lambertian surface requires higher dimensions. Also, the basis function can be represented using spherical harmonics, wavelets, and other orthogonal representations. Our derivation does not need a specific choice, only that it is a function of the surface normal. Assumption (A3) is again reasonable for many objects. Assumption (A1) is made since we consider a local region of the image appearance space. This assumption is reasonable for most video sequences captured under frame rates between 15 and 30 fps, which can be used to validate the theoretical model.

2.3.3 Gradual change of illumination and texture

In the above derivation, Assumption (A1) imposes the constraint that the change of pose and deformation between the two consecutive frames is small. On the other hand, we did not place such constraints upon the illumination and texture. If we further assume that both the illumination and texture do not change drastically within the time interval Δt, we can have the following results

Corollary 1 *When the illumination and texture change gradually, the image space of a rigidly moving and deforming object under varying illumination and texture becomes linear.*

Outline of the Proof: As illumination and texture changes gradually, we have

$$\mathbf{l}(t_2) = \mathbf{l}(t_1) + \frac{\partial \mathbf{l}}{\partial t}|_{t_1} \Delta t,$$
$$\mathbf{b}_\rho(t_2) = \mathbf{b}_\rho(t_1) + \frac{\partial \mathbf{b}_\rho}{\partial t}|_{t_1} \Delta t. \tag{2.29}$$

Substituting (2.29) back into (2.28), we have

$$\mathcal{I}_{t_2} = \mathcal{I}_{t_1} + \tilde{\mathcal{B}}_{l\rho md} \begin{pmatrix} \frac{\partial \mathbf{l}}{\partial t}|_{t_1} \\ \frac{\partial \mathbf{b}_\rho}{\partial t}|_{t_1} \\ \mathbf{V} \\ \omega \\ \mathbf{b}_d \end{pmatrix} \Delta t. \tag{2.30}$$

Thus, under the assumption that the illumination and texture do not change drastically, the image space becomes a linear subspace of all the factors around $\mathcal{I}_{t_1}$. Geometrically, the local multi-linear manifold degenerates into the tangent plane. □

2.3.4 Drastic change of the pose and the shape

Usually, the pose and shape of the object cannot change drastically in a small time interval (e.g. a video sequence). Thus the cross and high-order terms of the rigid motion $\mathbf{V}\Delta t$, $\omega \Delta t$ and deformation $\mathbf{b}_d \Delta t$ can be neglected as per Assumption (A1). If we relax assumption (A1), the change of pose and object shape (i.e., $\mathbf{V}\Delta t$, $\omega \Delta t$ and $\mathbf{b}_d \Delta t$) can be large and the higher order terms of Δt should be retained. In this case, we can show the following:

Corollary 2 *If the second order terms of Δt are retained and given the parameters we used, the image space of a rigidly moving and deforming object under varying illumination and*

texture will not be multi-linear.

Outline of the Proof: In equation (2.9), we kept the first order term of Δ and ignored the higher order terms. From (2.23), we know $\Delta \sim O(\Delta t)$. Thus when we keep the higher order terms of Δt, higher order terms of Δ needs to be kept. Thus, from (2.9), the term

$$\Delta^{\mathbf{T}} \nabla \Phi_\rho|_{u_1,v_1} r_i \nabla Y_i|_{\mathcal{N}(u_1,v_1,t_1)} \mathbf{J}_{\mathcal{N}}|_{u_1,v_1,t_1} \Delta \times_\rho \mathbf{b}_\rho(t_2) \tag{2.31}$$

should be kept. Substituting the expression of Δ in (2.23) into (2.31), we then have

$$(\Delta\mathbf{T}, \mathbf{\Omega}, \mathbf{b}_d \Delta t) \begin{pmatrix} \mathbf{E} \\ \mathbf{D} \\ \mathbf{F} \end{pmatrix} \nabla \Phi_\rho|_{u_1,v_1} r_i \nabla Y_i|_{\mathcal{N}(u_1,v_1,t_1)} \mathbf{J}_{\mathcal{N}}|_{u_1,v_1,t_1} (\mathbf{E}, \mathbf{D}, \mathbf{F}) \begin{pmatrix} \Delta\mathbf{T} \\ \Delta\mathbf{\Omega} \\ \mathbf{b}_d \Delta t \end{pmatrix} \times_\rho \mathbf{b}_\rho(t_2). \tag{2.32}$$

Thus, keeping high order terms of Δt will introduce not only the cross terms between $\Delta\mathbf{T}, \Delta\mathbf{\Omega}, \text{and} \mathbf{b}_d$, but also their squares, leading to the image space not being multilinear. □

We would like to emphasize that these results are for the chosen parameters to represent motion, lighting and texture. The reason for the choice of these parameters is their clear physical reasoning. Other choices can lead to simpler models, but the derived bases may be difficult to interpret physically.

2.4 Modeling the Face Image Space

When confined to face images of a single person, the variations of the texture and shape are usually small while the change due to illumination may still be drastic. Thus

from **Corollary 1**, the image space of faces becomes bilinear with the illumination being bilinearly combined with direct sum of the motion, deformation and texture parameters, i.e.

$$\mathcal{I}_t = (\mathcal{B}_{l\bar{\rho}md} + \mathcal{B}_{l\rho md} \times_{m\rho} \begin{pmatrix} \frac{\partial \mathbf{b}_\rho}{\partial t} \\ \mathbf{V} \\ \omega \\ \mathbf{b}_d \end{pmatrix} \Delta t) \times_l \mathbf{l}(t), \text{where} \mathcal{B}_{l\bar{\rho}md} = \mathcal{B}_{l\rho md} \times_\rho \mathbf{b}_{\bar{\rho}}. \quad (2.33)$$

$\mathbf{b}_{\bar{\rho}}$ is the mean face texture coefficient. Thus, (2.33) models face appearance locally around the neutral mean shape and mean texture of faces at the cardinal poses $\mathbf{p}_j$, while globally along the illumination dimension.

Although the result in (2.33) is locally multi-linear along pose dimension, in Appendix F, we show that this piecewise locally multi-linear manifold can be embedded into a higher dimensional globally multi-linear manifold of much higher dimension.

2.4.1 Relation to Existing Methods

This theoretical study provides an understanding of the validity of many linear/multi-linear models of object appearance/shape representation used recently in computer vision. We can also understand the conditions under which these popular models can be applied. We provide below such an analysis, taking face representation and recognition as an example (since all of the models have been applied to faces).

PCA: From (2.33) we can see that, when the illumination and pose are fixed, the image space is linear in the shape and texture parameters, which encrypt the identity. This

proves the validity of the use of PCA under such scenarios. It explains the relatively good performance of PCA when applied to the face recognition problem under fixed pose and illumination and poor performance when illumination is changing.

AAM/ASM: AAM/ASM [14] represent shape and appearance using a linear set of basis vectors, which are then mapped non-linearly to the image space. Using our analytically derived bases, the image space can be obtained as in (2.33) even with pose and illumination variations. This is a simpler form than the AAM/ASM models.

MLM: In MLM [81, 85], different factors (illumination, pose, identity) are assumed to be globally multi-linearly combined. We show that lighting and texture are indeed trilinearly combined with the direct sum of the motion and deformation subspaces. Since this multi-linearity property is local, MLM methods will be more efficient and accurate when modeling local regions of the image space. However, from Appendix F, we see that a global MLM is also valid so long as we are willing to use higher dimensions.

Local Linearization: Probabilistic Appearance Model (PAM) [41] uses a series of tangent planes along pose to approximate the manifold - thus it is also locally linear. Our theoretical result provides an analytical description of this space. In [96], the authors locally linearize the appearance manifold for tracking, but they obtain the linearized basis from a learning algorithm. Again, we provide an analytical description of this linear subspace, which can be used to obtain the bases in a manner that is not dependent on the training data. The same reasoning is valid for locally linear models like [63, 80].

Non-linear approaches: In 3DMM, once the textured 3D shape is obtained, it is combined with the illumination and camera projection model, and thus the image pixel inten-

sities are nonlinear in the shape and texture coefficients. This is a more accurate representation (**Corollary 2**), but comes at the cost of higher computation due to optimization on a non-linear manifold. Non-linear manifolds is also the approach taken in [40].

2.5 Experimental Results

2.5.1 Synthetic Data

We used a 3D mean face model with uniform texture obtained from the 3DMM dataset to compute the analytically derived bases in (2.28), and to validate the synthesis of images using these bases.

In Fig. 2.2, we show some representative basis images. The first column in the motion bases shows the bases for translation along the vertical axis, while the second column shows in-plane rotation bases. Some representative bases of the deformation and texture using 2D DCT basis functions are shown in the following columns.

We show the comparison between the images synthesized with our theory, and the ones synthesized by simulating the PDEs in (2.4) and (2.6) using the 3D face model in Fig. 2.3. We fix the illumination and pose, and then apply deformations on the cheeks and around mouth using 2D DCT basis functions. The texture change is effected over the entire face. Again, there is very little visual difference between the two.

2.5.2 Numerical Accuracy Analysis

To evaluate the theory in a more precise manner, we performed a numerical error analysis. We chose some typical range of rigid motion, deformation, and texture variation

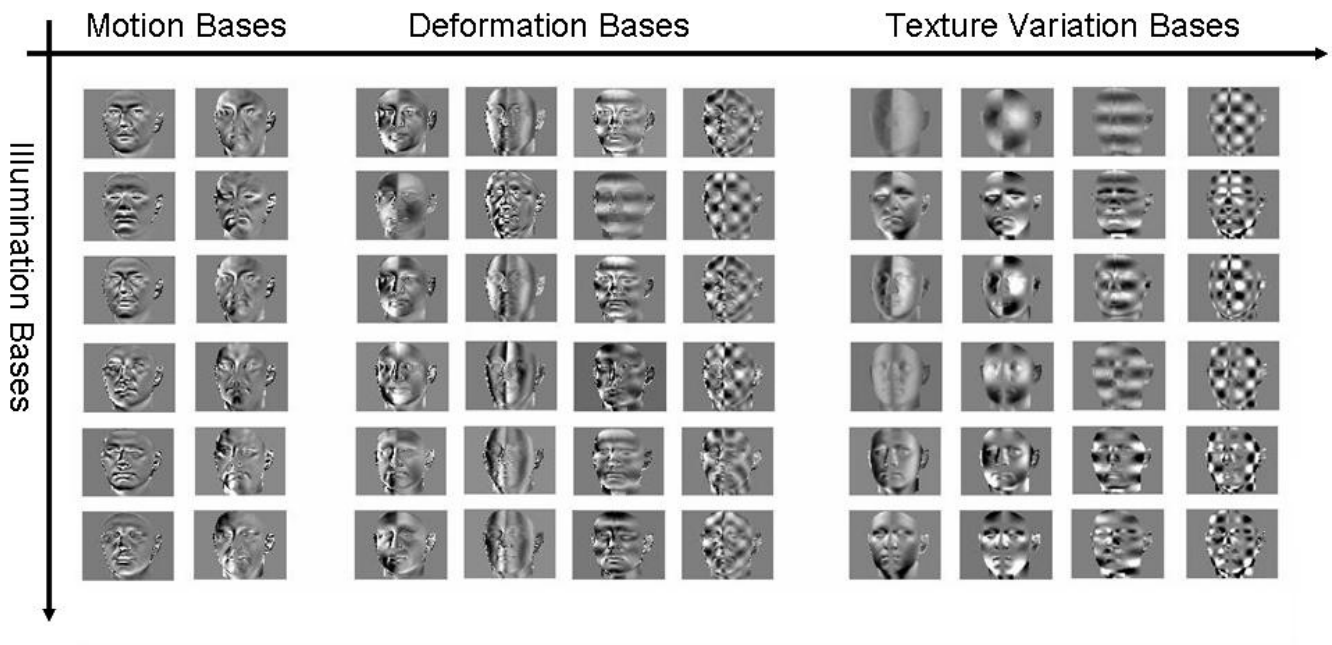


Figure 2.2: Some representative illumination, motion, deformation and texture variation basis images of a 3D face model.

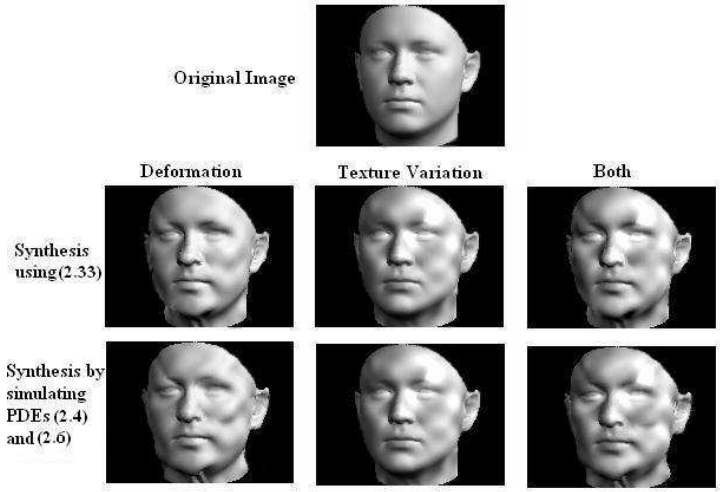


Figure 2.3: Comparison between the images synthesized with our theory, and the ones synthesized by simulating the PDEs in (2.4) and (2.6) using a 3D face model

between two consecutive frames in a video sequence. We computed the difference between the theoretically predicted pixel intensities and the true pixel intensities, normalized by the true values, and took the mean of this normalized error over the face region in the image. Assuming the face to be a hemisphere, we assumed that in one second, the deformation will not exceed 5% of the radius of this hemisphere, and set $\frac{5\%}{30 \text{ frames}}$ as one unit on the axis of deformation. Similarly, for the texture change, we assume the variance of the change will not exceed 5% of the square of the mean value of the original texture. For the rotation, we let that the maximum degree the object can rotate in one second to be 30°, which means 1° between two consecutive frames.

In Fig. 2.4, we plot the normalized error versus (a) deformation and texture variation, (b) deformation and rigid motion, and (c) texture variation and motion. We choose rotation along the vertical axis for the motion (as that is a common motion of the face in video). Fig. 2.4 indicates that, within a typical range of motion, deformation, and texture variation, the normalized error between the predicted value and the true value will not exceed 6%. This is the worst case performance and happens when the object is deforming and rotating. This is in accordance with the theory since we neglect higher order changes due to deformation and rigid motion in equations (2.14) and (2.22).

2.6 Conclusions

In this chapter, we analyzed the accuracy of linear and multi-linear object representation models from the fundamental physical laws of object motion and image formation. We proved that the image appearance space is multilinear, with the illumination and texture

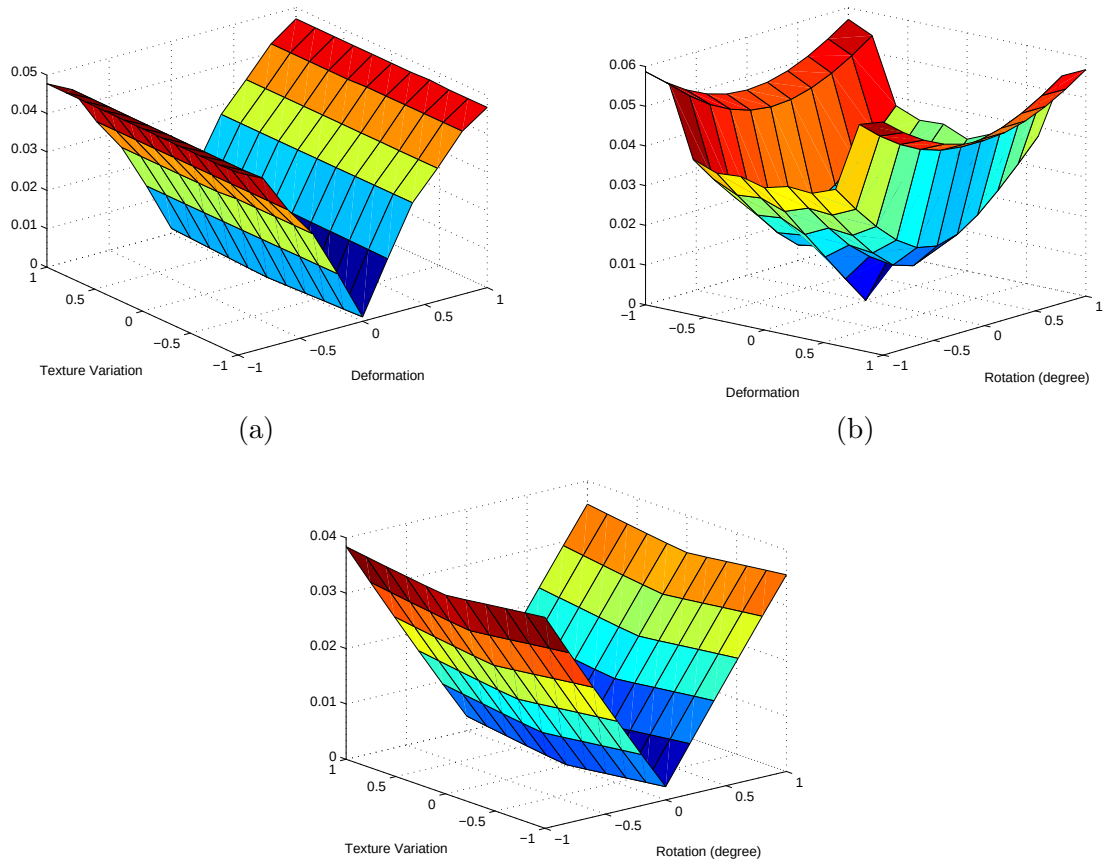


Figure 2.4: Accuracy analysis of the theoretical model. The error is computed as the squared difference between the theoretically predicted pixel intensities and the true pixel intensities, normalized by the true values, and taking its mean over the face region.

subspaces being trilinearly combined with the direct sum of the motion and deformation subspaces. Using this result, we discussed the validity of many of the linear and multi-linear approaches existing in the computer vision literature, including PCA, AAM/ASM, MLM, locally linear models and 3DMM. Experimental accuracy analysis of the theoretical results was presented.

Chapter 3

Combining Analytical and Statistical Models: Geometry-Integrated Appearance Manifold (GAM)

3.1 Introduction

In the previous chapter, we showed the derivation of the analytical image appearance model from first principles. However, its not easy to analytically describe the variation of the shape and texture due to the change of identity and expression in terms of the deformation and texture variation coefficients. Using general purpose bases, like the 2D cosine bases used in Chapter 2, for modeling such variations will lead to the requirement of a large

number of bases to capture a satisfactory percentage of the variation energy. In this chapter, we show how to combine the analytical model of the illumination and motion derived above with statistical learning models for identity and deformation so that we can benefit from the accuracy of the analytical model, and the robustness of statistical learning approaches. We demonstrate that it is possible to estimate low-dimensional manifolds that describe object appearance while retaining the geometrical information about the 3D structure of the object using a much smaller training set than most of the existing approaches. Specifically, we derive a quadrilinear manifold of object appearance that can represent the effects of illumination, pose, identity and deformation, and the basis functions of the tangent space to this manifold depend on the 3D surface normals of the objects.

Low dimensional representations of object appearance have proved to be one of the successful strategies in computer vision for applications in tracking, modeling and recognition. Principle Components Analysis is one of the early low dimensional representations that assumed the data to be approximately spanning a linear subspace. It works well for different people, but the performance deteriorates when illumination and pose changes. Active Appearance Model/Active Shape Model (AAM/ASM) [46, 14] represent shape and appearance of faces using a linear set of basis vectors, which are then mapped non-linearly to the image space. They can model the pose variation to some extent; however, illumination variation destroys the applicability of the AAM. To simultaneously model all these factors, Multi-linear model (MLM) assumes the image space to be multi-linear [85]. Then the Mode-N SVD is applied for learning the multi-linear bases for this space. To handle the pose variation, Probabilistic Appearance Manifold (PAM) works by clustering the images

and then for each cluster apply PCA to learn the linear bases [41]. Basically, PAM uses a series of tangent planes to approximate the whole image manifold along the pose dimension. The idea of local linearization is also proposed by the authors in [96]. They use a camera cluster for capturing the images at neighboring poses, assuming them to be locally multi-linear, and linearize these image sets for tracking.

These methods have two characteristics that may be limitations in many circumstances. *First*, in all these approaches, the construction of the underlying low-dimensional manifold relies upon obtaining different instances of the object's appearance under various conditions (e.g., pose, lighting, identity and deformations) and then using statistical data analysis and machine learning tools to approximate the appearance space. This approach requires first collecting a large number of examples of the object's appearance, and the accuracy of the method depends upon the examples that have been chosen for the training phase. Representation of appearances that have not been seen during the training phase can be inaccurate. *Second*, these representations do not retain any information about the 3D structure of the object, although the appearance must depend upon the 3D shape. In mathematical modeling terms, this is a purely *data-driven* approach.

On the other hand, physical models of the image formation process have been integrated in recent work. 3D morphable model (3DMM) [11] has achieved great success in facial image synthesis and recognition. It uses a linear subspace for modeling the 3D shape of the face, and the texture as well. Then physical models, including illumination models and camera projection models are applied for obtaining the final image. Although it achieves great success, it does not analyze the form of the image space. Due to the same

reason, the computation cost is also very intensive. Recently, some researchers started from Lambertian's reflectance law and analytically derived the form of the image space under different illumination conditions [8, 60]. They showed that, the image space of a Lambertian object under varying illumination lies approximately close to a 9D linear subspace. In addition, the bases to this subspace can be analytically constructed using the spherical harmonics functions. However, these theories focus on static images, ignoring the spatio-temporal coherence within the video sequences, as explained in Chapter 1. In Chapter 2, we modeled the scenario of the dynamic scene by integrating the illumination, motion, deformation and texture variation models. We showed that the form of the image appearance of a general object is actually a multilinear function of the illumination, motion, deformation and texture parameters. However, in the problem of face appearance modeling, it is difficult to express mathematically and precisely the change of shape and texture between different identities and expressions.

In this chapter, we will show how to combine the analytical image appearance model we derived in the previous chapter with statistical learning approaches for obtaining a quadrilinear manifold of illumination, pose, identity and expression parameters modeling the facial images. This result was also presented in [93]. Efficient estimation of these parameters will be presented in the next chapter. The rest of the chapter is organized as follows. Section 3.2 gives an overview of the GAM. Section 3.3 presents the multi-linear object representation framework by combining the analytically derived illumination/motion bases and statistically learned bases over identity and deformation. In Section 3.4, some analysis of two GAM examples is given. The comparison of the size of the training data

needed for constructing GAM and other face appearance models is also given. Finally, section 3.5 concludes the chapter.

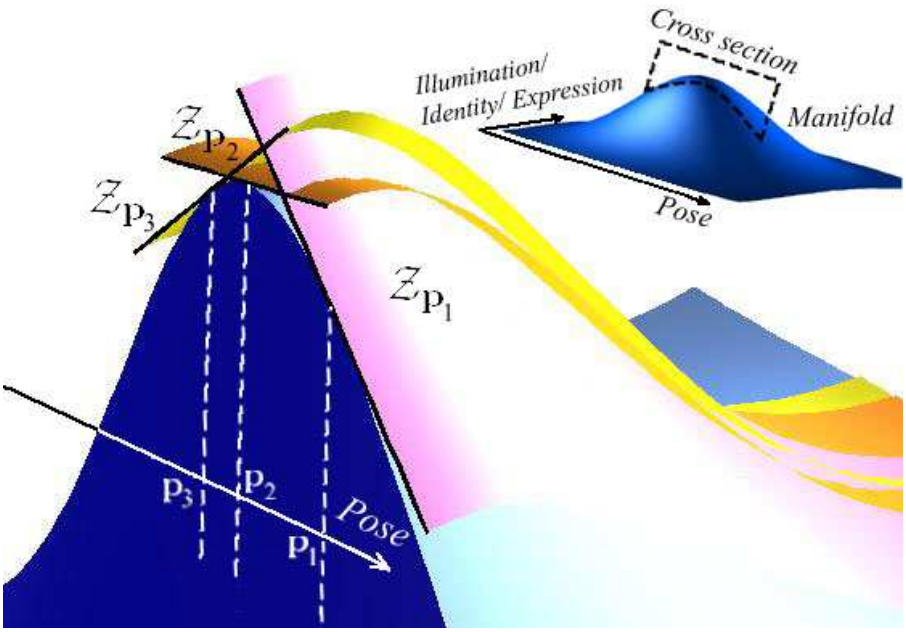


Figure 3.1: Pictorial representation of a GAM cross-section. Only two axes are shown for simplicity. The GAM can be visualized as a collection of locally linear tangent planes along the pose dimension.

3.2 Overview of Proposed Approach

We combine analytically derived geometrical models that represent the effects of motion, lighting and 3D shape [8, 61, 90], with statistical learning approaches that are used to model the other effects like identity (e.g., faces of different people) and non-rigidity which are not easy to represent analytically. Lighting is modeled using a spherical harmonics based linear subspace representation [8, 61]. We start from our earlier result in (2.3) where we proved that the appearance of an image is bilinear in the 3D rigid motion

and illumination parameters, with the 3D shape determining the basis vectors of the space [90]. The variations of this analytically derived bilinear basis over identity and deformation are then learned using multilinear SVD [39], and they together form a *quadrilinear* space of illumination, pose, identity and deformation. The GAM can be visualized (see Figure 3.1) as a collection of locally linear tangent planes along the pose dimension, where each tangent plane represents 3D motion in a local region around each pose. Thus the GAM is able to model the local tangent space around a pose.

The major difference of GAMs with other methods for computing appearance manifolds and subspaces [17, 41, 46, 85, 87, 96] is that the object appearance space is derived using a combination of geometrical models and data analysis tools, while the previous approaches relied purely on data analysis. This significantly reduces the data collection requirements for computing such manifolds, makes analysis on these manifolds less dependent on the actual data used to learn them in the first place, and allows representations of appearances that were not included in the learning phase. We will provide some concrete numerical examples to justify these in the experimental section. *Thus our method combines the precision and generalizability of model-based approaches with the robustness provided by statistical learning methods to deviations from the model predictions.*

3.3 Method for Learning GAMs

We will start with the image appearance manifold representation with variations in pose and lighting we derived in (2.3). Then N-mode SVD, a multilinear generalization of SVD, is applied to learn the variation of this manifold due to changes of identity and object

deformations. We will show that the image appearance due to variations of illumination, pose, and deformation, is quadrilinear.

3.3.1 Analytically Derived Manifold for Motion and Illumination - The Geometrical Approach

We start from the results in (2.3) which showed that the images of a moving object can be approximated by a bilinear subspace of nine illumination coefficients and six motion variables. Let the pose of the object in the camera reference frame to be defined as $\mathbf{p} = (\mathbf{T^T}, \mathbf{\Omega^T})^\mathbf{T}$. Representing by $\Delta\mathbf{T}$ the translation of the centroid of the object, by $\Delta\mathbf{\Omega}$ the rotation about the centroid, and by $\mathbf{l} \in \mathbb{R}^{N_l}$ ($N_l \approx 9$ for Lambertian objects with attached shadow) the illumination coefficients in a spherical harmonics basis (see [8] for details), we showed in Section 2.2.3 that under small motion, the reflectance image at $t_2 = t_1 + \delta t$ can be expressed as

$$I_{t_2}(\mathbf{u}) = \sum_{i=1}^{N_l} l_{i|t_2} b_{i|t_2}(\mathbf{u}), \tag{3.1}$$

where

$$b_{i|t_2}(\mathbf{u}) = b_{i|t_1}(\mathbf{u}) + \mathbf{A}_{t_1}(\mathbf{u}, \mathbf{n})\Delta\mathbf{T} + \mathbf{B}_{t_1}(\mathbf{u}, \mathbf{n})\Delta\mathbf{\Omega}. \tag{3.2}$$

In the above equations, $\mathbf{u}$ represents the image point projected from the 3D surface with surface normal $\mathbf{n}$, and $b_{i|t_1}(\mathbf{u})$ are the original basis images before motion. $\mathbf{A}_{t_1}$ and $\mathbf{B}_{t_1}$ contain the structure and camera intrinsic parameters, and are functions of $\mathbf{u}$ and the 3D surface normal $\mathbf{n}$. For each pixel $\mathbf{u}$, both $\mathbf{A}_{t_1}$ and $\mathbf{B}_{t_1}$ are N_l by 3 matrices.

It will be useful for us to represent this result using tensor notation as

$$\hat{\mathcal{I}}_{t_2} = \left(\mathcal{B}_{t_1} + \mathcal{C}_{t_1} \times_2 \begin{pmatrix} \Delta\mathbf{T} \\ \Delta\boldsymbol{\Omega} \end{pmatrix} \right) \times_1 \mathbf{l}_{t_2}. \tag{3.3}$$

For an image of size $M \times N$, $\mathcal{C}_{t_1}$ is a tensor of size $N_l \times 6 \times M \times N$. For each pixel (p, q) in the image, $\mathcal{C}_{klpq|t_1} = [\ \mathbf{A}_{t_1}(\mathbf{u}, \mathbf{n}) \quad \mathbf{B}_{t_1}(\mathbf{u}, \mathbf{n})\ ]$ of size $N_l \times 6$, $\mathcal{B}_{t_1}$ is a sub-tensor of dimension $N_l \times 1 \times M \times N$, comprised of the basis images $b_{i|t_1}$, and $\mathcal{I}_{t_2}$ is a sub-tensor of dimension $1 \times 1 \times M \times N$, representing the image (see Chapter 2).

3.3.2 Identity and Deformation Manifold - The Statistical Learning Approach

The above bilinear space of 3D motion and illumination is derived by using the knowledge of the 3D model of the object (tensor $\mathcal{C}$ contains the surface normals). However, the 3D shape is a function of the identity of the object (e.g., the identity of a face or a particular model of a car) and possible non-rigid deformations. The model in (2.3) cannot handle these cases. The challenge now is to generalize the above analytical model so that it can be used to represent a wide variety of appearances within a class of objects. We achieve this by learning multilinear appearance models.

Main Approach: Rather than directly modeling the variation in the appearance images, *we will model the bilinear bases of motion and illumination derived analytically in Section 3.3.1, and then combine all these different variations to obtain a multilinear model of object appearance. This will allow us to retain information about the geometry of the object.*

Using $[\bullet]_v$ to denote the vectorization operation, we can vectorize $\mathcal{B}$ and $\mathcal{C}$ in (3.3),

and concatenate them, as

$$\mathbf{v} = \begin{bmatrix} [\mathcal{B}]_v \\ [\mathcal{C}]_v \end{bmatrix}. \tag{3.4}$$

Note that $\mathcal{B}$ and $\mathcal{C}$ can be obtained from the 3D model of the object. This $\mathbf{v}$ is the vectorized bilinear basis for one shape (i.e., one object) with dimension $I_v \times 1$, where $I_v = 7N_lMN$ (N_lMN for $\mathcal{B}$ and $6N_lMN$ for $\mathcal{C}$). Given the 3D shape of I_i objects with I_e different deformations, we can compute this vectorized bilinear basis $\mathbf{v}$ for every combination. For faces, these instances can be obtained from any 3D face modeling algorithm or by direct acquisition of 3D data. With the application to faces in mind, we will sometimes use the words deformation and expression interchangeably.

We use $\mathbf{v}_e^i$ to represent the vectorized bilinear basis of identity i with expression e. Let us rearrange them into a training data tensor $\mathcal{D}$ of size $I_i \times I_e \times I_v$ with the first dimension for identity, second dimension for expression (deformation) and the third dimension for the vectorized, analytically derived bilinear basis for each training sample. Applying the *N-Mode SVD* algorithm [39], the training data tensor can be decomposed as

$$\begin{aligned} \mathcal{D} &= \mathcal{Y} \times_1 U_i \times_2 U_e \times_3 U_v = \mathcal{Z} \times_1 U_i \times_2 U_e, \\ \text{where} \qquad & \mathcal{Z} = \mathcal{Y} \times_3 U_v. \end{aligned} \tag{3.5}$$

$\mathcal{Y}$ is known as the core tensor of size $N_i \times N_e \times N_v$, and N_i and N_e are the number of bases we use for the identity and expression. With a slight abuse of terminology, we will call $\mathcal{Z}$, which is decomposed only along the identity and expression dimension with size $N_i \times N_e \times I_v$, to be the core tensor. U_i and U_e, with sizes of $I_i \times N_i$ and $I_e \times N_e$, are the left matrices of the SVD of

$$\mathcal{D}_{(1)} = \begin{pmatrix} \mathbf{v}_1^{1\mathbf{T}} & \cdots & \mathbf{v}_{I_e}^{1\ \mathbf{T}} \\ & \cdots & \\ \mathbf{v}_1^{I_i\mathbf{T}} & \cdots & \mathbf{v}_{I_e}^{I_i\mathbf{T}} \end{pmatrix}$$

$$\text{and } \mathcal{D}_{(2)} = \begin{pmatrix} \mathbf{v}_1^{1\mathbf{T}} & \cdots & \mathbf{v}_1^{I_i\mathbf{T}} \\ & \cdots & \\ \mathbf{v}_{I_e}^{1\ \mathbf{T}} & \cdots & \mathbf{v}_{I_e}^{I_i\mathbf{T}} \end{pmatrix}, \tag{3.6}$$

where the subscripts of tensor $\mathcal{D}$ indicate the tensor unfolding operation along the first and second dimension (please refer to Appendix A for detail of the tensor unfolding operation). According to the *N-mode SVD algorithm* and equation (3.3.2), the core tensor $\mathcal{Z}$ can be expressed as

$$\mathcal{Z} = \mathcal{D} \times_1 U_i^{\mathbf{T}} \times_2 U_e^{\mathbf{T}}. \tag{3.7}$$

3.3.3 Lighting, Motion, Identity and Deformation Manifold - Unifying Geometrical and Statistical Approaches

The core tensor $\mathcal{Z}$ contains the basis of identity and expression (or deformation) for $\mathbf{v}$ as

$$\mathbf{v}_i^{eT} = \mathcal{Z} \times_1 \mathbf{c}_i^{\mathbf{T}} \times_2 \mathbf{c}_e^{\mathbf{T}}, \tag{3.8}$$

where $\mathbf{c}_i$ and $\mathbf{c}_e$ are the coefficient vectors encoding the identity and expression. As $\mathbf{v}_i^e$ are the vectorized, bilinear basis functions of the illumination and 3D motion, the core tensor $\mathcal{Z}$ is *quadrilinear* in illumination, motion, identity and expression. As an example, this core tensor $\mathcal{Z}$ can describe all the face images of identity $\mathbf{c}_i$ with expression $\mathbf{c}_e$ and motion ($\Delta\mathbf{T}$, $\Delta\boldsymbol{\Omega}$) under illumination $\mathbf{l}$.

Due to the small motion assumption (A1) made in the derivation of the analytical model of motion and illumination in Section 2.2.3, the core tensor $\mathcal{Z}$ can only represent the image of the object whose pose is close to the pose $\mathbf{p}$ under which the training samples of $\mathbf{v}$ are computed. To emphasize that $\mathcal{Z}$ is a function of pose $\mathbf{p}$, we denote it as $\mathcal{Z}_{\mathbf{p}}$ in the following derivation.

Since $\mathbf{v}$ is obtained by concatenating $[\mathcal{B}]_v$ and $[\mathcal{C}]_v$, $\mathcal{Z}_{\mathbf{p}}$ also contains two parts, $\mathcal{Z}_{\mathbf{p}}^{\mathcal{B}}$ with size $(N_i \times N_e \times N_l MN)$ and $\mathcal{Z}_{\mathbf{p}}^{\mathcal{C}}$ with size $(N_i \times N_e \times 6N_l MN)$. The first part encodes the variation of the image due to changes of identity, deformation and illumination at the pose $\mathbf{p}$, and the second part encodes the variation due to motion around $\mathbf{p}$, i.e., the tangent plane of the manifold along the motion direction. Rearranging the two sub-tensors according to the illumination and motion basis into sizes of $N_l \times 1 \times N_i \times N_e \times MN$ and $N_l \times 6 \times N_i \times N_e \times MN$ (this step is needed to undo the vectorization operation of equation (3.4)), we can represent the quadrilinear basis of illumination, 3D motion, identity, and deformation along the first, second, third and forth dimensions respectively.

The image with identity $\mathbf{c}_{i|t_2}$ and expression $\mathbf{c}_{e|t_2}$ after motion ($\Delta\mathbf{T}$,$\Delta\boldsymbol{\Omega}$) around pose $\mathbf{p}_{t_1}$ under illumination $\mathbf{l}_{t_2}$ can be obtained by

$$\begin{aligned} \mathcal{I}_{t_2} &= \mathcal{Z}_{\mathbf{p}_{t_1}}^{\mathcal{B}} \times_1 \mathbf{l}_{t_2} \times_3 \mathbf{c}_{i|t_2} \times_4 \mathbf{c}_{e|t_2} \\ &\quad + \mathcal{Z}_{\mathbf{p}_{t_1}}^{\mathcal{C}} \times_1 \mathbf{l}_{t_2} \times_2 \begin{pmatrix} \boldsymbol{\Delta}\mathbf{T} \\ \\ \Delta\boldsymbol{\Omega} \end{pmatrix} \times_3 \mathbf{c}_{i|t_2} \times_4 \mathbf{c}_{e|t_2}. \end{aligned} \tag{3.9}$$

Note that we did not need examples of the object at different lighting conditions to construct this manifold. Also, the appearance variation due to rigid motion around each pose was modeled without any training examples. These parts of the manifold

came from the analytical expressions in (3.3).

To represent the manifold at all the possible poses, we do not need such a tensor at every pose. Effects of 3D translation can be removed by centering and scale normalization, while in-plane rotation to a pre-defined pose can mitigate the effects of rotation about the z-axis. Thus, the image of object under arbitrary pose, $\mathbf{p}$, can always be described by the multilinear object representation at a pre-defined $(\mathbf{T}_x^{pd}, \mathbf{T}_y^{pd}, \mathbf{T}_z^{pd}, \mathbf{\Omega}_z^{pd})$, with only $\mathbf{\Omega}_x$ and $\mathbf{\Omega}_y$ depending upon the particular pose. Thus, the image manifold under any pose can be approximated by the collection of a few tangent planes on distinct $\mathbf{\Omega}_x^j$ and $\mathbf{\Omega}_y^j$, denoted as $\mathbf{p}_j$.

3.4 Experimental Results

3.4.1 Analysis of the GAM

As discussed above, the advantages of using the GAMs are (i) ease of construction due to the need for significantly less number of training images, (ii) ability to represent objects at all poses and lighting conditions from only a few examples during training, and (iii) accuracy and efficiency of tracking. We will now show results to justify these claims.

• **Constructing GAM of faces:** In the case of faces, we will need at least one image for every person. We then estimate the face model and compute the vectorized tensor $\mathbf{v}$ at a pre-defined collection of poses $\mathbf{p}_j$. For each expression, we will need at least one image per person. Thus for N_i people with N_e expressions, we need $N_i N_e$ images. In our experiments,

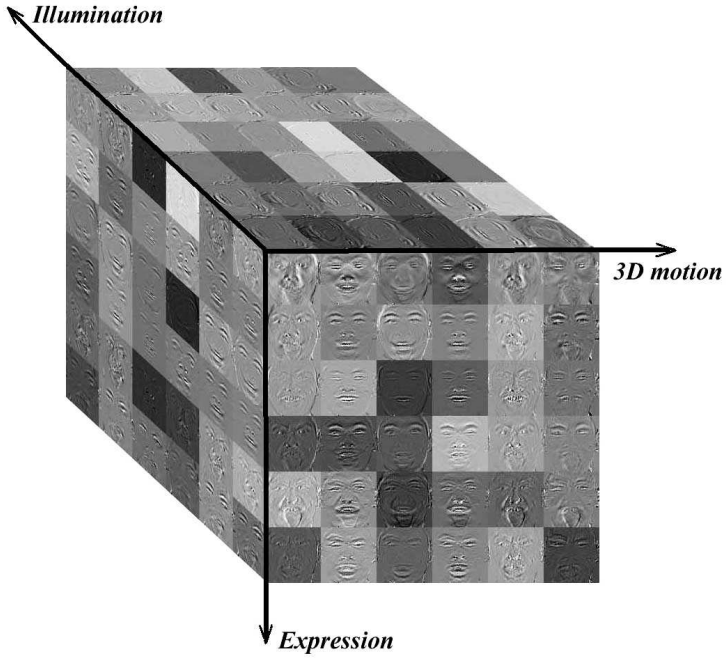


Figure 3.2: The basis images of the face GAM on illumination, expression, and the 3D motion around the frontal cardinal pose for a specific person.

$N_i = 100$ and $N_e = 7$ thus requiring 700 images for all the people and every expression. To compare with other methods modeling the appearance manifolds, we list the number of the example images needed for training in Table 3.1. Moreover, *the GAM can model the appearance space not only at these discrete poses, but also the manifold in a local region around each pose.* In our experiments, the pose collection $\mathbf{p}_j$ is chosen to be every 15° along the vertical rotational axis, and every 20° along the horizontal rotational axis. In Figure 3.2, we show some basis images of the face GAM along illumination, 3D motion, identity and expression dimensions. As we can show only 3 dimensions, identity is fixed to one particular person.

• **Constructing GAM of vehicles:** We now show an example of building a GAM for a non-face object. Assume we are interested in building the GAM for sedans. We need the training samples, $\mathbf{v}$ in (3.4), for different cars. The chief variation for different sedans is in their shape (cars usually have uniform texture which can be recovered separately from RGB space and is not considered here). The shape can be obtained by fitting a generic car model onto a few training images (from different views) for each model [43]. Thus, to build the GAM of sedan, we just need a few images (typically less than five) for different types of sedans (a few dozen at most), perform the fitting, compute the vectorized tensor $\mathbf{v}$ in (3.4) at a pre-defined collection of poses $\mathbf{p}_j$ and apply the method in Section 3.3. In contrast, purely learning-based methods based on [41, 85] would require hundreds, possibly thousands, of images at different poses, lighting conditions and car models.

Table 3.1: Comparison of the size of the training set needed for constructing face appearance models

AAM [14]	One image per person, but illumination and expression are not modeled. Pose variation is achieved through a shape normalization warping.
PAM [41]	300 images per person modeling only pose variation.
Multilinear model [85]	225 images per person modeling pose and illumination variations. No expression is modeled.
GAM	One image per person while modeling both pose and illumination variation. When modeling N_e kinds of expressions, N_e images per person needed.

3.5 Conclusions

In this chapter, we showed that it is possible to estimate low-dimensional manifolds that describe object appearance with a small number of training samples using a combi-

nation of analytically derived geometrical models and statistical data analysis. We derived a quadrilinear space of object appearance that can represent the effects of illumination, motion, identity and deformation, and termed it as the Geometry-Integrated Appearance Manifold. We showed specific examples on how to construct this manifold, and compared the size of the training set with other methods. The method for estimating pose and lighting parameters and the tracking result on real data will be presented in the next chapter.

Chapter 4

Efficient Parameter Estimation on GAM

4.1 Introduction

Per design, Geometry-Integrated Appearance Manifold can represent the image appearance variation due to the change of pose, illumination and deformation. Using this model, we can recover the pose, illumination and deformation parameters simultaneously. Although there are numerous methods for estimating motion and shape of an object from video sequences, and many of them can handle significant changes in the illumination conditions by *compensating* for the variations [23, 25, 32], there do not exist many methods that can *recover* the 3D motion *and* time-varying global illumination conditions from video sequences of moving objects. In this chapter, we propose an accurate and efficient method whereby the parameters of the illumination, pose and deformation are recovered in *con-*

tinuous time from video sequences, which was also presented in [93, 92]. The work has important applications in a number of areas, most importantly object recognition and inverse rendering.

4.1.1 Relation To Previous Work

A well-known approach for 2D motion estimation and registration in monocular sequences is Lucas-Kanade tracking [44]. Building upon this framework, a very efficient tracking algorithm was proposed in [25] by inverting the role of the target image and the template. However, their algorithm can only be applied to a restricted class of warps between the target and template (see [6] for details). A forward compositional algorithm was proposed in [74] by estimating an incremental warp for image alignment. Baker et al [6] proposed an inverse compositional (IC) algorithm for efficient implementation of the Lucas-Kanade algorithm to save computational cost in re-evaluation of the derivatives in each iteration. The inverse compositional algorithm was then used for efficiently fitting Active Appearance Models (AAMs) [46] and the well-known 3D Morphable Model (3DMM) [62] to face images under large pose variations. A dual inverse compositional algorithm was also proposed for dealing with both the geometric and photometric transformations in image registration when lighting varies [7].

None of the above estimate the lighting conditions in the images. An earlier version of 3DMM fitting [11] used a Phong illumination model, estimation of whose parameters in the presence of extended light sources can be difficult. The method in [16] dealt with point sources and did not consider the effect of attached shadows. Specular reflection was

taken into consideration in [24], but it dealt with tracking feature points. To handle cast shadows, a physical model incorporating the visible spectrum was introduced for removing the shadows in [21]. Based upon this theory, a shadow resistant image registration method was proposed using the Gauss-Newton method in [54]. Neither of them used an IC approach for motion and lighting estimation.

Our lighting estimation can account for extended lighting sources and attached shadows. Also, we estimate 3D motion, unlike 2D motion in [23, 25, 34, 72, 74]. The warping function in this chapter is different from [6, 62] as we explain in Section 4.3. For applications on faces, our approach can be combined with the 3DMM method. Since our inverse compositional approach estimates 3D motion, it allows us to perform the expensive computations only once every few frames (unlike once for every frame as in the image alignment approaches of [6]). Specifically, these computations are done only when there is a significant change of pose.

4.1.2 Contributions

The following are the major contributions of this chapter.

• We propose a novel 3D model-based warping function for estimating 3D motion and lighting from a video sequence. This involves a $2D \rightarrow 3D \rightarrow 2D$ transformation, which is different from the warping functions used in [6, 62]. This function can be used in future for developing other IC-based tracking algorithms to estimate 3D motion from image sequences.

• Due to this novel warping function, we are able to extend two-frame IC tracking methods

to multiple frames without any significant sacrifice in accuracy.

• We show that IC approaches can be used not only for estimating 3D motion, but also the time-varying lighting conditions in the scene, including the effects of attached shadows. Existing inverse compositional methods have focused on 2D motion or fitting a 3D model to an image.

• We rigorously prove the accuracy of the motion and lighting estimates from first principles, analyze the computational savings, and provide results on the numerical correctness of the estimates.

For simplicity of explanation, we will first consider a bilinear model of pose and illumination variation in (2.3). Then we will show the IC algorithm for estimating illumination, 3D pose and deformation parameters on GAM. The rest of the chapter is organized as follows. Section 4.2 presents the algorithm for learning the motion and illumination parameters from video using the bilinear model of motion and illumination we presented in Section 2.2.3. Derivation and analysis of the efficient inverse compositional estimation of motion and illumination algorithm is presented in section 4.3. Then, the IC algorithm for the parameter estimation on the GAM will be presented in section 4.4. In Section 4.5, comparison with controlled experiments as well as the real-data tracking results on the GAM are shown. Section 4.6 concludes the chapter.

4.2 Pose and Illumination Estimation Using Bilinear model of Motion and Illumination

In this section, we will briefly review the result described in Section 2.2.3 helping to lay the background and notation for the mathematical derivation. Let $\mathbf{p} = (\mathbf{T^T}, \mathbf{\Omega^T})^\mathbf{T}, \mathbf{p} \in \mathbb{R}^6$, denote the pose of the object. It was proved in Section 2.2.3 that if the motion of the object (defined as the translation of the object centroid $\Delta\mathbf{T} \in \mathbb{R}^3$ and the rotation vector $\Delta\mathbf{\Omega} \in \mathbb{R}^3$ about the centroid in the camera frame) from time t_1 to new time instance $t_2 = t_1 + \delta t$ is small, then upto a first order approximation, the reflectance image $I(x, y)$ at t_2 can be expressed in the form

$$\mathcal{I}_{t_2} = \left(\mathcal{B}_{t_1} + \mathcal{C}_{t_1} \times_2 \begin{pmatrix} \Delta\mathbf{T} \\ \\ \Delta\mathbf{\Omega} \end{pmatrix} \right) \times_1 \mathbf{l}_{t_2}, \tag{4.1}$$

where $\mathbf{l}_{t_2} \in \mathbb{R}^{N_l}$. Thus, the image at t_2 can be represented using the parameters computed at t_1. For each pixel (p, q) in the image, $\mathcal{C}_{klpq|t_1} \triangleq [\ \mathbf{A}_{t_1} \quad \mathbf{B}_{t_1}\ ]$ of size $N_l \times 6$. Thus for an image of size $M \times N$, $\mathcal{C}$ is $N_l \times 6 \times M \times N$, $\mathcal{B}_{t_1}$ is a sub-tensor of dimension $N_l \times 1 \times M \times N$, comprising the basis images $b_{i|t_1}(\mathbf{u})$, and $\mathcal{I}_{t_2}$ is a sub-tensor of dimension $1 \times 1 \times M \times N$, representing the image.

Equation (4.1) provides us an expression relating the reflectance image $\mathcal{I}_{t_2}$ with the illumination coefficients $\mathbf{l}_{t_2}$ and motion variables $\Delta\mathbf{T}, \Delta\mathbf{\Omega}$. Letting $\mathbf{m} \triangleq \mathbf{\Delta p} = [\mathbf{\Delta T^T}, \mathbf{\Delta \Omega^T}]^\mathbf{T}$, we can estimate 3D motion and illumination as

$$(\hat{\mathbf{l}}_{t_2}, \hat{\mathbf{m}}_{t_2}) = \arg \min_{\mathbf{l_{t_2}}, \mathbf{m_{t_2}}} \|\mathcal{I}_{t_2} - (\mathcal{B}_{t_1} + \mathcal{C}_{t_1} \times_2 \mathbf{m}_{t_2}) \times_1 \mathbf{l}_{t_2}\|^2 + \alpha ||\mathbf{m}_{t_2}||^2 \tag{4.2}$$

where $\hat{x}$ denotes an estimate of x. Since the motion between consecutive frames is small, but

illumination can change suddenly, we add a regularization term to the above cost function with the form of $\alpha||\mathbf{m}_{t_2}||^2$.

Since the image $\mathcal{I}_{t_2}$ lies approximately in a bilinear space of illumination and motion variables (ignoring the regularization term for now), such a minimization problem can be achieved by alternately estimating the motion and illumination parameters. Assuming that we have tracked the sequence upto some frame at t_1 for which we can estimate the motion (hence, pose) and illumination, we calculate the basis images $\mathcal{B}_{t_1}$ and $\mathcal{C}_{t_1}$ at the current pose. Unfolding $\mathcal{B}_{t_1}$ and the image $\mathcal{I}_{t_2}$ along the first dimension, which is the illumination dimension (see Appendix A for the definition of unfolding), the illumination can be estimated as

$$\hat{\mathbf{l}}_{t_2} = (\mathcal{B}_{t_1(1)}\mathcal{B}_{t_1(1)}^T)^{-1}\mathcal{B}_{t_1(1)}\mathcal{I}_{t_2(1)}^T. \tag{4.3}$$

Keeping the illumination coefficients fixed, the bilinear space in equation (4.1) becomes a linear subspace, i.e.,

$$\mathcal{I}_{t_2} = \mathcal{B}_{t_1} \times_1 \mathbf{l}_{t_2} + \mathcal{G} \times_2 \mathbf{m}_{t_2}, \text{ where } \mathcal{G} = \mathcal{C}_{t_1} \times_1 \mathbf{l}_{t_2}, \tag{4.4}$$

and motion can be estimated as

$$\hat{\mathbf{m}}_{t_2} = \left(\mathcal{G}_{(2)}\mathcal{G}_{(2)}^T + \alpha\mathbf{I}\right)^{-1}\mathcal{G}_{(2)}(\mathcal{I}_{t_2} - \mathcal{B}_{t_1} \times_1 \mathbf{l}_{t_2})_{(2)}^T, \tag{4.5}$$

where $\mathbf{I}$ is an identity matrix of dimension 6×6. When we apply the Levenberg-Marquardt method [79] to minimize the difference between the input frame and the rendered frame in (4.1), we will have exactly the same expression as in (4.5) with α being the corresponding damping factor. When the regularization term is ignored, the result becomes that of the Gauss-Newton method.

4.3 Inverse Compositional Tracking

The method described in Section 4.2 requires iteration between equations (4.3) and (4.5). In each iteration, as pose is updated, the tensors $\mathcal{B}_t$ and $\mathcal{G}_t$ need to be recomputed, which is very expensive computationally (since they require finding the point of intersection of the ray through each point with the 3D surface). In this section, we will derive an inverse compositional approach for efficient and accurate estimation of 3D motion and illumination. We start by showing that (4.2) is equivalent to a Lucas-Kanade algorithm for estimation of 3D motion and lighting which leads to the inverse compositional approach. Finally, we show how to extend it to a sequence of frames. In keeping the standard notation used in tracking, we assume $\delta t = 1$, and consider two frames at t and $t-1$.

4.3.1 Lucas-Kanade Estimation of 3D Motion and Lighting

Let us initially start with the condition that illumination does not change between two frames. We will then consider the varying illumination condition. Also, we ignore the regularization term in (4.2), which can be easily added back later. The image synthesis process can be considered as a rendering function of the object at pose $\mathbf{p}$ in the camera frame to the pixel coordinates $\mathbf{v}$ in the image plane as $f(\mathbf{v}, \mathbf{p}_t)$. Using the bilinear model described above, it can be implemented with (4.4). Given an input image $I_t(\mathbf{v})$, we want to align the synthesized image with it so as to obtain

$$\hat{\mathbf{p}}_t = \arg\min_{\mathbf{p}_t} \sum_{\mathbf{v}} \left(f(\mathbf{v}, \mathbf{p}_t) - I_t(\mathbf{v}) \right)^2, \tag{4.6}$$

where $\hat{\mathbf{p}}_t$ denotes the estimated pose for this input image $I_t(\mathbf{v})$. This is the cost function of Lucas-Kanade tracking in [6] modified for 3D motion estimation.

Let us now consider the problem of estimating the pose change, $\mathbf{m}_t = \triangle\mathbf{p}_t$, between two consecutive frames, $I_t(\mathbf{v})$ and $I_{t-1}(\mathbf{v})$ as

$$\hat{\mathbf{m}}_t = \arg\min_{\mathbf{m}_t} \sum_{\mathbf{v}} \left(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \mathbf{m}_t) - I_t(\mathbf{v})\right)^2, \text{ and } \hat{\mathbf{p}}_t = \hat{\mathbf{p}}_{t-1} + \hat{\mathbf{m}}_t. \tag{4.7}$$

The optimization of the above equation can be achieved by assuming a current estimate of $\hat{\mathbf{m}}_t$ is known and iteratively solving for increments $\triangle\mathbf{m}$ ($\triangle\mathbf{m}$ are the increments between two iterations, where multiple iterations will be needed to get $\mathbf{m}_t$) such that

$$\sum_{\mathbf{v}} \left(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \mathbf{m}_t + \triangle\mathbf{m}) - I_t(\mathbf{v})\right)^2 \tag{4.8}$$

is minimized. Applying the first order Taylor expansion on (4.8), we can rewrite it as

$$\sum_{\mathbf{v}} \left(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \mathbf{m}_t) + \frac{\partial f(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}}^{\mathbf{T}} |_{\mathbf{p}=\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t} \triangle\mathbf{m} - I_t(\mathbf{v}) \right)^2. \tag{4.9}$$

Recall that equation (4.4) linearizes the image intensity I with respect to the motion parameter $\mathbf{m}$ when illumination parameter $\mathbf{l}$ is fixed. Thus, from equation (4.4), we have

$$\begin{aligned} \frac{\partial f(\mathbf{v}, \mathbf{p}(\mathbf{m_t}))}{\mathbf{m}_t}|_{\mathbf{p}(\mathbf{m_t})\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t} &= \frac{\partial f(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}} \frac{\partial \mathbf{p}(\mathbf{m_t})}{\partial \mathbf{m}_t}|_{\mathbf{p}(\mathbf{m_t})=\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t} = \frac{\partial f(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}}|_{\mathbf{p}=\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t} \\ &= \mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}+\mathbf{m_t}}, \end{aligned} \tag{4.10}$$

where $\mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}+\mathbf{m_t}}$ denotes the components of $\mathcal{G}$ at the pixel coordinate $\mathbf{v}$ computed at the pose $\hat{\mathbf{p}}_{t-1} + \mathbf{m}_t$, and $\mathbf{p}(\mathbf{m_t})$ is used to clearly show that pose $\mathbf{p}$ depends on the $\mathbf{m}_t$ (see (4.7)). Physically, $\mathcal{G}_{\mathbf{v}}$ contains the information of the object structure and the camera

model. Since $\mathcal{C}$ is a tensor of size $N_l \times 6 \times M \times N$ and $\mathcal{G} = \mathcal{C} \times_1 \mathbf{l}$, therefore $\mathcal{G}$ is of size $1 \times 6 \times M \times N$. At a specific pixel $\mathbf{v}$, $\mathcal{G}_{\mathbf{v}}$ degenerates to a 6×1 vector. Substituting (4.10) into (4.9), taking the derivative with respect to $\triangle\mathbf{m}$ and setting it to be zero, we get

$$\sum_{\mathbf{v}} \big(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \mathbf{m}_t) + \mathcal{G}^{\mathbf{T}}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} \triangle\mathbf{m} - I_t(\mathbf{v})\big)\mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} = 0. \tag{4.11}$$

Then solving for $\triangle\mathbf{m}$, we have

$$\begin{aligned} \triangle\mathbf{m} &= \mathbf{H} \sum_{\mathbf{v}} \mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} \big(I_t(\mathbf{v}) - f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \mathbf{m}_t)\big), \\ \text{where } \mathbf{H} &= \left[\sum_{\mathbf{v}} (\mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} \mathcal{G}^{\mathbf{T}}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}})\right]^{-1}. \end{aligned} \tag{4.12}$$

Let us now reintroduce the illumination variation which was ignored for simplicity of explanation. The image synthesis function f can be replaced with the analytical expression in (4.1). Although $\mathcal{G}_{\bullet|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}}$ varies with the illumination condition $\mathbf{l}_t$ according to (4.4), $\mathcal{C}_{\bullet|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}}$ is not a function of $\mathbf{l}_t$. Thus, given $\mathbf{l}_t$, (4.12) becomes:

$$\begin{aligned} \triangle\mathbf{m} &= \mathbf{H} \sum_{\mathbf{v}} (\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} \times_1 \mathbf{l}_t) \big(I_t(\mathbf{v}) - \mathcal{B}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} \times_1 \mathbf{l}_t\big), \\ \text{where} \quad \mathbf{H} &= \left[\sum_{\mathbf{v}} (\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} \times_1 \mathbf{l}_t)(\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}} \times_1 \mathbf{l}_t)^{\mathbf{T}}\right]^{-1}, \end{aligned} \tag{4.13}$$

which is effectively equation (4.5) when α is zero. Once motion is known, lighting can be easily estimated by computing $\mathcal{B}$ in (4.3). Thus, the direct method we described in Section 4.2 is equivalent to Lucas-Kanade 3D tracking and illumination estimation algorithm.

4.3.2 Inverse Compositional Estimation of 3D Motion and Lighting

In the above method, the motion $\mathbf{m}$ is updated in each iteration and $\mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t}-\mathbf{1}}+\mathbf{m}_\mathbf{t}}$ needs to be reevaluated. This requires exhaustively visiting every intersection point of each

ray with the surface and computing the derivatives, which extracts a huge computational cost. Thus, it is inefficient to use $\mathcal{G}_{\bullet|\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t}$ in each step of motion and lighting estimation.

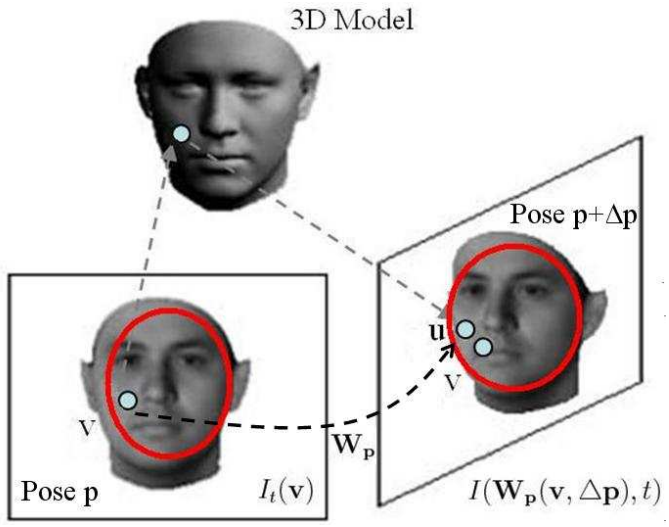


Figure 4.1: Illustration of the warping function $\mathbf{W}$. A point $\mathbf{v}$ in image plane is projected onto the surface of the 3D object model. After the pose transformation with $\triangle\mathbf{p}$, the point on the surface is back projected onto the image plane at a new point $\mathbf{u}$. The warping function maps from $\mathbf{v} \in \mathbb{R}^2$ to $\mathbf{u} \in \mathbb{R}^2$. The red ellipses show the common part in both frames that the warping function $\mathbf{W}$ is defined upon.

Let us now introduce a warp operator $\mathbf{W} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ such that, if we denote the pose of $I_t(\mathbf{v})$ as $\mathbf{p}$, the pose of $I_t(\mathbf{W_p}(\mathbf{v}, \triangle\mathbf{p}))$ is $\mathbf{p} + \triangle\mathbf{p}$. Specifically, a 2D point on the image plane is projected onto the 3D object surface. Then we transform the pose of the object surface by $\triangle\mathbf{p}$ and back project the point from the 3D surface onto the image plane. Thus, $\mathbf{W}$ represents the displacement in the image plane due to a pose transformation of the 3D model. Note that this warping involves a 3D pose transformation (unlike [6]). In [62], the warping was from a point on the 3D surface to the image plane, and was used for fitting a 3D model to an image. We propose a new warping function for the inverse compositional estimation of 3D rigid motion and illumination in video sequence, which is

not addressed in [6] or [62].

Using this warp operator, for any frame $I_t(\mathbf{v})$, the cost function (4.7) can be written as

$$\hat{\mathbf{m}}_t = \arg\min_{\mathbf{m}_t} \sum_{\mathbf{v}} \big(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t))\big)^2. \tag{4.14}$$

Rewriting the cost function (4.14) in the inverse compositional framework [6], we consider minimizing

$$\arg\min_{\triangle\mathbf{m}} \sum_{\mathbf{v}} \Big(f(\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t}-1}}(\mathbf{v}, \triangle\mathbf{m}), \hat{\mathbf{p}}_{t-1}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t))\Big)^2 \tag{4.15}$$

with the update rule

$$\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t) \leftarrow \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t) \circ \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, \triangle\mathbf{m})^{-1}. \tag{4.16}$$

We will first derive the solution to (4.15), then we will prove its equivalence to (4.14) in Section 4.3.3. The compositional operator $\circ$ in (4.16) means the second warp is composed into the first warp, i.e., $\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m_t}) \equiv \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, \triangle\mathbf{m})^{-1}, -\mathbf{m_t})$. The inverse of the warp $\mathbf{W}$ is defined to be the $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ mapping such that if we denote the pose of $I_t(\mathbf{v})$ as $\mathbf{p}$, the pose of $I_t(\mathbf{W_p}(\mathbf{W_p}(\mathbf{v}, \triangle\mathbf{p}), \triangle\mathbf{p})^{-1})$ is $\mathbf{p}$ itself. As the warp $\mathbf{W_p}(\mathbf{v}, \triangle\mathbf{p})$ transforms the pose from $\mathbf{p}$ to $\mathbf{p} + \triangle\mathbf{p}$, the inverse $\mathbf{W_p}(\mathbf{v}, \triangle\mathbf{p})^{-1}$ should transform the pose from $\mathbf{p} + \triangle\mathbf{p}$ to $\mathbf{p}$, i.e. $\mathbf{W_p}(\mathbf{v}, \triangle\mathbf{p})^{-1} = \mathbf{W}_{\mathbf{p}+\triangle\mathbf{p}}(\mathbf{v}, -\triangle\mathbf{p})$. Thus $\{\mathbf{W_p}\}$ is a group.

According to the definition of $\mathbf{W}$, we can approximate $f(\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t}-1}}(\mathbf{v}, \triangle\mathbf{m}), \hat{\mathbf{p}}_{t-1})$ in (4.15) with $f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \triangle\mathbf{m})$. This is because $f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \triangle\mathbf{m})$ is the image synthesized at $\hat{\mathbf{p}}_{t-1} + \triangle\mathbf{m}$, while $f(\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t}-1}}(\mathbf{v}, \triangle\mathbf{m}), \hat{\mathbf{p}}_{t-1})$ is the image synthesized at $\hat{\mathbf{p}}_{t-1}$ followed with the warp of the pose increments $\triangle\mathbf{m}$. Although illumination is rotated by

$\Delta\mathbf{m}$ in $f(\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t-1}}}(\mathbf{v}, \triangle\mathbf{m}), \hat{\mathbf{p}}_{t-1})$, for Lambertian objects it is not difficult to show that $f(\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t-1}}}(\mathbf{v}, \triangle\mathbf{m}), \hat{\mathbf{p}}_{t-1}) - f(\mathbf{v}, \hat{\mathbf{p}}_{t-1} + \triangle\mathbf{m}) \sim O(\Delta\mathbf{m}) = o(\Delta\hat{\mathbf{p}}_{t-1})$. Neglecting this amounts is neglecting second order pose variations, which is the same approximation as the one used for the proof of the IC algorithm in Section 4.3.3. Thus this substitution is valid for our case. Applying the first order Taylor expansion on it, we have

$$\sum_{\mathbf{v}} \left(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1}) + \frac{\partial f(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}}|_{\mathbf{p}=\hat{\mathbf{p}}_{t-1}} \triangle\mathbf{m} - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t)) \right)^2. \tag{4.17}$$

Taking the derivative of (4.17) with respect to $\triangle\mathbf{m}$ and setting it to be zero, we have

$$\sum_{\mathbf{v}} \left(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1}) + \mathcal{G}^{\mathbf{T}}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \triangle\mathbf{m} - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t)) \right) \mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} = 0. \tag{4.18}$$

Solving for $\triangle\mathbf{m}$, we get:

$$\triangle\mathbf{m} = \mathbf{H_{IC}} \sum_{\mathbf{v}} \mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \left(I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t)) - f(\mathbf{v}, \hat{\mathbf{p}}_{t-1}) \right),$$

$$\text{where } \mathbf{H_{IC}} = \left[\sum_{\mathbf{v}} \mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \mathcal{G}^{\mathbf{T}}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \right]^{-1}. \tag{4.19}$$

Comparing with equation (4.12), the derivative $\mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}}$ and Hessian $\mathbf{H_{IC}}$ in (4.19) do not depend upon the updating variable $\mathbf{m}_t$, which is moved into the warp operator $\mathbf{W}$. The computational complexity of $\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t-1}}}(\mathbf{v}, -\mathbf{m_t})$ will be significantly lower than that of re-computing $\mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}+\mathbf{m_t}}$ and Hessian $\mathbf{H}$ in every iteration (see Section 4.3.6 for details on the computational cost).

Reintroducing the illumination variation and following the same derivation as

(4.13), we have

$$\begin{aligned} \triangle \mathbf{m} &= \mathbf{H_{IC}} \sum_{\mathbf{v}} (\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \times_1 \mathbf{l}_t) \left(I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m}_t)) - \mathcal{B}_{\mathbf{v}|\hat{\mathbf{p}}_{t-1}} \times_1 \mathbf{l}_t \right), \\ \text{where} \qquad \mathbf{H_{IC}} &= \left[\sum_{\mathbf{v}} (\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \times_1 \mathbf{l}_t)(\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \times_1 \mathbf{l}_t)^{\mathbf{T}} \right]^{-1}. \end{aligned} \tag{4.20}$$

4.3.3 Proof of the Convergence of the IC Estimation Algorithm

Using the above update rule, we will now show the equivalence of (4.15) to (4.14), which is equivalent to the cost function (4.7) in the Lucas-Kanade 3D tracking method.

Considering (4.15), the continuous version of which can be written as

$$\int_V (f(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(v, \triangle \mathbf{m}), \hat{\mathbf{p}}_{t-1}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(v, -\mathbf{m}_t)))^2 dv, \tag{4.21}$$

where V is the collection of all the pixels within the image at the pose $\hat{\mathbf{p}}_{t-1}$. Let $u \triangleq \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(v, \triangle \mathbf{m})$, thus $v = \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(u, \triangle \mathbf{m})^{-1} = \mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(u, -\triangle \mathbf{m})$. Plugging it into (4.21), we have

$$\int_U (f(u, \hat{\mathbf{p}}_{t-1}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(u, -\triangle \mathbf{m}), -\mathbf{m}_t)))^2 |\frac{d\mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(u, -\triangle \mathbf{m})}{du}| du \tag{4.22}$$

Note that with $\mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(u, 0) = u$, it follows that

$$\frac{d\mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(u, -\triangle \mathbf{m})}{du} = 1 + O(\triangle \mathbf{m}) = 1 + o(\mathbf{m}_t) = 1 + o(\triangle \hat{\mathbf{p}}_{t-1}). \tag{4.23}$$

Recall that $u = \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(v, \triangle \mathbf{m})$, i.e., U is the image of V after warping with $\mathbf{W}$. Since V is the collection of all the pixels within the image at pose $\hat{\mathbf{p}}_{t-1}$, U is the collection of all the pixels within the image at pose $\hat{\mathbf{p}}_{t-1} + \triangle \mathbf{m}$. For a video sequence, the motion $\mathbf{m}$ between the consecutive frames is usually small, thus the increments $\triangle \mathbf{m}$ should be even smaller.

With such small increments, the change of the image region should be small, i.e.

$$U = V + O(\triangle \mathbf{m}) = V + o(\mathbf{m}_t) = V + o(\triangle \hat{\mathbf{p}}_{t-1}). \tag{4.24}$$

Thus U and V differ only in the second order pose variation terms. Also, in the warp composition $\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(\mathbf{u}, -\triangle \mathbf{m}), -\mathbf{m_t})$, the inner warp transforms the pose of the object from $\hat{\mathbf{p}}_{t-1} + \triangle \mathbf{m}$ to $\hat{\mathbf{p}}_{t-1}$, while the outer warp transforms the pose from $\hat{\mathbf{p}}_{t-1}$ to $\hat{\mathbf{p}}_{t-1} - \mathbf{m}_t$. Thus, it can be simplified as $\mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(u, -\mathbf{m_t} - \triangle \mathbf{m})$. Neglecting the second order variation of the pose with respect to $\hat{\mathbf{p}}_{t-1}$, i.e., neglecting $\triangle \mathbf{m}$ w.r.t. $\hat{\mathbf{p}}_{t-1}$, but not w.r.t. $\mathbf{m}$, we get

$$\begin{aligned} \mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle \mathbf{m}}(u, -\mathbf{m}_t - \triangle \mathbf{m})) &\approx \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(u, -\mathbf{m}_t - \triangle \mathbf{m})) + o(\mathbf{m}_t) \\ &= \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(u, -\mathbf{m}_t - \triangle \mathbf{m})) + o(\triangle \hat{\mathbf{p}}_{t-1}). \end{aligned} \tag{4.25}$$

To achieve the above derivation, consider the warp $\mathbf{W}_{\mathbf{p}+\xi_1}(u, \xi_2)$, where $\xi_\mathbf{1}$ and $\xi_\mathbf{2}$ are both small w.r.t. $\mathbf{p}$. Let $\xi_\mathbf{1} = \varpi_\mathbf{1} \triangle \theta_\mathbf{1}$ and $\xi_\mathbf{2} = \varpi_\mathbf{2} \triangle \theta_\mathbf{2}$, where $\varpi_\mathbf{1}$ and $\varpi_\mathbf{2}$ are unit vectors. $\mathbf{x}$ is the 3D coordinate of a vertex on the 3D model. Using an orthographic or weak perspective camera model, the first dimension of the warp can be expressed as $(e^{\xi_\mathbf{1}} e^{\mathbf{p}} \mathbf{x})^{(1)} - (e^{\xi_\mathbf{2}} e^{\xi_\mathbf{1}} e^{\mathbf{p}} \mathbf{x})^{(1)} \approx ((I + \tilde{\varpi_\mathbf{1}} \sin\theta_\mathbf{1}) e^{\mathbf{p}} \mathbf{x})^{(1)} - ((I + \tilde{\varpi_\mathbf{2}} \sin\theta_\mathbf{2})(I + \tilde{\varpi_\mathbf{1}} \sin\theta_\mathbf{1}) e^{\mathbf{p}} \mathbf{x})^{(1)} = -(\tilde{\varpi_\mathbf{2}} \sin\theta_\mathbf{2} e^{\mathbf{p}} \mathbf{x})^{(1)} + o(\xi_\mathbf{1}, \xi_\mathbf{2})$, where $\tilde{\varpi}$ denotes the skew symmetric matrix with entries $\begin{pmatrix} 0 & -\varpi^{(3)} & \varpi^{(2)} \\ \varpi^{(3)} & 0 & -\varpi^{(1)} \\ -\varpi^{(2)} & \varpi^{(1)} & 0 \end{pmatrix}$, and the superscript $^{(1)}$ indicates the first dimension of the vector. Similar operations can be applied on the second dimension of warp. Thus, when both $\xi_\mathbf{1}$ and $\xi_\mathbf{2}$ are small terms w.r.t. $\mathbf{p}$, $\mathbf{W}_{\mathbf{p}+\xi_\mathbf{1}}(\mathbf{u}, \xi_\mathbf{2}) \approx \mathbf{W}_\mathbf{p}(\mathbf{u}, \xi_\mathbf{2})$.

Consequently, using (4.23), (4.24) and (4.25), and neglecting the second order pose variations, (4.22) can be approximated with

$$\int_V (f(v, \hat{\mathbf{p}}_{t-1}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(v, -\mathbf{m}_t - \triangle\mathbf{m})))^2 dv. \tag{4.26}$$

Note that this assumption of ignoring the second order pose variations is similar to the assumption in [6] of neglecting the second order variation in the parameter set.

Rewriting (4.26) in the discrete format, we have

$$\arg\min_{\triangle\mathbf{m}} \sum_{\mathbf{v}} \left(f(\mathbf{v}, \hat{\mathbf{p}}_{t-1}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\mathbf{m_t} - \triangle\mathbf{m}))\right)^2, \tag{4.27}$$

which is the solution strategy for minimizing (4.14) using the additive update rule $\mathbf{m}_t \leftarrow \mathbf{m}_t + \triangle\mathbf{m}$. Thus the cost functions (4.14) and (4.15) are equivalent, and the inverse compositional update rule can be approximated with the additive rule.

4.3.4 Inverse Compositional Estimation Over A Sequence of Frames

The computational complexity in the above derivation is reduced by pre-computing the derivative $\mathcal{G}$ and Hessian $\mathbf{H_{IC}}$ for reuse in each iteration. For the new input frame at time t, although $\mathcal{G}_{\bullet|\hat{\mathbf{p}}_t}$ would be close to $\mathcal{G}_{\bullet|\hat{\mathbf{p}}_{t-1}}$, it still needs to be recomputed. To further save computation complexity in the video sequence context, we can apply a similar idea by choosing a cardinal pose $\mathbf{p_c}$, pre-compute the derivatives $\mathcal{G}_{\mathbf{v}|\mathbf{p_c}}$ and $\mathbf{H_{IC|p_c}}$, and then reuse them for consequent frames.

Let us consider a sequence of frames $I(\bullet, 1), \ldots I(\bullet, t), \ldots I(\bullet, N)$. Without loss of generality, let us assume that the cardinal pose, $\mathbf{p_c}$, is at frame $I(\bullet, 1)$, i.e. $\mathbf{p_c} = \hat{\mathbf{p}}_\mathbf{1}$. Assume we already know the estimated motion upto time instance $t-1$, $\hat{\mathbf{m}}_\mathbf{1}, \ldots \hat{\mathbf{m}}_\mathbf{t-1}$. For

the input frame $I_t(\mathbf{v})$, we use the pose transformation operator $\mathbf{W}$ to normalize the pose to the cardinal pose based on $\hat{\mathbf{p}}_\mathbf{1}, \ldots \hat{\mathbf{p}}_{\mathbf{t-1}}$, i.e.,

$$\hat{\mathbf{m}}_t = \arg\min_{\mathbf{m}_t} \sum_{\mathbf{v}} \left(f(\mathbf{v}, \mathbf{p_c}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t)) \right)^2. \tag{4.28}$$

Rewriting the cost function (4.28) in the inverse compositional framework, we consider minimizing

$$\arg\min_{\triangle\mathbf{m}} \sum_{\mathbf{v}} \left(f(\mathbf{W}_{\mathbf{p_c}}(\mathbf{v}, \triangle\mathbf{m}), \mathbf{p_c}) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t)) \right)^2 \tag{4.29}$$

with the update rule

$$\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, (\mathbf{p_c} - \hat{\mathbf{p}}_{\mathbf{t-1}}) - \mathbf{m_t}) \leftarrow \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, (\mathbf{p_c} - \hat{\mathbf{p}}_{\mathbf{t-1}}) - \mathbf{m_t}) \circ \mathbf{W}_{\mathbf{p_c}}(\mathbf{v}, \triangle\mathbf{m})^{-1}. \tag{4.30}$$

Note that (4.29) is similar to (4.15), except that the warping for f is computed at the cardinal pose. Following the derivation of equations (4.17) - (4.20) and reintroducing the illumination variation, we have

$$\begin{aligned} \triangle\mathbf{m} &= \mathbf{H_{IC}} \sum_{\mathbf{v}} (\mathcal{C}_{\mathbf{v}|\mathbf{p_c}} \times_1 \mathbf{l}_t) \left(I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t)) - \mathcal{B}_{\mathbf{v}|\mathbf{p_c}} \times_1 \mathbf{l}_t \right), \\ \text{where} \qquad \mathbf{H_{IC}} &= \left[\sum_{\mathbf{v}} (\mathcal{C}_{\mathbf{v}|\mathbf{p_c}} \times_1 \mathbf{l}_t)(\mathcal{C}_{\mathbf{v}|\mathbf{p_c}} \times_1 \mathbf{l}_t)^\mathbf{T} \right]^{-1}. \end{aligned} \tag{4.31}$$

The proof of (4.31) can be done in a way similar to that of Section 4.3.3. Rewriting (4.29) in continuous domain and substituting $u \triangleq \mathbf{W}_{\mathbf{p_c}}(v, \triangle\mathbf{m})$ (conversely, $v = \mathbf{W}_{\mathbf{p_c}}(u, \triangle\mathbf{m})^{-1} = \mathbf{W}_{\mathbf{p_c}+\triangle\mathbf{m}}(u, -\triangle\mathbf{m})$),

$$\int_U (f(u, \mathbf{p}_c) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{W}_{\mathbf{p_c}+\triangle\mathbf{m}}(u, -\triangle\mathbf{m}), (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t)))^2 | \frac{d\mathbf{W}_{\mathbf{p_c}+\triangle\mathbf{m}}(u, -\triangle\mathbf{m})}{du} | du. \tag{4.32}$$

Assuming that pose $\hat{\mathbf{p}}_{t-1}$ does not deviate from $\mathbf{p_c}$ too much, and from Section 4.3.3 we have

$$\begin{aligned}
& \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{W}_{\mathbf{p_c}+\triangle\mathbf{m}}(u, -\triangle\mathbf{m}), (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t) \\
\approx\ & \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle m}(u, -\triangle\mathbf{m}), (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t) \\
=\ & \mathbf{W}_{\hat{\mathbf{p}}_{t-1}+\triangle\mathbf{m}}(u, (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t - \triangle\mathbf{m}) \\
\approx\ & \mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(u, (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t - \triangle m). \qquad (4.33)
\end{aligned}$$

Using the same reasoning as in (4.23)-(4.24), and under the assumption of neglecting second and higher order pose variations, (4.32) can be approximated as

$$\int_V (f(v, \mathbf{p}_c) - I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(v, (\mathbf{p}_c - \hat{\mathbf{p}}_{t-1}) - \mathbf{m}_t - \triangle\mathbf{m})))^2 dv, \qquad (4.34)$$

which is equivalent to (4.28) with the additive update rule.

In a video sequence, $\mathbf{p}_c - \hat{\mathbf{p}}_{t-1}$ might become large as t increases. This invalidates the assumption used in deriving (4.32). Thus, the cardinal pose needs to be changed within a long sequence. In our experiments, we found that this reinitialization was needed for every $15^\circ - 20^\circ$. The physical interpretation of this is that the visibility of a significant portion of the object will change due to the difference between $\mathbf{p_c}$ and $\hat{\mathbf{p}}_{t-1}$, and thus $\mathbf{W}$ will no longer be reliable.

4.3.5 Overall Algorithm

Consider a sequence of image frames I_t, $t = 0, ..., N-1$.

Initialization: Take the first frame of the video sequence, register the 3D model onto this

frame and map the texture onto the 3D model. Take this pose as cardinal pose $\mathbf{p_c}$. Pre-compute the $\mathcal{C}_{\bullet|\mathbf{p_c}}$ and $\mathcal{B}_{\bullet|p_c}$ at this pose. Assume that we know the pose and illumination estimates for frame $t-1$, i.e., $\hat{\mathbf{p}}_{t-1}$ and $\hat{\mathbf{l}}_{t-1}$.

• Step 1. For the new input frame $I_t(\mathbf{v})$, apply the pose transformation operator to get the pose normalized version of the frame $I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, \mathbf{p_c} - \hat{\mathbf{p}}_{t-1}))$. Let $\hat{\mathbf{l}}_t = \hat{\mathbf{l}}_{t-1}$, and $\hat{\mathbf{m}}_t = 0$.

• Step 2. Compute the increments of motion $\triangle\mathbf{m}$ using (4.31), and update the motion $\hat{\mathbf{m}}_t \leftarrow \hat{\mathbf{m}}_t + \triangle\mathbf{m}$.

• Step 3. Use $\hat{\mathbf{m}}_t$ to update the pose normalized image $I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, \mathbf{p_c} - \hat{\mathbf{p}}_{t-1} - \hat{\mathbf{m}}_t))$.

• Step 4. Use pre-computed $\mathcal{B}_{\bullet|p_c}$ and equation (4.3) to estimate the illumination vector $\hat{\mathbf{l}}_t$ of the updated pose normalized image $I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, \mathbf{p_c} - \hat{\mathbf{p}}_{t-1} - \hat{\mathbf{m}}_t))$.

• Step 5. Repeat Steps 2, 3 and 4 with the new estimated $\hat{\mathbf{l}}_t$ for that input frame till the difference error between the input frame and the rendered frame can be reduced lower than an acceptable threshold.

• Step 6. If the $\hat{\mathbf{p}}_t - \mathbf{p_c}$ is larger than a threshold, re-initialize $\hat{\mathbf{p}}_t$ as the new cardinal pose $\mathbf{p_c}$. Re-compute $\mathcal{C}_{\bullet|\mathbf{p_c}}$ and $\mathcal{B}_{\bullet|p_c}$ at this new cardinal pose.

• Step 7. Set t = t + 1. Repeat Steps 1, 2, 3, 4, 5 and 6. Continue till t = N - 1.

4.3.6 Computational Complexity Analysis

The computation of $\mathcal{B}$ and $\mathcal{C}$ needs to exhaustively search over all the pixels, while the IC algorithm saves significant computational cost by pre-computing the derivatives $\mathcal{B}|_{\mathbf{p_c}}$ and $\mathcal{C}|_{\mathbf{p_c}}$ at the cardinal pose $\mathbf{p_c}$. In both approaches, a number of iterations will be needed

to track each frame. As shown in section 4.3.3 and 4.3.4, the increments $\triangle\mathbf{m}$ obtained from (4.31) in IC approach is approximately at the same order as the $\triangle\mathbf{m}$ obtained from (4.13) in the direct approach, thus about the same number of iterations will be needed. In each iteration, the direct approach needs to compute the derivatives $\mathcal{B}_{\mathbf{v}|\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t}$ and $\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t}$, while the IC approach needs to compute the 3D warping $\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t-1}}}(\mathbf{v}, \triangle\mathbf{p})$. According to the definition of $\mathcal{B}$ and $\mathcal{C}$ in Section 2.2.3, we need 24 multiplications plus 2 additions for computing $\mathcal{B}_{\mathbf{v}|\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t}$ and 93 multiplications plus 24 additions for computing $\mathcal{C}_{\mathbf{v}|\hat{\mathbf{p}}_{t-1}+\mathbf{m}_t}$ at only one pixel $\mathbf{v}$, while only one assignment operation (mapping the intensity at $I_t(\mathbf{v})$ to $I_t(\mathbf{W}_{\mathbf{p}}(\mathbf{v}, \triangle\mathbf{p}))$) will be needed for computing $\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t-1}}}$ at the same pixel $\mathbf{v}$. Thus, by precomputing the the derivatives $\mathcal{B}|_{\mathbf{p_c}}$ and $\mathcal{C}|_{\mathbf{p_c}}$ at the cardinal pose $\mathbf{p_c}$, a significant amount of computation can be saved. The savings will depend upon the implementation, and our experimental results show that the IC algorithm has an average speed-up of > 50 times (maximum > 100) over the direct approach for the controlled data. For the real data, the average speed-up is over 30 times with maximum of 75.9 times, while maintaining the same estimation accuracy.

4.4 Robust and Efficient Tracking on GAMs

We now show how the IC algorithm can be applied for estimation of 3D motion, lighting, identity and expression parameters, which we broadly refer to as tracking, on the GAM. These estimates of motion and lighting can be used for novel view synthesis which has applications in object recognition and inverse rendering. GAMs are particularly applicable for tracking because they do not require a large number of training images at different pose

and lighting conditions, unlike AAMs, PAMs and MLMs.

4.4.1 Direct approach

The GAM, as described in (3.3.3), provides a function of image appearance using image formation parameters, including the pose, lighting, object identity and shape. Using this new result, we can directly have a method for tracking on the image appearance manifold by minimizing a cost function

$$(\hat{\mathbf{l}}_t, \hat{\mathbf{m}}_t, \hat{\mathbf{c}}_{i|t}, \hat{\mathbf{c}}_{e|t}) = \arg\min_{\mathbf{l}_t, \mathbf{m}_t, \mathbf{c}_{i|t}, \mathbf{c}_{e|t}} \|\mathcal{I}_t - (\mathcal{Z}^{\mathcal{B}}_{\hat{\mathbf{p}}_{t-1}} + \mathcal{Z}^{\mathcal{C}}_{\hat{\mathbf{p}}_{t-1}} \times_2 \mathbf{m}_t) \times_1 \mathbf{l}_t \times_3 \mathbf{c}_{i|t} \times_4 \mathbf{c}_{e|t}\|^2 \quad (4.35)$$

where $\hat{x}$ denotes an estimate of x. This cost function is quadrilinear in illumination, motion, identity and deformation variables. The optimization of (4.35) can be done by optimizing over each dimension one by one while keeping the others fixed. Given the condition that motion $\mathbf{m}_t$ is small (the assumption for (3.3.3)), the multilinearity of the image appearance manifold can be satisfied, and thus such alternative minimization can achieve local minimum.

The illumination coefficients can be estimated using least squares (since the illumination bases after motion are not orthogonal), while the identity and expression coefficients can be estimated by projection of the image onto the corresponding basis. Using k to indicate the iteration number, we have

$$\begin{aligned} \hat{\mathbf{l}}_t^k &= (\mathcal{B}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}_t^{k-1}(1)} \mathcal{B}^{\mathbf{T}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}_t^{k-1}(1)})^{-1} \mathcal{B}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}_t^{k-1}(1)} \mathcal{I}^{\mathbf{T}}_{t(1)}, \\ \text{where} \quad \mathcal{B}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}_t^{k-1}} &= \left[\mathcal{Z}^{\mathcal{B}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}_t^{k-1}} \times_3 \hat{\mathbf{c}}_{i|t}^{k-1} \times_4 \hat{\mathbf{c}}_{e|t}^{k-1}\right]_v^{-1}, \end{aligned} \quad (4.36)$$

$$\begin{aligned} \hat{\mathbf{c}}^k_{i|t} &= \mathcal{E}^{\mathbf{T}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t(3)} \mathcal{I}^{\mathbf{T}}_{t(3)}, \\ \text{where} \quad & \mathcal{E}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t} = \left[\mathcal{Z}^{\mathcal{B}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t} \times_1 \hat{\mathbf{l}}^{k-1}_t \times \hat{\mathbf{c}}^{k-1}_{e|t} \right]^{-1}_v, \end{aligned} \tag{4.37}$$

$$\begin{aligned} \hat{\mathbf{c}}^k_{e|t} &= \mathcal{F}^{\mathbf{T}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t(4)} \mathcal{I}^{\mathbf{T}}_{t(4)}, \\ \text{where} \quad & \mathcal{F}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t} = \left[\mathcal{Z}^{\mathcal{C}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t} \times_1 \hat{\mathbf{l}}^{k-1}_t \times_3 \hat{\mathbf{c}}^{k-1}_{i|t} \right]^{-1}_v, \end{aligned} \tag{4.38}$$

Fixing $\hat{\mathbf{l}}^k_t$, $\hat{\mathbf{c}}^k_{i|t}$ and $\hat{\mathbf{c}}^k_{e|t}$, the image becomes a linear function of motion $\mathbf{m}_t$, and using least squares we can estimate $\hat{\mathbf{m}}^k_t$ as

$$\begin{aligned} \Delta\hat{\mathbf{m}}^k_t &= \left(\mathcal{G}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t(2)} \mathcal{G}^{\mathbf{T}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t(2)} \right)^{-1} \mathcal{G}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t(2)} (\mathcal{I}_t - \mathcal{B}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t} \times_1 \hat{\mathbf{l}}^k_t)^T_{(2)}, \\ \text{where} \quad & \mathcal{G}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t} = \mathcal{Z}^{\mathcal{C}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t} \times_1 \hat{\mathbf{l}}^k_t \times_3 \hat{\mathbf{c}}^k_{i|t} \times_4 \hat{\mathbf{c}}^k_{e|t} \\ \text{and} \quad & \hat{\mathbf{m}}^k_t = \hat{\mathbf{m}}^{k-1}_t + \Delta\hat{\mathbf{m}}^k_t. \end{aligned} \tag{4.39}$$

This is essentially Newton's method using the tangent of the manifold. Each iteration requires recomputing the bases $\mathcal{Z}^{\mathcal{B}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t}$ and $\mathcal{Z}^{\mathcal{C}}_{\hat{\mathbf{p}}_{t-1}+\hat{\mathbf{m}}^{k-1}_t}$. As GAMs are computed at only a collection of discrete poses and we do not have an analytical description of the manifold, the direct approach is difficult to apply on GAMs.

The inverse compositional algorithm [6] works by moving the updating terms out of the iterative process. In the following part of this section, we will show how the inverse compositional algorithm can be applied upon GAMs by using only the cardinal poses for tracking a sequence in video.

4.4.2 Inverse Compositional Estimation of 3D Motion on GAM

For simplicity of explanation, let us initially start with the condition that $\mathbf{l}$, $\mathbf{c}_i$, $\mathbf{c}_e$ do not change between two frames. We will then consider the varying illumination and

deformation case. Following the derivation in Section 4.3, we get

$$\begin{aligned}
\triangle \mathbf{m}_t^k &= \mathbf{H_{IC}} \sum_{\mathbf{v}} \mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \left(I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, -\hat{\mathbf{m}}_t^{k-1})) - \mathcal{Z}^{\mathcal{B}}_{\mathbf{v}|\hat{\mathbf{p}}_{t-1}} \times_1 \hat{\mathbf{l}}_t^k \times_3 \hat{\mathbf{c}}_{i|t}^k \times_4 \hat{\mathbf{c}}_{e|t}^k \right), \\
\text{where} \qquad \mathbf{H_{IC}} &= \left[\sum_{\mathbf{v}} \mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \mathcal{G}^{\mathbf{T}}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}} \right]^{-1}, \qquad (4.40)
\end{aligned}$$

and $\hat{\mathbf{m}}_t^k$ can be updated using

$$\hat{\mathbf{m}}_t^k = \hat{\mathbf{m}}_t^{k-1} + \Delta \mathbf{m}_t^k. \qquad (4.41)$$

The proof of the convergence of the update rule (4.41) can be done in a similar way to that in section 4.3. Note that the derivative $\mathcal{G}_{\mathbf{v}|\hat{\mathbf{p}}_{\mathbf{t-1}}}$ and Hessian $\mathbf{H_{IC}}$ in (4.40) do not depend upon the updating variable $\hat{\mathbf{m}}_t^k$.

Using

$$\begin{aligned}
\hat{\mathbf{l}}_t^k &= (\mathcal{B}_{\hat{\mathbf{p}}_{t-1}(1)} \mathcal{B}^{\mathbf{T}}_{\hat{\mathbf{p}}_{t-1}(1)})^{-1} \mathcal{B}_{\hat{\mathbf{p}}_{t-1}(1)} (\mathcal{I}_{t(1)}^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(-\hat{\mathbf{m}}_t^{k-1})})^{\mathbf{T}}, \\
\text{where} \qquad \mathcal{B}_{\hat{\mathbf{p}}_{t-1}} &= \left[\mathcal{Z}^{\mathcal{B}}_{\hat{\mathbf{p}}_{t-1}} \times_3 \hat{\mathbf{c}}_{i|t}^{k-1} \times_4 \hat{\mathbf{c}}_{e|t}^{k-1} \right]_v^{-1}, \qquad (4.42) \\
\hat{\mathbf{c}}_{i|t}^k &= \mathcal{E}^{\mathbf{T}}_{\hat{\mathbf{p}}_{t-1}(3)} (\mathcal{I}_{t(3)}^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(-\hat{\mathbf{m}}_t^{k-1})})^{\mathbf{T}}, \\
\text{where} \qquad \mathcal{E}_{\hat{\mathbf{p}}_{t-1}} &= \left[\mathcal{Z}^{\mathcal{B}}_{\hat{\mathbf{p}}_{t-1}} \times_1 \hat{\mathbf{l}}_t^{k-1} \times \hat{\mathbf{c}}_{e|t}^{k-1} \right]_v^{-1}, \qquad (4.43) \\
\hat{\mathbf{c}}_{e|t}^k &= \mathcal{F}^{\mathbf{T}}_{\hat{\mathbf{p}}_{t-1}(4)} (\mathcal{I}_{t(4)}^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(-\hat{\mathbf{m}}_t^{k-1})})^{\mathbf{T}}, \\
\text{where} \qquad \mathcal{F}_{\hat{\mathbf{p}}_{t-1}} &= \left[\mathcal{Z}^{\mathcal{C}}_{\hat{\mathbf{p}}_{t-1}} \times_1 \hat{\mathbf{l}}_t^{k-1} \times_3 \hat{\mathbf{c}}_{i|t}^{k-1} \right]_v^{-1}, \qquad (4.44)
\end{aligned}$$

for estimating $\hat{\mathbf{l}}_t^k$, $\hat{\mathbf{c}}_{i|t}^k$, $\hat{\mathbf{c}}_{e|t}^k$, we can recompute $\mathcal{G}^k_{\hat{\mathbf{p}}_{t-1}}$ as

$$\mathcal{G}^k_{\hat{\mathbf{p}}_{t-1}} = \mathcal{Z}^{\mathcal{C}}_{\hat{\mathbf{p}}_{t-1}} \times_1 \hat{\mathbf{l}}_t^k \times_3 \hat{\mathbf{c}}_{i|t}^k \times_4 \hat{\mathbf{c}}_{e|t}^k. \qquad (4.45)$$

Substituting $\hat{\mathbf{l}}_t^k$, $\hat{\mathbf{c}}_{i|t}^k$, $\hat{\mathbf{c}}_{e|t}^k$, and $\mathcal{G}^k_{\hat{\mathbf{p}}_{t-1}}$ back into (4.40), we can alternately estimate all the

parameters. Although $\mathcal{G}^k_{\hat{\mathbf{p}}_{t-1}}$ needs to be updated in each iteration, the core tensor $\mathcal{Z}^{\mathcal{C}}_{\hat{\mathbf{p}}_{t-1}}$, which is the most computational intensive part, does not need to be updated.

4.4.3 Inverse Compositional Estimation on GAMs Over A Sequence of Frames

Similar to Section 4.3.4, to further save on computational complexity in the video sequence context, we can obtain the IC tracking algorithm for GAM over a sequence of frames as:

$$\begin{aligned}
\triangle \mathbf{m}^k_t &= \mathbf{H_{IC}} \sum_{\mathbf{v}} \mathcal{G}_{\mathbf{v}|\mathbf{p}_c} \left(I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{v}, (\mathbf{p_c} - \hat{\mathbf{p}}_{t-1}) - \hat{\mathbf{m}}^{k-1}_t)) - \mathcal{Z}^{\mathcal{B}}_{\mathbf{v}|\mathbf{p_c}} \times_1 \hat{\mathbf{l}}^k_t \times_3 \mathbf{c}^k_{i|t} \times_4 \mathbf{c}^k_{e|t} \right), \\
\text{where} \qquad \mathbf{H_{IC}} &= \left[\sum_{\mathbf{v}} \mathcal{G}_{\mathbf{v}|\mathbf{p}_c} \mathcal{G}^{\mathbf{T}}_{\mathbf{v}|\mathbf{p}_c} \right]^{-1} \text{and } \hat{\mathbf{m}}^k_t = \hat{\mathbf{m}}^{k-1}_t + \Delta \mathbf{m}^k_t. \qquad (4.46)
\end{aligned}$$

The proof of the convergence of the IC algorithm over a sequence of frames directly follows the one in Section 4.3.3.

Using

$$\begin{aligned}
\hat{\mathbf{l}}^k_t &= (\mathcal{B}_{\mathbf{p}_c(1)} \mathcal{B}^{\mathbf{T}}_{\mathbf{p}_c(1)})^{-1} \mathcal{B}_{\mathbf{p}_c(1)} (\mathcal{I}^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{p}_c - \hat{\mathbf{p}}_{t-1} - \hat{\mathbf{m}}^{k-1}_t)}_{t(1)})^{\mathbf{T}}, \\
\text{where} \qquad \mathcal{B}_{\mathbf{p}_c} &= \left[\mathcal{Z}^{\mathcal{B}}_{\mathbf{p}_c} \times_3 \hat{\mathbf{c}}^{k-1}_{i|t} \times_4 \hat{\mathbf{c}}^{k-1}_{e|t} \right]^{-1}_v, \qquad (4.47) \\
\hat{\mathbf{c}}^k_{i|t} &= \mathcal{E}^{\mathbf{T}}_{\mathbf{p}_c(3)} (\mathcal{I}^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{p}_c - \hat{\mathbf{p}}_{t-1} - \hat{\mathbf{m}}^{k-1}_t)}_{t(3)}), \\
\text{where} \qquad \mathcal{E}_{\mathbf{p}_c} &= \left[\mathcal{Z}^{\mathcal{B}}_{\mathbf{p}_c} \times_1 \hat{\mathbf{l}}^{k-1}_t \times \hat{\mathbf{c}}^{k-1}_{e|t} \right]^{-1}_v, \qquad (4.48) \\
\hat{\mathbf{c}}^k_{e|t} &= \mathcal{F}^{\mathbf{T}}_{\mathbf{p}_c(4)} (\mathcal{I}^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{p}_c - \hat{\mathbf{p}}_{t-1} - \hat{\mathbf{m}}^{k-1}_t)}_{t(4)}), \\
\text{where} \qquad \mathcal{F}_{\mathbf{p}_c} &= \left[\mathcal{Z}^{\mathcal{C}}_{\mathbf{p}_c} \times_1 \hat{\mathbf{l}}^{k-1}_t \times_3 \hat{\mathbf{c}}^{k-1}_{i|t} \right]^{-1}_v, \qquad (4.49)
\end{aligned}$$

for estimating $\hat{\mathbf{l}}_t^k$, $\hat{\mathbf{c}}_{i|t}^k$ and $\hat{\mathbf{c}}_{e|t}^k$, $\mathcal{G}_{\mathbf{p}_c}$ can be computed as

$$\mathcal{G}_{\mathbf{p}_c}^k = \mathcal{Z}_{\mathbf{p}_c}^{\mathcal{C}} \times_1 \hat{\mathbf{l}}_t^k \times_3 \hat{\mathbf{c}}_{i|t}^k \times_4 \hat{\mathbf{c}}_{e|t}^k. \tag{4.50}$$

Substituting $\hat{\mathbf{l}}_t^k$, $\hat{\mathbf{c}}_{i|t}^k$, $\hat{\mathbf{c}}_{e|t}^k$, and $\mathcal{G}_{\hat{\mathbf{p}}_c}^k$ back into (4.46), we can alternately estimate all the parameters. The bases $\mathcal{Z}_{\mathbf{p}_c}^{\mathcal{B}}$ and $\mathcal{Z}_{\mathbf{p}_c}^{\mathcal{C}}$ are computed at the discrete cardinal poses $\mathbf{p}_c$, allowing us to track the video efficiently on the GAM. This is unlike (4.39) where we need the bases at the pose of each iteration.

4.4.4 The IC Algorithm on GAMs

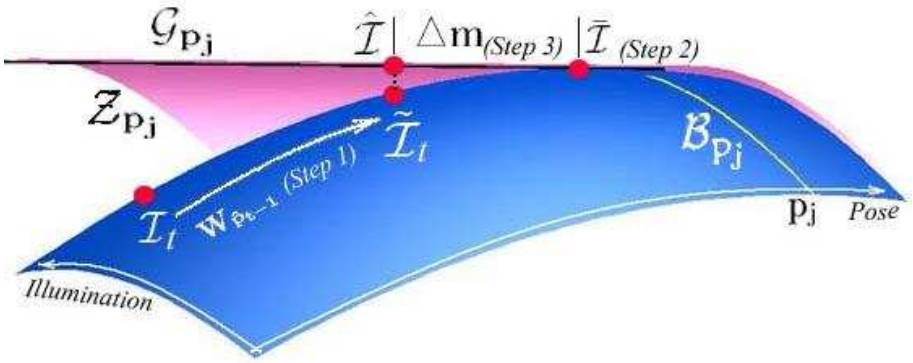


Figure 4.2: Pictorial representation of the inverse compositional tracking scheme on GAMs.

Due to the fact that GAM is constructed at the pre-defined values of $(\mathbf{T}_x^{pd}, \mathbf{T}_y^{pd}, \mathbf{T}_z^{pd}, \mathbf{\Omega}_z^{pd})$ and using a series of tangents modeling along the $\mathbf{\Omega}_x^{pd}$ and $\mathbf{\Omega}_y^{pd}$ directions, for each input frame I_t, we will use the pose estimated at previous time instance, $\hat{\mathbf{p}}_{t-1}$, or the initialization for centering and scale normalizing the image to these pre-defined values. Then, the Inverse Compositional algorithm on GAMs will be applied on this normalized image. The 3D estimates will be combined with the above values to obtain the final pose estimates.

A pictorial representation of the IC tracking algorithm on GAMs is shown in Fig. 4.2. Consider a sequence of image frames $\mathcal{I}_t$, $t = 0, ..., N-1$. Assume that we know the pose and illumination estimates for frame $t-1$, i.e., $\hat{\mathbf{p}}_{t-1}$ and $\hat{\mathbf{l}}_{t-1}$.

• *Step 1.* For the new input frame $\mathcal{I}_t$, normalize it to the pre-defined values $(\mathbf{T}_x^{pd}, \mathbf{T}_y^{pd}, \mathbf{T}_z^{pd}, \mathbf{\Omega}_z^{pd})$ using the pose estimates at $t-1$, i.e. $\hat{\mathbf{p}}_{t-1}$. Find the closest $\mathbf{p_j}$ to $\hat{\mathbf{p}}_{t-1}^{pd} \triangleq (\mathbf{T}_x^{pd}, \mathbf{T}_y^{pd}, \mathbf{T}_z^{pd}, \hat{\mathbf{\Omega}}_{x|t-1}, \hat{\mathbf{\Omega}}_{y|t-1}, \mathbf{\Omega}_z^{pd})$. Set the iteration index $k = 1$. Assume motion $\hat{\mathbf{m}}_t^{pd|0}$ at this pre-defined pose to be zero, illumination condition $\hat{\mathbf{l}}_t^0 = \hat{\mathbf{l}}_{t-1}$, identity coefficient $\hat{\mathbf{c}}_{i|t}^0 = \hat{\mathbf{c}}_{i|t-1}$, and expression coefficient $\hat{\mathbf{c}}_{e|t}^0 = \hat{\mathbf{c}}_{e|t-1}$.

• *Step 2.* Apply the pose transformation operator $\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}$ to get the pose normalized version of the frame $\tilde{\mathcal{I}}_t^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}(\mathbf{p_j} - \hat{\mathbf{p}}_{t-1}^{pd} - \hat{\mathbf{m}}_t^{pd|k-1})}$, i.e., $I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}(\mathbf{u}, \mathbf{p_j} - \hat{\mathbf{p}}_{t-1}^{pd} - \hat{\mathbf{m}}_t^{pd|k-1}))$. This is shown in Figure 4.2, where the input frame $\mathcal{I}_t$ on the manifold is first normalized and warped to $\tilde{\mathcal{I}}_t$ which is within a nearby region of pose $\mathbf{p_j}$.

• *Step 3.* Use (4.47), (4.48) and (4.49) to alternately estimate $\hat{\mathbf{l}}_t^k$, $\hat{\mathbf{c}}_{i|t}^k$ and $\hat{\mathbf{c}}_{e|t}^k$ of the pose normalized image $\tilde{\mathcal{I}}_t^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}(\mathbf{p_j} - \hat{\mathbf{p}}_{t-1}^{pd} - \hat{\mathbf{m}}_t^{pd|k-1})}$. In Figure 4.2, the curve $\mathcal{B}_{\mathbf{p_j}}$ shows the manifold of the image at pose $\mathbf{p_j}$ with motion as zero, but varying illumination, identity or deformation. By iteratively minimizing along the illumination, identity, and deformation directions, we find the point

$$\bar{\mathcal{I}}_t^k = \mathcal{Z}_{\mathbf{p}_j}^{\mathcal{B}} \times_1 \hat{\mathbf{l}}_t^k \times_3 \hat{\mathbf{c}}_{i|t}^k \times_4 \hat{\mathbf{c}}_{e|t}^k \tag{4.51}$$

on the curve $\mathcal{B}_{\mathbf{p_j}}$ which has the minimum distance to the pose normalized point $\tilde{\mathcal{I}}_t$.

• *Step 4.* With the estimated $\hat{\mathbf{l}}_t^k$, $\hat{\mathbf{c}}_{i|t}^k$ and $\hat{\mathbf{c}}_{e|t}^k$ from Step 3, use (4.46) to estimate the motion increment $\triangle\mathbf{m}_t^{pd|k}$. Update $\hat{\mathbf{m}}_t^{pd|k}$ with $\hat{\mathbf{m}}_t^{pd|k} \leftarrow \hat{\mathbf{m}}_t^{pd|k-1} + \triangle\mathbf{m}_t^{pd|k}$. In Figure 4.2, we compute the tangent along the motion direction, shown as the black line $\mathcal{G}_{\mathbf{p_j}}$, from

the core tensor shown as the surface $\mathcal{Z}$. $\triangle \mathbf{m}_t^{pd|k}$ is shown to be the distance from point $\bar{\mathcal{I}}_t$ to $\hat{\mathcal{I}}_t$, the projection of $\tilde{\mathcal{I}}_t$, onto the motion tangent.

• *Step 5.* Use the updated $\hat{\mathbf{m}}_t^{pd|k}$ from Step 4 to update the pose normalized image as $\tilde{\mathcal{I}}_t^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}(\mathbf{p_j}-\hat{\mathbf{p}}_{t-1}^{pd}-\hat{\mathbf{m}}_t^{pd|k})}$, i.e. $I_t(\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}(\mathbf{u}, \mathbf{p_j} - \hat{\mathbf{p}}_{t-1}^{pd} - \hat{\mathbf{m}}_t^{pd|k}))$.

• *Step 6.* Set $k = k + 1$. Repeat Steps 2, 3, 4 and 5 for that input frame till the difference error ε between the pose normalized image $\tilde{\mathcal{I}}_t^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}(\mathbf{p_j}-\hat{\mathbf{p}}_{t-1}^{pd}-\hat{\mathbf{m}}_t^{pd|k})}$ and the rendered image $\bar{\mathcal{I}}_t^k$ can be reduced below an acceptable threshold.

• Step 7. Undo the normalization of Step 1 to inverse transform $\hat{\mathbf{m}}_t^{pd|k}$ to $\hat{\mathbf{m}}_t$ and update $\hat{\mathbf{p}}_t = \hat{\mathbf{p}}_{t-1} + \hat{\mathbf{m}}_t$.

• Step 8. Set t = t + 1. Repeat Steps 1, 2, 3, 4, 5, 6 and 7. Continue till t = N - 1.

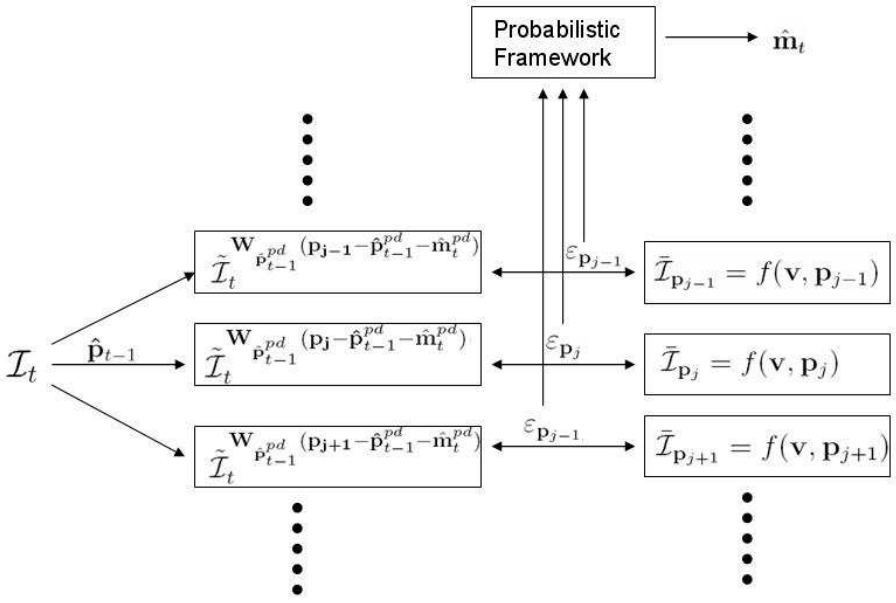


Figure 4.3: Pictorial representation of the probabilistic inverse compositional tracking scheme on GAMs.

4.4.5 Probabilistic Inverse Compositional (PIC) Estimation

To ensure that the tracking is robust to estimation errors, we embed the IC approach within a probabilistic framework. For ease of explanation, let us denote the current cardinal pose to be $\mathbf{p}_j$, and the set of the nearby cardinal poses as $\{\mathbf{p}_{\mathbb{N}(j)}\}$, where $\mathbb{N}(j)$ is the set of all the neighboring cardinal poses around $\mathbf{p}_j$. Denote the nearest-neighbor partition region on the multilinear manifold for cardinal pose $\mathbf{p}_j$ to be $\Theta_{\mathbf{p}_j}$. Given the estimated pose at the previous time instance $\hat{\mathbf{p}}_{t-1}$, the average velocity $\bar{\mathbf{m}}$ and variation σ_m^2 of it within a recent history, we can model the distribution of the current pose $\mathbf{p}_t \sim \mathcal{N}(\hat{\mathbf{p}}_{t-1} + \bar{\mathbf{m}}, \sigma_m^2)$, where $\mathcal{N}$ is the normal distribution. Assume the likelihood distribution of the difference between pose normalized image $\tilde{\mathcal{I}}_t^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}^{pd}}(\mathbf{p_j} - \hat{\mathbf{p}}_{t-1}^{pd} - \hat{\mathbf{m}}_t^{pd|k})}$ and the rendered image $\bar{\mathcal{I}}_{\mathbf{p}_j}$, ε, at pose $\mathbf{p}_j$ is $P(\varepsilon_{\mathbf{p}_j}|\mathbf{p}_t \in \Theta_{\mathbf{p}_j}) \sim \mathcal{N}(0, \sigma)$. Using Bayes rule, we get

$$P(\mathbf{p}_t \in \Theta_{\mathbf{p}_j}|\varepsilon_{\mathbf{p}_j}) = \frac{P(\varepsilon_{\mathbf{p}_j}|\mathbf{p}_t \in \Theta_{\mathbf{p}_j}) \int_{\Theta_{\mathbf{p}_j}} p(\mathbf{p}_t) d\mathbf{p}_t}{P(\varepsilon_{\mathbf{p}_j})}. \tag{4.52}$$

Denoting the estimate of motion, illumination, identity, and expression with the tangent at $\mathbf{p}_j$ as $\hat{\mathbf{x}}_t^{\mathbf{p}_j}$, the final estimate can be obtained as

$$\hat{\mathbf{x}}_t = E(\mathbf{x}_t|\mathcal{I}) = \frac{\sum_{i \in \mathbb{N}(j)} \hat{\mathbf{x}}_t^{\mathbf{p}_i} P(\mathbf{p}_t \in \Theta_{\mathbf{p}_i}|\varepsilon_{\mathbf{p}_i})}{\sum_{i \in \mathbb{N}(j)} P(\mathbf{p}_t \in \Theta_{\mathbf{p}_i}|\varepsilon_{\mathbf{p}_i})}. \tag{4.53}$$

A pictorial representation of the PIC algorithm is shown in Fig. 4.3.

4.5 Experimental Results

4.5.1 Accuracy Analysis on Controlled Data

To show the tracking accuracy of the IC tracking algorithms, we first do a synthetic experiment with the Stanford Bunny rabbit model under varying illumination conditions. The bunny rabbit is rotating along the vertical axis at some specific angular velocity, and the illumination is changing both in direction (from right-bottom corner to the left-top corner) and in brightness (from dark to bright to dark). The first row in Fig. 4.4 shows the back projection of some feature points on the 3D model back onto the input frames using the estimated motion with the IC tracking algorithm under three different illumination conditions. The second row shows the synthesis images with the motion and illumination estimates. There is no perceptual difference between the original frame and the synthesized ones.

In Fig. 4.5, we compare the IC algorithm with the direct approach described in Section 4.2. We show the comparison of the computational cost between the two approaches in (a), the motion estimation accuracy in (b), and the frequency of convergence in (c). The computational cost is measured by the processing time needed for each frame on a standard PC with 1.8GHz CPU, 2G RAM with a Matlab implementation. The average processing time for each frame in direct approach is 9.7 seconds, while in IC algorithm it is 0.18 seconds per frame. Thus, IC algorithm has an 52.1 folds speeding up while sacrificing little in the estimation accuracy. The frequency of convergence is computed as the percentage of the frames among the 180 frames in the control experiment that converge to the specific

accuracy of the pose estimates measured in degree. On the average, the direct approach and the IC algorithm have the same frequency of convergence, validating the equivalence between the two. The divergence of the two curves is due to the relatively small number of the frames used for measuring the percentage of the convergence.

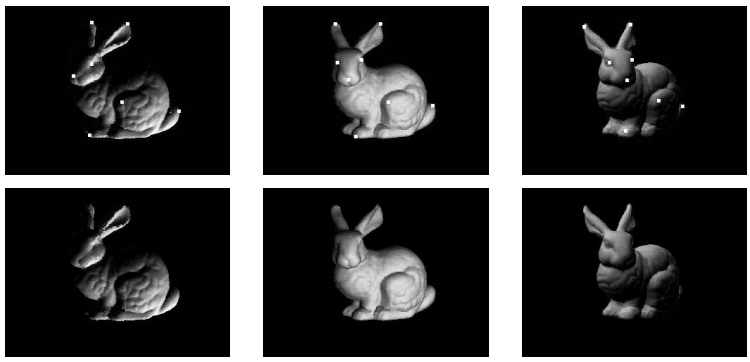


Figure 4.4: Top: the back projection of the mesh vertices of the 3D bunny rabbit model using the estimated 3D motion onto some input frames. Bottom: Synthesized images with estimated motion and illumination.

In Fig. 4.6, we show some accuracy analysis of the motion and illumination estimation. We designed three experiments: Expt. A - estimate both motion and illumination simultaneously; Expt. B - estimate motion with known illumination; Expt. C - estimate illumination with known motion.

Note that illumination bases $\mathcal{B}$ are functions of pose, while the motion bases $\mathcal{C}$ do not rely upon illumination. Thus, knowing motion should be helpful for estimating the illumination. This is seen in Fig. 4.6 (b) where the illumination estimation error in Expt. C is consistently lower than that of Expt. A. Due to the same reason, the synthesis error in Expt. C is consistently lower than that in Expt. A, as shown in Fig. 4.6 (c). On the other hand, knowing illumination does not help as much in motion estimation, since the motion

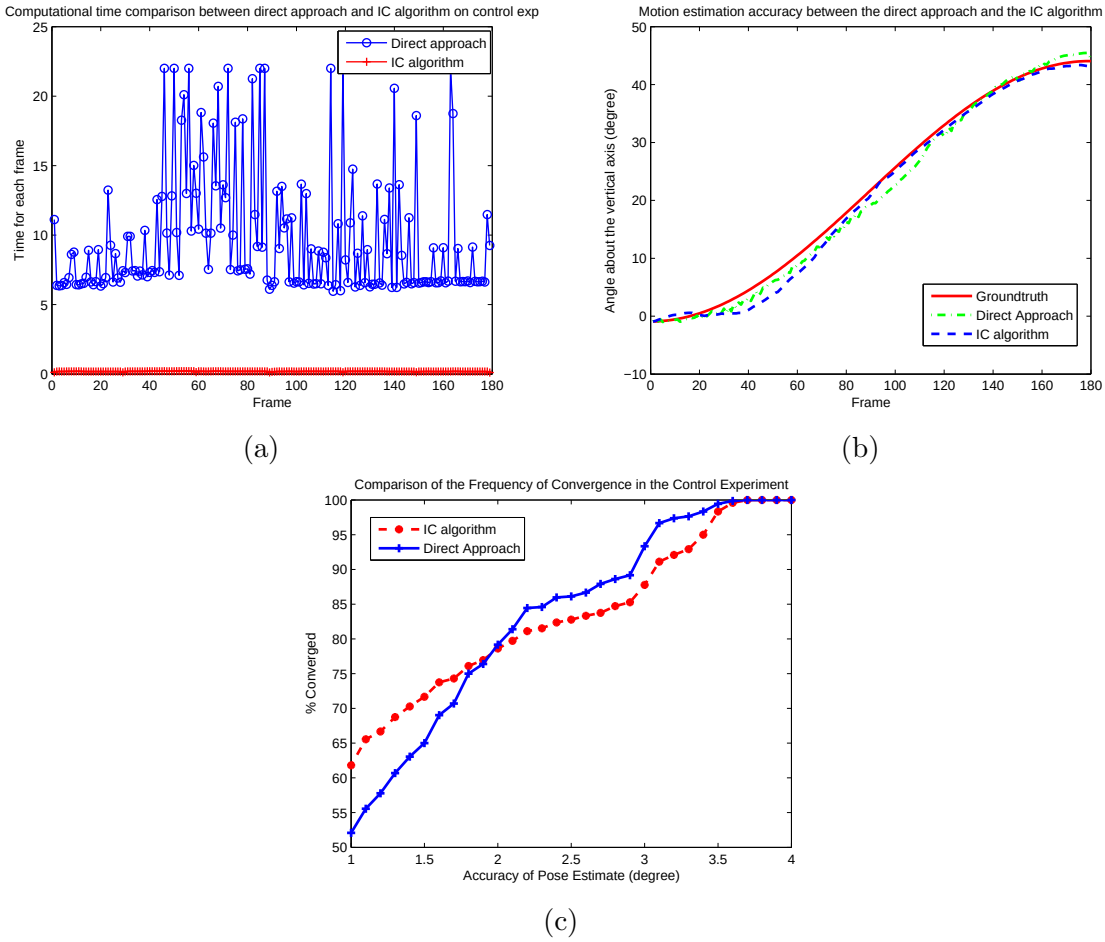


Figure 4.5: (a) shows the comparison of the computational time needed for each frame in the direct approach and the IC algorithm. (b) shows the comparison of the motion estimation accuracy obtained by the direct approach and the IC algorithm. (c) shows the comparison of the frequency of convergence in the control experiment between the direct approach and the IC algorithm.

bases do not depend upon illumination. Thus, the motion estimates of Expt. A are neither consistently better nor worse than those of Expt. B as shown in Fig. 4.6 (a), and the same is true for the synthesis errors, shown in Fig. 4.6 (c). Thus, knowing the ground truth motion can lead to more accurate estimates of illumination (the average synthesis error is 2.51%), while knowledge of illumination produces an average synthesis error of 3.78%. In

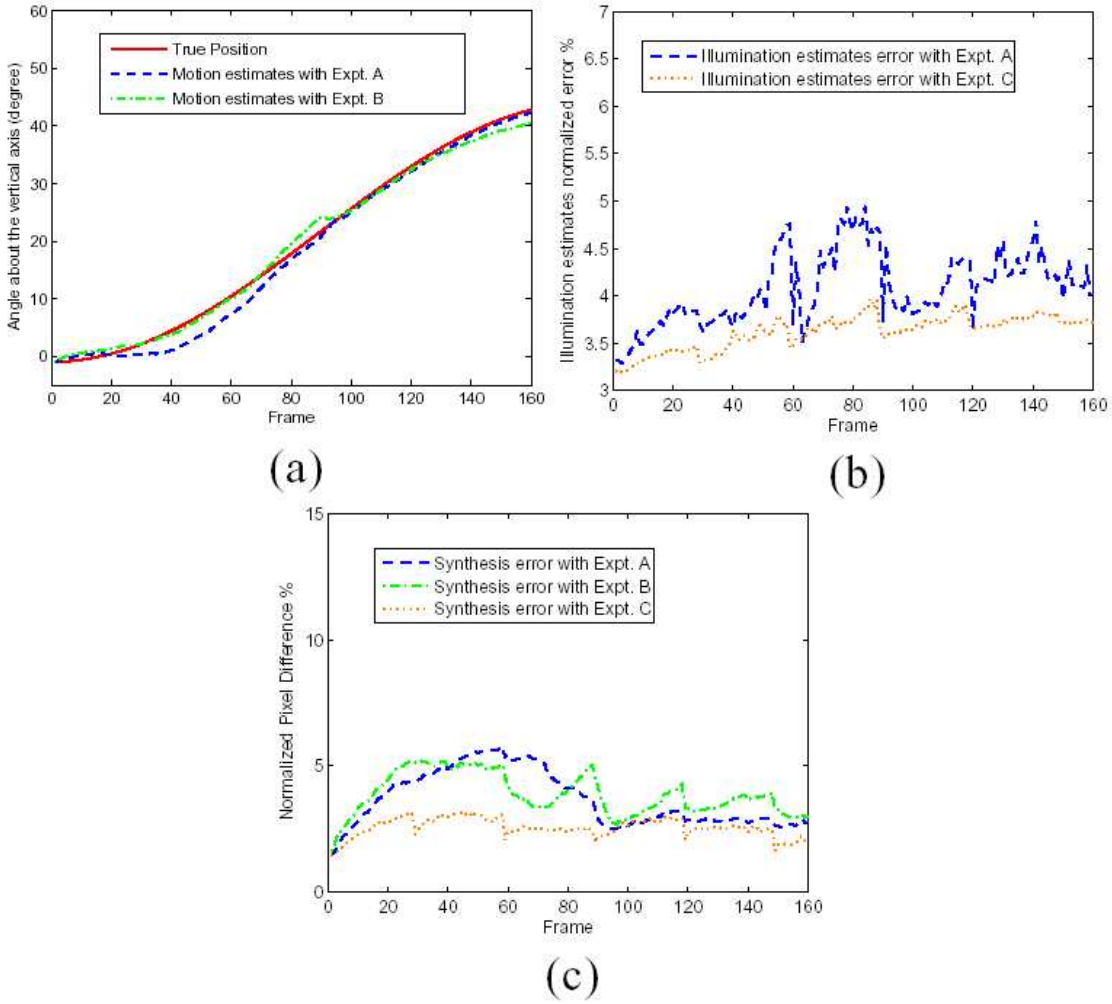


Figure 4.6: (a) shows the comparison between the pose estimates with known illumination, unknown illumination, and the ground truth, (b) shows the normalized error of the illumination estimates without knowing the true motion and with the true motion known, (c) shows the normalized synthesis error with unknown illumination and motion, unknown motion but known illumination, and unknown illumination but known motion.

Fig. 4.7, we show the plots of six illumination coefficients [1].

[1]It has been shown that from a specific viewing point, the spherical harmonic functions will not be orthogonal to each other; therefore, not all the illumination coefficients will be observable [58]. We orthogonalize the spherical harmonic basis functions by taking their principal components, and estimate the illumination condition with this principal component basis.

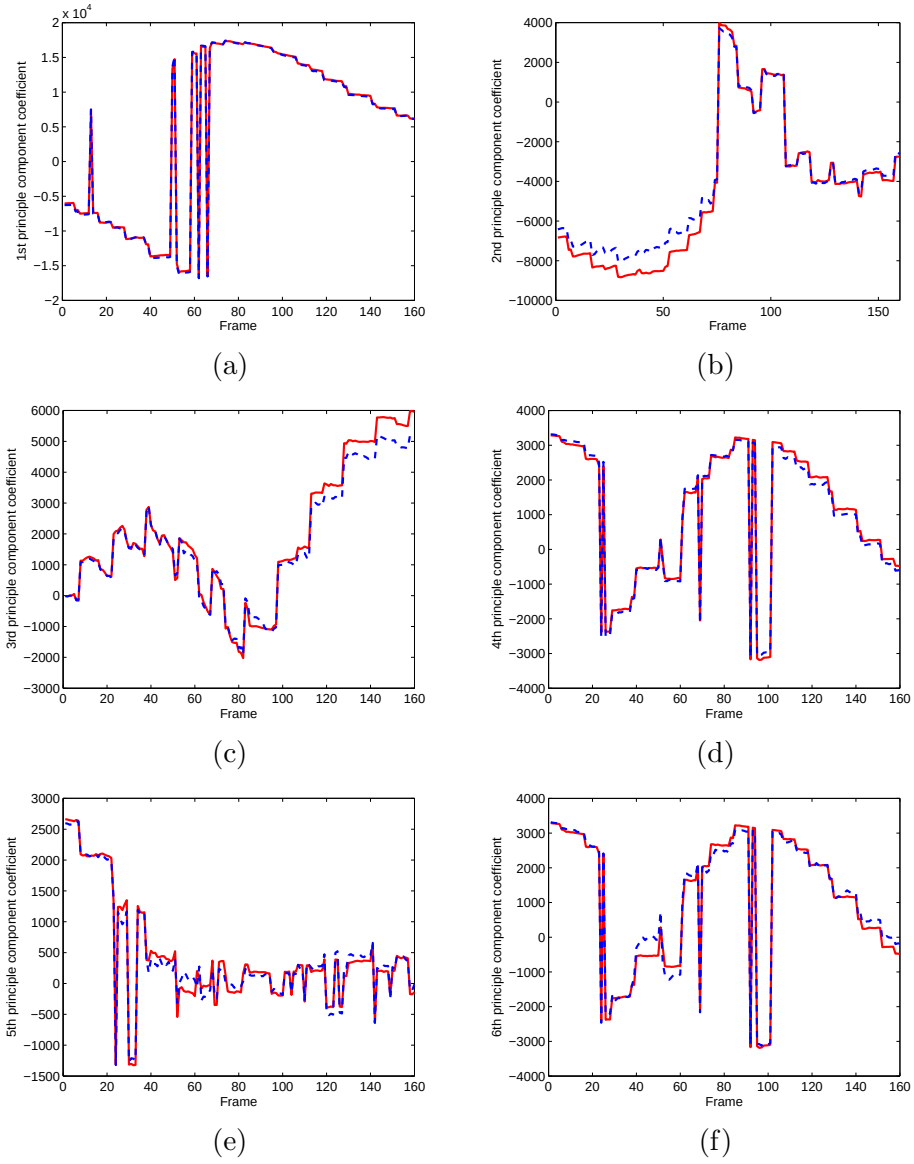


Figure 4.7: (a) to (f) show the plots of the true and the estimated coefficients from the 1st to the 6th illumination principle components. The solid red plots are for the true illumination vector, the dotted blue ones are the illumination coefficients estimated from the inverse compositional algorithm.

4.5.2 Accuracy Analysis on Real-Life Face Data

Fig. 4.8 shows the motion and illumination estimates on two real data examples. The images in the first row are the input frames with the back projection of some feature

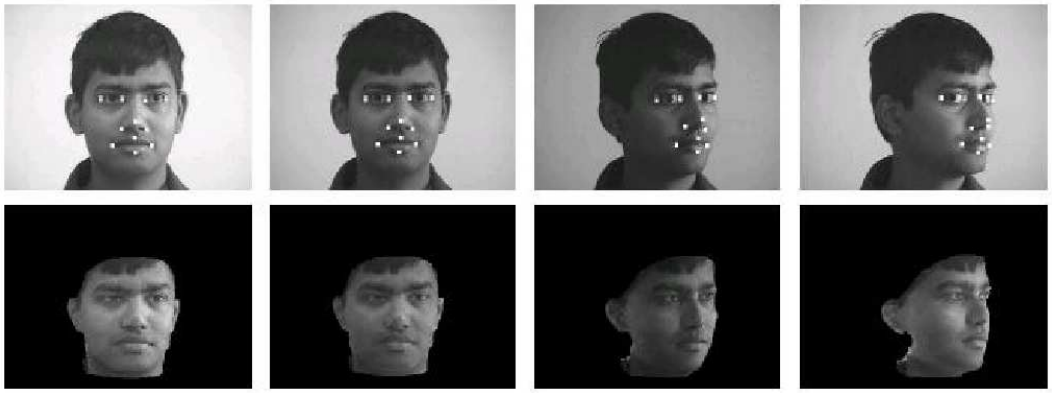
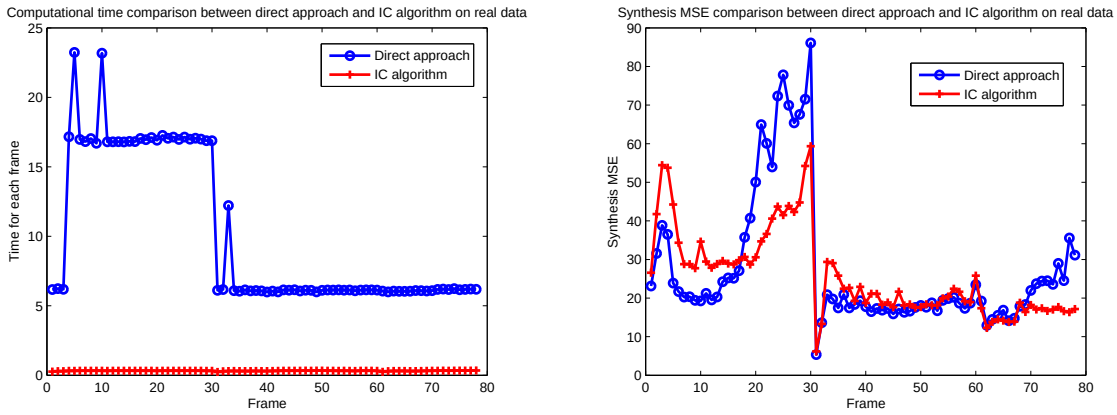


Figure 4.8: The comparison between the original frames and the synthesized ones with the estimated motion and illumination variables. The first rows show the original frames, and the second row shows the synthesized frames with the estimated illumination and motion from the images in the same column.



(a) (b)

Figure 4.9: Computational cost and the estimation accuracy comparison between the direct approach and the inverse compositional algorithm on the real data. (a) The vertical axis shows the processing time needed for each frame, while the horizontal axis shows the index of frames. By taking the mean of the processing time for all the frames in each approach, the direct approach has an average processing time of 10.11 seconds for each frame, while the IC algorithm uses 0.32 seconds per frame. (b) the vertical axis shows the MSE between the input frame and the synthesized frames using the estimated motion and illumination parameters.

mesh vertices, and the ones in the second row are synthesized with the estimated illumination and motion. This result shows that it is possible to synthesize images with the motion

and illumination parameters learned from natural videos. This is useful for applications in video-based rendering and object recognition.

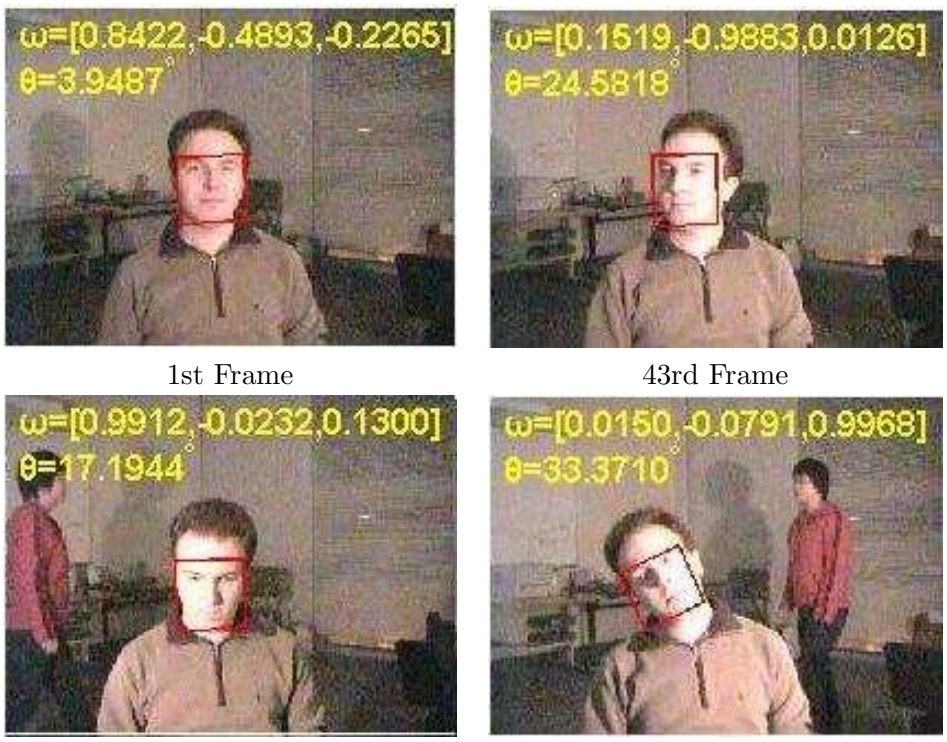


Figure 4.10: An example of face tracking using GAMs under changes of pose and lighting. The estimated pose is shown on the top of the frames. (Should be viewed on a monitor)

In Fig. 4.9, we show the comparison of the computational cost and the estimation accuracy between the direct approach described in Section 4.2 and the inverse compositional approach described in Section 4.3 on the sequence shown in the first row of Fig. 4.8. We use totally 80 frames, in which the head rotates from frontal pose to about 45 degree along the vertical axis. To assess the quality of the motion and illumination estimation accuracy on the real data, we synthesize the images with the estimated motion and illumination parameters, and take the pairwise pixel intensity difference between the synthesized frame

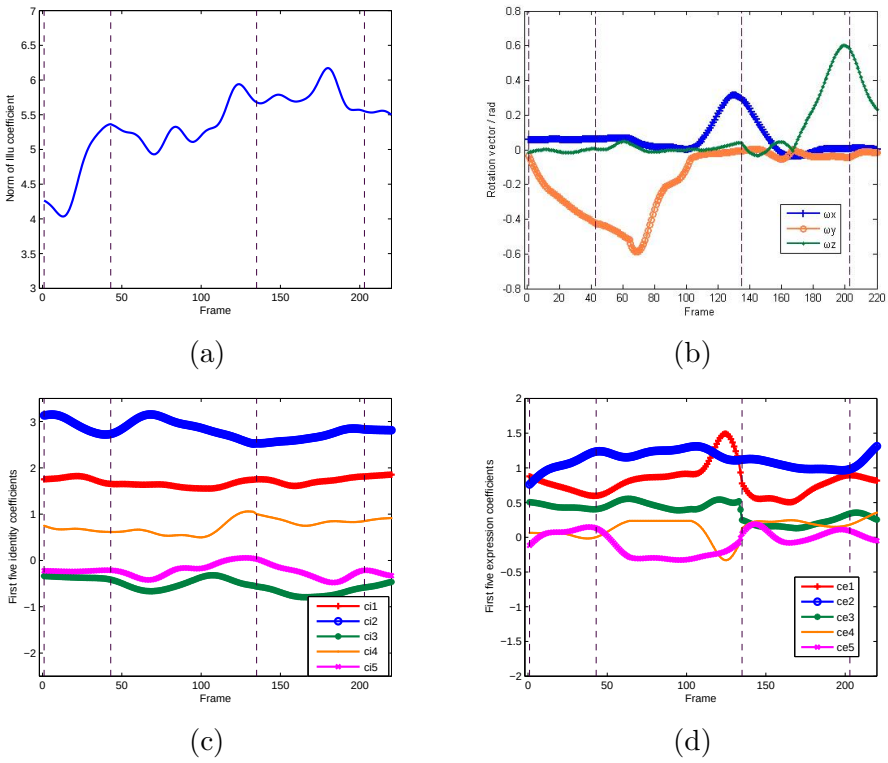
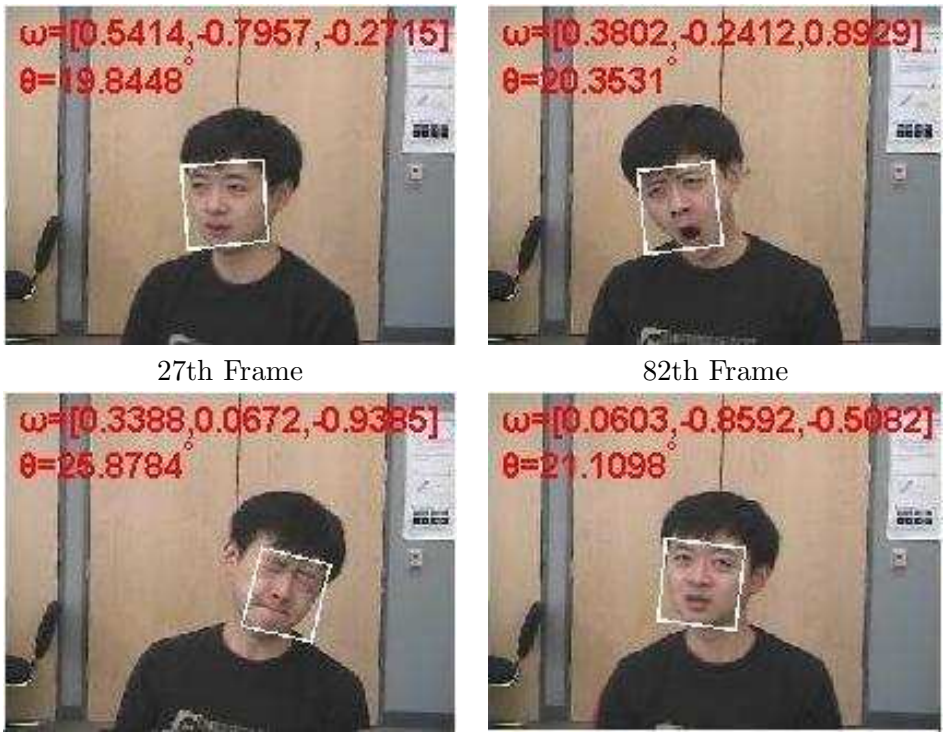


Figure 4.11: Parameter estimates of the sequence shown in Fig. 4.10 using GAMs. (a) shows the norm of the estimated illumination coefficients as a function of time, (b) shows the estimated pose as rotation vector. (c) shows the first five estimated coefficients of the identity dimension, while (d) shows the first five estimated coefficients of the expression dimension. The key frames shown in Fig. 4.10 are marked using dash lines.

and the input frame. Some peaks and plateaus in the plot of the direct approach in Fig. 4.9 (a) indicate that at those frames more iterations are needed for convergence. It is also shown in the plot that usually it takes about 6 seconds for one computation of the bilinear bases; thus the processing time for each frame is approximately a multiple of this time. From Fig. 4.9 (a), we can see that more iterations are used in the first 30 frames. This is because the motion between these frames is large and hence more iterations are needed. Around frame 30, the synthesis error was above a threshold and a new cardinal pose was chosen. After this, the inter-frame motion is smaller and the computation time and synthesis error are

low. By taking the mean of the processing time for the 80 frames, the inverse compositional approach is 31.6 times faster than the direct approach, while the synthesis error is about the same in both approaches. The maximum improvement is 75.9 faster than the direct approach at specific frame. This shows the significant improvement of the IC algorithm over the direct approach.

4.5.3 IC Tracking on GAM using Real Data



27th Frame 82th Frame

143th Frame 187th Frame

Figure 4.12: Another example of face tracking using GAMs under changes of pose and expressions. The estimated pose is shown on the top of the frames.

Figure 4.10 and 4.12 show results of face tracking under large changes of pose, lighting, expression and background using the IC approach on GAM. The images in Figure

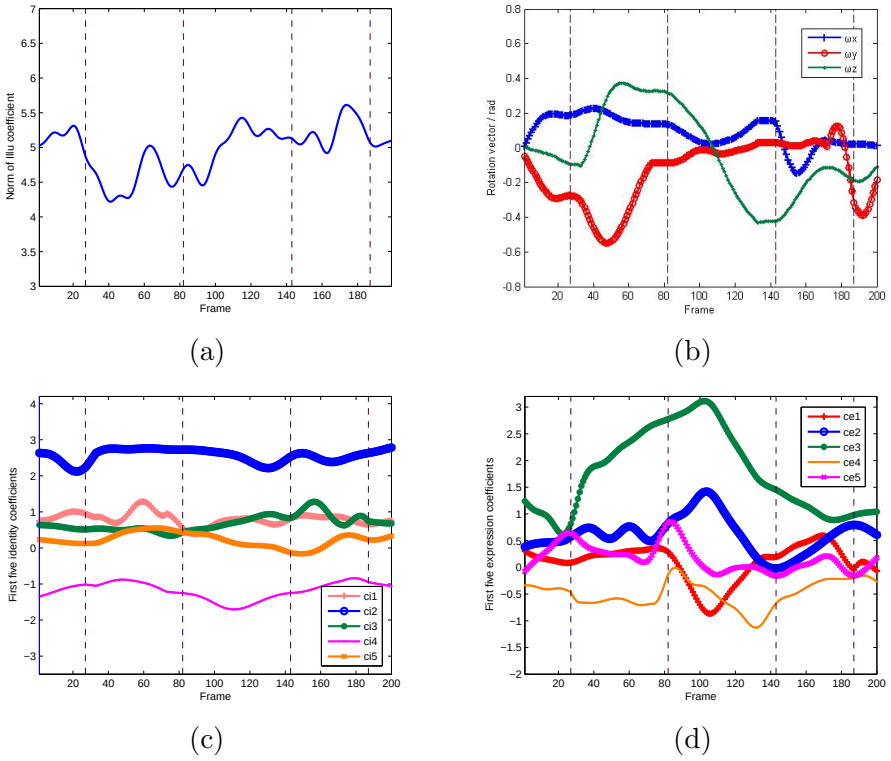


Figure 4.13: Parameter estimates of the sequence shown in Fig. 4.12 using GAMs. (a) shows the norm of the estimated illumination coefficients as a function of time, (b) shows the estimated pose as rotation vector, (c) shows the first five estimated coefficients of the identity dimension, while (d) shows the first five estimated coefficients of the expression dimension. The key frames shown in Fig. 4.12 are marked using dash lines.

4.10 show tracking under illumination variations. The images in Figure 4.12 show tracking on the GAM with expression variations. On the top of the frames, we show the estimated pose of the face at the current frame. The pose is represented as a unit vector for the rotation axis, and the rotation angle in degrees, where the reference is taken to be the frontal face (i.e., we can get the rotation matrix $\mathbf{R} = e^{\hat{\omega}\theta}$). We did not require a texture-mapped 3D model as is common in many 3D model-based tracking methods. Our method outputs not only the 2D locations of the face (shown as the boxes in the figures) but also the 3D pose (shown on top of the figures), expression, and lighting parameters. In Figure

4.11 and 4.13, we show the estimates of the illumination, pose, identity and expression parameters as a function of time. The illumination parameter estimates are shown in terms of the norm of $\hat{\mathbf{l}}_t$, which indicates the intensity of the illumination. The larger the norm is, the brighter the illumination is. Pose parameters are shown as rotation vector $[\hat{\omega}_x, \hat{\omega}_y, \hat{\omega}_z]$. We see that the estimates of identity and expression are different in these two sequences, as should be expected. The key frames shown in Figs. 4.10 and 4.12 are marked using dash lines in Figures 4.11 and 4.13. We are able to obtain close to real-time performance using a MATLAB implementation.

Figure 4.14: Synthesized frames in the second row with the same motion and illumination from the first row. A generic face model is used for the synthesis

4.5.4 Application in Inverse Rendering:

We now show the accuracy of the motion and lighting estimates obtained from the GAM in novel view synthesis. The motion and illumination estimates from one video can be used for synthesizing the sequence of another object. This is termed as inverse rendering in the computer graphics community. Figure 4.14 shows an example, where the pose and

illumination estimates from the first row are used to synthesize the images in the second row.

4.6 Conclusions

In this chapter, we presented an accurate and efficient inverse compositional approach for estimating illumination, 3D motion and deformation from a video sequence using GAM. We proposed a new warping function, proved the converge of the IC approach, and showed experimental results on accuracy and computational efficiency. We presented experimental evaluation on controlled data with known ground truth, tracking results on real data and results in video synthesis.

Chapter 5

Video-based Face Recognition - An Analysis-by-Synthesis Framework

5.1 Introduction

It is believed by many that video-based face recognition systems hold promise in certain applications where motion can be used as a cue for face segmentation and tracking, and the presence of more data can increase recognition performance [101]. However, these systems have their own challenges. They require tracking the video sequence, as well as recognition algorithms that are able to integrate information over the entire video.

In this chapter, we present a novel *analysis-by-synthesis* framework for *pose and illumination invariant, video-based face recognition* that is based on (i) learning joint illumination and motion models from video, (ii) synthesizing novel views based on the learned parameters, and (iii) designing measurements that can compare two time sequences

while being robust to outliers. This is achieved by utilizing the bilinear pose and illumination model with the IC tracking algorithms presented above. We can handle a variety of lighting conditions, including the presence of multiple point and extended light sources, which is natural in outdoor environments (where face recognition performance is still relatively poor [101, 52, 53]). We can also handle gradual and sudden changes of lighting patterns over time. The pose and illumination conditions in the gallery and probe can be *completely disjoint*. We show experimentally that our method achieves high identification rates under extreme changes of pose and illumination. The main results were presented in [94].

5.1.1 Previous Work

The proposed approach touches upon aspects of face recognition, tracking and illumination modeling. We place our work in the context of only the most relevant ones.

A broad review of face recognition is available in [101]. Recently there have been a number of algorithms for pose and/or illumination invariant face recognition, many of which are based on the fact that the image of an object under varying illumination lies in a lower-dimensional linear subspace. In [99], the authors proposed a 3D Spherical Harmonic Basis Morphable Model (SHBMM) to implement a face recognition system given one single image under arbitrary unknown lighting. Another 3D face Morphable Model (3DMM) based face recognition algorithm was proposed in [10], but they use the Phong illumination model, estimation of whose parameters can be more difficult in the presence of multiple and extended light sources. The authors in [57] proposed to use Eigen Light-Fields and Fisher Light-Fields to do pose invariant face recognition. The authors in [45] introduced a

probabilistic version of Fisher Light-Fields to handle the differences of face images due to within-individual variability. Another method of learning statistical dependency between image patches was proposed for pose invariant face recognition in [55]. Correlation filters, which analyze the image frequencies, have been proposed for illumination invariant face recognition from still images in [69]. A novel method for multilinear independent component analysis was proposed in [85] for pose and illumination invariant face recognition.

All of the above methods deal with recognition in a single image or across discrete poses and do not consider continuous video sequences. Video-based face recognition requires integrating the tracking and recognition modules and exploitation of the spatio-temporal coherence in the data. The authors in [41] deal with the issue of video-based face recognition, but concentrate mostly on pose variations. Similarly [42] used adaptive Hidden Markov Models for pose-varying video-based face recognition. The authors of [18] proposed to use a 3D model of the entire head for exploiting features like hairline and handled large pose variations in head tracking and video-based face recognition. However, the application domain is consumer video and requires recognition across a few individuals only. The authors in [2] proposed to perform face recognition by computing the Kullback-Leibler divergence between testing image sets and a learned manifold density. Another work in [1] learns manifolds of face variations for face recognition in video. A method for video-based face verification using correlation filters was proposed in [88], but the pose in the gallery and probe have to be similar.

Except [18] (which is not aimed at face recognition on large datasets), all the rest are 2D approaches, in contrast to our 3D model-based method. The advantage of

using 3D models in face recognition has been highlighted in [12], but their focus is on acquiring 3D models directly from the sensors. The main reason for our use of 3D models is invariance to large pose changes and more accurate representation of lighting compared to 2D approaches. We do not need to learn models of appearance under different pose and illumination conditions. *This makes our recognition strategy independent of training data needed to learn such models, and allows the gallery and probe conditions to be completely disjoint.*

There are numerous methods for tracking objects in video in the presence of illumination changes [37, 25, 32, 38, 16]. However, most of them *compensate* for the illumination conditions of each frame in the video (as opposed to *recovering* the illumination conditions). In [8] and [60], the authors independently derived a low order (9D) spherical harmonics based linear representation to accurately approximate the reflectance images produced by a Lambertian object with attached shadows. In [27, 59], the authors discussed the advantage of this 3D model-based illumination representation compared to some image-based representations. Their methods work only for a single image of an object that is fixed relative to the camera, and do not account for changes in appearance due to motion. In this chapter, we show how the IC estimation algorithm of Chapter 4 can be applied for video-based face recognition.

5.1.2 Overview of the Approach

The underlying concept of this chapter is a method for learning joint illumination and motion models of objects from video. We assume that a 3D model of each face in the

gallery is available. For our experiments, the 3D model is estimated from images, but any 3D modeling algorithm, including directly acquiring the model through range sensors, can be used for this purpose. Given a probe sequence, we track the face automatically in the video sequence under arbitrary pose and illumination conditions using the bilinear model of the illumination and motion we developed in (2.3) in Section 2.2.3. This is achieved by the inverse compositional estimation algorithm in Section 4.3.4. The illumination-invariant model based tracking algorithm allows us not only to estimate the 3D motion, but also *recover* the illumination conditions as a function of time. The learned illumination parameters are used to synthesize video sequences for each gallery under the motion and illumination conditions in the probe. The distance between the probe and synthesized sequences is then computed for each frame. Different distance measurements are explored for this purpose. Next, the synthesized sequence that is at a minimum distance from the probe sequence is computed and is declared to be the identity of the person.

Experimental evaluation is carried out on a database of 57 people that we collected for this purpose. We compare our approach against other image-based and video-based face recognition methods. One of the challenges in video-based face recognition is the lack of a good dataset, unlike in image-based approaches [101]. The dataset in [41] is small and consists mostly of pose variations. The dataset described in [51] has large pose variations under constant illumination, and illumination changes in (mostly) fixed frontal/profile poses (these are essentially for gait analysis). The XM2VTS dataset (http:// www.ee.surrey.ac.uk /CVSSP/xm2vtsdb/) does not have any illumination variations, which is one of the main contributions of our work. An ideal dataset for us would be similar to the CMU PIE dataset

[75], but with video sequences instead of discrete poses. This is the reason why we collected our own data, which has large, simultaneous pose, illumination and expression variations. It is similar to the PIE dataset though the illumination change is random and uses pre-existing and natural indoor and outdoor lighting.

5.1.3 Contributions

The following are the main contributions of this chapter.

- We propose an *analysis-by-synthesis* framework for video-based face recognition that can work with large pose and illumination changes that are normal in natural imagery.
- We propose different metrics to obtain the identity of the individual in a probe sequence by integrating over the entire video and compare their merits and demerits.
- Our overall strategy does not require learning an appearance variation model, unlike many existing methods [1, 41, 42, 2, 85, 88]. Thus, the proposed strategy is not dependent on the quality of the learned appearance model and can handle situations where the pose and illumination conditions in the probe are completely independent of the gallery and training data.
- We perform a thorough evaluation of our method against well-known image-based approaches like Kernel PCA + LDA [5] and 3D model-based approaches like 3DMM [10, 99].

5.2 Estimating Illumination and Motion Parameters from Video

In the previous chapters, we showed that, using the image appearance bilinear model

$$\mathcal{I}_{t_2} = \left(\mathcal{B}_{t_1} + \mathcal{C}_{t_1} \times_2 \begin{pmatrix} \Delta\mathbf{T} \\ \Delta\mathbf{\Omega} \end{pmatrix} \right) \times_1 \mathbf{l}, \tag{5.1}$$

of Section 2.2.3 and introducing the warping operator $\mathbf{W_p} : \mathbb{R}^\mathbf{2} \rightarrow \mathbb{R}^\mathbf{2}$ of Section 4.3.2, we can estimate the pose and illumination parameters in an inverse compositional framework as

$$\hat{\mathbf{l}} = (\mathcal{B}_{\mathbf{p}_j} \mathcal{B}_{\mathbf{p}_j}^\mathbf{T})^{-1} \mathcal{B}_{\mathbf{p}_j}^\mathbf{T} \tilde{\mathcal{I}}_{t_{(1)}}^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{p_j} - \hat{\mathbf{p}}_{t-1} - \mathbf{m})}, \tag{5.2}$$

$$\text{and} \triangle\hat{\mathbf{m}} = \left[\mathcal{G}_{\mathbf{p_j}} \mathcal{G}_{\mathbf{p_j}}^\mathbf{T} \right]^{-1} \mathcal{G}_{\mathbf{p_j}} (\tilde{\mathcal{I}}_t^{\mathbf{W}_{\hat{\mathbf{p}}_{t-1}}(\mathbf{p}_j - \hat{\mathbf{p}}_{t-1} - \mathbf{m})} - \mathcal{B}_{\mathbf{p}_j} \times_1 \hat{\mathbf{l}}), \tag{5.3}$$

where

$$\mathcal{G}_{\mathbf{p_j}} = \mathcal{C}_{\mathbf{p_j}} \times_1 \hat{\mathbf{l}}, \tag{5.4}$$

$\mathbf{m} \triangleq [\mathbf{\Delta T}^\mathbf{T}, \mathbf{\Delta \Omega}^\mathbf{T}]^\mathbf{T}$, $\triangle\mathbf{m}$ is the change of $\mathbf{m}$ in each iteration, and $\mathbf{m}$ will be updated iteratively with $\mathbf{m} \leftarrow \mathbf{m} + \triangle\mathbf{m}$. Pose is represented with $\mathbf{p} = (\mathbf{T}^\mathbf{T}, \mathbf{\Omega}^\mathbf{T})^\mathbf{T} \in \mathbb{R}^6$, and $\mathbf{p}_j$ is called cardinal pose (see Section 4.3.4).

We select a set of cardinal poses $\{\mathbf{p}_j\}$ with interval of 20 degrees in pan and tilt angles, and precompute the bases $\mathcal{B}$ and $\mathcal{C}$ at these poses. All frames that are close

to a particular pose $\mathbf{p}_j$ will use the $\mathcal{B}$ and $\mathcal{C}$ at that pose, and the warp $\mathbf{W}_{\hat{\mathbf{p}}_{\mathbf{t_1}}}$ should be performed to normalize the pose to $\mathbf{p}_j$. The pictorial representation of the inverse compositional tracking scheme is shown in Fig. 4.2. While most of the existing inverse compositional methods move the expensive update steps out of the iterations for two-frame matching, we go even further and perform these expensive computations only once every few frames (Section 4.3.4). This is by virtue of the fact that we estimate 3D motion.

5.3 Face Recognition From Video

We now explain the face recognition algorithm and analyze the importance of different measurements for integrating the recognition performance over a video sequence. In our method, the gallery is represented by a textured 3D model of the face. The model can be built from a single image [11], a video sequence [64] or obtained directly from 3D sensors [12]. In our experiments, the face model will be estimated from the gallery video sequence for each individual. Face texture is obtained by normalizing the illumination of the first frame in the gallery sequence to an ambient condition, and mapping it onto the 3D model. Given a probe sequence, we will estimate the motion and illumination conditions using the algorithms described in Section 5.2. Note that the tracking does not require a person-specific 3D model - a generic face model is usually sufficient. Given the motion and illumination estimates, we will then render images from the 3D models in the gallery. The rendered images can then be compared with the images in the probe sequence. For this purpose, we will design robust measurements for comparing these two sequences. A feature of these measurements will be their ability to integrate the identity over all the frames,

ignoring some frames that may have the wrong identity.

Let $I_i, i = 0, ..., N-1$ be the ith frame from the probe sequence. Let $S_{i,j}, i = 0, ..., N-1$ be the frames of the synthesized sequence for individual j, where $j = 1, ..., M$ and M is the total number of individuals in the gallery. Note that the number of frames in the two sequences to be compared will always be the same in our method. By design, each corresponding frame in the two sequences will be under the same pose and illumination conditions, dictated by the accuracy of the estimates of these parameters from the probes sequences. Let d_{ij} be the Euclidean distance between the i^{th} frames I_i and $S_{i,j}$. We now compare three distance measures that can be used for obtaining the identity of the probe sequence.

1. $ID = \arg\min_j \min_i d_{ij}, \quad (5.5)$

2. $ID = \arg\min_j \max_i d_{ij}, \quad (5.6)$

3. $ID = \arg\min_j \frac{1}{N} \sum_i d_{ij}. \quad (5.7)$

The first alternative computes the distance between the frames in the probe sequence and each synthesized sequence that are the most similar and chooses the identity as the individual with the smallest distance. The second distance measure can be interpreted as minimizing the maximum separation between the frames in the probe sequence and synthesized sequences. Both of these measures suffer from a lack of robustness, which can be critical for their performance since the correctness of the frames in the synthesized sequences depends upon the accuracy of the illumination and motion parameter estimates. For this purpose, we replace the max by the f^{th} percentile and the min (in the inner dis-

tance computation of 1) by the $(1-f)^{th}$ percentile. In our experiments, we choose f to be 0.8.

The third option (5.7) chooses the identity as the minimum mean distance between the frames in the probe sequence and each synthesized sequence. Under the assumptions of Gaussian noise and uncorrelatedness between frames, this can be interpreted as choosing the identity with the maximum a-posterior probability given the probe sequence.

As the images in the synthesized sequences are pose and illumination normalized to the ones in the probe sequence, d_{ij} can be computed directly using the Euclidean distance. Other distance measurements, like [70, 3], can be considered in situations where the pose and illumination estimates may not be reliable or in the presence of occlusion and clutter. We will look into such issues in our future work.

5.3.1 Video-Based Face Recognition Algorithm

Using the above notation, let $I_i, i = 0, ..., N-1$ be N frames from the probe sequence. Let $G_1, ..., G_M$ be the 3D models with texture for each of M galleries.

• **Step 1.** Register a 3D generic face model to the first frame of the probe sequence. This is achieved using the method in [89]. Estimate the illumination and motion model parameters for each frame of the probe sequence using the method described in Section 5.2.

• **Step 2.** Using the estimated illumination and motion parameters, synthesize, for each gallery, a video sequence using the generative model of (5.1). Denote these as $S_{i,j}, i = 1, ..., N$ and $j = 1, ..., M$.

• **Step 3.** Compute d_{ij} as above.

• **Step 4.** Obtain the identity using a suitable distance measure as in (5.5) or (5.6) or (5.7).

5.4 Experimental Results

5.4.1 Face Database and Experimental Setup

Our database consists of videos of 57 people. Each person was asked to move his/her head as they wished (mostly rotate their head from left to right, and then from down to up), and the illumination was changed randomly. The illumination consisted of ceiling lights, lights from the back of the head and sunlight from a window on the left side of the face. Random combinations of these were turned on and off and the window was controlled using dark blinds. There was no control over how the subject moves his/her head or on facial expression. Sample frames of these video sequences are shown in Figure 5.1. The images are scale normalized and centered. Some of the subjects had expression changes also, e.g., the last row of the Figure 5.1. The average size of the face was about 70 x 70, with the minimum size being 50 x 50. Videos are captured with uniform background. We recorded 2 to 3 sessions of video sequences for each individual. All the video sessions are recorded within one week. The first session is used as the gallery for constructing the 3D textured model of the head, while the remaining are used for testing. We used a simplified version of the method in [64] for this purpose. We would like to emphasize that any other 3D modeling algorithm would also have worked. Texture is obtained by normalizing the illumination of the first frame in each gallery sequence to an ambient illumination condition,

and mapping onto the 3D model.

Figure 5.1: Sample frames from the video sequences collected for our database (best viewed on a monitor).

As can be seen from Figure 5.1, the pose and illumination vary randomly in the video. For each subject, we designed three experiments by choosing different probe sequences:

Expt. A: A video was used as the probe sequence with the average pose of the face in the video being about 15° from frontal;

Expt. B: A video was used as the probe sequence with the average pose of the face in the video being about 30° from frontal;

Expt. C: A video was used as the probe sequence with the average pose of the face in the video being about 45° from frontal.

Each probe sequence has about 20 frames around the average pose. The variation

of pose in each sequence was less than 15°, so as to keep pose in the experiments disjoint. The probe sequences are about 5 seconds each. This is because we wanted to separate the probes based on pose of the head (every 15 degrees) and it does not take the subject more than 5 seconds to move 15 degrees when continuously rotating the head. To show the benefit of video-based methods over image-based approaches, we designed three new Expts. D, E and F by taking random single images from A, B and C respectively.

5.4.2 Tracking and Synthesis Results

The results on tracking and synthesis on two of the probe sequences in our database (described next) are shown in Figure 5.2. The inverse compositional tracking algorithm can track about 20 frames per second on a standard PC using a MATLAB implementation. Real-time tracking could be achieved through better software and hardware optimization.

5.4.3 Recognition Results

We plot the Cumulative Match Characteristic (CMC) [101, 52] for experiments A, B, and C with measurement 1 (5.5), measurement 2 (5.6), and measurement 3 (5.7) in Figure 5.3. In Expt. A, where pose is 15° away from frontal, all the videos with large and arbitrary variations of illumination are recognized correctly. In Expt. B, we achieve about 95% recognition rate, while for Expt. C it is 93% using the distance measure (5.5). Irrespective of the illumination changes, the recognition rate decreases consistently with large difference in pose from frontal (which is the gallery), a trend that has been reported by other authors [10, 99]. *Note that the pose and illumination conditions in the probe*

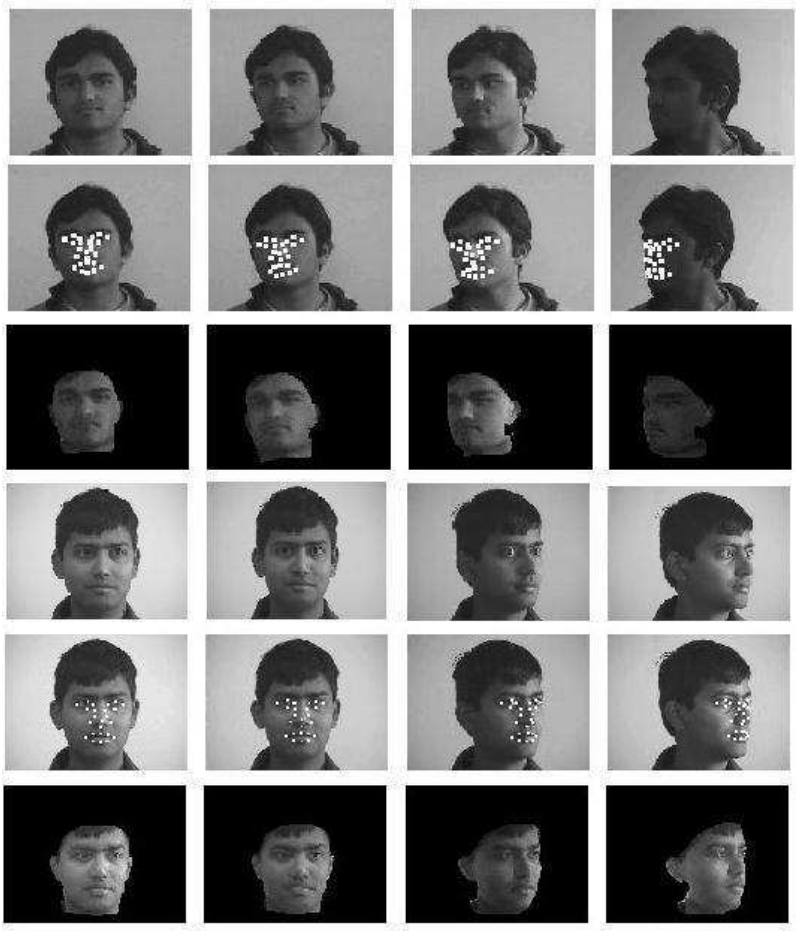


Figure 5.2: Original images, tracking and synthesis results are shown in three successive rows for two of the probe sequences.

and gallery sets can be completely disjoint.

5.5 Performance Analysis

5.5.1 Performance with changing pose

Figures 5.3 (a), (b) and (c) show the recognition rate with the measurements in (5.5), (5.6), and (5.7). Measurement 1 in (5.5) gives the best result. This is consistent

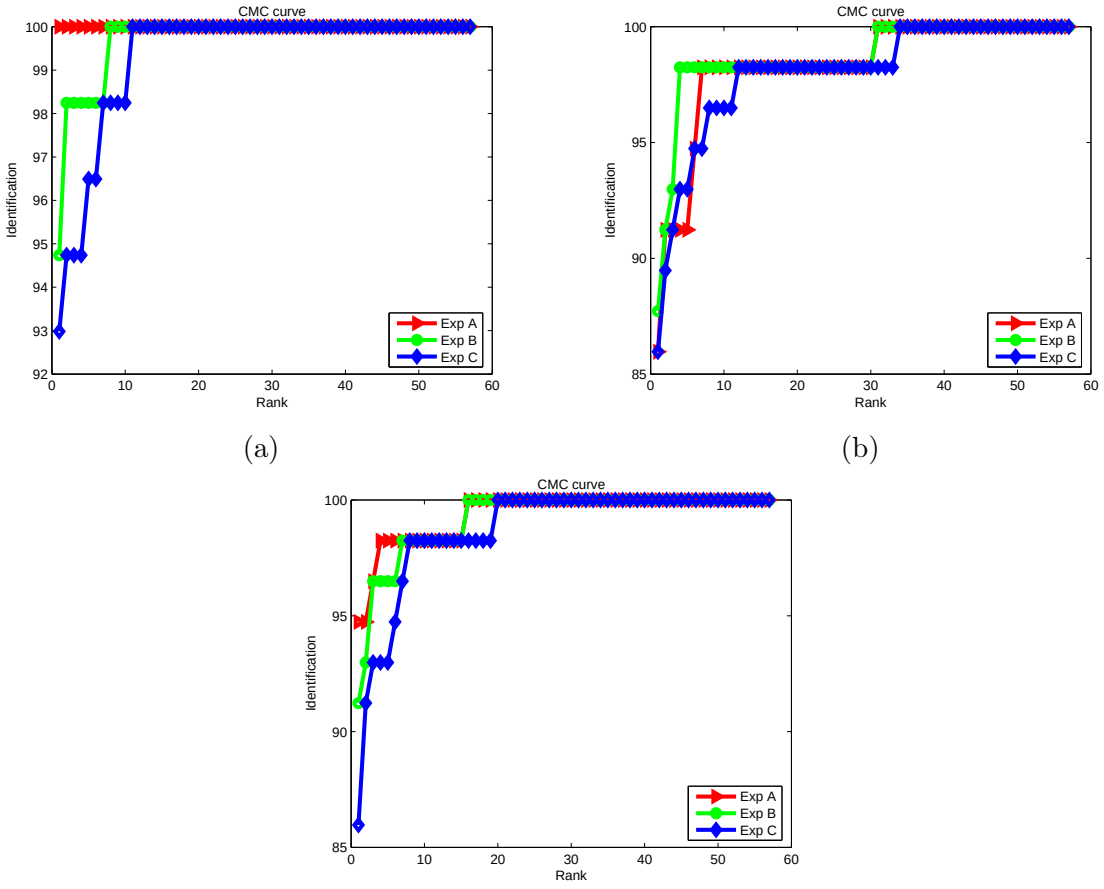


Figure 5.3: CMC curve for video-based face recognition experiments A to C. (a): with distance measure 1 in (5.5); (b): with distance measure 2 in (5.6); (c): with distance measure 3 in (5.7).

with our expectation, as (5.5) is not affected by the few frames in which the motion and illumination estimation error is relatively high. The recognition result is affected mostly by registration error which increases with non-frontal pose (i.e. A $\rightarrow$ B $\rightarrow$ C). On the other hand, measurement 2 in (5.6) is mostly affected by the errors in the motion and illumination estimation and registration, and thus the recognition rate in Fig. 5.3 (b) is lower than that of Fig. 5.3(a). Ideally, measurement 3 should give the best recognition rate as this is the

MAP estimation. However, the assumptions of Gaussianity and uncorrelatedness may not be valid. This affects the recognition rate for measurement 3, causing it perform worse than measurement 1 (5.5) but better than measurement 2 (5.6). We also found that small errors in 3D shape estimation have negligible impact on the motion and illumination estimates and the overall recognition result.

5.5.2 Effect of registration and tracking errors

There are two major error sources: registration and motion/illumination estimation. The error in registration may affect the motion and illumination estimation accuracy in subsequent frames, while robust motion and illumination estimation may regain tracking back after some time if the registration errors are small.

In Figure 5.4 (a), (b) and (c), we show the plots of error curves under three different cases. Figure 5.4 (a) is the ideal case, in which the registration is accurate and the error in motion and illumination estimation is consistently small through the whole sequence. The distance d_{ik} from the probe sequence I_i with the true identity k to the synthesized sequence with the correct model $S_{i,k}$, will always be smaller than $d_{ij}, j = 1, \ldots k-1, k+1, \ldots M$. In this case, all the measurements 1, 2 and 3 in (5.5), (5.6) or (5.7) will work. In the case shown in Figure 5.4 (b), the registration is correct but the error in the motion and illumination estimation accumulates. Finally, the drift error causes d_{ik}, the distance from the probe sequence to the synthesized sequence with the correct model (shown in bold red) to be higher than some other distance $d_{ij}, j \neq k$ (shown in green). In this case, measurement 2 in (5.6) will be wrong but measurements 1 and 3 in (5.5) or (5.7) still work. In Figure 5.4

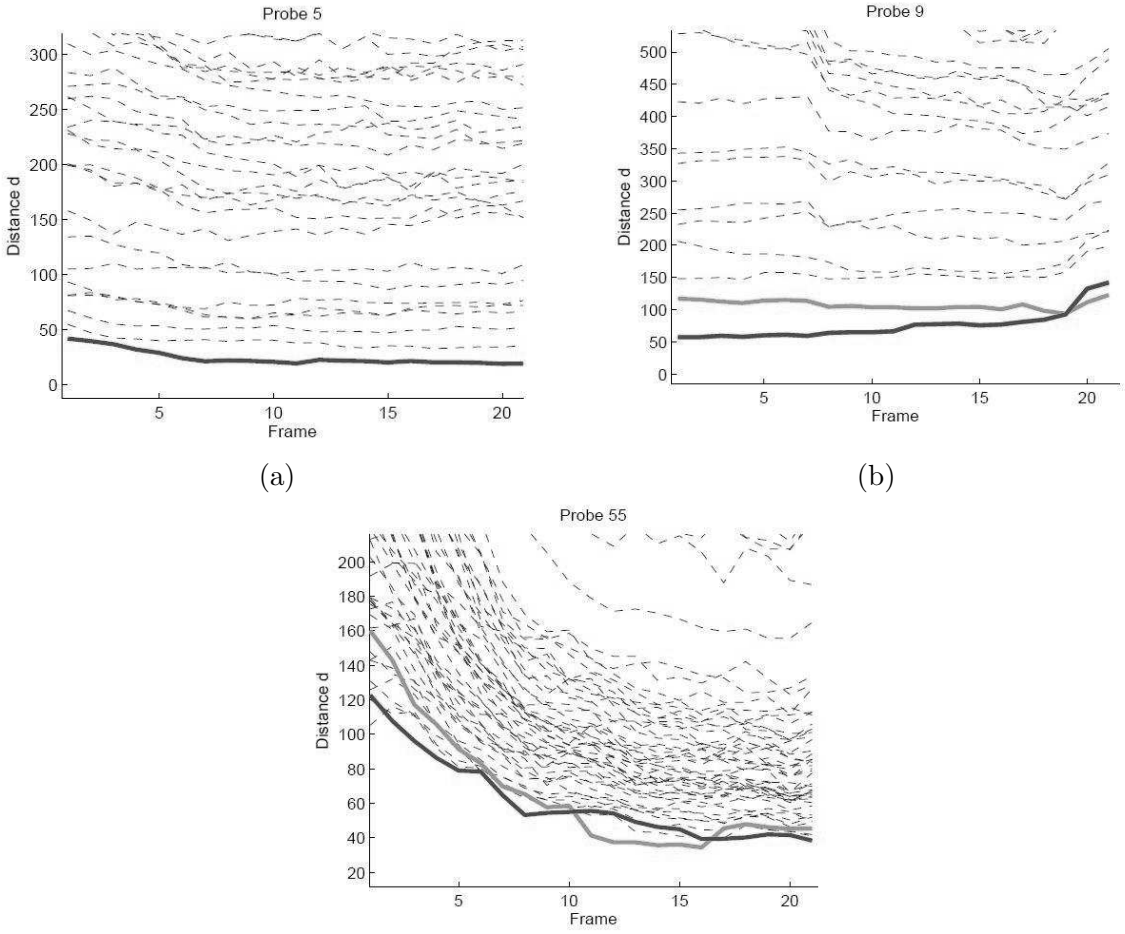


Figure 5.4: The plots of error curves under three different cases: (a) both registration and motion/illumination estimation are correct; (b) registration is correct but motion/illumination estimation has drift error; (c) registration is inaccurate, but robust motion/illumination estimation can regain tracking after a number of frames. The black, bold curve shows the distance of the probe sequence with the synthesized sequence of the correct identity, while both the gray bold and dotted curves show the distance with the synthesized sequences using the incorrect identity.

(c), the registration is not accurate (the error d_{ik} at the first frame is significantly higher than in (a) and (b)), but the motion and illumination estimation is able to regain tracking after a number of frames where the error decreases. Under this case, both measurements

1 and 2 in (5.5) (5.6) will not work, as it is not any individual frame that reveals the true identity, but the behavior of the error over the collection of all frames. Measurement 3 in (5.7) computes the overall distance by taking every frame into consideration, thus it works in such cases. This shows the importance of using different distance measurements based on the application scenario. Also, the effect of obtaining the identity by integrating over time is seen. It should be noted that the choice of distance measures may need to be revisited in an application scenario consisting of a much larger dataset.

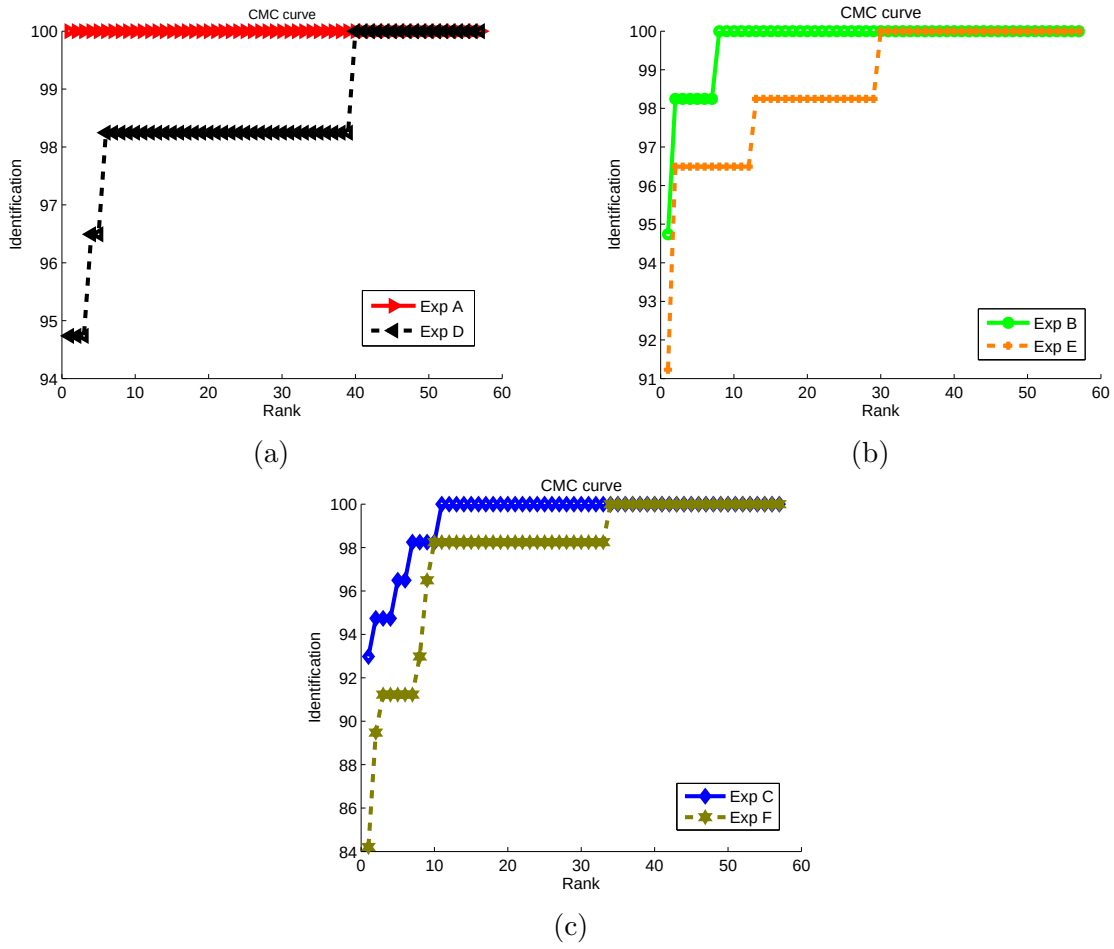


Figure 5.5: Comparison between the CMC curves for the video-based face experiments A to C with distance measurement 1 against SHBMM method of [99].

5.6 Comparison with other Approaches

The area of video-based face recognition is less standardized than image-based approaches. There is no standard dataset on which both image and video-based methods have been tried, thus we do the comparison on our own dataset. This dataset can be used for such comparison by other researchers in the future.

5.6.1 Comparison with 3DMM based approaches

3DMM has achieved a significant impact in the face biometrics area, and obtained impressive results in pose and illumination varying face recognition. It is similar to our proposed approach in the sense that both methods are 3D approaches, estimate the pose, illumination, and do synthesis for recognition. However, 3DMM [10] method uses the Phong illumination model, thus it cannot model extended light sources (like the sky) accurately. To overcome this, the authors in [99] proposed the SHBMM (3D Spherical Harmonics Basis Morphable Model) that integrates the spherical harmonics illumination representation into the 3DMM. Also, 3DMM and SHBMM methods have been applied to single images only. Although it is possible to repeatedly apply 3DMM or SHBMM approach to each frame in the video sequence, it is inefficient. Registration of the 3D model to each frame will be needed, which requires a lot of computation and manual work. None of the existing 3DMM approaches integrate tracking and recognition. Our proposed method, which integrates 3D motion into SHBMM, is a unified approach for modeling lighting and motion in a face video sequence.

Using our dataset, we now compare our proposed approach against the SHBMM

method of [99], which was shown to give better results than 3DMM in [10]. We will also compare our results with the published results of SHBMM method [99] in the later part of this section.

Recall that we designed three new Expts. D, E and F by taking random single images from A, B and C respectively. In Figure 5.5, we plot the CMC curve with measurement 1 in equation (5.5) (which has the best performance for Expt. A, B and C) for the Expts. D, E, F and compare them with the ones of the Expt. A, B, and C. The image-based approach recognition was achieved by integrating spherical harmonics illumination model with the 3DMM (which is essentially the idea in SHBMM [99]) on our data. For this comparison, we randomly chose images from the probe sequences of Expts. A, B, C and computed the recognition performance over multiple such random sets. Thus the Expts. D, E and F average the image-based performance over different conditions. By analyzing the plots in Figure 5.5, we see that the recognition performance with the video-based approach is consistently higher than the image-based one, both in Rank 1 performance as well as the area under the CMC curve. This trend is magnified as the average facial pose becomes more non-frontal. Also, we expect that registration errors, in general, will affect image-based methods more than video-based methods (since robust tracking maybe able to overcome some of the registration errors, as shown in section 4.4).

It is interesting to compare these results against the results in [99], for image-based recognition. The size of the databases in both cases is close (though ours is slightly smaller). Our recognition rate with a video sequence at average 15 degrees facial pose (with a range of 15 degrees about the average) is 100%, while the average recognition rate for approximately

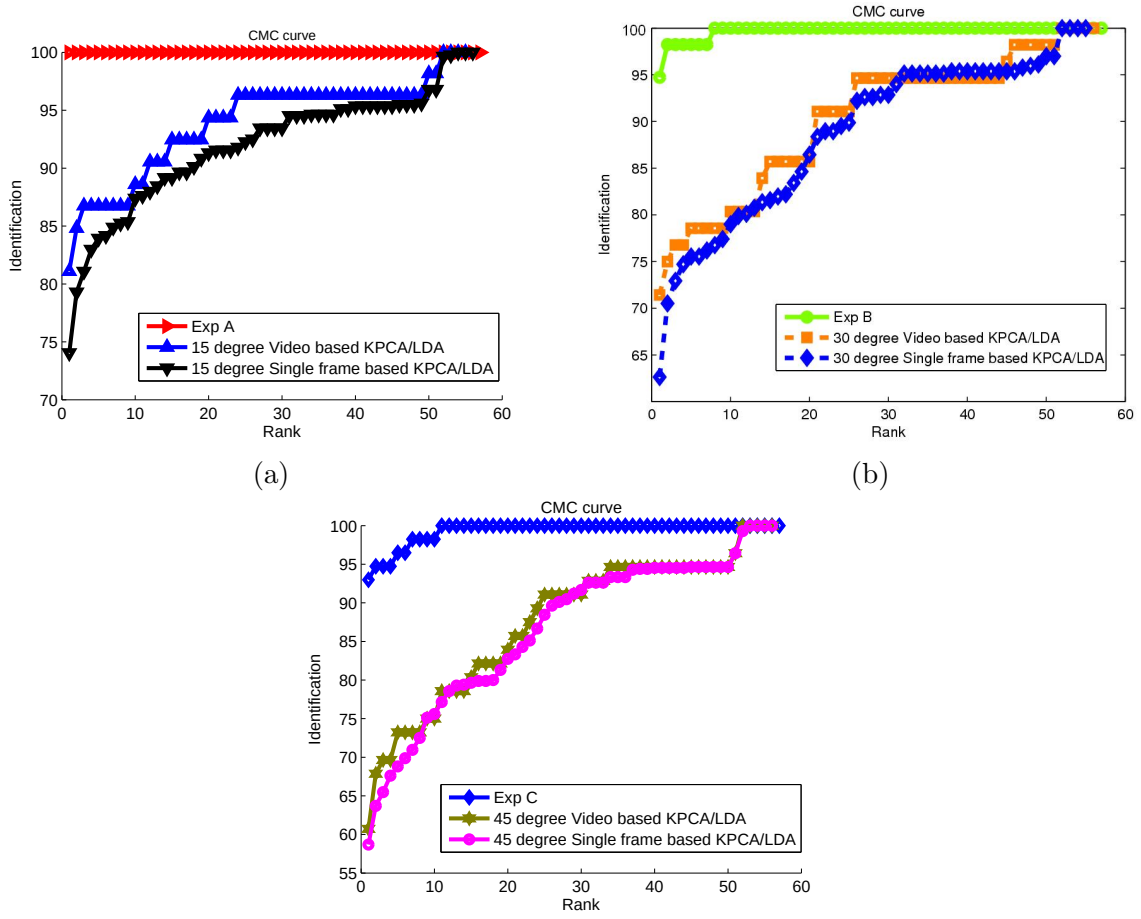


Figure 5.6: Comparison between the CMC curves for the video-based face experiments A to C with distance measurement 1 in (5.5) against KPCA+LDA based 2D approaches.

20 degrees (called side view) in [99] is 92.4%. For the Exp. B and C, [99] does not have comparable cases and goes directly to profile pose (90 degrees), which we don't have. Our recognition rate at 45° average pose is 93%. In [99], the quoted rates at 20° is 92% and at 90° is 55%. Thus the trend of our video-based recognition results are significantly higher than image-based approaches that deal with both pose and illumination variations.

We would like to emphasize that the above paragraph shows a comparison of recognition rates on two different datasets. While this may not seem completely fair, we are

constrained by the lack of a standard dataset on which to compare image- and video-based methods. We have shown a comparison on our dataset using our implementation in Fig. 5.5. The objective of the above paragraph is just to point out some trends with published results on other datasets that do not have video - these should be taken as very definitive statements.

5.6.2 Comparison with 2D approaches

In addition to comparing with 3DMM based methods, we also do the comparison against traditional 2D methods. We choose the Kernel PCA [5] based approaches as it has performed quite well in many applications. [1] In the training phase, we applied KPCA using the polynomial kernel and decrease the dimension of the training samples to 56. Then multi-class LDA is used for separating between different people. For each individual, we use the same images that we used for constructing the 3D shape in our proposed 3D approach as the training set. With this KPCA/LDA approach, we tested the recognition performance using single frames and the whole video sequences.

When we have a single frame as probe, we use k-Nearest Neighbor for the recognition, while in the case of video sequence, we compute the distance from every frame in the probe sequence to the centroid of the training samples in each class, take the summation over time, and then rank the distance of the sequence to each class. Here we show the results of recognition with the described 2D approach using single frames and video sequences about 15 degree (comparable to Exps. A and D.), 30 degree (comparable to Exps. B and

[1] We downloaded the Kernel PCA code from http://asi.insa-rouen.fr/ arakotom/toolbox/index.html, and implemented the Kernel PCA with the LDA in Matlab.

E.), and 45 degree (comparable to Exps. C and F.) in Fig. 5.6. For the comparison, we also show the results of our approach with video sequences in Exps. A, B, and C. Note that testing frames and sequences are the same as those used in Exps. A/B/C and D/E/F. Since 2D approaches cannot model the pose and illumination variation well, the recognition results are much worse compared to 3D approaches under arbitrary pose and illumination variation. However, we can still see the advantage of integrating the video sequences in Fig. 5.6.

5.7 Conclusions

In this chapter, we proposed an "analysis-by-synthesis" framework for video-based face recognition that relies upon the analytical image appearance model for integrating illumination and motion for describing the appearance of a video sequence. We started with a brief exposition of this theoretical result, followed by methods for learning the model parameters. Then, we described our recognition algorithm that relies on synthesis of video sequences under the conditions of the probe. We collected a face video database consisting of 57 people with large and arbitrary variation in pose and illumination, and demonstrated the effectiveness of the method on this new database. A detailed analysis of performance are also carried out. Future work on video-based face recognition will require experimentation on large datasets, design of suitable metrics and tight integration of the tracking and recognition phases.

Chapter 6

Conclusions and Future Work

In this thesis, we derived an analytical image appearance model, showed the applications of it in facial image modeling, efficient pose and illumination estimation, and proposed a new framework of video-based face recognition based on this theory.

We analyzed the accuracy of linear and multi-linear object representation models from the fundamental physical laws, and proved that the image appearance space is multilinear, with the illumination and texture subspaces being trilinearly combined with the direct sum of the motion and deformation subspaces. Using this result, we discussed the validity of many of the linear and multi-linear approaches existing in the computer vision literature, including PCA, AAM/ASM, MLM, locally linear models and 3DMM.

To combine the accuracy of the analytical methods with the robustness of statistical methods, we showed how to combine the analytically derived geometrical model with the statistical data analysis methods to obtain a Geometry-Integrated Appearance Manifold. GAM is a quadrilinear space of object appearance that can represent the effects of

illumination, motion, identity and deformation. The comparison of the size of the training set for GAM with other methods shows that GAM can be learned with a significantly less number of training samples. We proposed an accurate and efficient inverse compositional approach for estimating the illumination, 3D motion and deformation parameters from a video sequence using GAM. We proved the convergence of the IC approach, and both the computational analysis and experimental results showed significant savings on the computational cost.

As an application, we proposed a method for video-based face recognition that relies upon the analytical image appearance model for describing the appearance of a video sequence. We collected a face video database consisting of 57 people with large and arbitrary variation in pose and illumination, and demonstrated the effectiveness of the method on this new database. Detailed analysis of performance was also carried out.

One potential improvement of the analytical image appearance model will be looking for the efficient bases representing the shape and texture variation. In the derivation in Chapter 2, we used the general purpose 2D cosine bases. However, when we are considering the model of a specific object, like face, usually the shape and texture variation will follow some fixed patterns. Using the general purpose bases for modeling such variations will lead to the requirement of a large number of bases to capture a satisfactory percentage of the variation energy. A set of potentially better bases can be found by applying PCA to the training shape and texture for obtaining a set of orthonormal bases. In addition, some comparison between the analytically constructed deformation and texture variation bases with the statistically learned identity and expression bases would be helpful.

Another potential research direction will be the application of GAM in face recognition. The video-based face recognition results in this thesis are achieved by utilizing the multi-linear pose and illumination model with the IC tracking algorithm. By integrating GAM into this framework, we can not only simultaneously estimate the illumination and pose parameters, but there is also the possibility of simultaneous identity and expression recognition under varying illumination.

Another improvement can be in learning specific distance metrics to compute distance between two or more face video sequences. Along this direction, the pose and illumination estimates can be replaced with other methods. Our video-based face recognition algorithm can be applied to unconstrained scenarios like surveillance videos. A future application is to track people in a camera network by using a combination of appearance features with pose and illumination invariant face recognition.

Appendix A

Basic Tensor Operations

Tensor is the high dimensional generalization of vector and matrix, widely used in multilinear algebra. In this thesis, we used tensor notations and operations to make the multilinear equations brief and succinct. Like vector and matrix, tensor has its own operations. Below are the two tensor operations we used in this thesis

Mode-N Product: The *mode-n product* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n \times \ldots \times I_N}$ by a vector $\mathbf{V} \in \mathbb{R}^{1 \times I_n}$, denoted by $\mathcal{A} \times_n \mathbf{V}$, is the $I_1 \times I_2 \times \ldots \times 1 \times \ldots \times I_N$ tensor

$$(\mathcal{A} \times_n \mathbf{V})_{i_1 \ldots i_{n-1} 1 i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} v_{i_n}.$$

Tensor Unfolding Operation: Assume an Nth-order tensor $\mathcal{A} \in \mathbf{C}^{I_1 \times I_2 \times \ldots \times I_N}$. The matrix unfolding $\mathbf{A}_{(n)} \in \mathbf{C}^{I_n \times (I_{n+1} I_{n+2} \ldots I_N I_1 I_2 \ldots I_{n-1})}$ contains the element $a_{i_1 i_2 \ldots i_N}$ at the position with row number i_n and column number equal to $(i_{n+1}-1) I_{n+2} I_{n+3} \ldots I_N I_1 I_2 \ldots I_{n-1} + (i_{n+2}-1) I_{n+3} I_{n+4} \ldots I_N I_1 I_2 \ldots I_{n-1} + \cdots + (i_N - 1) I_1 I_2 \ldots I_{n-1} + (i_1 - 1) I_2 I_3 \ldots I_{n-1} + \cdots + i_{n-1}$.

Appendix B

Derivation of (2.3)

For the moment, we assume that at time instance t_1 we know the 3D model of the object, its pose, and the illumination condition in terms of the coefficients $l_{ij}^{t_1}$, which is the each element of $\mathbf{l}$ at time instance t_1. We will first consider the case of Lambertian object, and then generalize to the condition stated in Assumption (A2). Without loss of generality, we also assume that the pixel (x, y) corresponds to the point $\mathbf{P_1}$ at t_1. Thus, from the Lambertian Reflectance Linear Subspace (LRLS) theory [8], we have the reflectance intensity for the pixel (x, y) as:

$$I(x, y, t_1) = \sum_{i=0,1,2} \sum_{j=-i,-i+1 \ldots i-1,i} l_{ij}^{t_1} b_{ij}(\mathbf{n_{P_1}}), \tag{B.1}$$

where i and j are the indicators of the spherical harmonics function order.

Let us define the the motion of the object in the above reference frame as the translation $\mathbf{T} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^T$ of the centroid of the object and the rotation $\mathbf{\Omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ about the centroid. At the new time instance t_2, the illumination can

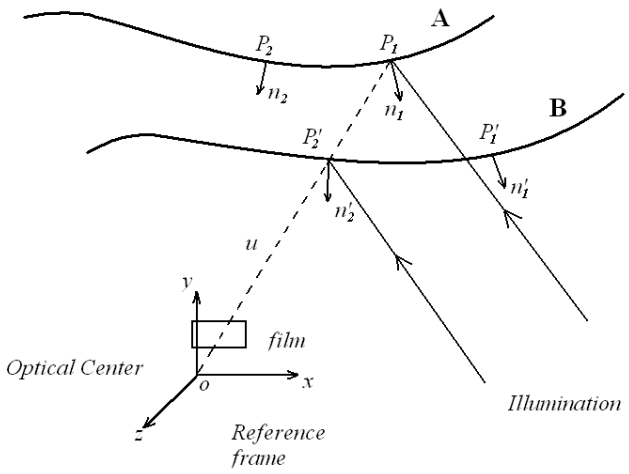


Figure B.1: Pictorial representation depicting imaging framework.

change and is represented in terms of the coefficients $l_{ij}^{t_2}$. We will now derive the relationship between $I(x, y, t_1)$, $I(x, y, t_2)$, $\mathbf{T}$, $\mathbf{\Omega}$, $l_{ij}^{t_1}$, and $l_{ij}^{t_2}$.

The overall derivation of the joint motion and illumination space will proceed as follows. We will first derive the new basis images taking into consideration the motion of the object. We show that the new bases are approximately of the form $(A\mathbf{T} + B\mathbf{\Omega})$, where A and B are suitably defined functions, the precise form of which we will derive. Next, incorporating the lighting parameters (which can be represented as a linear expansion using the LRLS theory), the joint motion and illumination space is shown to be bilinear.

B.1 Computation of the new basis image

Let A and B represent the same object before and after motion respectively, as shown in Fig. B.1. Consider the ray from the optical center to a particular pixel (x, y). We

can find its intersection with the surface of the object by extending the ray. With respect to the camera, the direction of this ray does not change. Before the object's motion, the ray intersects with the surface at $\mathbf{P_1}$ (on A), and after motion, it intersects at $\mathbf{P_2}'$ (on B). $\mathbf{P_1}$ (on A) moves to $\mathbf{P_1}'$ (on B), and $\mathbf{P_2}$ (on A) moves to $\mathbf{P_2}'$ (on B). Note that $\mathbf{P_2}'$ may not overlap with $\mathbf{P_1}$; they are just on the same projection ray. We will follow the convention of representing a point after motion with a prime ($'$).

We first define some notation required for our derivation. Let

$$\mathbf{J_{P_1}} = \mathbf{J}\left(\frac{\partial \mathbf{n_{P_1}}}{\partial \mathbf{P}}\right) \text{ and } \mathbf{\Delta} = \mathbf{P_2} - \mathbf{P_1} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix},$$

where $\mathbf{J_{P_1}}$ is the Jacobian matrix of the norm, $\mathbf{n_{P_1}}$, at point $\mathbf{P_1}$, with respect to $\mathbf{P} \triangleq (x, y, z)^T$, and $\mathbf{\Delta}$ is the difference in the coordinates of $\mathbf{P_2}$ and $\mathbf{P_1}$. Henceforth we will refer to $\mathbf{\Delta}$ as the coordinate change.

From (2.1) and (2.2), we see that when the illumination coefficients, l_{ij}, are known, only the norm and the albedo of the surface point of interest affects the reflection intensity at a particular pixel. The change in norm and albedo is obtained using the Jacobian matrix and gradient at the point of interest, as well as the coordinate change, which in turn is obtained from the motion information.

The norm changes from $\mathbf{P_1}$ to $\mathbf{P_2}$, and again from $\mathbf{P_2}$ to $\mathbf{P_2}'$. The first change is due to the fact that $\mathbf{P_2}$ is a different point on the surface, while the second change is due to the motion of the surface. Hence the difference of $\mathbf{n_{P_1}}$ and $\mathbf{n_{P_2'}}$ is a function of the spatial (from $\mathbf{n_{P_1}}$ to $\mathbf{n_{P_2}}$) and temporal (from $\mathbf{n_{P_2}}$ to $\mathbf{n_{P_2'}}$) changes. Using the coordinate change

$\mathbf{\Delta}$ and the Jacobian matrix of norm at $\mathbf{P_1}$, we are able to calculate the first order difference between $\mathbf{n_{P_1}}$ and $\mathbf{n_{P_2}}$. Using the motion information, we can obtain the difference between $\mathbf{n_{P_2}}$ and $\mathbf{n'_{P_2}}$. The albedo changes from $\mathbf{P_1}$ to $\mathbf{P_2}$, but is the same for $\mathbf{P_2}$ and $\mathbf{P_2}'$. Hence the difference of $\rho_{\mathbf{P_1}}$ and $\rho_{\mathbf{P'_2}}$ is a function of spatial coordinates only, and can be obtained using the gradient of albedo. We can express the change in norm and albedo upto a first order approximation as

$$\Delta\mathbf{n} = \mathbf{n_{P'_2}} - \mathbf{n_{P_1}} = \mathbf{J_{P_1}}\mathbf{\Delta} + \frac{\partial \mathbf{n_{P_2}}}{\partial t}\Delta t, \tag{B.2}$$

and

$$\Delta\rho = \rho_{\mathbf{P'_2}} - \rho_{\mathbf{P_1}} = \nabla\rho_{\mathbf{P_1}}\mathbf{\Delta}, \tag{B.3}$$

where $\nabla\rho_{\mathbf{P_1}}$ is the gradient of ρ at point $\mathbf{P_1}$. Thus, $\Delta\mathbf{n}$ and $\Delta\rho$ can be substituted into the expression for the basis images in (2.2), which can be rewritten as

$$\begin{aligned} b_{ij}(\mathbf{n_{P'_2}}) &= (\rho_{\mathbf{P_1}} + \Delta\rho) r_i Y_{ij}(\mathbf{n_{P_1}} + \Delta\mathbf{n}) \\ &= b_{ij}(\mathbf{n_{P_1}}) + \nabla\rho_{\mathbf{P_1}} r_i Y_{ij}(\mathbf{n_{P_1}})\mathbf{\Delta} \\ &\quad + \rho_{\mathbf{P_1}} r_i \nabla Y_{ij}(\mathbf{n_{P_1}})\Delta\mathbf{n} + o(\mathbf{\Delta}). \end{aligned} \tag{B.4}$$

The last term is a higher order term and can be ignored when $\mathbf{\Delta}$ is small. Substituting $\Delta\mathbf{n}$ from (B.2) into (B.4), we see that the basis image is a linear function of $\mathbf{\Delta}$.

$$\begin{aligned} b_{ij}(\mathbf{n_{P'_2}}) &= b_{ij}(\mathbf{n_{P_1}}) \\ &\quad + \left(\nabla\rho_{\mathbf{P_1}} r_i Y_{ij}(\mathbf{n_{P_1}})\mathbf{\Delta} + \rho_{\mathbf{P_1}} r_i \nabla Y_{ij}(\mathbf{n_{P_1}})\mathbf{J_{P_1}}\right)\mathbf{\Delta} \\ &\quad + \rho_{\mathbf{P_1}} r_i \nabla Y_{ij}(\mathbf{n_{P_1}})\frac{\partial \mathbf{n_{P_2}}}{\partial t}\Delta t + o(\mathbf{\Delta}). \end{aligned} \tag{B.5}$$

$\frac{\partial \mathbf{n}_{\mathbf{P_2}}}{\partial t}$ is not a function of $\mathbf{\Delta}$, as we will show latter in Section B.3. We next show how to solve for $\mathbf{\Delta}$.

B.2 Computation of coordinate change $\mathbf{\Delta}$

Since $\mathbf{P_2}'$ and $\mathbf{P_1}$ are on the same ray, we can represent the difference between them using a unit vector $\mathbf{u}$ under the perspective camera model, i.e.,

$$\mathbf{P_2}' - \mathbf{P_1} = k\mathbf{u}, \tag{B.6}$$

where

$$\mathbf{u} = \frac{1}{\sqrt{x^2 + y^2 + f^2}} \begin{pmatrix} x \\ y \\ f \end{pmatrix}, \tag{B.7}$$

and k is a scalar. Since the motion of the object is considered as a pure rotation with respect to its centroid and a pure translation of the centroid, the new coordinate of $\mathbf{P_2}$ can be expressed as

$$\mathbf{P_2}' = \mathbf{R}(\mathbf{P_2} - \mathbf{T_0}) + \mathbf{T_0} + \mathbf{T}, \tag{B.8}$$

where $\mathbf{R}$ is the Rodrigues rotation matrix obtained from the rotation $\mathbf{\Omega}$ with respect to the centroid, and $\mathbf{T_0}$ is the position of the centroid of the object. Substituting it into (B.6), we get

$$k\mathbf{u} = \mathbf{R}(\mathbf{P_2} - \mathbf{T_0}) + \mathbf{T_0} + \mathbf{T} - \mathbf{P_1}. \tag{B.9}$$

Under the assumption of small motion, we have an additional constraint. We may consider the new point $\mathbf{P_2}$ to be on the tangent plane that passes through the original intersection

point $\mathbf{P_1}$, i.e.,

$$\mathbf{n}_{\mathbf{P_1}}^T(\mathbf{P_1} - \mathbf{P_2}) = 0. \tag{B.10}$$

Using (B.9) and (B.10) and after some algebraic manipulation (see the appendices in [90]), we can show that

$$\begin{aligned}\boldsymbol{\Delta} &= \left(\mathbf{R}^{-1} - \mathbf{I}\right)(\mathbf{P_1} - \mathbf{T_0}) - \mathbf{R}^{-1}\mathbf{T} \\ &\quad -\mathbf{R}^{-1}\frac{\mathbf{n}_{\mathbf{P_1}}^T\left(\left(\mathbf{R}^{-1} - \mathbf{I}\right)(\mathbf{P_1} - \mathbf{T_0}) - \mathbf{R}^{-1}\mathbf{T}\right)}{\mathbf{n}_{\mathbf{P_1}}^T\mathbf{R}^{-1}\mathbf{u}}\mathbf{u}.\end{aligned} \tag{B.11}$$

The coordinate change, $\boldsymbol{\Delta}$, obtained in (B.11) captures the effect of the motion. However, as it is a nonlinear function of the object motion variables $\mathbf{T}$ and $\boldsymbol{\Omega}$, its complex form makes it difficult to analyze. Henceforth we will denote this as $\boldsymbol{\Delta}_{nl}$.

Since the motion is small, we can simplify the above equation using certain approximations that neglect terms with small magnitude with respect to terms with large magnitudes. This will allow us to interpret the joint effect of motion and illumination analytically, while sacrificing little in terms of accuracy. Using a series of mathematical calculations, we can obtain $\boldsymbol{\Delta}$ as a linear function of the motion variables (see the appendices in [90]) as:

$$\begin{aligned}\boldsymbol{\Delta} &\cong \hat{\mathbf{P}}\boldsymbol{\Omega} + \mathbf{T} - \frac{1}{\mathbf{u}^T\mathbf{n}_{\mathbf{P_1}}}\mathbf{u}\mathbf{n}_{\mathbf{P_1}}^T\hat{\mathbf{P}}\boldsymbol{\Omega} - \frac{1}{\mathbf{u}^T\mathbf{n}_{\mathbf{P_1}}}\mathbf{u}\mathbf{n}_{\mathbf{P_1}}^T\mathbf{T} \\ &= \left(\mathbf{I} - \frac{1}{\mathbf{n}_{\mathbf{P_1}}^T\mathbf{u}}\mathbf{u}\mathbf{n}_{\mathbf{P_1}}^T\right)\left(\hat{\mathbf{P}}\boldsymbol{\Omega} - \mathbf{T}\right) \\ &\triangleq \mathbf{C}(\hat{\mathbf{P}}\boldsymbol{\Omega} - \mathbf{T}),\end{aligned} \tag{B.12}$$

where $\hat{\mathbf{P}} = (\mathbf{P_1} - \mathbf{T_0})^{\wedge}$ [1]

We will refer to this as $\mathbf{\Delta}_l$. Henceforth, when we use $\mathbf{\Delta}$ we will refer to $\mathbf{\Delta}_l$; when required to be specific, we will mention $\mathbf{\Delta}_l$ or $\mathbf{\Delta}_{nl}$.

B.3 Temporal change of norm

In order to obtain the change of norm $\Delta\mathbf{n}$, we still need to compute the effect of temporal change on the right hand side of (B.2). Using the assumption of small motion, we can compute:

$$\begin{aligned} \frac{\partial \mathbf{n_{P_2}}}{\partial t}\Delta &= \frac{\partial(\mathbf{n_{P_1}} + \mathbf{J_{P_1}}\mathbf{\Delta})}{\partial t} = \mathbf{\Omega} \times (\mathbf{n_{P_1}} + \mathbf{J_{P_1}}\mathbf{\Delta}) \\ &= \mathbf{\Omega} \times \mathbf{n_{P_1}} + o(\mathbf{\Omega T}) \cong (-\mathbf{n_{P_1}})^{\wedge}\mathbf{\Omega} \\ &\triangleq -\hat{\mathbf{N}}\mathbf{\Omega}. \end{aligned} \tag{B.13}$$

As $\mathbf{\Delta}$ is a linear function of the motion variables $\mathbf{\Omega}$ and $\mathbf{T}$, the cross product of $\mathbf{\Omega}$ and $\mathbf{J_{P_1}}\mathbf{\Delta}$ is a second order term and can be ignored when the motion is small. Thus, the temporal change is not a function of $\mathbf{\Delta}$, a fact that was used in equation (B.5).

B.4 Bilinear space of motion and illumination

Substituting (B.12) and (B.13) into (B.2), we get a linear expression for $\Delta\mathbf{n}$ as a function of motion variables, i.e.,

$$\Delta\mathbf{n} = \left(\mathbf{J_{P_1}}\mathbf{C}\hat{\mathbf{P}} - \hat{\mathbf{N}}\right)\mathbf{\Omega} - \mathbf{J_{P_1}}\mathbf{CT}. \tag{B.14}$$

[1] We define the skew symmetric matrix of a vector $\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ as $\mathbf{X}^{\wedge} = \hat{\mathbf{X}} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}$.

So far, we have expressed the coordinate and norm change as linear expressions of the motion variables. Substituting (B.12) and (B.14) into (2.1) and (B.5), which contain the illumination variables, we have

$$I(x, y, t_2) = \sum_{i=0,1,2} \sum_{j=-i,-i+1 \ldots i-1,i} l_{ij}^{t_2} b_{ij}(\mathbf{n}_{\mathbf{P}'_2}), \tag{B.15}$$

where

$$b_{ij}(\mathbf{n}_{\mathbf{P}'_2}) = b_{ij}(\mathbf{n}_{\mathbf{P}_1}) + \mathbf{AT} + \mathbf{B\Omega}, \tag{B.16}$$

$$\mathbf{A} = -r_i \left(\nabla \rho_{\mathbf{P}_1} Y_{ij}(\mathbf{n}_{\mathbf{P}_1}) + \rho_{\mathbf{P}_1} \nabla Y_{ij}(\mathbf{n}_{\mathbf{P}_1}) \mathbf{J}_{\mathbf{P}_1} \right) \mathbf{C}, \tag{B.17}$$

and

$$\mathbf{B} = -\mathbf{A}\hat{\mathbf{P}} - r_i \rho_{\mathbf{P}_1} \nabla Y_{ij}(\mathbf{n}_{\mathbf{P}_1}) \hat{\mathbf{N}}. \tag{B.18}$$

In (B.16), $b_{ij}(\mathbf{n}_{\mathbf{P}'_2})$ are the basis images after motion. The first term, $b_{ij}(\mathbf{n}_{\mathbf{P}_1})$, are the original basis images before motion. They are only determined by the object model and do not change with the variation of illumination. The illumination change is reflected in the change of the coefficients from $l_{ij}^{t_1}$ to $l_{ij}^{t_2}$. The effect of the motion is reflected in $\mathbf{AT} + \mathbf{B\Omega}$, where the first term describes the effect of translation, and the second term describes the effect of rotation. Substituting (B.16) into (B.15), we see that the new image spans a bilinear space of the motion variables and illumination variables.

When the illumination changes gradually, we may use the Talyor series to approximate the illumination coefficients as $l_{ij}^{t_2} = l_{ij}^{t_1} + \Delta l_{ij}$. Ignoring the higher order terms, the bilinear space now becomes a combination of two linear subspaces, defined by the motion

and illumination variables.

$$I(x,y,t_2) = I(x,y,t_1) + \sum_{i=0,1,2} \sum_{j=-i,\ldots,i} l_{ij}^{t_1}(\mathbf{AT} + \mathbf{B\Omega}) + \sum_{i=0,1,2} \sum_{j=-i,\ldots,i} \Delta l_{ij} b_{ij}(\mathbf{n_{P_1}}). \quad \text{(B.19)}$$

If the illumination does not change from t_1 to t_2 (often a valid assumption for a short interval of time) , we see that the new image at t_2 spans linear space of the motion variables, since the third term in (B.19) is zero.

B.5 Discussion on the Theoretical Result

Physical Interpretation: This bilinear space result integrates the effects of illumination and motion in generating an image from a 3D object using a perspective camera. When the object does not move, the second and third motion terms of the basis image $b_{ij}(\mathbf{n_{P'_2}})$ are zero, and the result is the same as the one in [8], a 9D Lambertian Reflectance Linear Subspace. When the illumination remains the same, the reflectance image spans a linear subspace of motion variables. When the illumination and motion variables all change, the image space is "close to" bilinear. Thus the joint illumination and motion space for a sequence of images is bilinear with (approximately) nine illumination variables and six motion variables. The shape of the object is encoded in the $\mathbf{A}$ and $\mathbf{B}$ matrices, and in $b_{ij}(\mathbf{n_{P_1}})$. The camera intrinsic parameters are implicitly present in $\mathbf{\Delta}$ (thus in $\mathbf{A}$ and $\mathbf{B}$) through $\mathbf{u}$. Therefore, Equations (B.15) and (B.16) integrate the motion, illumination, 3D structure, albedo and camera intrinsic parameters into one single framework.

Generalizations of the theory: Even though the above result is derived using previous work on the LRLS theory, the basic result (i.e., the joint motion and illumination space is bilinear with the bases of this space determined by the surface normals and camera intrinsic parameters) is valid in more general circumstances. If we can write the image appearance as a linear dot product of lighting coefficients and basis images, and if the basis images change linearly with the 3D rigid motion parameters, the joint motion and illumination space will be bilinear. This could be achieved using higher order coefficients in the spherical harmonics representation of illumination or a different set of basis functions [82, 48]. However, for other basis functions, the precise form of the expression would have to be rederived, while using higher order spherical harmonics coefficients would require imposing additional constraints to enforce non-negativity of the lighting function (see [31] for details). Also, for glossy surfaces, the gradient of the albedo can have high frequency components which can affect the parameter estimates in scene understanding applications.

Effect of scale changes: To understand this, we consider that the motion is purely in the direction of the optical axis, i.e., zooming effect. Irrespective of how the objects moves, equation (B.9), is satisfied. Thus, even when the object moves towards the camera, the intersection points of a ray with the object surface at two consecutive time instances, should still be close to each other, provided the motion is small. Therefore, $\mathbf{P_2}$ can still be considered to be on the tangent plane passing through $\mathbf{P_1}$. So, equation (B.9) and (B.10) are satisfied, and the coordinate change $\mathbf{\Delta}$, which completely determines the change of norm and albedo, can be calculated accurately.

The motion of a plane: When the plane moves with pure translation, there is no difference between $\mathbf{n_{P_2}}$ and $\mathbf{n_{P_2'}}$ (in Figure B.1); thus the change of norm is completely due to the spatial component in (B.2). When the object plane moves with pure rotation confined to the image plane, the rotation axis is parallel to the norm on the object plane; thus $\mathbf{n_{P_2}}$ and $\mathbf{n_{P_2'}}$ are the same, so the change of norm again has only the spatial component. When the object plane purely rotates but the rotation is not confined to the image plane, there will be both spatial and temporal change of the norm. So, the change of norm, which determines the reflectance intensity, can be described by the theory. The albedo change is the same as in the main theory (see equation (B.3)).

Pixels for which the unit vector u is perpendicular to the norm: In this case, equation (B.9) is still satisfied; however, because the ray is now coplanar with the tangent plane passing through $\mathbf{P_1}$, there is an infinite number of solutions for equation (B.10). In implementing the theory, this affects all points for which the angle between the unit vector $\mathbf{u}$ and the norm $\mathbf{n_P}$ are very close to 90°, making the denominators in equations (B.11,B.12) very small. In this case, the two constraints (equations (B.9),(B.10)) for calculating the coordinate change $\mathbf{\Delta}$ become only one, and it is not possible to compute $\mathbf{\Delta}$. However, this happens only at a very few points near the object's edge (e.g., near the edge of a face) and is not a serious impediment to the application of the theory in practical problems. In the implementation, the value of the pixels where this happens is replaced with values from nearby pixels. This is not a shortcoming of our theory, since it is not possible to view a point if the viewing direction and the surface normal are perpendicular (there is no light

reflected along the viewing direction) .

Appendix C

Derivation of (2.14)

Substituting (2.11) into (2.10), we have

$$\mathcal{C}(u_2, v_2, t_2) - \mathcal{C}(u_1, v_1, t_1) = k\mathbf{R}^{-1}\mathbf{r}. \tag{C.1}$$

Substituting (2.12) and (2.13) into (C.1), we have

$$\alpha_u \mathcal{T}_u + \alpha_v \mathcal{T}_v + \mathbf{b}_d^{\mathbf{T}} \Phi_d(u_2, v_2)\mathcal{N}(u_2, v_2, t_1)\Delta t = k\mathbf{R}^{-1}\mathbf{r}. \tag{C.2}$$

Applying Taylor expansion, we have

$$\mathbf{b}_d^{\mathbf{T}} \Phi_d(u_2, v_2) = \mathbf{b}_d^{\mathbf{T}} \Phi_d(u_1, v_1) + \mathbf{b}_d^{\mathbf{T}} \nabla \Phi_d|_{u_1, v_1, t_1} \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix},$$

$$\mathcal{N}(u_2, v_2, t_1) = \mathcal{N}(u_1, v_1, t_1) + \mathbf{J}_{\mathcal{N}|u_1, v_1} \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix}. \tag{C.3}$$

Thus, $\mathbf{b}_d^{\mathbf{T}}\Phi_d(u_2,v_2)\mathcal{N}(u_2,v_2,t_1)$ can be expressed as

$$\left(\mathbf{b}_d^{\mathbf{T}}\Phi_d+\mathbf{b}_d^{\mathbf{T}}\nabla\Phi_d\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}\right)\left(\mathcal{N}+\mathbf{J}_{\mathcal{N}}\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}\right) = \mathbf{b}_d^{\mathbf{T}}\Phi_d\mathcal{N}+\mathbf{b}_d^{\mathbf{T}}\Phi_d\mathbf{J}_{\mathcal{N}}\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix} + \mathcal{N}\mathbf{b}_d^{\mathbf{T}}\nabla\Phi_d\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}+o(\alpha_u,\alpha_v), \quad \text{(C.4)}$$

where all the terms are computed at (u_1,v_1,t_1), and the last term is a high order term thus can be ignored. Substituting (C.4) into (C.2) we have

$$\begin{aligned}\mathbf{A}\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix} &= k\mathbf{R}^{-1}\mathbf{r}-\mathbf{b}^{\mathbf{T}}\Phi(u_1,v_1)\mathcal{N}(u_1,v_1,t_1), \text{where}\\ \mathbf{A} &= (\mathcal{T}_u,\mathcal{T}_v)+\mathbf{b}^{\mathbf{T}}\Phi(u_1,v_1)\mathbf{J}_{\mathcal{N}}|u_1,v_1+\mathcal{N}(u_1,v_1,t_1)\mathbf{b}^{\mathbf{T}}\nabla\Phi|_{u_1,v_1,t_1}. \quad \text{(C.5)}\end{aligned}$$

Solving for k, we have

$$k=\frac{\mathbf{b}^{\mathbf{T}}\Phi+\mathbf{b}^{\mathbf{T}}\Phi\mathcal{N}^{\mathbf{T}}\mathbf{J}_{\mathcal{N}}\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}+\mathbf{b}^{\mathbf{T}}\nabla\Phi\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}}{\mathcal{N}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{r}}, \quad \text{(C.6)}$$

where all the dependent variables of $\mathcal{N},\mathbf{J}_{\mathcal{N}},\Phi,\nabla\Phi$ are discarded as they are now all at (u_1,v_1,t_1). Thus, substituting (C.6) back into (C.5), and after some algebraic manipulations, we have (2.14).

Appendix D

Derivation of (2.16)

As

$$\mathcal{N} = \frac{\frac{\partial \mathcal{C}}{\partial u} \times \frac{\partial \mathcal{C}}{\partial v}}{\|\frac{\partial \mathcal{C}}{\partial u} \times \frac{\partial \mathcal{C}}{\partial v}\|} = \frac{\mathcal{C}_u \times \mathcal{C}_v}{\|\mathcal{C}_u \times \mathcal{C}_v\|} = \frac{\hat{\mathcal{C}}_u \mathcal{C}_v}{\sqrt{\mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v}}, \tag{D.1}$$

where $\hat{\mathcal{C}}$ denote the skew symmetric matrix with entries $\begin{pmatrix} 0 & -\mathcal{C}^{(3)} & \mathcal{C}^{(2)} \\ \mathcal{C}^{(3)} & 0 & -\mathcal{C}^{(1)} \\ -\mathcal{C}^{(2)} & \mathcal{C}^{(1)} & 0 \end{pmatrix}$, and the superscript $\mathcal{C}^{(1)}$ indicates the first dimension of the vector. Taking the partial derivative of $\mathcal{N}$ with respect to t, we have

$$\frac{\partial \mathcal{N}}{\partial t} = \frac{\frac{\partial \hat{\mathcal{C}}_u}{\partial t} \mathcal{C}_v + \hat{\mathcal{C}}_u \frac{\partial \mathcal{C}_v}{\partial t}}{\sqrt{\mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v}} - \frac{\hat{\mathcal{C}}_u \mathcal{C}_v \frac{\partial (\mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v)}{\partial t}}{2(\mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v)^{\frac{3}{2}}}. \tag{D.2}$$

Taking the partial derivative of (2.4) with respect to u and v, and assuming $\frac{\partial^2 \mathcal{C}}{\partial u \partial t}$ and $\frac{\partial^2 \mathcal{C}}{\partial t \partial u}$ exist and are smooth (which is assumption (A3)), we have

$$\begin{aligned} \frac{\partial^2 \mathcal{C}}{\partial u \partial t} &= \frac{\partial \beta}{\partial u} \mathcal{N} + \beta \frac{\partial \mathcal{N}}{\partial u} = \beta_u \mathcal{N} + \beta \mathcal{N}_u = \frac{\partial \mathcal{C}_u}{\partial t}, \\ \frac{\partial^2 \mathcal{C}}{\partial v \partial t} &= \frac{\partial \beta}{\partial v} \mathcal{N} + \beta \frac{\partial \mathcal{N}}{\partial v} = \beta_v \mathcal{N} + \beta \mathcal{N}_v = \frac{\partial \mathcal{C}_v}{\partial t}. \end{aligned} \tag{D.3}$$

As the skew symmetric matrix $\hat{\mathcal{C}}$ is linear with respect to the original vector $\mathcal{C}$, we have

$$\begin{aligned}\frac{\partial \hat{\mathcal{C}}_u}{\partial t} &= \beta_u \hat{\mathcal{N}} + \beta \hat{\mathcal{N}}_u, \\ \frac{\partial \hat{\mathcal{C}}_v}{\partial t} &= \beta_v \hat{\mathcal{N}} + \beta \hat{\mathcal{N}}_v. \end{aligned} \tag{D.4}$$

Substitute (D.3) and (D.4) back into the numerator of the first term in the right hand side of (D.2), we have

$$\begin{aligned}\frac{\partial \hat{\mathcal{C}}_u}{\partial t}\mathcal{C}_v + \hat{\mathcal{C}}_u \frac{\partial \mathcal{C}_v}{\partial t} &= (\beta_u \hat{\mathcal{N}} + \beta \hat{\mathcal{N}}_u)\mathcal{C}_v + \hat{\mathcal{C}}_u(\beta_v \mathcal{N} + \beta \mathcal{N}_v) \\ &= \beta_u \hat{\mathcal{N}}\mathcal{C}_v + \beta \hat{\mathcal{N}}_u \mathcal{C}_v + \hat{\mathcal{C}}_u \beta_v \mathcal{N} + \hat{\mathcal{C}}_u \beta \mathcal{N}_v \\ &= \beta_u \mathcal{N} \times \mathcal{C}_v + \beta \mathcal{N}_u \times \mathcal{C}_v + \beta_v \mathcal{C}_u \times \mathcal{N} + \beta \mathcal{C}_u \times \mathcal{N}_v. \end{aligned} \tag{D.5}$$

Similarly, the numerator of the second term in the right hand side of (D.2) can be simplified as

$$\begin{aligned}\frac{\partial(\mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v)}{\partial t} &= (\beta_v \mathcal{N}^{\mathbf{T}} + \beta \mathcal{N}_v^{\mathbf{T}})\hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v + (\beta_u \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{N}}^{\mathbf{T}} + \beta \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{N}}_u^{\mathbf{T}})\hat{\mathcal{C}}_u \mathcal{C}_v \\ &\quad + (\beta_u \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{N}} + \beta \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{N}}_u)\mathcal{C}_v \\ &\quad + (\beta_v \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{N} + \beta \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{N}_v). \end{aligned} \tag{D.6}$$

Note that

$$\mathcal{N}^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v = (\mathcal{C}_u \times \mathcal{N})^{\mathbf{T}} (\mathcal{C}_u \times \mathcal{C}_v). \tag{D.7}$$

Because $\mathcal{C}_u \parallel \mathcal{T}_u$ and $\mathcal{C}_v \parallel \mathcal{T}_v$, thus $(\mathcal{C}_u \times \mathcal{N}) \perp \mathcal{N}$ while $(\mathcal{C}_u \times \mathcal{C}_v) \parallel \mathcal{N}$. Consequently, the inner product between the two terms in (D.7) is zero. Similarly, we have

$$\mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{N}}^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{C}_v = \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{N}} \mathcal{C}_v = \mathcal{C}_v^{\mathbf{T}} \hat{\mathcal{C}}_u^{\mathbf{T}} \hat{\mathcal{C}}_u \mathcal{N} = 0. \tag{D.8}$$

Thus, (D.6) can be simplified as

$$\begin{aligned} & \beta\mathcal{N}_v^{\mathbf{T}}\hat{\mathcal{C}}_u^{\mathbf{T}}\hat{\mathcal{C}}_u\mathcal{C}_v + \beta\mathcal{C}_v^{\mathbf{T}}\hat{\mathcal{N}}_u^{\mathbf{T}}\hat{\mathcal{C}}_u\mathcal{C}_v + \beta\mathcal{C}_v^{\mathbf{T}}\hat{\mathcal{C}}_u^{\mathbf{T}}\hat{\mathcal{N}}_u\mathcal{C}_v + \beta\mathcal{C}_v^{\mathbf{T}}\hat{\mathcal{C}}_u^{\mathbf{T}}\hat{\mathcal{C}}_u\mathcal{N}_v \\ = \; & \beta(\mathcal{C}_u \times \mathcal{N}_v)^{\mathbf{T}}(\mathcal{C}_u \times \mathcal{C}_v) + \beta(\mathcal{N}_u \times \mathcal{C}_v)^{\mathbf{T}}(\mathcal{C}_u \times \mathcal{C}_v) \\ & +\beta(\mathcal{C}_u \times \mathcal{C}_v)^{\mathbf{T}}(\mathcal{N}_u \times \mathcal{C}_v) + \beta(\mathcal{C}_u \times \mathcal{C}_v)^{\mathbf{T}}(\mathcal{C}_u \times \mathcal{N}_v)) \\ = \; & 2\beta(\mathcal{C}_u \times \mathcal{C}_v)^{\mathbf{T}}(\mathcal{C}_u \times \mathcal{N}_v + \mathcal{N}_u \times \mathcal{C}_v). \end{aligned} \tag{D.9}$$

Thus, substituting (D.5) and (D.9) back into (D.2), we have

$$\begin{aligned} \frac{\partial \mathcal{N}}{\partial t} = \; & \frac{\beta_u \mathcal{N} \times \mathcal{C}_v + \beta_v \mathcal{C}_u \times \mathcal{N}}{\|\mathcal{C}_u \times \mathcal{C}_v\|} \\ & +\beta \frac{\|\mathcal{C}_u \times \mathcal{C}_v\|^2 \mathbf{I} - (\mathcal{C}_u \times \mathcal{C}_v)(\mathcal{C}_u \times \mathcal{C}_v)^{\mathbf{T}}}{\|\mathcal{C}_u \times \mathcal{C}_v\|^3} (\mathcal{C}_u \times \mathcal{N}_v + \mathcal{N}_u \times \mathcal{C}_v). \end{aligned} \tag{D.10}$$

Because $\mathcal{N}_u \parallel \mathcal{C}_u$ and $\mathcal{N}_v \parallel \mathcal{C}_v$, thus $(\mathcal{C}_u \times \mathcal{N}_v) \parallel (\mathcal{N}_u \times \mathcal{C}_v) \parallel \mathcal{N}$. Let $\mathcal{C}_u \times \mathcal{N}_v + \mathcal{N}_u \times \mathcal{C}_v = p\mathcal{N}$ and $\mathcal{C}_u \times \mathcal{C}_v = q\mathcal{N}$, where p and q are scalars. Thus the second term in the right hand side of (D.10) becomes

$$\beta \frac{q^2 p\mathcal{N} - q^2 \mathcal{N}\mathcal{N}^{\mathbf{T}} p\mathcal{N}}{q^3} = \beta \frac{q^2 p\mathcal{N} - q^2 p\mathcal{N}}{q^3} = 0. \tag{D.11}$$

Thus, (D.10) can be simplified as

$$\frac{\partial \mathcal{N}}{\partial t} = \frac{\beta_u \mathcal{N} \times \mathcal{C}_v + \beta_v \mathcal{C}_u \times \mathcal{N}}{\|\mathcal{C}_u \times \mathcal{C}_v\|}. \tag{D.12}$$

If $\beta_u = 0$ and $\beta_v = 0$, the surface evolve isotropically, and the norm does not change over deformation. By choosing proper parameters u and v, we can let $\|\mathcal{C}_u\| = 1$, $\|\mathcal{C}_v\| = 1$, and $\mathcal{C}_u \perp \mathcal{C}_v$. Use this set of parameterization and assume the right hand coordinate system to be $(u \times v) \parallel \mathcal{N}$, (D.12) can be simplified as

$$\frac{\partial \mathcal{N}}{\partial t} = -(\beta_u \mathcal{C}_u + \beta_v \mathcal{C}_v). \tag{D.13}$$

Thus, the second term in the right hand side of (2.7), i.e., temporal change of norm due to the deformation, can be simplified as

$$\begin{aligned}
\frac{\partial \mathcal{N}}{\partial t}|_{u_2,v_2,t_1}\mathbf{\Delta}t &= -(\mathbf{b}_d^{\mathbf{T}}\Phi_u\mathcal{C}_u + \mathbf{b}_d^{\mathbf{T}}\Phi_v\mathcal{C}_v)|_{u_2,v_2,t_1} \\
&= -(\mathcal{C}_u,\mathcal{C}_v)|_{u_2,v_2,t_1}\begin{pmatrix}\Phi_u^{\mathbf{T}}\\ \\ \Phi_v^{\mathbf{T}}\end{pmatrix}|_{u_2,v_2,t_1}\mathbf{b}_d \\
&= -\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))|_{(u_2,v_2,t_1)}\mathbf{J}_{\mathcal{N}}(\Phi|(u,v))|^{\mathbf{T}}_{(u_2,v_2,t_1)}\mathbf{b}_d. \qquad \text{(D.14)}
\end{aligned}$$

Due to the fact that the change of the norm is not affected by the texture variation, for the simplicity of notation, we use Φ to denote Φ_d. Substituting

$$\begin{aligned}
\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))|_{(u_2,v_2,t_1)} &= \mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))|_{(u_1,v_1,t_1)} + \frac{\partial \mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))}{\partial(u,v)}|_{(u_1,v_1,t_1)} \times_3 \begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}, \\
\mathbf{J}_{\mathcal{N}}(\Phi|(u,v))|_{(u_2,v_2,t_1)} &= \mathbf{J}_{\mathcal{N}}(\Phi|(u,v))|_{(u_1,v_1,t_1)} + \frac{\partial \mathbf{J}_{\mathcal{N}}(\Phi|(u,v))}{\partial(u,v)}|_{(u_1,v_1,t_1)} \times_3 \begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix},
\end{aligned} \qquad \text{(D.15)}$$

into (D.14), we have

$$\begin{aligned}
\frac{\partial \mathcal{N}}{\partial t}|_{u_2,v_2,t_1}\boldsymbol{\Delta} t &= -(\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v)) + \frac{\partial \mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))}{\partial(u,v)} \times_3 \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix}) \\
&\quad \left(\mathbf{J}_{\mathcal{N}}(\Phi|(u,v)) + \frac{\partial \mathbf{J}_{\mathcal{N}}(\Phi|(u,v))}{\partial(u,v)} \times_3 \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} \right)^{\mathbf{T}} \mathbf{b} \\
&= -\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))\mathbf{J}_{\mathcal{N}}(\Phi|(u,v))^{\mathbf{T}}\mathbf{b}_d \\
&\quad -\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v)) \left(\frac{\partial \mathbf{J}_{\mathcal{N}}(\Phi|(u,v))}{\partial(u,v)} \times_3 \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} \right)^{\mathbf{T}} \mathbf{b}_d \\
&\quad -\frac{\partial \mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))}{\partial(u,v)} \times_3 \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} \mathbf{J}_{\mathcal{N}}(\Phi|(u,v))^{\mathbf{T}}\mathbf{b}_d \\
&\quad -\frac{\partial \mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))}{\partial(u,v)} \times_3 \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} \left(\frac{\partial \mathbf{J}_{\mathcal{N}}(\Phi|(u,v))}{\partial(u,v)} \times_3 \begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} \right)^{\mathbf{T}} \mathbf{b}_d. \quad \text{(D.16)}
\end{aligned}$$

From (2.15) or (2.23), we know $\begin{pmatrix} \alpha_u \\ \\ \alpha_v \end{pmatrix} = O(\Delta t)$. In addition, as $\mathbf{b}_d = O(\Delta t)$, the first term in the right hand side of (D.16) is $O(\Delta t)$ while the other terms are $O(\Delta t^2)$. Using Assumption (A2), we can neglect $O(\Delta t^2)$ with respect to $O(\Delta t)$, and (D.16) becomes,

$$\frac{\partial \mathcal{N}}{\partial t}|_{u_2,v_2,t_1}\boldsymbol{\Delta} t \approx -(\mathbf{J}_{\mathcal{N}}(\mathcal{C}|(u,v))\mathbf{J}_{\mathcal{N}}(\Phi|(u,v))^{\mathbf{T}}\mathbf{b}_d. \quad \text{(D.17)}$$

Appendix E

Derivation of (2.22)

Substituting (2.20) and (2.21) into (2.19), we have

$$\mathbf{\Delta R}(\mathcal{C}(u_1,v_1,t_1)+\mathbf{b_d^T}\Phi_d(u_2,v_2)\mathcal{N}(u_2,v_2,t_1)\Delta t+\alpha_u\mathcal{T}_u+\alpha_v\mathcal{T}_v)-\mathcal{C}(u_1,v_1,t_1)$$
$$=k\mathbf{R}^{-1}\mathbf{r}-\mathbf{R^{-1}\Delta T}. \quad \text{(E.1)}$$

Using (C.4) to approximate $\mathbf{b_d^T}\Phi_d(u_2,v_2)\mathcal{N}(u_2,v_2,t_1)$, we have

$$\left(\Delta\mathbf{R}\left(\mathcal{T}_u,\mathcal{T}_v\right)+\mathbf{b_d^T}\Phi_d\mathbf{\Delta R J}\Delta t+\mathbf{\Delta R}\mathcal{N}\mathbf{b_d^T}\nabla\Phi_d\Delta t\right)\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}$$
$$=(\mathbf{I}-\mathbf{\Delta R})\mathcal{C}(u_1,v_1,t_1)-\mathbf{b_d^T}\Phi_d\mathbf{\Delta R}\mathcal{N}\Delta t-\mathbf{R^{-1}\Delta T}+k\mathbf{R}^{-1}\mathbf{r}, \quad \text{(E.2)}$$

where all the $\mathcal{N},\mathbf{J}_\mathcal{N},\Phi\text{and}\nabla\Phi$ are at (u_1,v_1,t_1) and subscripts are discarded. Solving for k, we have

$$k\approx\frac{\mathcal{N}^\mathbf{T}\mathbf{R}^{-1}\mathbf{\Delta T}+\mathcal{N}^\mathbf{T}(\mathbf{I}-\Delta\mathbf{R}^{-1})\mathcal{C}(u_1,v_1)+\mathbf{b^T}\Phi+(\mathbf{b^T}\Phi\mathcal{N}^\mathbf{T}\mathbf{J}_\mathcal{N}+\mathbf{b^T}\nabla\Phi)\begin{pmatrix}\alpha_u\\ \\ \alpha_v\end{pmatrix}}{\mathcal{N}^\mathbf{T}\mathbf{R}^{-1}\mathbf{r}} \quad \text{(E.3)}$$

Substituting back into (E.2), we have (2.22).

Appendix F

Piecewise Multi-linear Manifold Embedding

A piecewise multi-linear manifold can be embedded into a higher dimensional globally multi-linear subspace.

Outline of the Proof: Without loss of generality, we prove the case of piecewise bilinear manifold. Assuming we have a collection of locally bilinear manifold in the form of $\mathcal{B}_j \times_1 \mathbf{a} \times_2 \mathbf{b}$, where j is the indicator of the local manifold, and $j = 1 \dots J$. This piecewise manifold can be embeded into

$$\begin{pmatrix} \mathcal{B}_1 & 0 & \cdots & 0 \\ 0 & \mathcal{B}_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{B}_J \end{pmatrix} \times_1 \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_J \end{pmatrix} \times_2 \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_J \end{pmatrix}, \tag{F.1}$$

where $\mathbf{a}_1$ to $\mathbf{a}_J$ and $\mathbf{b}_1$ to $\mathbf{b}_J$ are the same size of $\mathbf{a}$ and $\mathbf{b}$. The jth piece of manifold can be obtained by setting all the $\mathbf{a}$s and $\mathbf{b}$s except $\mathbf{a}_j$ and $\mathbf{b}_j$ to be zero, while (F.1) forms a globally bilinear subspace.

Bibliography

[1] O. Arandjelovic and R. Cipolla. An illumination invariant face recognition system for access control using video. *British Machine Vision Conference*, 2004.

[2] O. Arandjelovic, J. Fisher G. Shakhnarovich, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. *IEEE Computer Vision and Pattern Recognition*, 2005.

[3] O. Arandjelovic, G. Shakhnarovich, J.W. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. *IEEE Computer Vision and Pattern Recognition*, 2005.

[4] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.

[5] A.J. Smola B. Scholkopf and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10:1299-1319, 1998.*, 1998.

[6] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255, Mar. 2004.

[7] A. Bartoli. Groupwise geometric and photometric direct image registration. In *7th British Machine Vision Conference, Edinburgh, UK*, Sep 2006.

[8] R. Basri and D.W. Jacobs. Lambertian Reflectance and Linear Subspaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(2):218–233, February 2003.

[9] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions? In *IEEE Conf. Computer Vision and Pattern Recognition*, 1996.

[10] V. Blanz, P. Grother, P. Phillips, and T. Vetter. Face Recognition Based on Frontal Views Generated From Non-Frontal Images. In *Computer Vision and Pattern Recognition*, 2005.

[11] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, September 2003.

[12] K.W. Bowyer and Chang. A survey of 3D and Multimodal 3D+2D Face Recognition. In *Face Processing: Advanced Modeling and Methods*. Academic Press, 2005.

[13] T.J. Broida and R. Chellappa. Estimating the Kinematics and Structure of a Rigid Object from a Sequence of Monocular Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:497–513, 1991.

[14] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active Appearance Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.

[15] K. Daniilidis and H. Nagel. Analytic results on error sensitivity of motion estimation from two views. *Image and Vision Computing*, 8(4):297–303, 1990.

[16] P. Eisert and B. Girod. Illumination compensated motion estimation for analysis sythesis coding. *3D Image Analysis and Synthesis, pages 61-66*, 1996.

[17] A.M. Elgammal and C.S. Lee. Separating style and content on a nonlinear manifold. In *Computer Vision and Pattern Recognition*, pages I: 478–485, 2004.

[18] M. Everingham and A. Zisserman. Identifying individuals in video by combinig 'generative' and discriminative head models. *IEEE International Conference on Computer Vision*, 2005.

[19] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 2002.

[20] C. Fermuller and Y. Aloimonos. *Foundations of Image Understanding*. Kluwer, 2001.

[21] G. Finlayson, S. Hordley, and M. Drew. Removing shadows from images. In *European Conference on Computer Vision*, 2002.

[22] R.T. Frankot and R. Challappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988.

[23] D. Freedman and M. Turek. Illumination-Invariant Tracking via Graph Cuts. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[24] M. Gouiffes, C. Collewet, C. Fernandez-Maloigne, and A. Trémeau. Feature points tracking : robustness to specular highlights and lighting changes. In *9th European Conference on Computer Vision, Graz, Austria*, May 2006.

[25] 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*, 20(10):1025–1039, 1998.

[26] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[27] J. Ho and D Kriegman. On the Effect of Illumination and Face Recognition. In *Face Processing: Advanced Modeling and Methods*. Academic Press, 2005.

[28] B.K.P. Horn and M.J. Brooks. The Variational Approach to Shape from Shading. *Computer Vision, Graphics and Image Processing*, 33(2):174–208, 1986.

[29] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[30] J. D. Jackson, A. J. Yezzi, and S. Soatto. Dynamic shape and appearance modeling via moving and deforming layers. *International Journal of Computer Vision*, 2008.

[31] D. Jacobs and S. Shirdhonkar. Non-negative lighting and specular object recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2005.

[32] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *IEEE Intl. Conf. on Computer Vision*, 2001.

[33] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63(3):175–189, 2005.

[34] A. Kale and C. Jaynes. A joint illumination and shape model for visual tracking. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 602–609, 2006.

[35] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Intl. Journal of Comp. Vision*, pages 321–331, 1988.

[36] I. Kemelmacher-Shlizerman, R. Basri, and B. Nadler. 3d shape reconstruction of mooney faces. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[37] Y.H. Kim, A.M. Martinez, and A.C. Kak. Robust Motion Estimation under Varying Illumination. *Image and Vision Computing*, 23, 2005.

[38] S. Koterba, S. Baker, I. Matthews, C. Hu, H. Xiao, J. Cohn, and T. Kanade. Multi-view aam fitting and camera calibration. In *IEEE Intl. Conf. on Computer Vision*, 2005.

[39] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A Multillinear Singular Value Decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[40] C. Lee and A. Elgammal. Nonlinear shape and appearance models for facial expression analysis and synthesis. *IEEE Conference on Computer Vision and Pattern Recognition*, I:313–320, 2003.

[41] K.C. Lee, J. Ho, M.H. Yang, and D.J. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *Computer Vision and Pattern Recognition*, pages I: 313–320, 2003.

[42] X. Liu and T. Chen. Video-based face recognition using adaptive hidden markov models. *IEEE Computer Vision and Pattern Recognition*, 2003.

[43] J. Lou, T. Tan, W. Hu, H. Yang, and S. Maybank. 3d model-based vehicle tracking. *IEEE Trans. on Image Proc.*, 14(10):1561–1569, 2005.

[44] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, April 1981.

[45] S. Lucey and T. Chen. Learning patch dependencies for improved pose mismatched face verification. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2006.

[46] I. Matthews and S. Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164, Nov. 2004.

[47] Y. Moses. *Face Recognition: Generalization to Novel Images*. PhD thesis, Weizmann Inst. of Sciences, 1993.

[48] R. Ng, R. Ramamoorthi, and P. Hanrahan. Wavelet triple product integrals for all-frequency relighting. In *SIGGRAPH*, pages 475–485, 2004.

[49] J. Oliensis. A critique of structure from motion algorithms. *Computer Vision and Image Understanding*, 80(2):172–214, 2000.

[50] J. Oliensis and P. Dupuis. Direct method for reconstructing shape from shading. In *Proc. SPIE Conf. 1570 on Geometric Methods in Computer Vision*, 1991.

[51] A. O'Toole et al. A video database of moving faces and people. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 812–816, May 2005.

[52] P.J. Phillips, P.J. Grother, R.J. Micheals, D.M. Blackburn, E. Tabassi, and J.M. Bone. Face recognition vendor test 2002: Evaluation report. Technical Report NISTIR 6965, http://www.frvt.org, 2003.

[53] P. J. Phillips et al. Overview of the face recognition grand challenge. In *Computer Vision and Pattern Recognition*, 2005.

[54] D. Pizarro and A. Bartoli. Shadow resistant direct image registration. In *SCIA'07 - Proceedings of the Fifteenth Scandinavian Conference on Image Analysis, Aalborg, Denmark*, Jun 2007.

[55] S.J.D. Prince and J.H. Elder. Probabilistic linear discriminant analysis for inferences about identity. *In Proc. IEEE International Conference on Computer Vision*, 2007.

[56] G. Qian, R. Chellappa, and Q. Zheng. Robust structure from motion estimation using inertial data. *The Journal of the Optical Society of America A*, 18(12):2982–2997, 2001.

[57] I. Matthews R. Gross and S. Baker. Appearance-based face recognition and light-fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[58] R. Ramamoorthi. Analytic PCA Construction for Theoretical Analysis of Lighting Variability in Images of a Lambertian Object. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002.

[59] R. Ramamoorthi. Modeling Illumination Variation With Spherical Harmonics. In *Face Processing: Advanced Modeling and Methods*. Academic Press, 2005.

[60] R. Ramamoorthi 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 A*, 18(10), Oct 2001.

[61] R. Ramamoorthi and P. Hanrahan. A signal processing framework for reflection. *ACM Trans. on Graphics*, pages 1004–1042, October 2004.

[62] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *IEEE International conference on Computer Vision 2003*, 2003.

[63] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec 2000.

[64] A. Roy-Chowdhury and R. Chellappa. Face Reconstruction From Monocular Video Using Uncertainty Analysis and a Generic Model. *Computer Vision and Image Understanding*, 91(1-2):188–213, July-August 2003.

[65] A. K. Roy-Chowdhury and R. Chellappa. Stochastic approximation and rate distortion analysis for robust structure and motion estimation. *International Journal on Computer Vision*, 55(1):27–53, 2003.

[66] A. K. Roy-Chowdhury and R. Chellappa. An information theoretic criterion for evaluating the quality of 3d reconstructions from video. *IEEE Trans. on Image Processing*, pages 960–973, Jul 2004.

[67] A. K. Roy-Chowdhury and R. Chellappa. Statistical bias in 3d reconstruction from a monocular video. *IEEE Trans. on Image Processing*, pages 1057–1062, Aug 2005.

[68] G. Sapiro. *Geometric Partial Differential Equations and Image Processing*. Cambridge University Press, January 2001.

[69] M. Savvides, B.V.K. Vijaya Kumar, and P.K. Khosla. Corefaces - robust shift invariant pca based correlation filter for illumination tolerant face recognition. In *Computer Vision and Pattern Recognition*, 2004.

[70] G. Shakhnarovich, J.W. Fisher, and T. Darrell. Face recognition from long-term observations. *European Conference on Computer Vision*, 2002.

[71] A. Shashua. On Photometric Issues in 3D Visual Recognition from a Single 2D Image. *International Journal of Computer Vision*, 21(1-2):99–122, 1997.

[72] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[73] S. Shirdhonkar and D. Jacobs. Non-negtive lighting and specular object recognition. *IEEE International Conference on Computer Vision*, I:1323–1330, Oct 2005.

[74] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 16(1):63–84, 2000.

[75] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination and expression database. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:1615–1618, December 2003.

[76] D. Simakov, D. Frolova, and R. Basri. Dense shape reconstruction of a moving object under arbitrary, unknown lighting. In *IEEE Intl. Conf. on Computer Vision*, 2003.

[77] S. Soatto, G. Doretto, and Y.N. Wu. Dynamic Textures. *International Conf. on Computer Vision*, 2:439–446, 2001.

[78] R. Szeliski and S. Kang. Recovering 3d shape and motion from image streams using non-linear least squares. *Journal of Visual Computation and Image Representation*, 5:10–28, 1994.

[79] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994.

[80] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Dec 2000.

[81] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[82] K. Thornber and D. Jacobs. Broadened, Specular reflection and linear subspaces. Technical Report 2001-033, NEC, 2001.

[83] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IEEE International Journal of Computer Vision*, 9(2):137–154, 1992.

[84] L. Torresani and C. Bregler. Space-time tracking. In *IEEE European Conference on Computer Vision*, 2002.

[85] M.A.O. Vasilescu and D. Terzopoulos. Multilinear Independent Components Analysis. In *Computer Vision and Pattern Recognition*, 2005.

[86] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human motion analysis. *IEEE Trans. on Pattern Analysis and Machine Intellitence*, pages 1896–1909, Dec 2005.

[87] H. Wang and N. Ahuja. Facial expression decomposition. *IEEE International Conference on Computer Vision*, 2:958 – 965, 2003.

[88] C. Xie, B.V.K. Vijaya Kumar, S. Palanivel, and B. Yegnanarayana. A still-to-video face verification system using advanced correlation filters. In *First International Conference on Biometric Authentication*, 2004.

[89] Y. Xu and A. Roy-Chowdhury. Pose and illumination invariant registration and tracking for video-based face recognition. *IEEE Computer Society Workshop on Biometrics (in association with CVPR)*, 2006.

[90] Y. Xu and A. Roy-Chowdhury. Integrating Motion, Illumination and Structure in Video Sequences, With Applications in Illumination-Invariant Tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, May 2007.

[91] Y. Xu and A. Roy-Chowdhury. A Theoretical Analysis of Linear and Multi-linear Moels of Image Appearance. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[92] Y. Xu and A. Roy-Chowdhury. Inverse compositional estimation of 3d motion and lighting in dynamic scenes. *IEEE Trans. on Pattern Analysis and Machine Intellitence*, pages 1300–1307, July 2008.

[93] Y. Xu and A. Roy-Chowdhury. Learning A Geometry Integrated Image Appearance Manifold From A Small Training Set. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[94] Y. Xu, A. Roy-Chowdhury, and K. Patel. Integrating illumination, motion and shape models for robust face recognition in video. *EURASIP Journal on Advances in Signal Processing: Advanced Signal Processing and Pattern Recognition Methods for Biometrics*, 2008.

[95] Y. Xu and A. K. Roy-Chowdhury. Integrating the effects of motion, illumination and structure in video sequences. In *International Conference on Computer Vision*, Oct. 2005.

[96] H. Yang, M. Pollefeys, G. Welch, J.-M. Frahm, and A. Ilie. Differential Camera Tracking through Linearizing the Local Appearance Manifold. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.

[97] G. Young and R. Chellappa. Statistical analysis of inherent ambiguities in recovering 3-d motion from a noisy flow field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:995–1013, 1992.

[98] L. Zhang, B. Curless, A. Hertzmann, and S.M. Seitz. Shape and Motion under Varying Illumination: Unifying Structure from Motion, Photometric Stereo, and Multi-view Stereo. In *Proc. of IEEE International Conference on Computer Vision*, 2003.

[99] L. Zhang and D. Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 351–363, March 2006.

[100] Z. Zhang. Determining the epipolar geometry and its unvertainty: A review. *International Journal of Computer Vision*, 27:161–195, 1998.

[101] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Transactions*, 2003.