Abstract

Title of dissertation STATISTICAL/GEOMETRIC TECHNIQUES FOR OBJECT REPRESENTATION AND RECOGNITION

Soma Biswas, Doctor of Philosophy, 2009

Directed by Professor Rama Chellappa
Department of Electrical and Computer Engineering

Object modeling and recognition are key areas of research in computer vision and graphics with wide range of applications. Though research in these areas is not new, traditionally most of it has focused on analyzing problems under controlled environments. The challenges posed by real life applications demand for more general and robust solutions. The wide variety of objects with large intra-class variability makes the task very challenging. The difficulty in modeling and matching objects also vary depending on the input modality. In addition, the easy availability of sensors and storage have resulted in tremendous increase in the amount of data that needs to be processed which requires efficient algorithms suitable for large-size databases. In this dissertation, we address some of the challenges involved in modeling and matching of objects in realistic scenarios.

Object matching in images require accounting for large variability in the appearance due to changes in illumination and view point. Any real world object is characterized by its underlying shape and albedo, which unlike the image intensity are insensitive to changes in illumination conditions. We propose a stochastic filtering framework for estimating object albedo from a single intensity image by formulating the albedo estimation as an image estimation problem. We also show how this albedo estimate can be used for illumination insensitive object matching and for more accurate shape recovery from a single image using standard shape from shading formulation. We start with the simpler problem where the pose of the object is known and only the illumination varies. We then extend the proposed approach to handle unknown pose in addition to illumination variations. We also use the estimated albedo maps for another important application, which is recognizing faces across age progression.

Many approaches which address the problem of modeling and recognizing objects from images assume that the underlying objects are of diffused texture. But most real world objects exhibit a combination of diffused and specular properties. We propose an approach for separating the diffused and specular reflectance from a given color image so that the algorithms proposed for objects of diffused texture become applicable to a much wider range of real world objects.

Representing and matching the 2D and 3D geometry of objects is also an integral part of object matching with applications in gesture recognition, activity classification, trademark and logo recognition, etc. The challenge in matching 2D/3D shapes lies in accounting for the different rigid and non-rigid deformations, large intra-class variability, noise and outliers. In addition, since shapes are usually represented as a collection of landmark points, the shape matching algorithm also has to deal with the challenges of missing or unknown correspondence across these data points. We propose an efficient shape indexing approach where the different feature vectors representing the shape are mapped to a hash table. For a query shape, we show how the similar shapes in the database can be efficiently retrieved without the need for establishing correspondence making the algorithm extremely fast and scalable. We also propose an approach for matching and registration of 3D point cloud data across unknown or missing correspondence using an implicit surface representation. Finally, we discuss possible future directions of this research.

Statistical/Geometric Techniques for Object Representation and Recognition

by

Soma Biswas

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:

Professor Rama Chellappa, Chair
Professor Larry S. Davis
Professor David W. Jacobs
Professor Ankur Srivastava
Professor Shuvra S. Bhattacharyya

This thesis is dedicated to
My parents

Acknowledgment

I would like to take this opportunity to thank everyone who made this dissertation possible. Firstly, I thank my advisor Prof. Rama Chellappa who was a constant source of inspiration for me. He always gave me full freedom to work on topics of my interest. Throughout this period, he helped and encouraged me think independently and come up with new ideas. He made sure that I never have to worry about financial support so that I could concentrate solely on my work. Also he supported me for all the travels to different workshops and conferences that considerably helped in my research. I would also like to thank my mentors Dr. Nalini Ratha (from IBM T.J. Watson Research Center, NY) and Dr. Niels Haering (from ObjectVideo, VA) who guided me during the internships and also helped shape this dissertation. I would also like to thank my committee members Prof. Larry Davis, Prof. David Jacobs, Prof. Shuvra Bhattacharyya and Prof. Ankur Srivastava for agreeing to be in my committee.

I would like to thank all my friends Kaushik, Sameer, Pavan, Kaustav, Dikpal, Mahesh, Aswin and all my roommates who made my stay here fun and memorable. Inspite of having their own work, they always had time to listen and give valuable suggestion whenever I needed them

I would like to thank my sister, Kaushik-da and niece Sneha for always been there for me. I thank my husband and best friend Gaurav for supporting me through all ups and downs and for always being so encouraging. Finally, this dissertation would not have been possible without the constant love, support and sacrifice of my parents over all these years and I dedicate this thesis to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Object modeling and recognition are key areas of research in computer vision and graphics with wide range of applications in many different fields (Figure 1.1). Automatic matching of different biometrics like face, iris, fingerprints is an integral part of security and surveillance for automatic border control, airport security, etc. Object matching is also useful for consumer applications like shopping using visual search engines which allows users to use input pictures to search, retrieve and compare all similar products. In the medical field, matching of different medical shapes is useful for research, diagnosis and treatment by analysis of similar cases present in the database.

Figure 1.1: A few applications of object modeling and recognition. (a) SmartGate installation at Sydney International Airport for automatic border control which uses automatic face recognition; (b) Shopping by visual search by like.com; (c) SPIRS-IRMA system from NIH for retrieving similar medical images for diagnosis and treatment.

Though research in these areas is not new, traditionally most of it has focused on analyzing problems under controlled environments. The challenges posed by the real life applications demand for more general and robust solutions. The wide variety of objects with large intra-class variability makes the task very challenging.

The difficulty in modeling and matching objects also varies depending on the input modality. In addition, the easy availability of sensors and storage have resulted in tremendous increase in the amount of data that needs to be processed which requires efficient algorithms suitable for large-size databases. In this dissertation, we address some of the challenges involved in modeling and matching of objects in realistic scenarios.

1.1 Object matching in images

Object matching in images require accounting for large variability in the appearance due to changes in illumination and view point. Figure 1.2 shows how drastically the facial appearance changes due to changes in light source direction and head pose or camera location. The sources of appearance variation also depends on the specific application. For example, for the application of face recognition, the appearance of the face also changes due to different expressions, aging, facial makeup, etc.

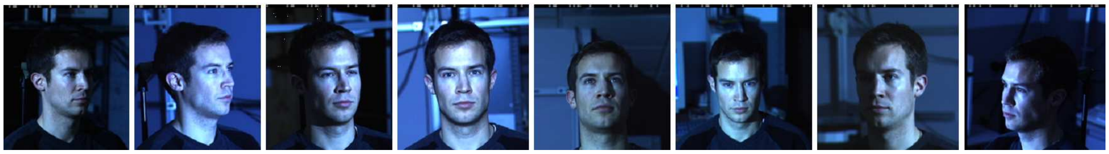


Figure 1.2: Illustration showing changes in facial appearance due to variations in illumination conditions and view-point.

Any real world object is characterized by its underlying shape and albedo. These are the intrinsic characteristics of the object and unlike the image intensity, they are insensitive to changes in illumination conditions and view-point. The first part of the dissertation deals with robustly recovering the albedo and the shape of an object from a single input image. We propose a stochastic filtering framework for estimating object albedo from an intensity image by formulating the albedo estimation as an image estimation problem. We also show how this albedo estimate can be used for illumination insensitive object matching and for more accurate shape recovery from a single image using standard shape from shading (SFS) formulation. We start with the simpler case where the pose of the object is known and only the illumination varies. We then extend the proposed approach to handle unknown pose in addition to illumination variations. We also show how the albedo estimate can be used for applications other than matching across different illuminations. Specifically we show its usefulness for the problem of matching face images across different ages, which has very important applications like finding of missing persons, automatic renewal of passport, visa, etc.

Lambertian reflectance for the imaged scenes/objects has been commonly assumed in a variety of computer vision algorithms, such as shape reconstruction, image matching, motion detection, as well as photometric and multi-view stereo. However, most real world surfaces exhibit a combination of diffuse and specular components making this assumption very restrictive in practice (Figure 1.3). Automatic separation of these components would enable these algorithms to be readily applied to a much wider class of non-Lambertian objects.

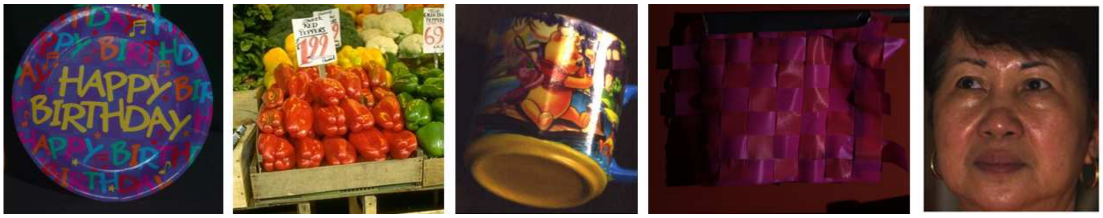


Figure 1.3: Illustration showing the importance of accounting for specularity for object matching.

The second part of the dissertation addresses the problem of specularity removal of complex textured surfaces from a single color image. First, we propose a Hough transform-based approach for automatic estimation of source color from a single color image. We further analyze the errors in source color estimation to perform robust separation of the diffused and specular reflection components. Both algorithms are completely automatic and do not need explicit color segmentation or color boundary detection as required by many existing methods.

1.2 Object matching from shapes

Representing and matching the 2D or 3D geometry of objects is also an integral part of object matching. Character recognition, trademark logo retrieval, activity recognition, object recognition and human pose estimation are a few of the challenging applications that can benefit from accurate and efficient shape matching techniques (Figure 1.4).

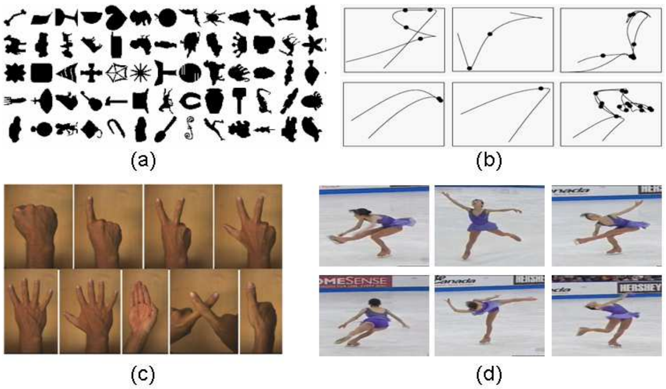


Figure 1.4: A few applications that can benefit from robust and efficient shape matching. (a) Matching and retrieval of shapes, like digit recognition, trademark retrieval, leaf recognition; (b) Activity Classification; (c) Gesture recognition; (d) Pose estimation.

Different applications require different representations and hence different matching algorithms to handle large variations in shapes. The challenge in matching 2D/3D shapes lies in accounting for the different rigid and non-rigid deformations along with large intra-class variability in addition to noise and outliers. Also since shapes are usually represented as a collection of landmark points, the shape matching algorithm also has to deal with the challenges of missing or unknown correspondence across these data points. In addition, with the recent advancement in technology and the availability of different kinds of sensors, the amount of data to be handled has increased tremendously over the last few decades. So even though research in the area of shape matching has matured, the challenges involved in achieving high performance in terms of both accuracy and computational complexity continues to interest researchers.

The third part of the dissertation deals with efficient representation and matching of 2D and 3D shapes. First, we propose an efficient shape indexing approach where the different feature vectors representing the shape are mapped to a hash table. For a query shape, we show how the similar shapes in the database can be retrieved efficiently without the the need for correspondence establishment making the algorithm extremely fast and scalable. We also propose an approach for matching and registration of 3D point cloud data across unknown or missing correspondence using an implicit surface representation.

1.3 Organization of the Thesis

The dissertation is organized as follows: Chapter 2 briefly summarizes previous work related to this dissertation. Chapter 3 describes the proposed approach for estimating the albedo from a single intensity input image and its applications in illumination invariant object matching, shape recovery, relighting, etc. Chapter 4 extends the approach developed in the previous chapter for the general case of unknown head pose. Chapter 5 provides a study of an important application, namely face recognition across aging and shows how the estimated albedo maps can be used to match face images across age progression. Chapter 6 describes our approach for estimating the illumination color from a given input image and separating the diffused and specular reflection components. Chapter 7 describes our approach for efficient representation and matching of shapes. Chapter 8 describes the future directions of this dissertation.

Chapter 2

Related Work

This chapter briefly reviews previous efforts related to this dissertation. The first part of the dissertation deals with estimating the albedo and surface normals from an image. Section 2.1 discusses the previous works on recovering the surface normals and albedo of an object from an intensity image. Related work for handling pose and misalignment issues is discussed in Section 2.2. Since our approach uses image estimation formulation, we also discuss related works in image estimation literature in Section 2.3. The second part of the dissertation deals with illumination color estimation and specularity removal from an input colored image and related work for this part is described in Section 2.4. The final part of the dissertation deals with efficient 2D/3D shape matching. Section 2.5 discusses related work on efficient representation and matching of shapes.

2.1 Related Work on Recovering Albedo and Shape

Recovery of surface normals and albedo of an object has been studied in the computer vision community for a long time. The approaches in the literature can broadly be classified into *SFS-based approaches* and *Model-based approaches.*

2.1.1 SFS-based Approach

SFS research [48] [140] aims at recovering the 3D shape of an object from a given image. Estimating the surface normals, albedo and illuminant direction given a single intensity image is inherently ill-posed. In order to make the problem more tractable, SFS approaches often make simplifying assumptions like constant or piecewise constant albedo and known illuminant direction for recovering the shape. But such assumptions often limit the applicability of the approaches for real world objects. Over the years, considerable advances have been made [143] [11] [29] [116] for dealing with objects with varying albedo. Much of the research has been directed towards the use of domain specific constraints to reduce the intractability of the problem for the analysis and estimation of general albedo maps. Often the use of such constraints bridges the gap between pure SFS approaches and the statistical model-based approaches leaving no clear demarcation between the two categories.

Zhao and Chellappa [143] propose an SFS approach to recover shape and albedo for the class of bilaterally symmetric Lambertian objects with a piecewise constant albedo field. Their approach combines the self-ratio image irradiance equation with the standard image irradiance equation to solve for the unknown derivatives of depth map. Atick *et al.* [11] reduce the SFS problem to that of parameter estimation in a low-dimensional space using Principal Component Analysis (PCA) over several hundred laser-scanned 3D heads. Dovgard *et al.* [29] combine the symmetric SFS formulation [143] with the statistical approach to facial shape reconstruction [11] to recover the 3D facial shape from a single image. Smith and Hancock [116] embed a statistical model of facial shape in an SFS formulation. Albedo estimation follows shape estimation to account for the differences between predicted and observed image brightness.

2.1.2 Model-Based Approach

Blanz and Vetter [20] propose a 3D morphable model based approach to recognize faces across pose and illumination variations. They represent each face as a linear combination of 3D basis exemplars. Recovery of shape and albedo parameters is formulated as an optimization problem that aims to minimize the difference between the input and the reconstructed image. Romdhani *et al.* [103] provide an efficient and robust algorithm for fitting a 3D morphable model using shape and texture error functions. Their algorithm uses linear equations to recover the shape and texture parameters irrespective of pose and lighting conditions of the face image. Zhang and Samaras [139] combine spherical harmonics illumination representation [17] [99] with 3D morphable models [20] to recover person-specific basis images. Feature-point based shape recovery is followed by an iterative estimation of albedo and illumination coefficients. Samaras and Metaxas [106] incorporate non-linear holonomic constraints in a deformable model to estimate shape and illuminant direction. Under the assumption of constant albedo, the light source direction and shape are estimated in an iterative manner by fixing one unknown and estimating the other until there is no more change in the illuminant estimate. Zhou *et al.* [145] impose a rank constraint on shape and albedo for the face class to separate the two from illumination using a factorization approach. Integrability and face symmetry constraints are employed to fully recover the class-specific albedos and surface normals. Lee and Moghaddam [71] propose a scheme for albedo estimation and relighting of human faces using a generic 3D face shape. First the average shape is used to determine the dominant light source direction which is then used to obtain an estimate of surface albedo for Lambertian objects. The problem of albedo estimation has also been addressed by lightness algorithms that recover an approximation to surface reflectance in independent wavelength channels [51].

2.2 Related Work on Handling Pose and Misalignment

One of the applications of albedo estimation across pose is object recognition across pose variations. Since most of the research on albedo estimation has taken place in the context of faces as objects, here we briefly describe some of the related work on face recognition across misalignment and pose. Sinilar to the work on 3D morphable model [20], Liu and Chen [76] propose a geometric approach in which they approximate a human head with a simpler 3D ellipsoid model and recognition is performed by comparing texture maps obtained by projecting the training and test images to the surface of the ellipsoid. Yue *et al.* [138] extend the spherical harmonics representation to encode pose information. Using the linear transformations that relate the 2D harmonic basis images at different poses, they project a non-frontal view test image onto the space of frontal view harmonic basis images to perform recognition across pose. For face recognition across pose, local patches are considered more robust than the whole face, and several patch-based approaches have been proposed [55] [9] [73]. Kanade and Yamada [55] proposed a face recognition system based on a probabilistic model of how the appearance of local subregions of a face changes with pose. Chen *et al.* [9] proposed an extension of this approach where in addition to modeling how a face patch varies in appearance, they also model how it deforms spatially with view-point change. Gao *et al.* [73] suggested measuring the similarity of patches between different poses by correlations in a media subspace, constructed by Canonical Correlation Analysis. Prince *et al.* [98] recently proposed a generative model for generating the observation space from the identity space using an affine mapping and pose information. Face recognition is performed with probabilistic distance modeling. Castillo and Jacobs [25] used the cost of stereo matching for 2-D face recognition across pose without performing 3-D reconstruction. Zhao and Gao [142] propose a new textural Hausdorff Distance which is a compound measurement integrating both spatial and textural features for performing pose and mis-alignment robust face recognition. In addition to these approaches, head pose estimation by itself is a separate research area [87].

2.3 Related Work on Image Estimation Methods

In our proposed approach, albedo estimation is formulated as an image estimation problem. Image estimation being a very mature area in the field of image processing [7], we provide pointers only to a few related papers. The standard

Wiener filter is known to be optimal for second order stationary processes [7]. In a stationary model, the statistical properties of the image are globally characterized which makes the stationary Wiener estimation algorithm blur the abrupt changes in the input image. Several modifications to the standard stationary image model have since been proposed. Hunt *et al.* [50] [129] proposed a non-stationary mean Gaussian image model in which an image is modeled as stationary fluctuations about a non-stationary ensemble mean. Lebedev and Mirkin [67] suggested a composite image model that assumes that an image is composed of many different stationary components, each having a distinct stationary correlation structure. Anderson and Netravali [6] used a subjective error criterion based on human visual system models. The derived non-recursive filter makes a trade-off between the loss of resolution and noise smoothing such that the same amount of subjective noise is suppressed throughout the image. Abramatic and Silverman [2] generalized this procedure and related it to the classical Wiener filter. Naderi and Sawchuk [88] derived a non-stationary discrete Wiener filter for a signal-dependent film-grain noise model which can adapt itself to the local signal statistics given the conditional noise statistics. Kuan *et al.* [58] proposed a non-stationary mean, non-stationary variance image model. A local linear minimum mean square error filter for images degraded by blur and a class of signal-dependent, uncorrelated noise is derived based on the proposed image model.

2.4 Related Work on Specularity Removal

This section describes the related work on illumination color estimation and separation of reflectance components. The dichromatic reflection model was proposed by Shafer [111] who suggested a method for separating reflection components from a single image by fitting a parallelogram to the pixel values in the RGB color space. Klinker *et al.* [57] showed that for a uniform colored surface, the combined spectral cluster of matte and highlight points form a skewed T-shape in the color space.

However, the presence of noise in real images makes the extraction of T-shape very challenging. Bajscy [12] assumed knowledge of illumination color to obtain improved results through color space transformation. Tan *et al.* [124] also used the illumination color information to reduce the problem of reflection component separation to that of identifying diffuse maximum chromaticity. All of these methods are basically designed for uniformly colored surfaces and thus require explicit color segmentation of the scene to deal with multi-colored surfaces.

Several methods have been proposed thereafter to deal with complex textured surfaces without requiring explicit color segmentation. Tan and Ikeuchi [121] proposed a specular-to-diffuse mechanism to iteratively reduce specularity of each pixel by comparing intensity logarithmic differentiation of the normalized input image to the *specular-free* image. Tan *et al.* [118] used an in-painting technique for highlight removal from a single image with user-specified highlight regions by minimizing an energy function. Malik *et al.* [78] utilized the partial separation of the reflection components provided by the SUV color space transformation [79]. The method involves solving a partial differential equation that iteratively erodes the specular component at each pixel. To avoid some of the limitations faced by local interaction methods, Tan and Ikeuchi [120] proposed a global non-iterative method by relating the specular pixels to diffuse pixels of the same color using the coefficients of reflectance linear basis functions. In another recent approach, Tan *et al.* [119] utilized additional information from outside the highlight region to determine the diffuse surface colors within the highlight. Most reflection component separation methods assume that the illumination source color is known *a priori* or can be estimated accurately.

Illumination color estimation has been extensively studied with regard to the problem of color constancy [14] [15] [69] [35] [36]. Here we provide pointers to some of the methods that deal with complex textured surfaces. Tan *et al.* [123] used highlight regions to estimate the illumination color by relating it to image chromaticity in the inverse-intensity chromaticity space. The approach relies on the presence of pixels with the same diffuse component for surface colors in the image. Toro and Funt [127] assumed a known list of candidate illumination colors and propose a multi-linear constraint to evaluate how well a candidate illumination color accounts for the observed colors in the input image.

2.5 Related Work on Shape Matching

The problem of shape matching has been around for quite sometime, probably due to its universality. Though significant advancements have been made, the requirement for computational efficiency and accuracy continues to interest researchers. In this section, we discuss some of the previous efforts that are related to representation and matching of 2D and 3D shapes proposed in the dissertation.

2.5.1 Representation and Matching of shapes

Shape context [18] based matching has been the theme of several recent works [86] [131] [125] [84] on shape matching. In the original version [18], each point is characterized by the spatial distribution of the other points relative to it. Similarity computation involves establishing correspondences using bipartite graph matching and thin plate spline (TPS) based alignment. The shape context framework has since been extended in various ways to suit different requirements of the shape matching problem. Mori and Malik [86] proposed using statistics of the tangent vectors along with the point counts to perform object recognition in clutter. A figural continuity constraint is incorporated in the feature correspondence estimation to yield reliable correspondences in cluttered scenes [125]. The constraint ensures that two points which are close on the model shape are close in the image. Tu and Yuille [131] incorporated softassign [26] in a shape context framework [131] for shape matching. One of its recent extensions by Ling and Jacobs [74] accounts for movement of part structures, by replacing the Euclidean distance in the classical version with inner distance, which is robust to articulations. In addition, the approach involves a dynamic programming (DP) based matching algorithm which helps it to outperform most previous methods. McNeill and Vijayakumar [82] proposed the hierarchical Procrustes matching algorithm which generalizes the idea of finding a point-to-point correspondence between two shapes to that of finding a segment-to-segment correspondence. In another recent work, Felzenszwalb and Schwartz [34] used a new hierarchical representation called *shape tree* for two-dimensional objects that captures shape information at multiple levels of resolution. Here a curve is described using a recursive selection of midpoints and DP is used to perform matching of two shapes.

There is another body of work for capturing part structures in which shapes are represented using shock graphs [112] [114]. The shock graph representation of a shape is based on the singularities of curve evolution process acting on the bounding contours. The shock graph grammar helps to reduce the shock graph representation to a unique rooted shock tree which is then matched using a tree matching algorithm. Sebastian *et al.* [109] proposed a shock graph-based method to handle shape deformations. They find the optimal deformation path of shock graphs that brings the two graphs (shapes) into correspondence.

2.5.2 Efficient matching and Indexing

Fast nearest neighbor searches in Euclidean space for finding closest points in metric spaces has a rich history [28]. Due to the tremendous increase in the amount of data that needs to be handled, indexing techniques are becoming increasingly popular for the development of fast retrieval algorithms for documents, images, etc. The indexing approach used in the dissertation is inspired by the work on fingerprint indexing using minutiae triangles as features [42]. Unlike the classical geometrical hashing technique [62], the triangle-based approach hashes a set of points based on local invariants (depends only on three minutiae, though need not be local spatially), which is more robust and leads to faster retrieval. For fast matching and retrieval of images, a vocabulary tree-based representation has been recently proposed by Nister and Stewenius [89]. Similar to their approach, our indexing system relies on invariant and robust shape representation, to make the retrieval process extremely fast. In [85], Mori *et al.* proposed solutions to improve the computational efficiency of shape contexts-based approaches. They show how pruning and vector quantization techniques can be utilized to make shape context useful for large databases.

Another approach for fast shape matching is to reduce the shape matching problem to the comparison of probability distributions, which does not require pose registration, feature correspondence, or model fitting. Osada *et al.* [93] use shape distributions sampled from a shape function and measured global geometric properties of an object for fast matching of 3D models. They experimented with different shape functions like distance between two random surface points, angle between three random points, etc. Ohbuchi *et al.* [91] used joint 2D histogram of distance and orientation of pairs of points for improved performance. Hamza and Krim [46] used geodesic shape distribution that measures the global geodesic distance between two arbitrary points on the surface to be able to better capture the (nonlinear) intrinsic geometric structure of the data. The idea of describing 3D models using distance between pairs of points and/or their mutual orientations has also appeared in [53] [77].

Chapter 3

Robust Estimation of Albedo for Illumination Insensitive Matching and Shape Recovery

A real world object is characterized by its underlying shape and surface properties. These characteristics define the way the object is perceived, irrespective of the view or illumination. Unlike image intensity, these characteristics of an object are invariant to changes in illumination conditions which makes them useful for illumination-invariant matching of objects. Realistic image-based rendering (IBR) is another application where accurate estimates of shape and albedo (texture) play a very important role. Thus estimating the shape and albedo of an object from an intensity image has been a very important area of research in computer vision and graphics. Though research on this topic has been underway for over two decades, the difficulty in obtaining accurate estimates and the wide range of applications continue to interest researchers.

3.1 Introduction

Albedo is the fraction of light that a surface point reflects when it is illuminated. It is an intrinsic property that depends on the material properties of the surface. In existing literature, albedo estimation has often been coupled with shape estimation. Given an input image, most methods follow the two-step approach of first recovering the shape of the object and then estimating the surface albedo [139]. A few others simultaneously estimate the shape and albedo of an object [20] [103]. There are also albedo estimation methods whose main goal is shape estimation and albedo is finally incorporated to account for the image reconstruction errors using the estimated shape [116]. Thus, in most approaches, albedo recovery depends on the accuracy of the estimated shape and illumination conditions. Errors in shape and illumination estimates lead to errors in albedo. Here we show how statistical characterization of the errors in estimates of normal and light source directions can be utilized to obtain robust albedo estimates.

The problem of albedo estimation is formulated as an image estimation problem. Given an initial albedo map (obtained using available domain-dependent average shape information), we obtain a robust albedo estimate by modeling the true unknown albedo as a non-stationary mean and non-stationary variance field. Unlike a stationary model, this model can account for the albedo variations present in most real objects. The initial albedo map can be expressed as a sum of the true unknown albedo and a signal-dependent non-stationary noise. The noise term incorporates the errors in surface normal and illumination information. Posing this as an image estimation problem, the albedo is estimated as the Linear Minimum Mean Square Error (LMMSE) estimate of the true albedo.

The theoretical formulation is extended to deal with images illuminated by multiple unknown light sources. We propose an algorithm for estimating the illumination conditions in such scenarios. The algorithm is based on an approximation to the linear subspace property of Lambertian surfaces [17] [99]. The estimated illumination information is used along with a domain-dependent average shape to obtain an initial albedo map. Similar to the single source framework, a robust albedo estimate is obtained by formulating it as an image estimation problem.

Like albedo, shape is another intrinsic property of an object which is invariant to changes in pose and illumination conditions. The importance of estimating the shape of an object has probably been the guiding force behind the vast amount of work that has been done to recover shape (shape-from-X) from images or videos. An example in this category is the work done on the problem of shape from shading [48] [140]. The goal of SFS research is to recover the 3D shape of an object from a single image. Dealing with an inherently ill-posed problem, SFS approaches typically make simplifying assumptions like constant or piece-wise constant albedo and/or known single distant light source. Such assumptions though useful to make the problem more tractable, often limit the applicability of the approaches for real world objects.

We focus on the general SFS problem of estimating the shape and albedo of an object with varying albedo map and unknown illuminant direction from a single image. To this end, we propose an algorithm that transforms the general SFS problem to one of estimating the shape of an object with unit albedo and known illuminant direction that can be addressed using a standard SFS approach.

3.1.1 Organization of the Chapter

The chapter is organized as follows. The proposed albedo estimation framework is detailed in Section 3.2. This section describes in detail the image estimation framework and the derivation of the required variances. Section 3.3 details the steps involved in shape recovery using the estimated albedo. Experimental results are presented in Section 3.4. Section 3.5 concludes the chapter with a brief summary.

3.2 Albedo Estimation

For Lambertian objects, the diffused component of the surface reflection is modeled using the *Lambert's Cosine Law* given by

$$I = \rho \max(\boldsymbol{n}^T \boldsymbol{s}, 0) \tag{3.1}$$

where I is the pixel intensity, $\boldsymbol{s}$ is the light source direction, ρ is the surface albedo and $\boldsymbol{n}$ is the surface normal of the corresponding surface point. The expression implicitly assumes a single dominant light source placed at infinity. It is worthwhile to note that the Lambert's law in its pure form is non-linear due to the max function which accounts for the formation of attached shadows. Shadow points do not reveal any information about their reflectivity and thus their albedo cannot be estimated from the image.

Let $\boldsymbol{n}_{i,j}^{(0)}$ and $\boldsymbol{s}^{(0)}$ be the initial values of the surface normal and illuminant direction. The initial surface normal can be the domain dependent average value or any estimate available or obtained from any other method. The Lambertian assumption imposes the following constraint on the initial albedo $\boldsymbol{\rho}^{(0)}$ obtained

$$\boldsymbol{\rho}_{i,j}^{(0)} = \frac{\boldsymbol{I}_{i,j}}{\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}} \tag{3.2}$$

where $\cdot$ is the standard dot product operator. We suppress the explicit max operator by considering only the pixels with positive intensity values. Clearly, more accurate the denominator $(\boldsymbol{n}^{(0)} \cdot \boldsymbol{s}^{(0)})$ is, the closer is the obtained initial albedo to its true value $\boldsymbol{\rho}$. For most applications, accurate initial estimates of normals and light source direction are not available leading to erroneous $\boldsymbol{\rho}^{(0)}$.

Figure 3.1 illustrates the nature of errors in the obtained albedo $\boldsymbol{\rho}^{(0)}$ for a synthetically generated face image using a frontal light source i.e. $\boldsymbol{s} = [0, 0, -1]$. True surface normal information is used for estimating the albedo in this example. One may expect the errors to be larger if inaccurate estimates or average value of surface normals are used. The light source direction is estimated using the method in [22] and the resulting $\boldsymbol{s}^{(0)}$ is $[0.1499, -0.0577, -0.9870]$. Interestingly, not only is $\boldsymbol{\rho}^{(0)}$ quite far from the true value for quite a few points, but even the error varies appreciably across pixels. The proposed estimation framework duly accounts for these variations to obtain a robust albedo estimate.

3.2.1 Image Estimation Framework

Here we present the image estimation framework to obtain a robust albedo estimate using the initial albedo map which is erroneous due to inaccuracies in surface

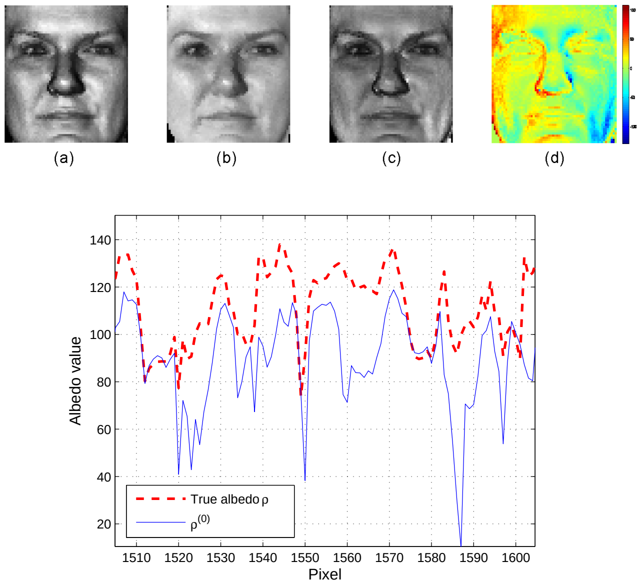


Figure 3.1: Illustration of the nature of errors in the initial albedo $\boldsymbol{\rho}^{(0)}$. (a) Input Image, (b) True Albedo, (c) Obtained Albedo, (d) Difference Image, (Bottom) Pixel-wise error variation in the initial albedo map $\boldsymbol{\rho}^{(0)}$.

normal and light source estimates. The expression in (3.2) can be rewritten as follows

$$\boldsymbol{\rho}_{i,j}^{(0)} = \frac{\boldsymbol{I}_{i,j}}{\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}} = \boldsymbol{\rho}_{i,j} \frac{\boldsymbol{n}_{i,j} \cdot \boldsymbol{s}}{\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}} \tag{3.3}$$

where $\boldsymbol{\rho}$, $\boldsymbol{n}$ and $\boldsymbol{s}$ are the true unknown albedo, normal and illuminant direction respectively. $\boldsymbol{\rho}^{(0)}$ can further be expressed as follows

$$\boldsymbol{\rho}_{i,j}^{(0)} = \boldsymbol{\rho}_{i,j} + \frac{\boldsymbol{n}_{i,j} \cdot \boldsymbol{s} - \boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}}{\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}} \boldsymbol{\rho}_{i,j} \tag{3.4}$$

Substituting $\boldsymbol{w}_{i,j} = \frac{\boldsymbol{n}_{i,j} \cdot \boldsymbol{s} - \boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}}{\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}} \boldsymbol{\rho}_{i,j}$, (3.4) simplifies to

$$\boldsymbol{\rho}_{i,j}^{(0)} = \boldsymbol{\rho}_{i,j} + \boldsymbol{w}_{i,j} \tag{3.5}$$

This can be identified with the standard image estimation formulation [7]. Here $\boldsymbol{\rho}$ is the original signal (true albedo), the initial estimate $\boldsymbol{\rho}^{(0)}$ is the degraded signal and $\boldsymbol{w}$ is the signal dependent additive noise.

3.2.2 LMMSE Estimate of Albedo

The Minimum Mean Square Error (MMSE) estimate of the albedo map $\boldsymbol{\rho}$ given noisy observed map $\boldsymbol{\rho}^{(0)}$ is the conditional mean

$$\hat{\boldsymbol{\rho}} = E(\boldsymbol{\rho}|\boldsymbol{\rho}^{(0)}) \tag{3.6}$$

In general, the MMSE estimate is non-linear and depends on the probability density functions of $\boldsymbol{\rho}$ and $\boldsymbol{w}$ and is difficult to estimate. Imposing linear constraint on the estimator structure, the LMMSE estimate is given by [105]

$$\hat{\boldsymbol{\rho}} = E(\boldsymbol{\rho}) + \mathrm{C}_{\rho\rho^{(0)}}\mathrm{C}^{-1}_{\rho^{(0)}}(\boldsymbol{\rho}^{(0)} - E(\boldsymbol{\rho}^{(0)})) \tag{3.7}$$

Here $\mathrm{C}_{\rho\rho^{(0)}}$ is the cross-covariance matrix of $\boldsymbol{\rho}$ and $\boldsymbol{\rho}^{(0)}$. $E(\boldsymbol{\rho}^{(0)})$ and $\mathrm{C}_{\rho^{(0)}}$ are the ensemble mean and covariance matrix of $\boldsymbol{\rho}^{(0)}$ respectively. The LMMSE filter requires the second order statistics of the signal and noise.

The expression for the signal-dependent noise $\boldsymbol{w}$ can be rewritten as follows

$$\boldsymbol{w}_{i,j} = \frac{(\boldsymbol{n}_{i,j} - \boldsymbol{n}^{(0)}_{i,j}) \cdot \boldsymbol{s} + \boldsymbol{n}^{(0)}_{i,j} \cdot (\boldsymbol{s} - \boldsymbol{s}^{(0)})}{\boldsymbol{n}^{(0)}_{i,j} \cdot \boldsymbol{s}^{(0)}}\boldsymbol{\rho}_{i,j} \tag{3.8}$$

Assuming the initial values of surface normal $\boldsymbol{n}^{(0)}$ and light source direction $\boldsymbol{s}^{(0)}$ to be unbiased, both $E(\boldsymbol{w})$ and $E(\boldsymbol{w}|\boldsymbol{\rho})$ are zero. Since noise is zero mean, $E(\boldsymbol{\rho}^{(0)}) = E(\boldsymbol{\rho})$. So $\mathrm{C}_{\rho\rho^{(0)}}$ can be written as

$$\begin{aligned} \mathrm{C}_{\rho\rho^{(0)}} &= E[(\boldsymbol{\rho} - E(\boldsymbol{\rho}))(\boldsymbol{\rho}^{(0)} - E(\boldsymbol{\rho}^{(0)}))^T] \\ &= \mathrm{C}_{\rho} + E[(\boldsymbol{\rho} - E(\boldsymbol{\rho}))\boldsymbol{w}^T] \end{aligned} \tag{3.9}$$

Similarly, if C_w is the covariance of the noise term, $\mathrm{C}_{\rho^{(0)}}$ can be written as

$$\begin{aligned}\mathrm{C}_{\rho^{(0)}} &= E[(\boldsymbol{\rho}^{(0)} - E(\boldsymbol{\rho}^{(0)}))(\boldsymbol{\rho}^{(0)} - E(\boldsymbol{\rho}^{(0)}))^T] \\ &= \mathrm{C}_\rho + \mathrm{C}_w + E[(\boldsymbol{\rho} - E(\boldsymbol{\rho}))\boldsymbol{w}^T] \\ &+ E[\boldsymbol{w}(\boldsymbol{\rho} - E(\boldsymbol{\rho}))^T] \end{aligned} \tag{3.10}$$

Recalling that $E(\boldsymbol{w})$ and $E(\boldsymbol{w}|\boldsymbol{\rho})$ are zero, $E\big((\boldsymbol{\rho} - E(\boldsymbol{\rho}))\boldsymbol{w}^T\big) = 0$. This simplifies (3.9) and (3.10) as follows

$$\mathrm{C}_{\rho\rho^{(0)}} = \mathrm{C}_\rho \;\text{ and }\; \mathrm{C}_{\rho^{(0)}} = \mathrm{C}_\rho + \mathrm{C}_w \tag{3.11}$$

In conventional image estimation problems, the original signal is assumed to be a wide-sense stationary random field. For albedo of real world objects, stationarity may be an oversimplified assumption. Figure 3.2(a) shows the albedo of a face. It is evident from the histogram (Figure 3.2(b)) which is not Gaussian that the albedo is not a stationary random field.

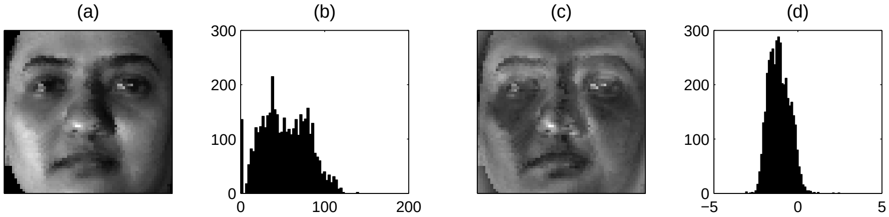


Figure 3.2: Non-stationary mean and non-stationary variance model for the true albedo. (a) True albedo, (b) Histogram of (a), (c) Normalized unit variance residual image, (d) Histogram of (c).

Here we assume a Non-stationary Mean Non-stationary Variance (NMNV) model [58] for the true albedo $\boldsymbol{\rho}$. Unlike the stationary model, the original signal is characterized by a non-stationary mean $E(\boldsymbol{\rho})$ which describes the gross structure of the signal. Under this model, the residual component $(\boldsymbol{\rho} - E(\boldsymbol{\rho}))$ which describes the signal variations is assumed to be a non-stationary white process, i.e. it is statistically uncorrelated and characterized by its non-stationary variance $\sigma^2_{i,j}(\boldsymbol{\rho})$. Figure 3.2(c) shows the normalized unit variance residual image. The normalized image seems to have Gaussian-like histogram (Figure 3.2(d)) which justifies the NMNV model for the true unknown albedo [58]. Under the NMNV assumption, C_ρ is a (non-constant) diagonal matrix. As C_ρ is diagonal, $\mathrm{C}_{\rho\rho^{(0)}}$ is diagonal. Since C_ρ is diagonal, C_w and thus $\mathrm{C}_{\rho^{(0)}}$ are also diagonal. Therefore, the LMMSE filtered output (3.7) simplifies to the following scalar (point) processor of the form

$$\begin{aligned} \hat{\boldsymbol{\rho}}_{i,j} &= E(\boldsymbol{\rho}_{i,j}) + \boldsymbol{\alpha}_{i,j}\big(\boldsymbol{\rho}^{(0)}_{i,j} - E(\boldsymbol{\rho}^{(0)}_{i,j})\big) \\ \text{where,} \qquad \boldsymbol{\alpha}_{i,j} &= \frac{\sigma^2_{i,j}(\boldsymbol{\rho})}{\sigma^2_{i,j}(\boldsymbol{\rho}) + \sigma^2_{i,j}(\boldsymbol{w})} \end{aligned} \tag{3.12}$$

where $\sigma^2_{i,j}(\boldsymbol{\rho})$ and $\sigma^2_{i,j}(\boldsymbol{w})$ are the non-stationary signal and noise variances respectively. Recalling that $E(\boldsymbol{\rho}^{(0)}) = E(\boldsymbol{\rho})$, (3.12) can be written as

$$\hat{\boldsymbol{\rho}}_{i,j} = (1 - \boldsymbol{\alpha}_{i,j})E\big(\boldsymbol{\rho}_{i,j}\big) + \boldsymbol{\alpha}_{i,j}\boldsymbol{\rho}^{(0)}_{i,j} \tag{3.13}$$

So the LMMSE albedo estimate is the weighted sum of the ensemble mean $E(\boldsymbol{\rho})$ and the observation $\boldsymbol{\rho}^{(0)}$, where the weight depends on the ratio of signal variance to the noise variance. For low signal to noise ratio (SNR) regions, more weight is given to the *a priori* mean $E(\boldsymbol{\rho})$ as the observation is too noisy to make an accurate estimate of the original signal. On the other hand, for high SNR regions, more weight is given to the observation.

3.2.3 Noise Variance

From (3.8), the signal-dependent noise $\boldsymbol{w}$ is

$$\boldsymbol{w}_{i,j} = \frac{(\boldsymbol{n}_{i,j} - \boldsymbol{n}^{(0)}_{i,j}) \cdot \boldsymbol{s} + \boldsymbol{n}^{(0)}_{i,j} \cdot (\boldsymbol{s} - \boldsymbol{s}^{(0)})}{\boldsymbol{n}^{(0)}_{i,j} \cdot \boldsymbol{s}^{(0)}}\boldsymbol{\rho}_{i,j} \tag{3.14}$$

We assume that the error in surface normal $(\boldsymbol{n}_{i,j} - \boldsymbol{n}_{i,j}^{(0)})$ is uncorrelated in x, y and z directions and their variances are same. A similar assumption on the error in the light source direction $(\boldsymbol{s} - \boldsymbol{s}^{(0)})$ leads to the following expression for the noise variance $\sigma^2(\boldsymbol{w})$

$$\sigma_{i,j}^2(\boldsymbol{w}) = \sigma_{i,j}^2(\boldsymbol{w}_1) + \sigma_{i,j}^2(\boldsymbol{w}_2) \tag{3.15}$$

where,

$$\sigma_{i,j}^2(\boldsymbol{w}_1) = \sigma_{i,j}^2(\boldsymbol{n}) \left(\frac{{\boldsymbol{s}_x}^2 + {\boldsymbol{s}_y}^2 + {\boldsymbol{s}_z}^2}{\left(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}\right)^2} \right) E\left(\boldsymbol{\rho}_{i,j}^2\right) \tag{3.16}$$

and

$$\sigma_{i,j}^2(\boldsymbol{w}_2) = \sigma^2(\boldsymbol{s}) \left(\frac{\left(n_x^{(0)}\right)^2 + \left(n_y^{(0)}\right)^2 + \left(n_z^{(0)}\right)^2}{\left(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}\right)^2} \right) E\left(\boldsymbol{\rho}_{i,j}^2\right) \tag{3.17}$$

Here $\sigma_{i,j}^2(\boldsymbol{n})$ and $\sigma^2(\boldsymbol{s})$ are the error variances in each direction of the surface normal and light source direction respectively. $\{s_x, s_y, s_z\}$ and $\{n_x^{(0)}, n_y^{(0)}, n_z^{(0)}\}$ are the three components of the illuminant direction and initial surface normal respectively. $[s_x, s_y, s_z]$ and $[n_x^{(0)}, n_y^{(0)}, n_z^{(0)}]$ being unit vectors, the expression for the noise variance can further be simplified as follows

$$\sigma_{i,j}^2(\boldsymbol{w}) = \frac{\sigma_{i,j}^2(\boldsymbol{n}) + \sigma^2(\boldsymbol{s})}{\left(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}\right)^2} E\left(\boldsymbol{\rho}_{i,j}^2\right) \tag{3.18}$$

Appropriately, the noise variance is proportional to the error variances of normal and light source estimates and the variance of the original signal. Interestingly, the noise variance is inversely proportional to $(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)})$ which is the cosine of the angle between the estimates of surface normal and light source direction.

We investigate the correctness of such a relation using a synthetically generated image. Figure 3.3 (left) shows the error in $\boldsymbol{\rho}^{(0)}$ for a synthetically generated face image. We see that the error actually varies inversely with $(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)})$ when all the other factors are constant. Such an observation can be attributed to the nature of the cosine function as shown in Figure 3.3 (right). When the angle is small,

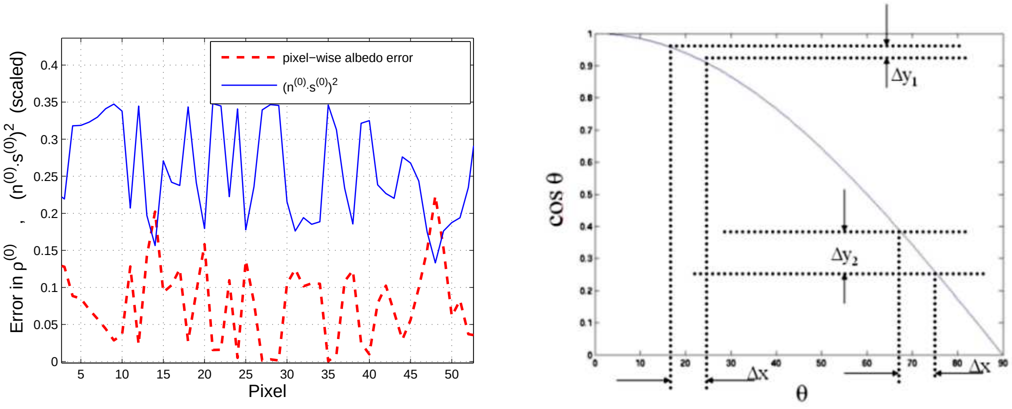


Figure 3.3: Left: Pixel-wise albedo error vs $\left(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}\right)^2$. Right: Cosine function explaining the error variation.

the cosine function changes slowly which implies that small errors in the angle estimate (Δx) will not adversely affect the accuracy of $\boldsymbol{\rho}^{(0)}$, i.e. Δy_1 is small. On the other hand, when the angle is large, even a small error in the angle estimate can lead to large errors in $\boldsymbol{\rho}^{(0)}$. The noise variance expression used in the proposed estimation framework is capable of accounting for this variation and thus has good potential to obtain a fairly accurate estimate of albedo. The various steps of the proposed algorithm to obtain the albedo estimate from an input intensity image are enumerated in Figure 3.4.

3.2.4 Illustration with synthetically generated data

Figure 3.5 shows the albedo maps obtained using the proposed algorithm for a face image. To facilitate comparisons with ground truth, the input face image is generated using 3D facial data [20]. Both correct and average facial surface normals are used as $\boldsymbol{n}^{(0)}$ to show the efficacy of our approach for a wide range of errors in surface normals. The other contextual information required to obtain the LMMSE estimate of the albedo is determined as follows

- The illuminant direction $s^{(0)}$ is estimated using [22]. $\sigma^2(\boldsymbol{s})$ is estimated by generating a large number of images under randomly selected lighting conditions

- Input: 2D intensity image $\boldsymbol{I}$ and average surface normal $\boldsymbol{n}^{(0)}$.
- Get initial estimate of source $\boldsymbol{s}^{(0)}$ in a least squares manner assuming unit albedo.
- Get initial raw estimate of albedo $\boldsymbol{\rho}^{(0)}$ using $\boldsymbol{n}^{(0)}$ and $\boldsymbol{s}^{(0)}$

$$\boldsymbol{\rho}_{i,j}^{(0)} = \frac{\boldsymbol{I}_{i,j}}{\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}}$$

- Estimate the non-stationary noise variance $\sigma_{i,j}^2(\boldsymbol{w})$ using

$$\sigma_{i,j}^2(\boldsymbol{w}) = \frac{\sigma_{i,j}^2(\boldsymbol{n}) + \sigma^2(\boldsymbol{s})}{\left(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}\right)^2} E\big(\boldsymbol{\rho}_{i,j}^2\big)$$

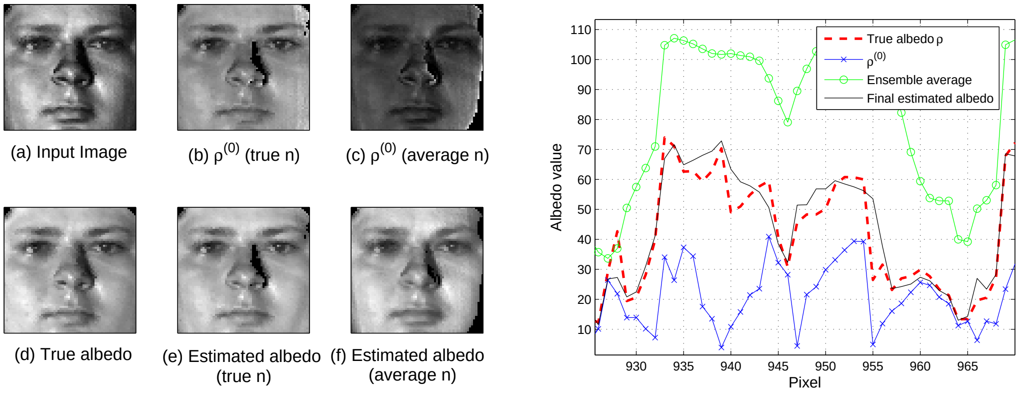
- Calculate the LMMSE estimate of the true unknown albedo by linearly combining the signal ensemble average $E\big(\boldsymbol{\rho}\big)$ and the initial raw albedo $\boldsymbol{\rho}^{(0)}$ as follows

$$\begin{aligned} \hat{\boldsymbol{\rho}}_{i,j} &= (1 - \boldsymbol{\alpha}_{i,j}) E\big(\boldsymbol{\rho}_{i,j}\big) + \boldsymbol{\alpha}_{i,j} \boldsymbol{\rho}_{i,j}^{(0)} \\ \text{where,} \qquad \boldsymbol{\alpha}_{i,j} &= \frac{\sigma_{i,j}^2(\boldsymbol{\rho})}{\sigma_{i,j}^2(\boldsymbol{\rho}) + \sigma_{i,j}^2(\boldsymbol{w})} \end{aligned}$$

Figure 3.4: Algorithm for finding the LMMSE estimate of albedo

and estimating their illumination directions.

- $\sigma^2(\boldsymbol{n})$ is estimated from 3D face data [20]. The data consists of surface normal information for 100 laser-scanned faces.
- Initial albedo $\boldsymbol{\rho}^{(0)}$ is obtained using (3.2).
- $E\big(\boldsymbol{\rho}\big)$, $\sigma^2(\boldsymbol{\rho})$ and $E\big(\boldsymbol{\rho}^2\big)$ are estimated from facial albedo data [20]. Figure 3.6 shows maps of $E\big(\boldsymbol{\rho}\big)$, $\sigma^2(\boldsymbol{\rho})$ and $\sigma^2(\boldsymbol{n})$.


(a) Input Image (b) $\rho^{(0)}$ (true n) (c) $\rho^{(0)}$ (average n)

(d) True albedo (e) Estimated albedo (true n) (f) Estimated albedo (average n)

Figure 3.5: Estimated albedo maps. Average per-pixel errors are in the ratio (b):(e):(c):(f) :: 17:9:26:12. The plot shows the final albedo estimate as compared to the true albedo, initial albedo $\boldsymbol{\rho}^{(0)}$ and the ensemble average.

The estimated albedo maps (Figure 3.5) seem to be free of *shadowy* effects present in the input image and are quite close to the true albedo map. As zero intensity pixels do not provide any albedo information, a few black regions can be seen in the estimated albedo maps.

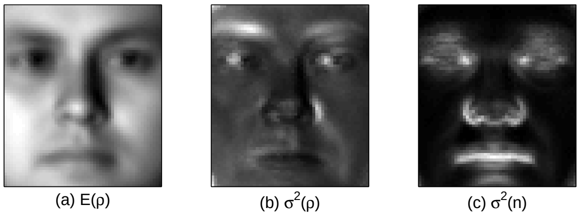

(a) E(ρ) (b) $\sigma^2(\rho)$ (c) $\sigma^2(n)$

Figure 3.6: Mean and variance maps used. (a) Ensemble mean of albedo, (b) Ensemble variance of albedo, (c) Error variance of the surface normal.

The improvement in the albedo maps can be explained using the plot in Figure 3.5. Though both the ensemble average and the initial albedo $\boldsymbol{\rho}^{(0)}$ are quite far from the true albedo, their linear combination follows the true value closely. Thus the approach does well in choosing appropriate combining coefficients $\boldsymbol{\alpha}_{i,j}$ in (3.13), so that the variation in the accuracy of $\boldsymbol{\rho}^{(0)}$ at different points is duly accounted for. The improvement obtained over $\boldsymbol{\rho}^{(0)}$ is significant and is consistent across images of different faces in several different challenging illumination conditions. When tried on 1000 images, the average reduction in per-pixel albedo error is observed to be over 33%. We observe that the improvement in albedo estimates is consistent and is not overly sensitive to the used statistics.

3.3 Shape Recovery

In this section, we focus on the general SFS problem of estimating the shape of an object with varying albedo map from a single image. This is an ill-posed problem with too many unknowns and just one constraint per pixel. Traditionally, assumptions like constant/piece-wise constant albedo and known illuminant direction are made to make the problem somewhat tractable. Though important to address the ever-elusive problem of shape recovery from a single image, these assumptions make the SFS approaches ineffective for real objects with varying albedo. In our approach, we transform the original problem of estimating shape of an object with varying albedo map and unknown illumination, to one of estimating the shape of an object with constant albedo and known light source direction that can be addressed using traditional SFS approaches.

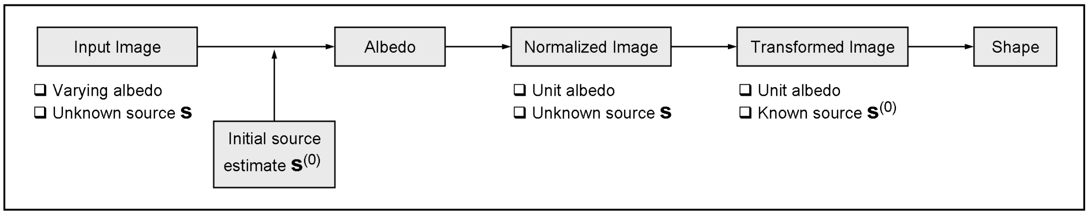


Figure 3.7: Schematic diagram of the proposed approach for shape recovery.

We describe in detail each step of the proposed algorithm using a face image as an example. We use 3D information of an average face model as the initial estimate. Using the average shape, we obtain an initial estimate of illuminant direction by formulating it as a linear Least Squares problem [22]

$$\boldsymbol{s}^{(0)} = \left(\sum_{i,j} \boldsymbol{n}_{i,j}^{(0)} \boldsymbol{n}_{i,j}^{(0)^T} \right)^{-1} \sum_{i,j} \boldsymbol{I}_{i,j} \boldsymbol{n}_{i,j}^{(0)} \tag{3.19}$$

where $\boldsymbol{n}^{(0)}$ is the average facial surface normal. Starting with these initial normal and illuminant estimates, the algorithm proceeds as follows (Figure 3.7)

3.3.1 Albedo Estimation and Image Normalization

Given an image, a robust albedo estimate is determined using the image estimation approach described in the preceding section. The albedo estimate $\hat{\boldsymbol{\rho}}$ is used to normalize the input image to obtain an *albedo-free* image $\boldsymbol{G}$ as follows

$$\boldsymbol{G}_{i,j} = \frac{\boldsymbol{I}_{i,j}}{\hat{\boldsymbol{\rho}}_{i,j}} \tag{3.20}$$

The normalized image $\boldsymbol{G}$ is an image of an object with the same shape as that of the original one but with unit albedo map. Figure 3.8 shows an example of the normalized image obtained from a synthetic face image. The normalized image appears quite close to the *true* normalized image obtained directly from the shape information. Also, both the images are quite different from the input image highlighting the importance of such a normalization step before shape estimation.

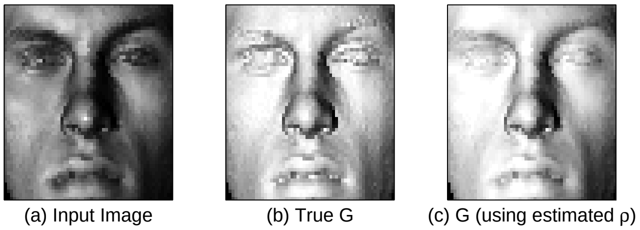

(a) Input Image
(b) True G
(c) G (using estimated ρ)

Figure 3.8: Normalized image obtained using the albedo estimate.

For a Lambertian object, $\boldsymbol{G}_{i,j}$ represents the cosine of the angle between the true unknown surface normal $\boldsymbol{n}_{i,j}$ and the true unknown illuminant direction $\boldsymbol{s}$ as illustrated in Figure 3.10 (right). So using $\boldsymbol{s}^{(0)}$ to recover the shape may introduce errors in the output depending on the errors in the source estimate. Though the normalized image can potentially be used to estimate the surface normal and refine the illuminant direction estimate using a suitable iterative optimization scheme (e.g., [22]), most traditional SFS approaches assume known light source direction because of possible stability issues in such iterative optimizations. Here, we propose an estimation framework to transform $\boldsymbol{G}$ to another *albedo-free* image illuminated by a known light source.

3.3.2 Image Transformation

In this step, we transform $\boldsymbol{G}_{i,j}$ to another image $\boldsymbol{H}_{i,j}$ that represents the cosine of the angle between the true unknown surface normal $\boldsymbol{n}_{i,j}$ and the known light source estimate $\boldsymbol{s}^{(0)}$ (Figure 3.10 (right)). An image estimation framework that utilizes the statistics of error in the source estimate is used for this task. The normalized image $\boldsymbol{G}$ can be written as

$$\boldsymbol{G} = \boldsymbol{n}^T \boldsymbol{s} \tag{3.21}$$

Now for each pixel, writing the true illuminant direction $\boldsymbol{s}$ in terms of the initial estimate $\boldsymbol{s}^{(0)}$ and the difference between the two, we obtain

$$\boldsymbol{G}_{i,j} = \boldsymbol{n}_{i,j} \cdot \boldsymbol{s}^{(0)} + \boldsymbol{n}_{i,j} \cdot (\boldsymbol{s} - \boldsymbol{s}^{(0)}) \tag{3.22}$$

Identifying that $\boldsymbol{H}_{i,j}$ represents the cosine of the angle between the true normal $\boldsymbol{n}_{i,j}$ and the initial estimate of the illuminant direction $\boldsymbol{s}^{(0)}$, (3.22) simplifies to

$$\boldsymbol{G}_{i,j} = \boldsymbol{H}_{i,j} + \boldsymbol{\nu}_{i,j} \tag{3.23}$$

where $\boldsymbol{\nu}_{i,j} = \boldsymbol{n}_{i,j} \cdot (\boldsymbol{s} - \boldsymbol{s}^{(0)})$. As before, this can be viewed as an image estimation problem to obtain an estimate of the transformed image $\boldsymbol{H}$. Here the normalized image $\boldsymbol{G}$ is the degraded signal and $\boldsymbol{\nu}$ is the observation noise.

Similar to the albedo estimation case, imposing linear constraint on the estimator structure, the LMMSE estimate is given by [105]

$$\begin{aligned} \hat{\boldsymbol{H}}_{i,j} &= (1 - \boldsymbol{\beta}_{i,j})E\big(\boldsymbol{H}_{i,j}\big) + \boldsymbol{\beta}_{i,j}\boldsymbol{G}_{i,j} \\ \text{where,} \qquad \boldsymbol{\beta}_{i,j} &= \frac{\sigma^2_{i,j}(\boldsymbol{H})}{\sigma^2_{i,j}(\boldsymbol{H}) + \sigma^2_{i,j}(\boldsymbol{\nu})} \end{aligned} \tag{3.24}$$

Here $\sigma^2(\boldsymbol{\nu})$ and $\sigma^2(\boldsymbol{H})$ are the non-stationary noise and signal variance respectively. The derivation for the expression for the LMMSE estimate of $\boldsymbol{H}$ in (3.24) follows in the same fashion as for albedo and hence omitted for brevity. As before, assuming that the error in the illuminant direction $(\boldsymbol{s} - \boldsymbol{s}^{(0)})$ is uncorrelated in the x, y and z directions with the same variance $\sigma^2(\boldsymbol{s})$, we have

$$\sigma^2_{i,j}(\boldsymbol{\nu}) = ({n_x}^2 + {n_y}^2 + {n_z}^2)\sigma^2(\boldsymbol{s}) = \sigma^2(\boldsymbol{s}) \tag{3.25}$$

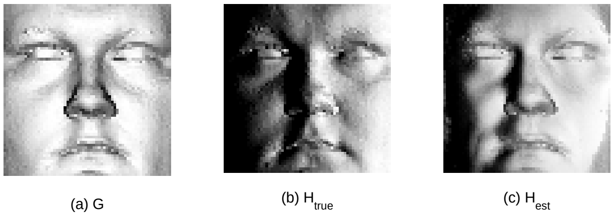


Figure 3.9: The transformed image (c) obtained using the proposed estimation framework. For comparison, the normalized image $\boldsymbol{G}$ and the *true* $\boldsymbol{H}$ (generated from the 3D data) are also shown. Average per-pixel errors in $\boldsymbol{G}$ and $\hat{\boldsymbol{H}}$ are in the ratio of 3:1.

Figure 3.9 shows the transformed image obtained using this approach. The true source direction in this case is [0, 0, -1] while $\boldsymbol{s}^{(0)}$ is taken to be [-0.86, 0, -0.52]. Clearly, the advantage gained using this image transformation step depends on the error in $\boldsymbol{s}^{(0)}$. We perform an experiment to observe the deviation in $\boldsymbol{G}$ and $\hat{\boldsymbol{H}}$ from

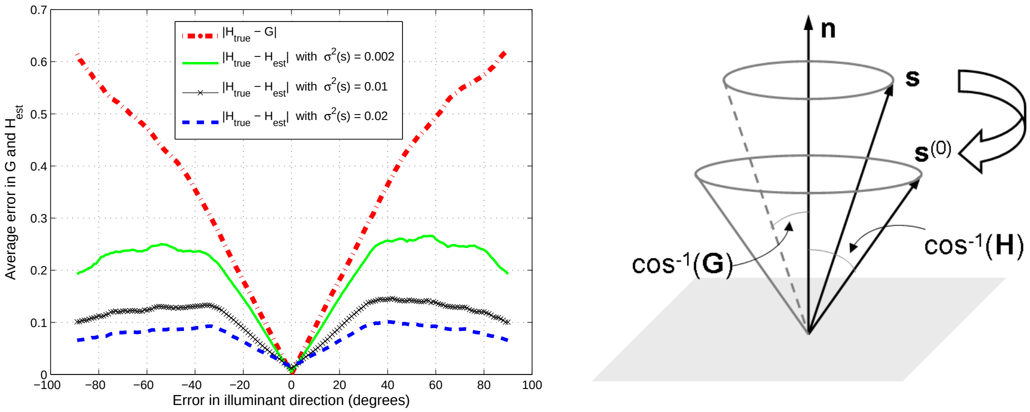


Figure 3.10: Left: The variation of error in $\boldsymbol{G}$ and $\hat{\boldsymbol{H}}$ with an increase in the angular difference between the estimated and true illuminant direction for different values of $\sigma^2(\boldsymbol{s})$, Right: Image transformation.

$\boldsymbol{H}$ with the increase in error in the estimated illuminant direction. As shown in Figure 3.10 (left), both $\boldsymbol{G}$ and $\hat{\boldsymbol{H}}$ are close to $\boldsymbol{H}$ when the error in $\boldsymbol{s}^{(0)}$ is small. The difference $|\boldsymbol{H} - \boldsymbol{G}|$ increases almost linearly with an increase in the source error. On the other hand, the error in $\hat{\boldsymbol{H}}$ saturates quickly, highlighting the advantage of the proposed estimation framework to obtain a reliable estimate of $\boldsymbol{H}$. The experiment is repeated with different values for the source error variance. The value of $\sigma^2(\boldsymbol{s})$ indicates the confidence in the estimated illuminant direction and thus the weight given to the normalized image $\boldsymbol{G}$ in comparison to the ensemble average $E\big(\boldsymbol{H}\big)$.

To evaluate the usefulness of the proposed transformation from $\boldsymbol{G}$ to $\boldsymbol{H}$, we estimate the typical errors in light source estimate using 1000 synthetic images and estimating illuminant direction using (3.19). The average error in the source direction estimate is around 16° indicating approximately 50% reduction (0.145 to 0.07) in average per-pixel error from $\boldsymbol{G}$ to $\hat{\boldsymbol{H}}$ for $\sigma^2(\boldsymbol{s}) = 0.01$, the value used in our experiments. The different contextual information required to obtain the transformed image is determined from the 3D facial surface normal data [20]. The estimated illuminant direction $\boldsymbol{s}^{(0)}$ is used for generating a large number of tran-

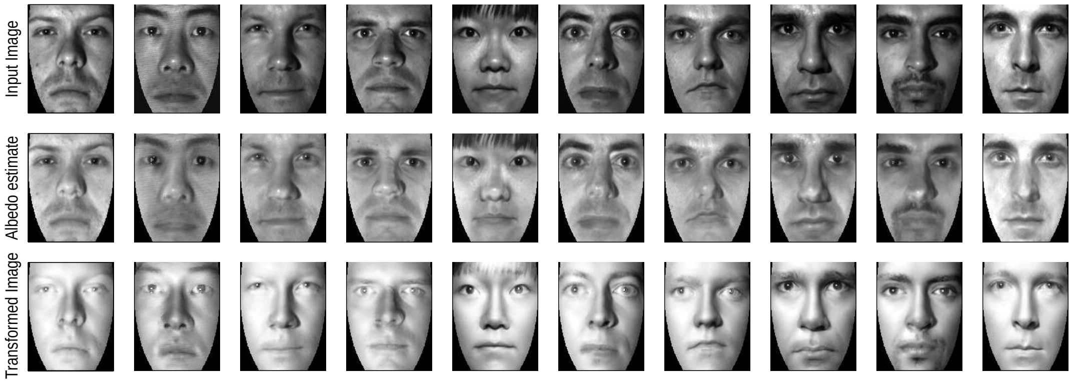


Figure 3.11: The estimated albedo maps $\hat{\boldsymbol{\rho}}$ and transformed images $\hat{\boldsymbol{H}}$ obtained for a few subjects from Yale dataset [41].

formed images which are then used for estimating the ensemble mean $E\big(\boldsymbol{H}\big)$ and variance $\sigma^2(\boldsymbol{H})$. Figure 3.10 (right) visually demonstrates the image transformation procedure.

Figure 3.11 shows the albedo maps and transformed images obtained using the proposed approach on real images from the Yale Face Database B [41]. As desired, the transformed images seem to be less affected due to variations in albedo than the original input images. Note that $\hat{\boldsymbol{H}}$ is the LMMSE estimate of the image of an object with same shape as the original object but with unit albedo map and illuminated by the light source $\boldsymbol{s}^{(0)}$. Thus one can use a suitable SFS algorithm to solve for the unknown shape.

3.3.3 Shape Estimation

In our implementation, we use the SFS approach by Tsai and Shah [130] that uses a linear approximation of the reflectance function. Here we provide a brief overview of the method for completion. For Lambertian surfaces, the reflectance function $\boldsymbol{R}$ has the following form

$$\boldsymbol{R}(\boldsymbol{p}_{i,j}, \boldsymbol{q}_{i,j}) = \frac{\boldsymbol{s} \cdot [\boldsymbol{p}_{i,j}, \boldsymbol{q}_{i,j}, 1]^T}{\sqrt{1 + \boldsymbol{p}_{i,j}^2 + \boldsymbol{q}_{i,j}^2}} \tag{3.26}$$

where $\boldsymbol{p}_{i,j}$ and $\boldsymbol{q}_{i,j}$ are the surface gradients. Employing discrete approximations for $\boldsymbol{p}$ and $\boldsymbol{q}$ using finite differences, we get

$$\begin{aligned} 0 &= f(\hat{\boldsymbol{H}}_{i,j}, \boldsymbol{Z}_{i,j}, \boldsymbol{Z}_{i-1,j}, \boldsymbol{Z}_{i,j-1}) \\ &= \hat{\boldsymbol{H}}_{i,j} - \boldsymbol{R}(\boldsymbol{Z}_{i,j} - \boldsymbol{Z}_{i-1,j}, \boldsymbol{Z}_{i,j} - \boldsymbol{Z}_{i,j-1}) \qquad (3.27) \end{aligned}$$

where $\boldsymbol{Z}_{i,j}$ denotes the depth values. For a given transformed image $\hat{\boldsymbol{H}}$, a linear approximation of the function f about a given depth map $\boldsymbol{Z}^{n-1}$ leads to a linear system of equations that can be solved using the Jacobi iterative scheme as follows

$$0 = f(\boldsymbol{Z}_{i,j}) \approx f(\boldsymbol{Z}^{n-1}_{i,j}) + (\boldsymbol{Z}_{i,j} - \boldsymbol{Z}^{n-1}_{i,j}) \frac{d}{d\boldsymbol{Z}_{i,j}} f(\boldsymbol{Z}^{n-1}_{i,j}) \qquad (3.28)$$

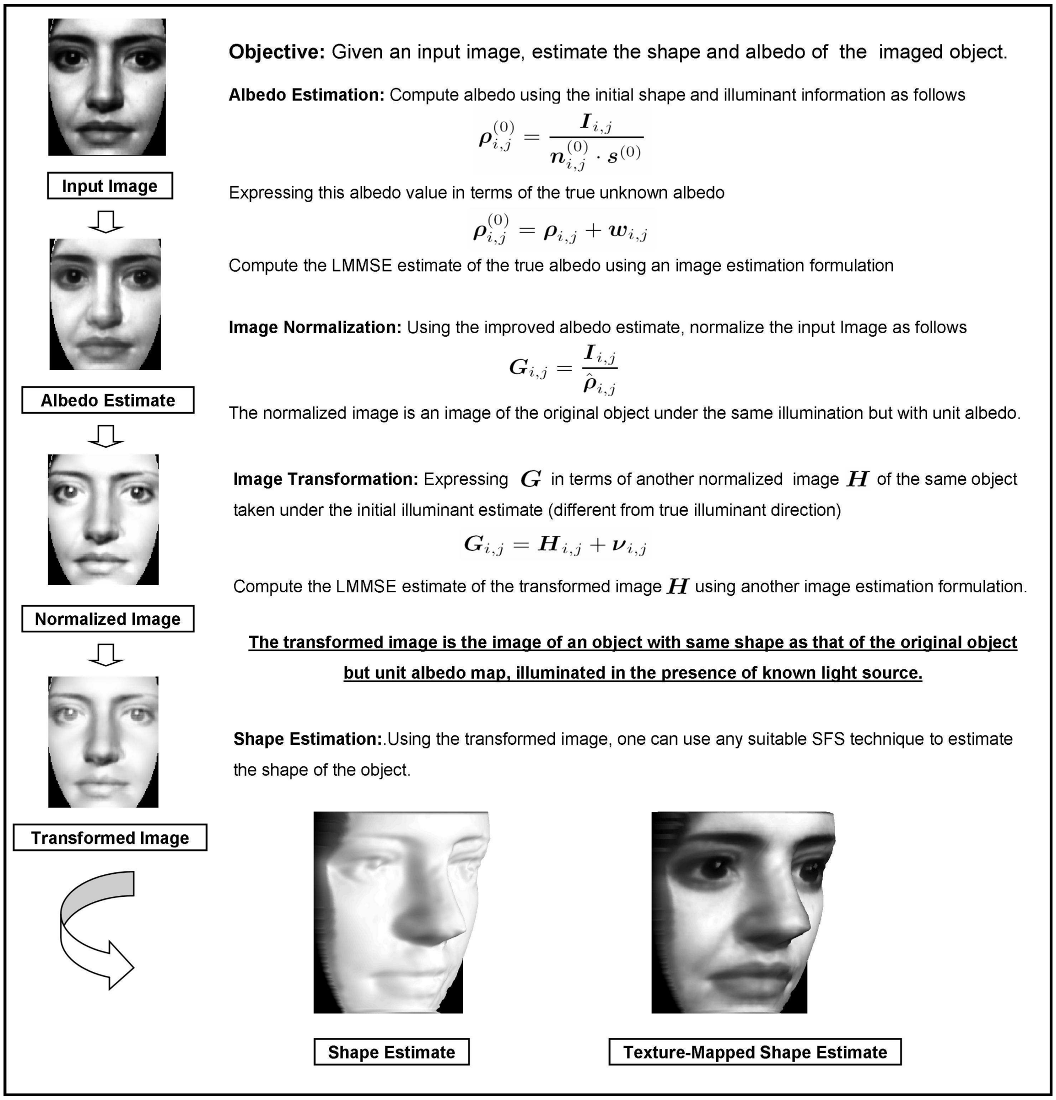
Now for $\boldsymbol{Z}_{i,j} = \boldsymbol{Z}^{n}_{i,j}$, the depth map at n-th iteration can be solved using

$$\boldsymbol{Z}^{n}_{i,j} = \boldsymbol{Z}^{n-1}_{i,j} + \frac{-f(\boldsymbol{Z}^{n-1}_{i,j})}{\frac{d}{d\boldsymbol{Z}_{i,j}} f(\boldsymbol{Z}^{n-1}_{i,j})} \qquad (3.29)$$

In our experiments, we use the domain-specific average shape as the initial depth map. Figure 3.12 shows the various steps of the proposed albedo estimation and shape recovery algorithm. Depending on the application, one can potentially repeat the sequence of steps with the updated estimates to further refine the albedo, shape and illuminant direction. In our experiments with face images, we do not see much improvement in the estimates after first (or first few) iteration(s). Therefore, all the results shown here are generated using a single parse through the proposed steps. The whole process takes around 2-3 seconds using an unoptimized MATLAB code on a regular desktop.

3.4 Experiments

We provide details of the experiments performed to evaluate the robustness and usefulness of the albedo and shape estimates obtained using the proposed image



Objective: Given an input image, estimate the shape and albedo of the imaged object.

Albedo Estimation: Compute albedo using the initial shape and illuminant information as follows

$$\boldsymbol{\rho}^{(0)}_{i,j} = \frac{\boldsymbol{I}_{i,j}}{\boldsymbol{n}^{(0)}_{i,j} \cdot \boldsymbol{s}^{(0)}}$$

Expressing this albedo value in terms of the true unknown albedo

$$\boldsymbol{\rho}^{(0)}_{i,j} = \boldsymbol{\rho}_{i,j} + \boldsymbol{w}_{i,j}$$

Compute the LMMSE estimate of the true albedo using an image estimation formulation

Image Normalization: Using the improved albedo estimate, normalize the input Image as follows

$$\boldsymbol{G}_{i,j} = \frac{\boldsymbol{I}_{i,j}}{\hat{\boldsymbol{\rho}}_{i,j}}$$

The normalized image is an image of the original object under the same illumination but with unit albedo.

Image Transformation: Expressing $\boldsymbol{G}$ in terms of another normalized image $\boldsymbol{H}$ of the same object taken under the initial illuminant estimate (different from true illuminant direction)

$$\boldsymbol{G}_{i,j} = \boldsymbol{H}_{i,j} + \boldsymbol{\nu}_{i,j}$$

Compute the LMMSE estimate of the transformed image $\boldsymbol{H}$ using another image estimation formulation.

The transformed image is the image of an object with same shape as that of the original object but unit albedo map, illuminated in the presence of known light source.

Shape Estimation:.Using the transformed image, one can use any suitable SFS technique to estimate the shape of the object.

Figure 3.12: Visual demonstration of the proposed algorithm.

estimation framework. We provide examples of albedo estimates obtained on real images. The albedo estimates are also used to relight images under frontal illumination condition. The illumination-insensitivity of the albedo estimates is highlighted by using them to recognize faces across illumination variations. The face recognition application requires albedo estimates from images illuminated by multiple illumination sources. Therefore, the proposed estimation framework is extended to deal with realistic multiple source scenarios.

Other than the image estimation framework to estimate albedo, the main contribution of the work is in transforming the input image to an albedo-free image taken in the presence of known light source which can be used for shape recovery using existing techniques. Therefore, we perform a recognition experiment to evaluate the efficacy of the transformed image before using it to recover 3D shape. The recovered shapes are compared with the available 3D information. The effectiveness of the approach is further highlighted by using the shapes recovered from images downloaded from the web to generate novel views taken under novel illumination.

3.4.1 Illumination-Insensitivity of Estimated Albedo

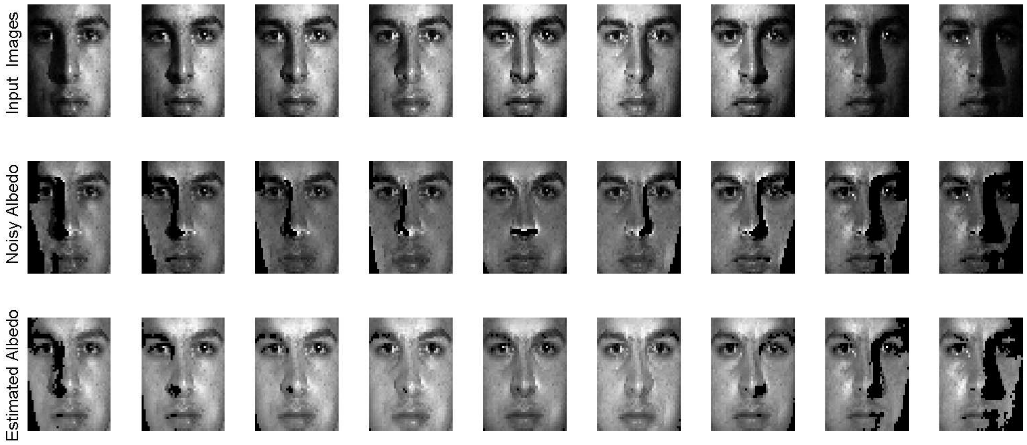


Figure 3.13: Albedo estimates obtained from several images of a subject from the PIE dataset [115].

Figure 3.13 shows the albedo maps obtained from several images of a subject from the PIE dataset [115] taken under different illumination conditions. Average facial surface normals are used as $\boldsymbol{n}^{(0)}$. The illuminant direction $\boldsymbol{s}^{(0)}$ is estimated using (3.19). The final albedo estimates obtained using the proposed approach appear much better than the initial erroneous ones and do not seem to have the *shadowy* effects present in the input images. In addition, as desired the estimated albedo maps appear quite similar to each other.

We also perform a relighting experiment using the estimated albedo maps to generate images under frontal illumination. The relighting is performed by combining the estimated albedo maps with average facial shape information. Figure 3.14 (second row) shows the relighted images and the corresponding input images (first row) taken under challenging illumination conditions. The relighted images seem quite similar to the actual frontally illuminated images of the same subjects from the dataset shown in the third row.

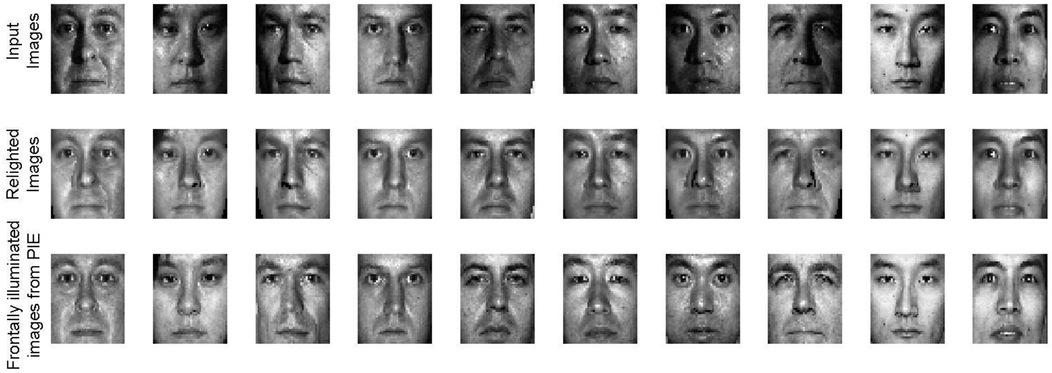


Figure 3.14: Relighting results on a few images from the PIE dataset [115].

3.4.2 Face Recognition

We now evaluate the usefulness of the estimated albedo maps as an illumination-insensitive signature. We perform recognition experiments on the PIE dataset that contains face images of 68 subjects taken under several different illumination conditions. Given the estimated albedo maps, the similarity between images is measured using Principal Component Analysis (PCA). FRGC [95] training data consisting of 366 face images is used to generate the (albedo) PCA space. Recognition is performed across illumination with images from one illumination condition forming the gallery while images from another illumination condition forming the probe set. In this experiment setting, each gallery and probe set contains just one image per subject. Table 3.1 shows the rank-1 recognition results obtained. Each entry in the table shows the rank-1 performance obtained for one choice of gallery and probe set. The albedo maps perform quite well as illumination-insensitive signatures with an overall average recognition accuracy of 94%.

As shown in the table, the performance compares favorably with those of [145] and [4] which follow similar experimental setting. The performance is also comparable to the one reported by Romdhani *et al.* [103] and Zhang and Samaras [139]. Using a 3D morphable model based algorithm, Romdhani *et al.* obtain an average recognition rate of 98% (same as ours with f_{12} as gallery) using the frontally illuminated images (flash f_{12}) as gallery. An average recognition rate of 99% (with f_{12} as gallery) is reported by Zhang and Samaras using a spherical harmonics-based approach. Initial albedo maps $\boldsymbol{\rho}^{(0)}$ perform poorly in this experiment with an overall average rank-1 performance of 27%. This may be due to the fact that average facial normals are far from the true normals leading to large errors. It is worthwhile to note that the proposed estimation framework is able to take care of such large errors leading to good recognition performance.

There are quite a few differences between [103, 139, 145, 4] and the proposed approach that warrant some clarifications. First, our experimental setup is restrictive with face images assumed to be in frontal pose (though it is potentially extensible to deal with non-frontal poses). However, there are quite a few advantages that the proposed approach offers as compared to other existing approaches. The proposed algorithm does not involve any costly optimization step and is easily and efficiently implementable. In addition, albedo estimation requires limited domain knowledge in the form of ensemble means and variances. In fact, as discussed in Section 3.4.6, one can replace the ensemble information by local statistics to obtain albedo estimates using the proposed framework. This makes the approach relatively easier to extend to general objects where the domain-dependent statistics is not available. Moreover, the proposed albedo estimation approach does not impose a linear statistical constraint on the unknown albedo and can be easily

Table 3.1: Recognition results on the PIE dataset [115] using the estimated albedo. We include averages from [145] and [4] for comparison. f_i denotes images with i^{th} flash ON as labeled in PIE. Each $(i, j)^{th}$ entry is the rank-1 recognition rate obtained with the images from f_i as gallery and f_j as probes. F denotes 100 percent.

Probe	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Avg	Avg [145]	Avg [4]
Gallery															
f_{08}	-	F	F	99	93	91	79	72	44	F	96	85	87	89	92
f_{09}	F	-	F	F	99	97	91	90	75	F	99	93	95	93	97
f_{11}	F	F	-	F	F	97	88	78	57	F	F	93	92	92	95
f_{12}	99	99	F	-	F	F	96	96	87	F	F	97	98	96	98
f_{13}	99	99	F	F	-	F	99	99	90	99	F	F	99	98	F
f_{14}	97	99	F	F	F	-	99	97	90	F	F	F	98	99	99
f_{15}	84	94	88	F	F	F	-	F	99	93	F	F	96	96	97
f_{16}	76	97	79	99	F	99	99	-	F	75	99	F	93	91	94
f_{17}	53	82	56	90	96	94	94	F	-	54	96	97	83	80	87
f_{20}	F	F	F	F	F	F	94	78	57	-	F	99	93	91	95
f_{21}	99	99	F	F	F	F	93	94	85	F	-	97	97	96	99
f_{22}	90	99	97	F	F	F	F	97	91	97	F	-	97	98	98
Avg	91	97	93	99	99	98	94	91	80	93	99	96	94	-	-
Avg [145]	88	94	93	97	99	99	96	89	75	93	98	98	-	93	-
Avg [4]	90	97	94	99	99	99	98	93	87	95	99	99	-	-	96

extended to realistic multiple illumination scenarios (Section 3.4.3).

3.4.3 Albedo Estimation in Multi-source Scenario

Our analysis so far assumes that the image is illuminated by a single light source. However, the assumption does not hold in many realistic scenarios. One of the main challenges in handling multiple light sources is the absence of *a priori* knowledge of the number and placement of the sources. To handle this, we use the result established by Lee *et al.* [70] that an image of an arbitrarily illuminated object can be approximated by a linear combination of the images of the same object in the same pose, illuminated by nine different light sources placed at pre-selected positions. Using this approximation, the image formation equation becomes

$$\boldsymbol{I} = \sum_{k=1}^{9} \boldsymbol{\gamma}_k \boldsymbol{I}_k \quad \text{where,} \;\; \boldsymbol{I}_k = \boldsymbol{\rho} \max\left(\boldsymbol{n} \cdot \boldsymbol{s}_k, 0\right) \tag{3.30}$$

$\{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_9\}$ are the pre-specified illumination directions. The following nine illumination directions [70] are used

$$\begin{aligned}\boldsymbol{\phi} &= \{0, 49, -68, 73, 77, -84, -84, 82, -50\}^o \\ \boldsymbol{\theta} &= \{0, 17, 0, -18, 37, 47, -47, -56, -84\}^o \qquad (3.31)\end{aligned}$$

Since the source directions are pre-specified, the only unknown to estimate the illumination conditions is $\boldsymbol{\gamma}$. Given an image $\boldsymbol{I}$ of an object and domain-dependent average surface normals $\boldsymbol{n}^{(0)}$, $\boldsymbol{\gamma}$ is estimated in a least squares sense as follows

$$\hat{\gamma} = \boldsymbol{W}^{\dagger}\boldsymbol{I} \qquad (3.32)$$

where $\boldsymbol{I}$ is the N dimensional vectorized image and $\boldsymbol{W}_{N\times 9}$ is given by

$$\boldsymbol{W} = \begin{pmatrix} \max(\boldsymbol{n}_1^{(0)} \cdot \boldsymbol{s}_1, 0) & \ldots & \max(\boldsymbol{n}_1^{(0)} \cdot \boldsymbol{s}_9, 0) \\ \max(\boldsymbol{n}_2^{(0)} \cdot \boldsymbol{s}_1, 0) & \ldots & \max(\boldsymbol{n}_2^{(0)} \cdot \boldsymbol{s}_9, 0) \\ \vdots & \ddots & \vdots \\ \max(\boldsymbol{n}_N^{(0)} \cdot \boldsymbol{s}_1, 0) & \ldots & \max(\boldsymbol{n}_N^{(0)} \cdot \boldsymbol{s}_9, 0) \end{pmatrix}$$

The coefficients $\hat{\boldsymbol{\gamma}}$ can be used to obtain an initial value of the albedo for each pixel as follows

$$\boldsymbol{\rho}_{i,j}^{(0)} = \frac{\boldsymbol{I_{i,j}}}{\sum_{k=1}^{9} \hat{\gamma}_k \max(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}_k, 0)} \qquad (3.33)$$

Robust albedo estimate is obtained by formulating an image estimation problem as follows

$$\boldsymbol{\rho}_{i,j}^{(0)} = \boldsymbol{\rho}_{i,j} + \boldsymbol{w}_{i,j} \qquad (3.34)$$

where the signal dependent noise $\boldsymbol{w}_{i,j}$ is given by

$$\boldsymbol{w}_{i,j} = \frac{\sum_{k=1}^{9} \left[(\boldsymbol{\gamma}_k - \hat{\boldsymbol{\gamma}}_k)\boldsymbol{n} \cdot \boldsymbol{s}_k + \hat{\boldsymbol{\gamma}}_k(\boldsymbol{n} - \boldsymbol{n}^{(0)}) \cdot \boldsymbol{s}_k \right]}{\sum_{k=1}^{9} \hat{\boldsymbol{\gamma}}_k \boldsymbol{n}^{(0)} \cdot \boldsymbol{s}_k} \boldsymbol{\rho}$$

Subscripts $(i,j)'s$ and explicit max operator have been dropped in the above expression for clarity. Similar to the analysis for the single light source case, the NMNV model for the true albedo leads to the following LMMSE albedo estimate

$$\hat{\boldsymbol{\rho}}_{i,j} = (1-\boldsymbol{\alpha}_{i,j})E\big(\boldsymbol{\rho}_{i,j}\big) + \boldsymbol{\alpha}_{i,j}\boldsymbol{\rho}^{(0)}_{i,j} \quad \text{where,} \;\; \boldsymbol{\alpha}_{i,j} = \frac{\sigma^2_{i,j}(\boldsymbol{\rho})}{\sigma^2_{i,j}(\boldsymbol{\rho}) + \sigma^2_{i,j}(\boldsymbol{w})} \tag{3.35}$$

Assuming that the errors in estimation of $\boldsymbol{\gamma}_k$'s are uncorrelated and have same variance $\sigma^2(\boldsymbol{\gamma})$, the noise variance for each pixel is given by

$$\begin{aligned}\sigma^2(\boldsymbol{w}) &= \frac{\sum_{k=1}^{9}\left[\sigma^2(\boldsymbol{\gamma})E[(\boldsymbol{n}\cdot\boldsymbol{s}_k)^2] + \sigma^2(\boldsymbol{n})\hat{\boldsymbol{\gamma}}_k^2\right]}{(\sum_{k=1}^{9}\hat{\boldsymbol{\gamma}}_k\boldsymbol{n}^{(0)}\cdot\boldsymbol{s}_k)^2}E(\boldsymbol{\rho}^2)\\ \text{where,} \quad E[(\boldsymbol{n}\cdot\boldsymbol{s}_k)^2] &= (\boldsymbol{n}^{(0)}\cdot\boldsymbol{s}_k)^2 + \sigma^2(\boldsymbol{\gamma})\sigma^2(\boldsymbol{n})\end{aligned}$$

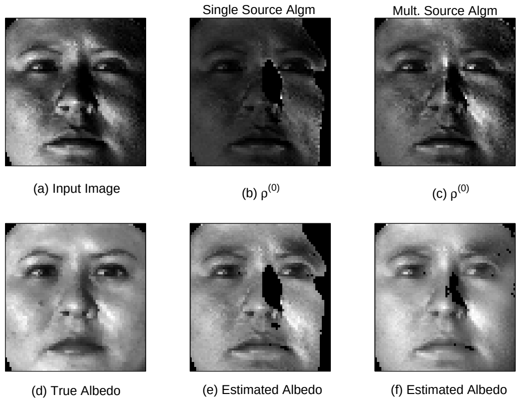


Figure 3.15: Comparison between albedo maps obtained using single and multi-source frameworks. Average per-pixel errors are in the ratio (b):(e):(c):(f)::30:19:22:12. The input image is illuminated by three light sources.

Figure 3.15 shows the albedo maps obtained using the proposed approach for a face image illuminated by three light sources. Average facial surface normals are used as $\boldsymbol{n}^{(0)}$ in this experiment. The proposed multi-source approach provides the best result and has less *shadowy* effects than the others. The single source estimation framework also improves the corresponding initial albedo $\boldsymbol{\rho}^{(0)}$, but

Table 3.2: Accuracy of albedo estimates using single and multiple source frameworks. Entries show average per-pixel albedo errors.

	Single source framework	Multiple source framework	Error reduction
$\boldsymbol{\rho}^{(0)}$	35	22	37%
$\hat{\boldsymbol{\rho}}$	23	14	39%
Error reduction	34%	36%	60%

the result is not as good as the one obtained using the multiple source algorithm. Table 3.2 compares the accuracy of various albedo estimates obtained from images illuminated by multiple light sources. As shown, the single light source assumption results in larger errors. Overall improvement in the albedo estimate obtained by the multiple light framework as compared to the initial albedo map obtained under the single light source assumption is about 60%. One thousand images were used to generate the statistics.

3.4.4 Face Recognition : Multiple Light Sources

In this section, we evaluate the usefulness of the proposed multi-source framework over the single source one when the images are illuminated by several light sources. In the absence of a controlled multi-light source dataset, we generate multi-light source scenarios using the PIE dataset by combining multiple images for each subject. Randomly chosen two, three or four images under different illumination conditions are combined to form twelve different multi-light source scenarios. For the recognition experiment, one image per subject is randomly selected from the twelve illumination conditions for gallery and another one for the probe set. The experiment is repeated 100 times for different random combinations of gallery and probe sets. Figure 3.16 shows Cumulative Match Characteristic (CMC) curves comparing the recognition performance obtained using the albedo maps estimated by the single source framework with that of multiple source framework. Error bars reflect the variations in recognition performance for the different trials. As expected, the multi-source framework significantly outperforms the single-source one. As before, the initial albedo maps $\boldsymbol{\rho}^{(0)}$ for both single and multiple source frameworks perform poorly in this experiment.

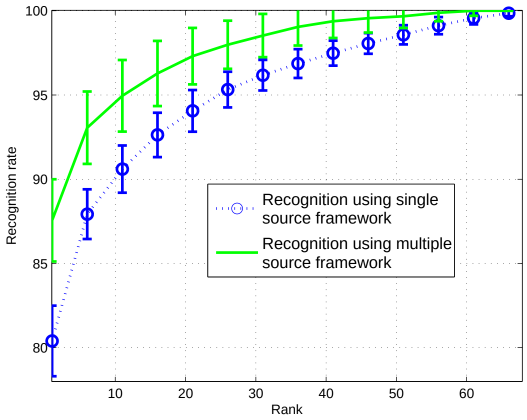


Figure 3.16: Recognition performance on *multiply*-illuminated PIE dataset.

3.4.5 Shape Recovery

As far as the problem of shape recovery is concerned, the main contribution of this work is in generating the *albedo-free* transformed image with known illuminant direction. Therefore, we first evaluate the robustness of the transformed images generated using the proposed approach. Unlike albedo estimates, the transformed images are not illumination-invariant. In fact, a transformed image represents the cosine of the angle between the true unknown surface normal and the known estimate of light source direction, which depends on the illumination in the input image (Figure 3.10 (right)). Therefore, unlike albedo, one cannot directly perform a recognition experiment to evaluate the accuracy of the transformed images. Instead, we make use of the statistical facial shape information to

Table 3.3: Recognition results on the PIE dataset [115] using the transformed images. f_i denotes images with i^{th} flash ON as labeled in PIE. Each $(i,j)^{th}$ entry is the rank-1 recognition rate obtained with the images from f_i as gallery and f_j as probes. F denotes 100 percent.

Probe	f_{08}	f_{09}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{20}	f_{21}	f_{22}	Avg
Gallery													
f_{08}	-	99	99	94	88	74	53	47	26	97	85	57	74
f_{09}	94	-	94	99	99	94	71	66	46	93	99	79	85
f_{11}	99	99	-	F	99	96	74	57	46	F	F	87	87
f_{12}	91	99	F	-	F	F	96	87	71	F	F	99	95
f_{13}	87	93	97	F	-	F	99	94	90	96	99	F	96
f_{14}	71	96	97	F	F	-	F	99	94	F	F	F	96
f_{15}	60	76	75	96	F	F	-	F	F	82	97	F	90
f_{16}	41	69	54	90	96	F	F	-	F	62	93	F	82
f_{17}	28	44	47	84	93	97	F	F	-	59	88	99	76
f_{20}	94	96	F	F	97	96	85	60	57	-	F	91	89
f_{21}	85	99	F	F	F	F	97	93	79	F	-	99	96
f_{22}	59	84	85	99	F	F	F	F	F	96	99	-	93
Avg	74	87	86	97	97	96	89	82	74	90	96	92	88

derive illumination-insensitive signatures from transformed images. In essence, we force a rank constraint on the unknown shape by writing the transformed image in terms of the basis surface normals as follows

$$\boldsymbol{H}_{i,j} = \max\left(\boldsymbol{n}_{i,j} \cdot \boldsymbol{s}^{(0)}, 0\right) = \sum_{k=1}^{K} \boldsymbol{a}_k \max\left(\boldsymbol{n}_{i,j}^k \cdot \boldsymbol{s}^{(0)}, 0\right) \tag{3.36}$$

where $\boldsymbol{n}^k$ is the k^{th} basis surface normal and $\boldsymbol{a}_k$ is the corresponding combining coefficient. The coefficient vector $\boldsymbol{a}$ being independent of $\boldsymbol{s}^{(0)}$, can be used to perform recognition across illumination variations.

Table 3.3 shows the recognition results obtained on the PIE dataset using the coefficient vectors $\boldsymbol{a}$ obtained from the corresponding transformed images. The basis normal vectors are derived from the 3D facial data [20] and the coefficient vectors are computed using a closed-form linear least square approach. The overall average recognition rate achieved in this experiment is 88%. This signifies the efficacy of the proposed approach in generating robust transformed images.

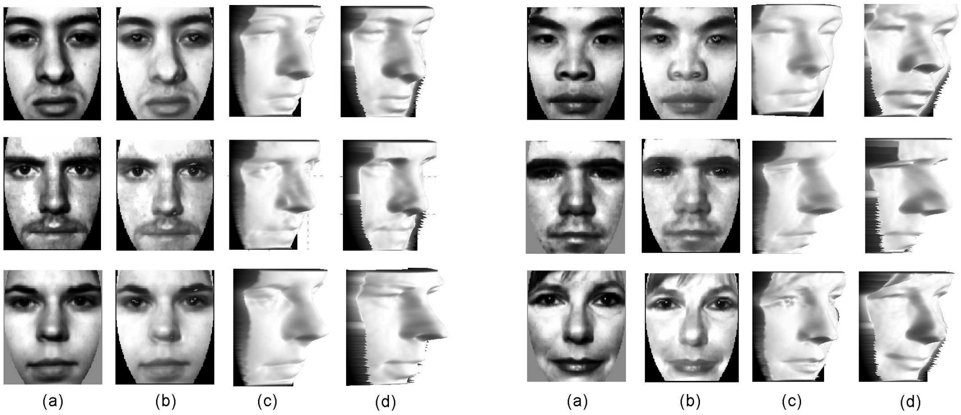


Figure 3.17: Comparison with the ground truth. (a) Input image (FRGC dataset), (b) Estimated albedo, (c) Recovered shape, (d) True shape.

We now demonstrate the usefulness of the transformed images for the task of shape recovery. Figure 3.17 shows a comparison of the recovered shapes with the 3D shapes of the corresponding subjects from the FRGC dataset [95]. The proposed approach seems to recover various person-specific facial features around lips, eyes, etc. Note that the 3D shapes available in the database are captured on a different day than the input intensity images, leading to slightly different facial expressions in the estimated and true shapes.

We perform another experiment to quantitatively evaluate the shape estimates obtained using the transformed images. We compare the shape estimates with the ones obtained using the approach in [130] that directly uses intensity images as input. One thousand synthetically generated images (using Vetter's 3D face data) are used to determine the angular error in the estimated normal maps for comparison. We observe an improvement of over 15% using our approach. We also test the efficacy of the approach on images downloaded from the web with little control over the illumination and other imaging conditions. Figure 3.18 shows the albedo and shape estimates obtained along with a few novel views synthesized under novel illumination conditions.

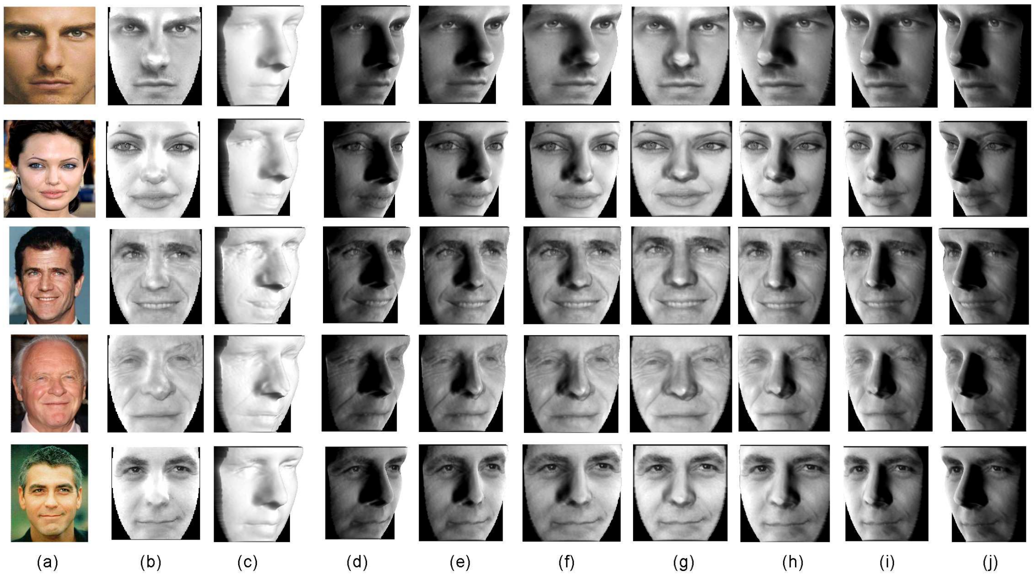


Figure 3.18: Novel view synthesis in the presence of novel illumination conditions. (a) Input image, (b) Estimated albedo, (c) Recovered shape, (d)-(j) Synthesized views under novel illumination conditions.

3.4.6 Application to general objects

All the experiments are conducted on faces but the approach is applicable to any domain in general where the required error statistics are available. In the absence of ensemble information, the required means and variances can possibly be approximated by local spatial statistics. Under such an approximation, the LMMSE albedo estimate is given by [58]

$$\begin{aligned} \hat{\boldsymbol{\rho}}_{i,j} &= \bar{\boldsymbol{\rho}}_{i,j} + \boldsymbol{\alpha}_{i,j}\big(\boldsymbol{\rho}^{(0)}_{i,j} - \bar{\boldsymbol{\rho}}^{(0)}_{i,j}\big) \\ \text{where,} \qquad \boldsymbol{\alpha}_{i,j} &= \frac{v^2_{i,j}(\boldsymbol{\rho})}{v^2_{i,j}(\boldsymbol{\rho}) + \sigma^2_{i,j}(\boldsymbol{w})} \end{aligned} \tag{3.37}$$

where $\bar{\boldsymbol{\rho}}_{i,j}$ and $\bar{\boldsymbol{\rho}}^{(0)}_{i,j}$ are the local spatial means of $\boldsymbol{\rho}_{i,j}$ and $\boldsymbol{\rho}^{(0)}_{i,j}$ respectively, and $v^2_{i,j}(\boldsymbol{\rho})$ is the local spatial variance of $\boldsymbol{\rho}_{i,j}$. The local statistics of $\boldsymbol{\rho}_{i,j}$ can be calculated from that of the degraded signal using the following relations

$$\bar{\boldsymbol{\rho}}_{i,j} = \bar{\boldsymbol{\rho}}_{i,j}^{(0)} \tag{3.38}$$

$$v_{i,j}^2(\boldsymbol{\rho}) = \frac{v_{i,j}^2(\boldsymbol{\rho}^{(\mathbf{0})}) - \left(\bar{\boldsymbol{\rho}}_{i,j}^{(0)}\right)^2 A_{i,j}}{1 + A_{i,j}}, \qquad \text{where} \qquad A_{i,j} = \frac{\sigma_{i,j}^2(\boldsymbol{n}) + \sigma^2(\boldsymbol{s})}{\left(\boldsymbol{n}_{i,j}^{(0)} \cdot \boldsymbol{s}^{(0)}\right)^2} \tag{3.39}$$

Though not too accurate, these approximations are probably the best one can do in the absence of any ensemble information. Figure 3.19 (middle row) shows the albedo estimates obtained this way for a few images (top row) from the Amsterdam Library of Object Images [43]. As desired, the illumination effects present in the input images are less visible in the albedo maps. This is further highlighted in the shadow maps shown in Figure 3.19 (bottom row). In these examples, the surface normals required to obtain the initial albedo map are assigned manually. For example, the cylindrical shape is used for the mug and the coke can while the cuboidal shape is used for the boxes shown in the figure. Figure 3.20 shows some more examples of the albedo estimates obtained along with a few zoomed-in regions to signify the usefulness of the approach for general objects.

3.5 Summary

We proposed an image estimation formulation for the task of albedo estimation from a single image. Errors in illumination and surface normal information lead to erroneous albedo maps. The proposed estimation framework effectively utilizes the statistics of error in illumination and normal information for robust estimation of albedo. Extensive experiments are performed to show the usefulness of the estimated albedo maps as illumination-insensitive signatures. The albedo maps are also used to obtain *albedo-free* images for shape recovery.

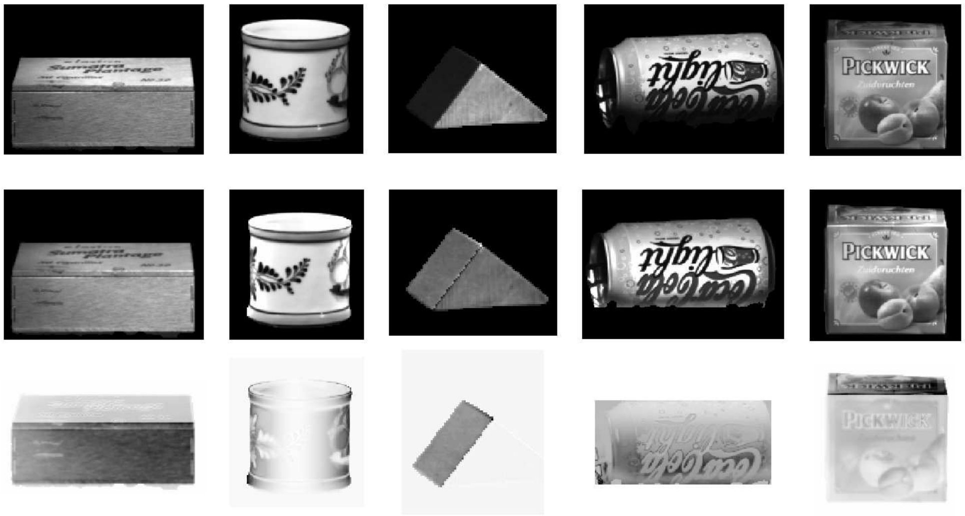


Figure 3.19: Top: Images from the Amsterdam Library of Object Images [43]; Middle: Estimated albedo; Bottom: Shadow maps (albedo-free images of the corresponding input images).

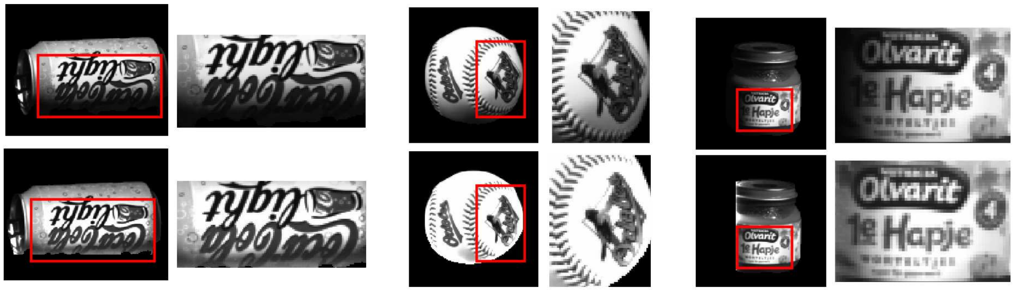


Figure 3.20: Top row: Original images of a few objects; Bottom row: Estimated albedo maps obtained using local statistics. The zoomed in regions are shown to highlight the difference between the input images and the corresponding albedo estimates.

Chapter 4

Pose-Robust Albedo Estimation from a Single Image

In Chapter 3, we proposed a model based approach based on the image formation model for robust estimation of albedo from a single face image. The approach uses a stochastic filtering framework for handling the errors due to inaccuracies in the surface normals and light source direction to estimate albedo across wide range of challenging illumination conditions. One limitation of the approach is that it requires accurate knowledge of the pose of the face that may not typically be available. To be able to recognize faces in real and unconstrained scenarios which is the ultimate goal, it may not be realistic to assume either frontal pose or an accurate knowledge of the pose since facial pose estimation is by itself a challenging research problem [87].

In this chapter, we build upon the formulation in the previous chapter to account for inaccurate pose information in addition to inaccuracies in light source and surface normal information. The proposed approach is an image estimation framework that utilizes class-specific statistics of the imaged object to iteratively improve pose and albedo estimates. In each iteration, given the current albedo estimate, 3D facial pose is estimated by solving a linear Least-Squares (LS) problem which is used to further improve the albedo estimate, and so on. The input to the algorithm is a face image in which face and eyes are automatically located using OpenCV's Haar-based detectors.

Extensive experiments are performed to evaluate the usefulness of the proposed approach. Experimental results on synthetic data in varying poses are provided to show the accuracy of the albedo and 3D pose estimates for different unknown poses. To show the usefulness of the estimated albedo maps as illumination insensitive measures, the estimated albedo maps are used for the task of face recognition. We also provide comparison with ground truth for the estimated 3D facial poses. Experiments on unconstrained real face images from the net further highlight the usefulness of the approach.

The rest of the chapter is organized as follows. The proposed albedo and pose estimation framework is described in Section 4.1. The details of the proposed algorithm are given in Section 4.2. The results of experimental evaluation are presented in Section 4.3. The chapter concludes with a summary and discussion.

Notation: Throughout the chapter, $\boldsymbol{\rho}$, $\boldsymbol{n}$, $\boldsymbol{s}$, Θ denote the true unknown albedo, surface normals and illuminant direction and pose of the object while $\bar{\boldsymbol{\rho}}$, $\bar{\boldsymbol{n}}$, $\bar{\boldsymbol{s}}$, $\bar{\Theta}$ represent the initial estimates of the corresponding variables.

4.1 Albedo Estimation from a Single Image

For the class of Lambertian objects, the diffused component of the surface reflection is modeled using the *Lambert's Cosine Law*

$$I = \rho \max(\boldsymbol{n}^T \boldsymbol{s}, 0) \tag{4.1}$$

where I is the pixel intensity, $\boldsymbol{s}$ is the light source direction, ρ is the surface albedo and $\boldsymbol{n}$ is the surface normal of the corresponding point. The max function in the relation accounts for the formation of attached shadows.

Let $\bar{\boldsymbol{n}}_{i,j}$ and $\bar{\boldsymbol{s}}$ be some initial estimate of the surface normal and illuminant direction respectively. Let $\bar{\Theta}$ represents initial knowledge of the pose. The Lambertian assumption imposes the following constraint on the initial albedo $\bar{\boldsymbol{\rho}}$ obtained at pixel (i, j)

$$\bar{\boldsymbol{\rho}}_{i,j} = \frac{\boldsymbol{I}_{i,j}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \tag{4.2}$$

where $\cdot$ is the standard dot product operator. In most real applications, the input is only a single intensity image and so we do not have accurate estimates of the (a) pose, (b) surface normals and the (c) light source direction. Inaccuracies in these initial estimates lead to considerable errors in the initial albedo estimate (Figure 4.1).

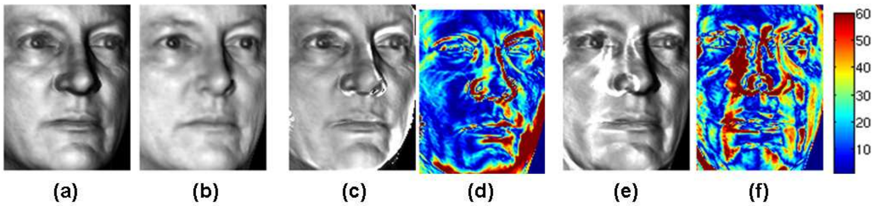


Figure 4.1: Illustration of errors in albedo due to errors in surface normals, illuminant direction and pose. (a) Input Image; (b) True albedo; (c) Albedo estimate using average facial surface normal, estimated illuminant direction and true pose; (d) Error map for (c); (e) Albedo estimate using true values of surface normal and illuminant direction and assuming frontal pose; (f) Error map for (e) due to inaccuracies in pose information.

As shown in the figure, even if surface normal and illuminant directions are accurately known, error in pose information can result in unacceptable errors in the albedo map. In Chapter 3, an image estimation formulation was proposed to account for the inaccuracies in the surface normals and the light source direction, but knowledge of the pose was assumed to be known *a priori*. In this work, we extend the framework to address the more general scenario where the pose is unknown. As a byproduct of the formulation, we also get an estimate of the 3D pose which is itself a challenging problem and an active area of research [87].

4.1.1 Image Estimation Formulation

Here we formulate the image estimation framework to obtain a robust albedo estimate using the initial albedo map which is erroneous due to inaccuracies in pose, surface normal and light source estimates. The expression in (4.2) can be rewritten as follows

$$\bar{\boldsymbol{\rho}}_{i,j} = \frac{\boldsymbol{I}_{i,j}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} = \boldsymbol{\rho}_{i,j} \frac{\boldsymbol{n}^{\Theta}_{i,j} \cdot \boldsymbol{s}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \tag{4.3}$$

where $\boldsymbol{\rho}$, $\boldsymbol{n}$ and $\boldsymbol{s}$ are the true unknown albedo, normal and illuminant direction respectively and Θ denotes the true unknown pose. $\bar{\boldsymbol{\rho}}_{i,j}$ can further be expressed as follows

$$\bar{\boldsymbol{\rho}}_{i,j} = \boldsymbol{\rho}_{i,j} \frac{\bar{\boldsymbol{n}}^{\Theta}_{i,j} \cdot \bar{\boldsymbol{s}}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} + \frac{\boldsymbol{n}^{\Theta}_{i,j} \cdot \boldsymbol{s} - \bar{\boldsymbol{n}}^{\Theta}_{i,j} \cdot \bar{\boldsymbol{s}}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \boldsymbol{\rho}_{i,j} \tag{4.4}$$

We substitute

$$\boldsymbol{w}_{i,j} = \frac{\boldsymbol{n}^{\Theta}_{i,j} \cdot \boldsymbol{s} - \bar{\boldsymbol{n}}^{\Theta}_{i,j} \cdot \bar{\boldsymbol{s}}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \boldsymbol{\rho}_{i,j} \quad , \qquad \boldsymbol{h}_{i,j} = \frac{\bar{\boldsymbol{n}}^{\Theta}_{i,j} \cdot \bar{\boldsymbol{s}}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \tag{4.5}$$

So equation (4.4) simplifies to

$$\bar{\boldsymbol{\rho}}_{i,j} = \boldsymbol{\rho}_{i,j} \boldsymbol{h}_{i,j} + \boldsymbol{w}_{i,j} \tag{4.6}$$

This can be identified with the standard image estimation formulation [7]. Here $\boldsymbol{\rho}$ is the original signal (true albedo), the rough albedo estimate $\bar{\boldsymbol{\rho}}$ is the degraded signal and $\boldsymbol{w}$ is the signal dependent additive noise. When the head pose is known accurately, i.e. if $\bar{\Theta} = \Theta$, $\boldsymbol{h}_{i,j} = 1$. So this is a generalization of the formulation proposed in Chapter 3 for the case of unknown head pose.

4.2 Albedo Estimate

Several methods have been proposed in literature to solve image estimation equations of the form (4.6). Here we compute the Linear Minimum Mean Squared Error (LMMSE) albedo estimate which is given by [105]

$$\boldsymbol{\rho}^{\mathrm{est}} = E(\boldsymbol{\rho}) + \mathrm{C}_{\rho\bar{\rho}} \mathrm{C}_{\bar{\rho}}^{-1} (\bar{\boldsymbol{\rho}} - E(\bar{\boldsymbol{\rho}})) \tag{4.7}$$

Here $\mathrm{C}_{\rho\bar{\rho}}$ is the cross-covariance matrix of $\boldsymbol{\rho}$ and $\bar{\boldsymbol{\rho}}$. $E(\bar{\boldsymbol{\rho}})$ and $\mathrm{C}_{\bar{\rho}}$ are the ensemble mean and covariance matrix of $\bar{\boldsymbol{\rho}}$ respectively. The LMMSE filter requires the second order statistics of the signal and noise.

From (4.5), the expression for the signal-dependent noise $\boldsymbol{w}_{i,j}$ can be rewritten as follows

$$\boldsymbol{w}_{i,j} = \frac{(\boldsymbol{n}^{\Theta}_{i,j} - \bar{\boldsymbol{n}}^{\Theta}_{i,j}) \cdot \boldsymbol{s} + \bar{\boldsymbol{n}}^{\Theta}_{i,j} \cdot (\boldsymbol{s} - \bar{\boldsymbol{s}})}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \boldsymbol{\rho}_{i,j} \tag{4.8}$$

Assuming the errors in illumination and surface normals to be unbiased, the noise $\boldsymbol{w}$ is zero-mean. Under this assumption, the expressions for $\mathrm{C}_{\rho\bar{\rho}}$ and $\mathrm{C}_{\bar{\rho}}$ simplify (details in the Appendix) to

$$\mathrm{C}_{\rho\bar{\rho}} = \mathrm{C}_{\rho} H^T \;\text{ and }\; \mathrm{C}_{\bar{\rho}} = H \mathrm{C}_{\rho} H^T + \mathrm{C}_w \tag{4.9}$$

where H is the matrix containing $\boldsymbol{h}$'s for the entire image as its diagonal entries and C_w is the covariance of the noise term.

Here, we assume a Non-stationary Mean Non-stationary Variance (NMNV) model for the original signal, which has been shown to be a reasonable assumption for many applications [58]. Under this model, the original signal is characterized by a non-stationary mean and a diagonal covariance matrix with non-stationary variance. Under the NMNV assumption, the LMMSE filtered output (4.7) simplifies (details in Appendix) to the following scalar (point) processor of the form

$$\begin{aligned} \boldsymbol{\rho}^{\text{est}}_{i,j} &= E(\boldsymbol{\rho}_{i,j}) + \boldsymbol{\alpha}_{i,j}\big(\bar{\boldsymbol{\rho}}_{i,j} - E(\bar{\boldsymbol{\rho}}_{i,j})\big) \\ \text{where,} \qquad \boldsymbol{\alpha}_{i,j} &= \frac{\sigma^2_{i,j}(\boldsymbol{\rho})\boldsymbol{h}_{i,j}}{\sigma^2_{i,j}(\boldsymbol{\rho})\boldsymbol{h}^2_{i,j} + \sigma^2_{i,j}(\boldsymbol{w})} \end{aligned} \tag{4.10}$$

where $\sigma^2_{i,j}(\boldsymbol{\rho})$ and $\sigma^2_{i,j}(\boldsymbol{w})$ are the non-stationary signal and noise variances respectively. Since noise $\boldsymbol{w}$ is zero-mean, $E(\bar{\boldsymbol{\rho}}_{i,j}) = \boldsymbol{h}_{i,j} E(\boldsymbol{\rho}_{i,j})$. Therefore, (4.10) can be written as

$$\boldsymbol{\rho}^{\text{est}}_{i,j} = (1 - \boldsymbol{h}_{i,j}\boldsymbol{\alpha}_{i,j}) E\big(\boldsymbol{\rho}_{i,j}\big) + \boldsymbol{\alpha}_{i,j}\bar{\boldsymbol{\rho}}_{i,j} \tag{4.11}$$

So the LMMSE albedo estimate is the weighted sum of the ensemble mean $E(\boldsymbol{\rho})$ and the observation $\bar{\boldsymbol{\rho}}$, where the weight depends on the ratio of signal variance to the noise variance. Now we derive the different entities in the expression for the albedo estimate.

4.2.1 Expression for the Noise Variance

From (4.8), assuming the errors in surface normal $(\boldsymbol{n}_{i,j} - \bar{\boldsymbol{n}}_{i,j})$ to be uncorrelated in the x, y and z directions and their variances are same, the expression for the noise variance can be shown to be (details in Appendix)

$$\sigma_{i,j}^2(\boldsymbol{w}) = \frac{\sigma_{i,j}^2(\boldsymbol{n}) + \sigma^2(\boldsymbol{s})}{\left(\bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}} \cdot \bar{\boldsymbol{s}}\right)^2} E\left(\boldsymbol{\rho}_{i,j}^2\right) \tag{4.12}$$

Here $\sigma_{i,j}^2(\boldsymbol{n})$ and $\sigma^2(\boldsymbol{s})$ are the error variances in each direction of the surface normal and light source direction respectively.

4.2.2 Expression for $\boldsymbol{h}_{i,j}$

The expression for $\boldsymbol{h}_{i,j}$ is given by

$$\boldsymbol{h}_{i,j} = \frac{\bar{\boldsymbol{n}}_{i,j}^{\Theta} \cdot \bar{\boldsymbol{s}}}{\bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}} \cdot \bar{\boldsymbol{s}}} = 1 + \frac{(\bar{\boldsymbol{n}}_{i,j}^{\Theta} - \bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}}) \cdot \bar{\boldsymbol{s}}}{\bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}} \cdot \bar{\boldsymbol{s}}} \tag{4.13}$$

The term $\boldsymbol{h}_{i,j}$ depends on the difference of the surface normal corresponding to the pixel location (i, j) between the initial pose information and the true pose. The term is present due to the fact that an incorrect pose $\bar{\Theta}$ is used to compute the initial albedo that is different from the true unknown pose Θ.

Let Figure 4.2 (a) represents the initial pose $\bar{\Theta}$ and Figure 4.2 (b) represents the true pose Θ. Let us consider the surface points corresponding to the same pixel location (i, j) for the two poses. Let P_1 be the surface point of the face in the initial pose which corresponds to pixel (i, j) (which is P_1' in the true pose) and P_2' be the surface point in the true pose for the same pixel (i, j) (which is P_2 in the initial pose). P_1 and P_2' which correspond to the same pixel location are physically different surface points since the initial pose is different from the true pose. Let us assume that the initial pose and the true pose are related by $(\boldsymbol{\Omega}, \boldsymbol{T})$. Here $\boldsymbol{\Omega} = [\Omega_x, \Omega_y, \Omega_z]$, denotes the rotation about the centroid of the face and $\boldsymbol{T} = [T_x, T_y, T_z]$ denotes the translation of the centroid.

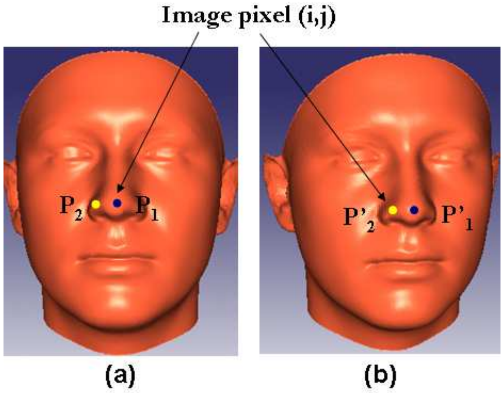


Figure 4.2: Illustration to explain the relation between surface normals of two different surface points corresponding to the same pixel location. (a) Initial pose; (b) True pose. Here P_1 and P_2' correspond to the same pixel location (i, j), though they are physically different points.

In this case, the difference between the normals can be expressed as

$$\Delta \boldsymbol{n} = \boldsymbol{n}_{P_2'} - \boldsymbol{n}_{P_1} = \boldsymbol{J}_{P_1} \boldsymbol{\Delta} + \Delta \boldsymbol{n}_{P_2, P_2'} \tag{4.14}$$

Here, $\boldsymbol{\Delta} = P_2 - P_1$ is the difference in the co-ordinates of P_2 and P_1 and $\boldsymbol{J}_{P_1}$ is the Jacobian matrix of the surface normal $\boldsymbol{n}_{P_1}$ at surface point P_1. The term $\Delta \boldsymbol{n}_{P_2, P_2'}$ denotes the difference in surface normals between $\boldsymbol{n}_{P_2}$ and $\boldsymbol{n}_{P_2'}$. The first term encodes that P_2 is a different surface point from P_1 and the second term takes care of the fact that the surface normal $\boldsymbol{n}_{P_2'}$ is a rotated version of the surface normal $\boldsymbol{n}_{P_2}$.

In [136], Xu and Roy-Chowdhury use a similar equation to relate different frames of a video sequence when the object under consideration was undergoing rotation and translation. They showed that under small motion assumption, the difference in normal can be expressed as a linear function of the object motion variables, i.e., the equation (4.14) can be expressed as

$$\Delta \boldsymbol{n}_{i,j} = \boldsymbol{A}_{i,j} \boldsymbol{\Omega} + \boldsymbol{B}_{i,j} \boldsymbol{T} \tag{4.15}$$

where the variables $\boldsymbol{A}$ and $\boldsymbol{B}$ can be computed from the average surface normal at the initial pose. The exact expressions for these variables are given in the Appendix. Figure 4.3 illustrates how well the linear expression for $\Delta \boldsymbol{n}_{i,j}$ approximates the true difference $\bar{\boldsymbol{n}}^{\Theta}_{i,j} - \bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j}$ for average 3D face model. The figure shows the average angular errors due to the linear approximation of $\Delta \boldsymbol{n}_{i,j}$ for different values of pitch and yaw. We see that for small rotations, the error is quite small which means that the approximation is quite good. Using (4.15), the expression for $\boldsymbol{h}_{i,j}$ can be written in terms of rotation and translation $(\boldsymbol{\Omega}, \boldsymbol{T})$ as

$$\boldsymbol{h}_{i,j} = 1 + \frac{(\boldsymbol{A}_{i,j} \boldsymbol{\Omega} + \boldsymbol{B}_{i,j} \boldsymbol{T}) \cdot \bar{\boldsymbol{s}}}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \tag{4.16}$$

4.2.3 Algorithm for Albedo and Pose Estimation

In this section, we describe the proposed algorithm for estimating the unknown albedo map and the pose using the described formulation. From (4.10) and (4.12), we can express the LMMSE albedo estimate as a function of pose and class-based statistics as follows

$$\boldsymbol{\rho}^{\text{est}} = f(S, \Theta) \tag{4.17}$$

where S represents the various statistics like $E\big(\boldsymbol{\rho}_{i,j}\big)$, $\sigma^2_{i,j}(\boldsymbol{\rho})$ and $\sigma^2_{i,j}(\boldsymbol{w})$ and Θ represents the pose which is given by rotation $\boldsymbol{\Omega}$ and translation $\boldsymbol{T}$. The statistics

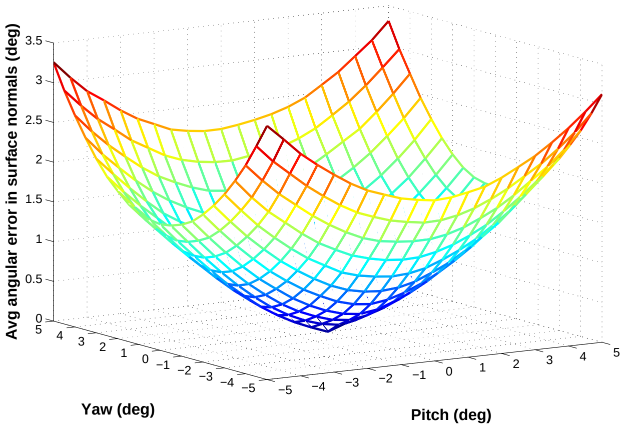


Figure 4.3: Average angular errors in surface normals for average 3D face model due to the linear approximation of $\bar{\boldsymbol{n}}_{i,j}^{\Theta} - \bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}}$ (4.15).

implicitly depends on the facial pose. If the pose is known, the LMMSE albedo estimate can be computed using the above relation and visa versa. Based on this, we propose an iterative algorithm to alternately estimate albedo and pose.

The input to the algorithm is a single intensity image and some initial estimates of surface normals and pose. In all our experiments, we use an average 3D face model as the initial estimate of the surface normals. Initial pose is assumed to be frontal in all our experiments. Given the image, OpenCV Haar-based detectors are used to obtain face and eye locations that serve to provide initial localization of the face region. Using the average shape and initial pose information, we obtain an initial estimate of illuminant direction as follows [22]

$$\bar{\boldsymbol{s}} = \left(\sum_{i,j} \bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}} {\bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}}}^{T} \right)^{-1} \sum_{i,j} \boldsymbol{I}_{i,j} \bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}} \tag{4.18}$$

where $\bar{\boldsymbol{n}}_{i,j}^{\bar{\Theta}}$ is the average facial surface normal at initial pose $\bar{\Theta}$. The required class statistics S is computed based on the initial pose information using Vetter's 3D face data [20]. The rest of the algorithm proceeds as follows

4.2 Albedo Estimate

1. Using the current estimate of the pose, the LMMSE albedo estimate $\boldsymbol{\rho}^{\text{est}}$ is computed using (4.11).

2. If the current pose estimate is very different from the true unknown pose, the current albedo estimate can be quite erroneous. So we perform a regularization step where the current albedo estimate is projected onto a statistical albedo model to ensure that the resulting albedo map lies within the space of allowable facial albedo maps. In our implementation, we use standard Principal Component Analysis (PCA)-based linear statistical model to preform this regularization. Let the regularized albedo map be denoted by $\boldsymbol{\rho}^{\text{reg}}$.

   To avoid computation of the statistical model for every intermediate pose, we bring the albedo map to the frontal pose before regularization. The albedo map at the frontal pose $\boldsymbol{\rho}^{\text{frontal}}$ is related to the albedo map at the current pose $\boldsymbol{\rho}^{\text{est}}$ as follows

   $$\boldsymbol{\rho}_{i,j}^{\text{frontal}} = \boldsymbol{\rho}_{i,j}^{\text{est}} - \Delta\boldsymbol{\rho}_{i,j} \tag{4.19}$$

   From Figure 4.2, the albedo changes from P_1 to P_2, but is the same for P_2 and P_2'. Therefore, $\Delta\boldsymbol{\rho} = \boldsymbol{\rho}_{P_2'} - \boldsymbol{\rho}_{P_1} = \Delta\rho_{P_1}\boldsymbol{\Delta}$ where $\Delta\rho_{P_1}$ is the gradient of $\boldsymbol{\rho}$ at point P_1. $\boldsymbol{\Delta\rho}$ can further be approximated as [136]

   $$\Delta\boldsymbol{\rho}_{i,j} = \boldsymbol{C}_{i,j}\boldsymbol{\Omega} + \boldsymbol{D}_{i,j}\boldsymbol{T} \tag{4.20}$$

   where the variables $\boldsymbol{C}$ and $\boldsymbol{D}$ are computed from the class statistics (details are in Appendix).

3. The regularized albedo map is further used to compute a revised estimate of the pose. From (4.17), we can express the pose in terms of the albedo

estimate as follows

$$\begin{pmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Omega} \\ \boldsymbol{T} \end{pmatrix} = (\bar{\boldsymbol{\rho}} - \boldsymbol{\rho}^{\mathrm{reg}}) \bar{\boldsymbol{n}}^{\bar{\Theta}} \cdot \bar{\boldsymbol{s}} \tag{4.21}$$

where $\boldsymbol{X}_1 = \boldsymbol{\rho}^{reg} \bar{\boldsymbol{s}}^T \boldsymbol{A} - \boldsymbol{C}$ and $\boldsymbol{X}_2 = \boldsymbol{\rho}^{reg} \bar{\boldsymbol{s}}^T \boldsymbol{B} - \boldsymbol{D}$. The subscript i, j has been omitted from (4.21) for clarity. (4.21) is used to obtain the new pose estimate using the LS method.

4. if the albedo and pose estimates between two successive iterations are below a pre-specified threshold, terminate the algorithm and output the current albedo and pose. Otherwise, using the updated pose and illuminant estimates, repeat the iteration.

As we have seen from Figure 4.3, the linear approximation for $\Delta \boldsymbol{n}_{i,j}$ in (4.15) works well for small difference between the initial pose and the true pose which imposes a limit on the pose difference which the above algorithm can handle. Experimentally, we have seen that the above algorithm can handle rotation about $5-6$ degrees. To generalize the method to larger pose difference, we *de-rotate* and *de-translate* the input image by the estimated rotation and translation after every iteration. Then we use the new *de-rotated* and *de-translated* image as input to the next iteration. Figure 4.4 shows the different steps of the proposed algorithm. As shown, we obtain pose and albedo (in frontal pose) estimates as the output of the algorithm. The number of iterations required depends on the pose but we observed that typically it takes around 5-6 iterations for a pose error of around 20°. Our MATLAB implementation of the algorithm converges in around 1.5 minutes on a Pentium M 1.60 GHz laptop out of which over 92% of the time is used in warping the class statistics (which can be made a lot faster using a GPU-based parallel implementaion).

Figure 4.5 shows the albedo map and 3D pose obtained using the proposed algorithm for a face image generated using 3D facial data [20]. The *derotated*

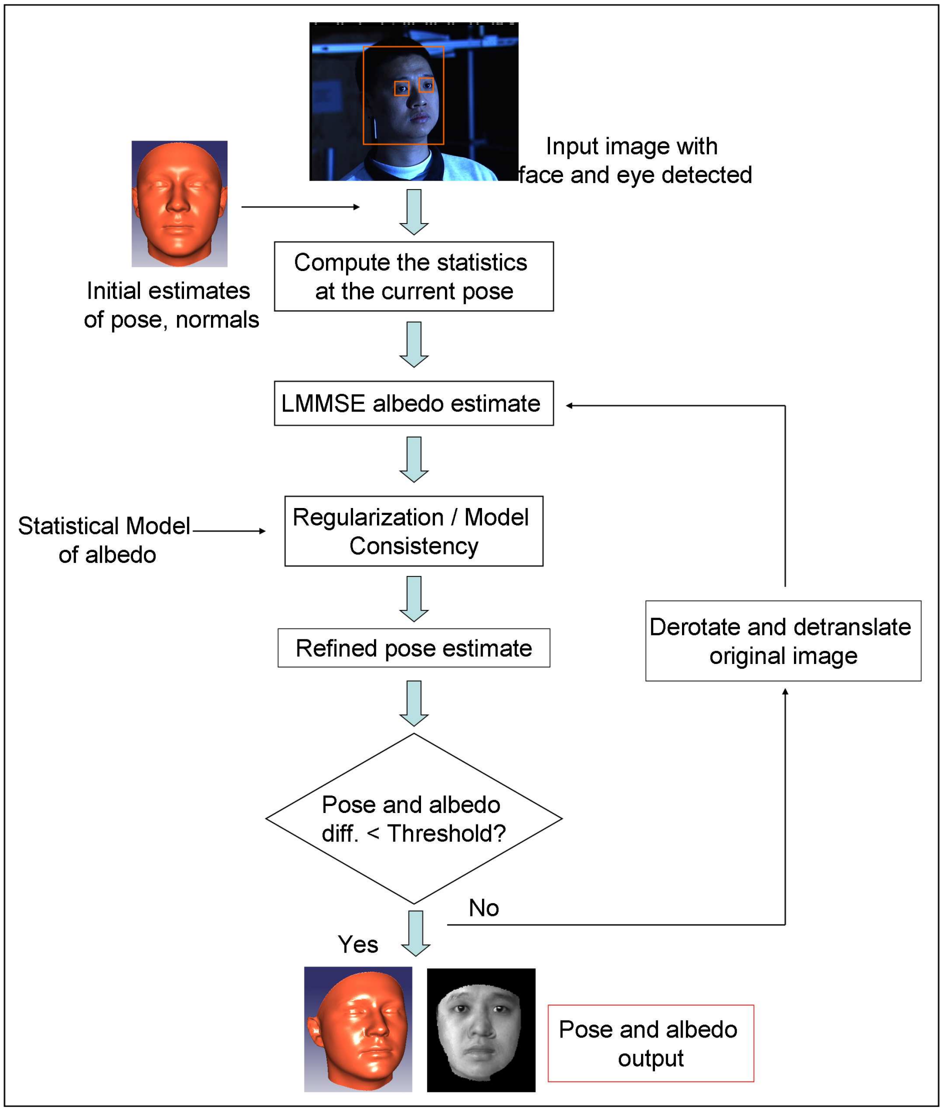


Figure 4.4: Flowchart illustrating the different steps of the proposed algorithm.

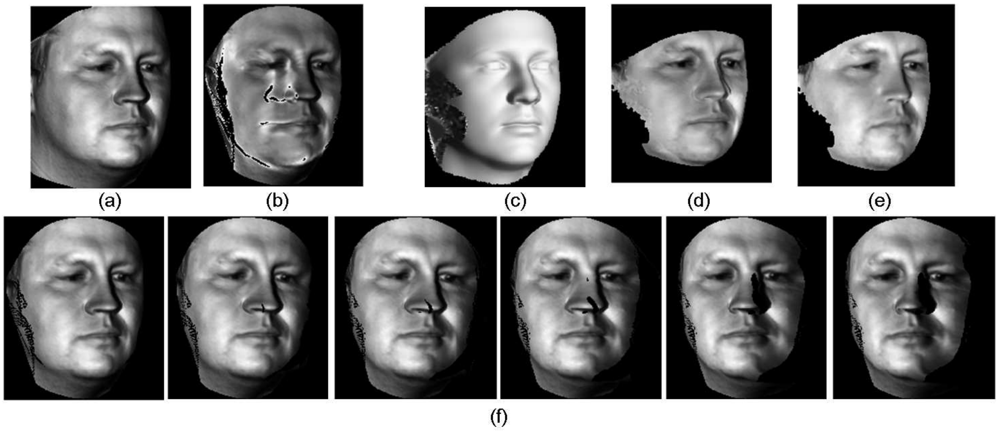


Figure 4.5: (a) Input image; (b) Initial rough albedo estimate using frontal pose; (c) Estimated 3D pose; (d) Estimated albedo map; (e) True albedo; (f) The *derotated* images after every iteration.

images after every iteration are shown in the second row. The albedo estimate at the true pose is obtained using the pose estimate and the estimated albedo at the frontal pose.

To further illustrate the working of the algorithm, we present the error surface along with the path traversed by the proposed iterative algorithm (Figure 4.6). The error surface is generated by computing average per-pixel albedo error for albedo estimates obtained for different pose hypotheses. The error is minimum at the true pose of 20 degrees yaw. The algorithm starts with the assumption of frontal pose and converges to a pose close to the true pose in 5 iterations (red line in the plot).

Discussion: We now analyze the reason for the proposed algorithm to work reliably for pose errors over 30° even though the linear approximation for $\Delta\boldsymbol{n}_{i,j}$ in (4.15) seems to be accurate only for much smaller angles. Note that the error plot in Figure 4.3 shows the errors averaged over the entire face but we observe that most of these errors come from the nose region. The linear approximation is fairly accurate for angles as large as 30° for face points that are not close to the

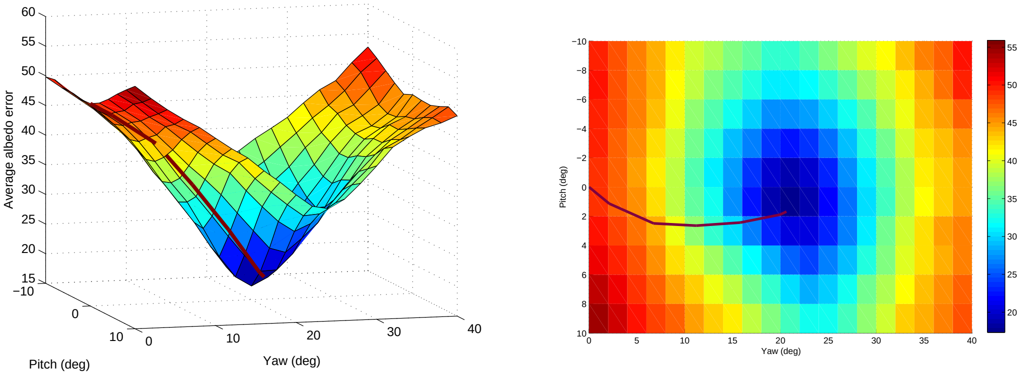


Figure 4.6: Visualization of the error surface for a synthetic image. (Left) Average per-pixel error in the albedo map for different poses. The path taken by our algorithm is shown in red. (Right) Top view of the error surface.

nose making the proposed algorithm capable to deal with such large pose errors.

4.3 Experimental Evaluation

4.3.1 Experiment on synthetically generated data

Table 4.1: Average accuracy in the pose estimates (in deg) for synthetic data under different illumination conditions and poses. The results are averaged over 1000 images. The initial pose is always taken to be frontal.

	Mean and std	5°	10°	15°	20°	25°	30°
Yaw	Mean	5.9	10.3	15.2	20.1	24.3	28.8
	Std	1.05	1.3	1.3	1.6	1.5	1.6
Pitch	Mean	5.4	10.3	14.9	20.1	24.9	29.2
	Std	1.3	1.5	1.9	1.6	1.6	2.1
Roll	Mean	4.7	9.7	14.5	19.5	24.1	28.6
	Std	1.3	1.5	1.6	1.5	1.5	1.4

For comparison with the ground truth, we first evaluate the proposed approach for images synthetically generated form 3D facial data [20]. Table 4.1 and Table 4.2 show the average accuracy in the pose and albedo estimates obtained for 1000 images generated under different illumination conditions and poses. For all the images, the initial pose was assumed to be frontal, so the table shows the results of the algorithm for increasing errors in the initial pose. The albedo estimates obtained are significantly more accurate (around 40%) compared to the initial noisy maps obtained assuming frontal pose.

Table 4.2: Average accuracy in the albedo estimates for the experiment described in Table 4.1. The entries in the table represent the average per-pixel errors in albedo estimates.

	5°	10°	15°	20°	25°	30°
Yaw	14.8	14.9	14.4	14.9	14.9	15.1
Pitch	14.3	14.4	15.2	15.4	15.9	16.1
Roll	14.7	14.8	14.8	15.2	15.3	15.9

4.3.2 Recognition across illumination and pose

Figure 4.7 shows the estimated frontal albedo maps for several images under different illumination conditions and poses for one subject from the PIE dataset [115]. As desired, the albedo maps look quite similar to each other with much of the illumination and viewpoint differences removed. We further use the estimated albedo maps as illumination and pose insensitive signatures in a face recognition experiment on the PIE dataset that contains face images of 68 subjects taken under several different illumination conditions and pose. The estimated albedo

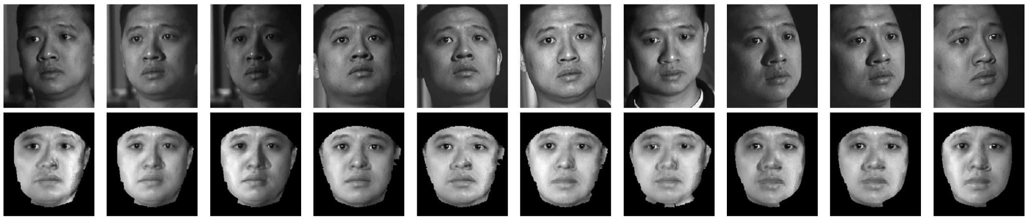


Figure 4.7: Albedo estimates obtained for several images of the same subject from the PIE dataset [115].

maps are then projected onto a albedo PCA space (generated from FRGC training data) to compute similarity across gallery and probe images. Here the gallery images are in frontal pose and frontal illumination f_{12} and the probe images are in side pose and 21 different illumination conditions. In this experiment, each gallery and probe set contains just one image per subject. Table 4.3 shows the rank-1 recognition results obtained. We see that the proposed algorithm compares favorably with the state-of-the-art [103] [139].

Table 4.3: Recognition results on the PIE dataset [115]. The recognition rates of [103] [139] are included for comparison.

	Illumination source from PIE																					
	$\mathbf{f_{02}}$	$\mathbf{f_{03}}$	$\mathbf{f_{04}}$	$\mathbf{f_{05}}$	$\mathbf{f_{06}}$	$\mathbf{f_{07}}$	$\mathbf{f_{08}}$	$\mathbf{f_{09}}$	$\mathbf{f_{10}}$	$\mathbf{f_{11}}$	$\mathbf{f_{12}}$	$\mathbf{f_{13}}$	$\mathbf{f_{14}}$	$\mathbf{f_{15}}$	$\mathbf{f_{16}}$	$\mathbf{f_{17}}$	$\mathbf{f_{18}}$	$\mathbf{f_{19}}$	$\mathbf{f_{20}}$	$\mathbf{f_{21}}$	$\mathbf{f_{22}}$	**Avg**
[103]	60	78	83	91	89	92	94	97	89	97	98	97	98	97	94	89	85	86	97	98	97	**90.8**
[139]	81	88	91	89	92	95	93	96	97	98	99	93	94	93	91	92	88	90	94	96	95	**92.6**
Our	68	84	91	96	97	97	97	97	99	97	99	97	97	97	93	90	96	97	97	96	97	**94.2**

4.3.3 Head pose estimation and comparison with ground truth

Figure 4.8 shows the results of head pose estimation using the proposed algorithm on a set of images from the BU data [60]. The sequence has 200 frames out of which we considered every alternate frame. For every frame, we started with the frontal pose as the initial pose. The first row in Figure 4.8 shows some of the frames from the sequence and the second row shows the comparison of the pose estimates obtained against the ground truth provided with the dataset. As can be seen, the proposed estimates are quite close to the ground truth.

We also use the proposed algorithm to estimate albedo and pose on images

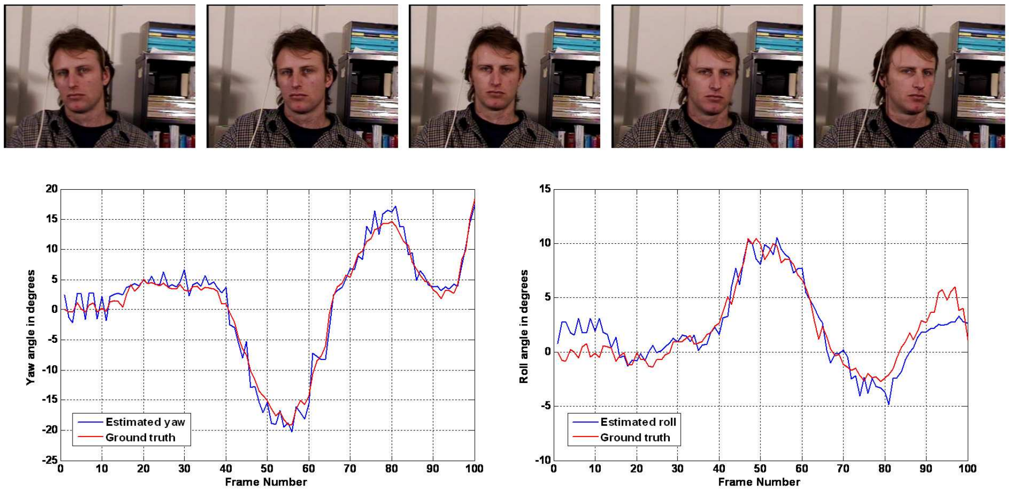


Figure 4.8: Comparison of the pose estimation results on the BU dataset [60] with the provided ground truth.

downloaded from the web with little control over the imaging conditions. Figure 4.9 shows the albedo and pose estimates obtained.

4.4 Summary and Discussion

In this chapter, we have proposed an approach for simultaneous estimation of albedo and 3D head pose from a single image. In all our experiments, we used OpenCV's Haar-based detectors to automatically detect faces and eyes for initial localization. Compared to most state-of-the-art approaches [104], the proposed approach does not require manually marked landmarks and is completely automatic. In addition, the method does not impose any linear statistical constraint on the unknown albedo and the statistical albedo model is used only for regularization. Currently, we do not estimate 3D shape of the input face image that will be part of our future research. The proposed algorithm works well for a wide range of poses (around 30° on either side for a total range of around 60°). Starting with a different canonical pose, the method can easily be extended for more extreme

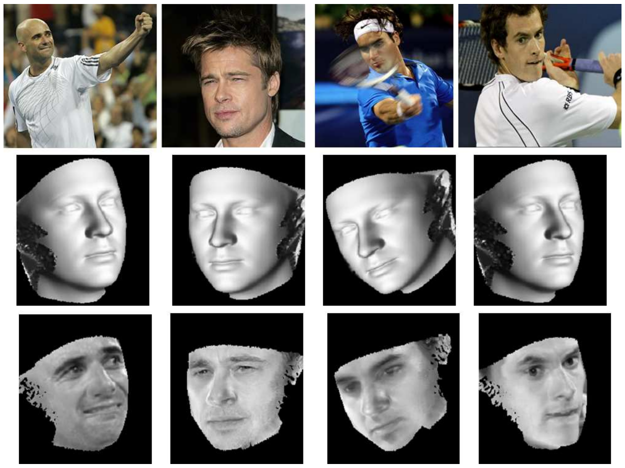


Figure 4.9: Row 1: A few images downloaded form the web; Row 2: Estimated 3D head pose; Row 3: Estimated albedo map.

poses. Multiple illumination sources can easily be incorporated in the proposed formulation as in the previous chapter.

Appendix

Expression for the various terms: Assuming P_1 is the 3D face point corresponding to the pixel i, j in the initial pose, the expressions for $\boldsymbol{A}$ and $\boldsymbol{B}$ in (4.15) are given by

$$\boldsymbol{A} = \boldsymbol{J}_{P_1} \boldsymbol{M} \hat{\boldsymbol{P}}_{\boldsymbol{1}} - \hat{\boldsymbol{n}}_{P_1}; \qquad \boldsymbol{B} = -\boldsymbol{J}_{P_1} \boldsymbol{M} \tag{4.22}$$

The subscript i, j has been omitted for clarity. Here,

$$\boldsymbol{M} = \boldsymbol{I} - \frac{1}{\boldsymbol{n}_{P_1}^T \boldsymbol{u}} \boldsymbol{u} \boldsymbol{n}_{P_1}^T$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{u}$ is the unit vector in the direction joining the optical center of the camera to the surface point P_1 corresponding to the pixel

(i, j). The skew symmetric matrix of a vector

$$\boldsymbol{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}; \qquad \text{is} \qquad \hat{\boldsymbol{X}} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}$$

The expressions for $\boldsymbol{C}$ and $\boldsymbol{D}$ in (4.20) are given by

$$\boldsymbol{C} = \Delta\rho_{P_1}\boldsymbol{M}\hat{\boldsymbol{P}}_{\boldsymbol{1}}; \qquad \boldsymbol{D} = \Delta\rho_{P_1}\boldsymbol{M}$$

The subscript i, j has been omitted for clarity. For derivations of these expressions, readers are referred to [136].

Derivation of LMMSE albedo estimate: Here we compute the Linear Minimum Mean Squared Error (LMMSE) albedo estimate which is given by [105]

$$\boldsymbol{\rho}^{\text{est}} = E(\boldsymbol{\rho}) + \mathrm{C}_{\rho\bar{\rho}}\mathrm{C}_{\bar{\rho}}^{-1}(\bar{\boldsymbol{\rho}} - E(\bar{\boldsymbol{\rho}})) \tag{4.23}$$

Here $\mathrm{C}_{\rho\bar{\rho}}$ is the cross-covariance matrix of $\boldsymbol{\rho}$ and $\bar{\boldsymbol{\rho}}$. $E(\bar{\boldsymbol{\rho}})$ and $\mathrm{C}_{\bar{\rho}}$ are the ensemble mean and covariance matrix of $\bar{\boldsymbol{\rho}}$ respectively. The LMMSE filter requires the second order statistics of the signal and noise.

The expression for the signal-dependent noise $\boldsymbol{w}_{i,j}$ can be rewritten as follows

$$\boldsymbol{w}_{i,j} = \frac{(\boldsymbol{n}^{\Theta}_{i,j} - \bar{\boldsymbol{n}}^{\Theta}_{i,j}) \cdot \boldsymbol{s} + \bar{\boldsymbol{n}}^{\Theta}_{i,j} \cdot (\boldsymbol{s} - \bar{\boldsymbol{s}})}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}}\boldsymbol{\rho}_{i,j} \tag{4.24}$$

Assuming the initial values of surface normal and light source direction to be unbiased, both $E(\boldsymbol{w})$ and $E(\boldsymbol{w}|\boldsymbol{\rho})$ are zero. Since noise $\boldsymbol{w}$ is zero-mean, $E(\bar{\boldsymbol{\rho}}_{i,j}) = \boldsymbol{h}_{i,j}E(\boldsymbol{\rho}_{i,j})$.

So, $\mathrm{C}_{\rho\bar{\rho}}$ can be written as

$$\begin{aligned} \mathrm{C}_{\rho\bar{\rho}} &= E[(\boldsymbol{\rho} - E(\boldsymbol{\rho}))(\bar{\boldsymbol{\rho}} - E(\bar{\boldsymbol{\rho}}))^T] \\ &= \mathrm{C}_\rho H^T + E[(\boldsymbol{\rho} - E(\boldsymbol{\rho}))\boldsymbol{w}^T] \end{aligned} \tag{4.25}$$

Similarly, if C_w is the covariance of the noise term, $\mathrm{C}_{\bar{\rho}}$ can be written as

$$\begin{aligned} \mathrm{C}_{\bar{\rho}} &= E[(\bar{\boldsymbol{\rho}} - E(\bar{\boldsymbol{\rho}}))(\bar{\boldsymbol{\rho}} - E(\bar{\boldsymbol{\rho}}))^T] \\ &= H\mathrm{C}_\rho H^T + \mathrm{C}_w + HE[(\boldsymbol{\rho} - E(\boldsymbol{\rho}))\boldsymbol{w}^T] \\ &+ E[\boldsymbol{w}(\boldsymbol{\rho} - E(\boldsymbol{\rho}))^T]H^T \end{aligned} \tag{4.26}$$

Recalling that $E(\boldsymbol{w})$ and $E(\boldsymbol{w}|\boldsymbol{\rho})$ are zero, $E\big((\boldsymbol{\rho} - E(\boldsymbol{\rho}))\boldsymbol{w}^T\big) = 0$. This simplifies (4.25) and (4.26) as follows

$$\mathrm{C}_{\rho\bar{\rho}} = \mathrm{C}_\rho H^T \text{ and } \mathrm{C}_{\bar{\rho}} = H\mathrm{C}_\rho H^T + \mathrm{C}_w \tag{4.27}$$

where H is the matrix containing $\boldsymbol{h}_{i,j}$'s for the entire image as its diagonal entries and C_w is the covariance of the noise term.

Here, we assume a Non-stationary Mean Non-stationary Variance (NMNV) model for the original signal. Under the NMNV assumption, C_ρ is a (non-constant) diagonal matrix. As C_ρ and H are both diagonal, $\mathrm{C}_{\rho\bar{\rho}}$ is diagonal. Since C_ρ is diagonal, C_w and thus $\mathrm{C}_{\bar{\rho}}$ are also diagonal. Therefore, the LMMSE filtered output (4.23) simplifies to the following scalar (point) processor of the form

$$\begin{aligned} \boldsymbol{\rho}^{\text{est}}_{i,j} &= E(\boldsymbol{\rho}_{i,j}) + \boldsymbol{\alpha}_{i,j}\big(\bar{\boldsymbol{\rho}}_{i,j} - E(\bar{\boldsymbol{\rho}}_{i,j})\big) \\ \text{where,} \qquad \boldsymbol{\alpha}_{i,j} &= \frac{\sigma^2_{i,j}(\boldsymbol{\rho})h_{i,j}}{\sigma^2_{i,j}(\boldsymbol{\rho})h^2_{i,j} + \sigma^2_{i,j}(\boldsymbol{w})} \end{aligned} \tag{4.28}$$

where $\sigma^2_{i,j}(\boldsymbol{\rho})$ and $\sigma^2_{i,j}(\boldsymbol{w})$ are the non-stationary signal and noise variances respectively. Since $E(\bar{\boldsymbol{\rho}}_{i,j}) = h_{i,j}E(\boldsymbol{\rho}_{i,j})$, (4.28) can be written as

$$\boldsymbol{\rho}^{\text{est}}_{i,j} = (1 - \boldsymbol{h}_{i,j}\boldsymbol{\alpha}_{i,j})E\big(\boldsymbol{\rho}_{i,j}\big) + \boldsymbol{\alpha}_{i,j}\bar{\boldsymbol{\rho}}_{i,j} \tag{4.29}$$

So the LMMSE albedo estimate is the weighted sum of the ensemble mean $E(\boldsymbol{\rho})$ and the observation $\bar{\boldsymbol{\rho}}$, where the weight depends on the ratio of signal variance to the noise variance.

Noise Variance: The expression for the signal-dependent noise $\boldsymbol{w}_{i,j}$ is given by

$$\boldsymbol{w}_{i,j} = \frac{(\boldsymbol{n}^{\Theta}_{i,j} - \bar{\boldsymbol{n}}^{\Theta}_{i,j}) \cdot \boldsymbol{s} + \bar{\boldsymbol{n}}^{\Theta}_{i,j} \cdot (\boldsymbol{s} - \bar{\boldsymbol{s}})}{\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}} \boldsymbol{\rho}_{i,j} \tag{4.30}$$

Assuming the errors in surface normal $(\boldsymbol{n}_{i,j} - \bar{\boldsymbol{n}}_{i,j})$ and the light source direction $(\boldsymbol{s} - \bar{\boldsymbol{s}})$ to be uncorrelated in the x, y and z directions and their variances are same, the expression for the noise variance can be shown to be

$$\sigma^2_{i,j}(\boldsymbol{w}) = \sigma^2_{i,j}(\boldsymbol{w}_1) + \sigma^2_{i,j}(\boldsymbol{w}_2) \tag{4.31}$$

where,

$$\sigma^2_{i,j}(\boldsymbol{w}_1) = \sigma^2_{i,j}(\boldsymbol{n}) \left(\frac{{s_x}^2 + {s_y}^2 + {s_z}^2}{\left(\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}\right)^2} \right) E\big(\boldsymbol{\rho}^2_{i,j}\big) \tag{4.32}$$

and

$$\sigma^2_{i,j}(\boldsymbol{w}_2) = \sigma^2(\boldsymbol{s}) \left(\frac{\left(\bar{n}^{\Theta}_x\right)^2 + \left(\bar{n}^{\Theta}_y\right)^2 + \left(\bar{n}^{\Theta}_z\right)^2}{\left(\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}\right)^2} \right) E\big(\boldsymbol{\rho}^2_{i,j}\big) \tag{4.33}$$

Here $\sigma^2_{i,j}(\boldsymbol{n})$ and $\sigma^2(\boldsymbol{s})$ are the error variances in each direction of the surface normal and light source direction respectively. $\{s_x, s_y, s_z\}$ and $\{\bar{n}^{\Theta}_x, \bar{n}^{\Theta}_y, \bar{n}^{\Theta}_z\}$ are the three components of the illuminant direction and initial surface normal respectively. $[s_x, s_y, s_z]$ and $[\bar{n}^{\Theta}_x, \bar{n}^{\Theta}_y, \bar{n}^{\Theta}_z]$ being unit vectors, the expression for the noise variance can further be simplified as follows

$$\sigma^2_{i,j}(\boldsymbol{w}) = \frac{\sigma^2_{i,j}(\boldsymbol{n}) + \sigma^2(\boldsymbol{s})}{\left(\bar{\boldsymbol{n}}^{\bar{\Theta}}_{i,j} \cdot \bar{\boldsymbol{s}}\right)^2} E\big(\boldsymbol{\rho}^2_{i,j}\big) \tag{4.34}$$

Chapter 5

Face Recognition across Aging

Face recognition is one of the most successful applications of decades of research on image analysis and understanding [144]. Research in this area has traditionally focused on analyzing and modeling changes in facial appearance due to variations in illumination conditions, facial pose, expressions, etc. Other than these commonly occurring variations, aging is another phenomenon that affects facial appearance significantly. Though effects of aging on facial appearance have been studied for a long time, it is only recently that efforts have been made to recognize faces across age progression. Automatic matching of faces as people age is particularly useful for tasks like passport/visa renewal where authorities need to verify if the old and new photographs belong to the same person. Unlike other variates like illumination conditions and viewpoint, there is no simple geometric/statistical model to analyze appearance changes due to aging. Changes in facial appearance due to aging typically depend on quite a few factors like race, geographical location, eating habits, stress level, etc., that makes the problem of matching faces across aging extremely difficult.

Most existing works [59,23,135,126,63,64,100,108,117,40,101] on facial aging focus primarily on modeling and simulating aging effects on human face and report impressive simulation results. Given the infinite different ways in which a person can age depending on his/her surroundings, habits, etc., it is difficult to predict how a person will appear at a different age. Also, simulating face images at target age assumes that both the base and target age are known or can be estimated which by itself is a difficult problem. But in spite of this large variability, humans are quite good at matching faces across age progression. This may mean that irrespective of the exact manner in which a person ages, there is a certain pattern in the way facial appearance changes with age.

Here, we propose an alternate approach for matching age-separated faces by analyzing whether the changes in shape and appearance can be attributed to aging effects. Facial aging effects manifest in the form of wrinkles, skin texture variation and facial shape change in addition to other intangibles. The relative effects of these factors typically depend on the age being considered [80]. Shape variations are more pronounced in children, while most facial aging effects in adults can be attributed to 1) wrinkles, 2) skin texture variations, and 3) drift of facial features due to subtle shape variation or sagging of underlying muscles.

First we show that if the two images belong to the same subject, the drift in features seems to follow a coherent pattern which is usually not the case if the images belong to different subjects. Unlike feature drift, textural variations cannot be modeled directly from face images due to differences in illumination conditions across images. To capture textural variations, we use ratio of albedo maps of the input images. Given two images of the same person, such a ratio captures the change in appearance due to wrinkles and other skin texture variations which follows a coherent pattern. The ratio is characterized using local histograms of LBP based features [92]. Using this LBP based histogram representation, we use the following two approaches to perform verification: 1) train a Support Vector Machine (SVM) to separate genuine and impostor image pairs, and 2) kernel density estimation approach [31] to learn the probability density function (pdf) of the aging functions. Illustrations and experimental evaluation show the efficacy of such the proposed features for matching faces across age progression.

The rest of chapter is organized as follows: The following section discusses a few related works from the literature. Section 5.2 gives the motivation for the work using drifts in facial features. Details of textural analysis are described in Section 5.3. Results of experiments performed to evaluate the efficacy of the approach are shown in Section 5.4. The chapter concludes with a discussion.

5.1 Related work

Facial aging has been an area of interest for decades [96, 97, 81, 80], but it is only recently that efforts have been made to address problems like age estimation, age transformation, etc. from a computational point of view [59, 23, 135, 126, 63, 64, 100, 108, 117, 40, 101]. Burt and Perrett [23] investigate visual cues to age using facial composites that blend facial shape and color from multiple faces. Kwon and Lobo [59] classify input images as babies, young adults and senior adults based on cranio-facial development and skin wrinkle analysis. Wu *et al.* [135] describe a skin deformation model to simulate face wrinkles using an elastic process assembled with visco and plastic units. Tiddeman *et al.* [126] present a wavelet-based method for prototyping and transforming facial features to increase the perceived age of the input images. Lanitis *et al.* [64] use PCA-based transformation models to explain the effects of aging on facial appearance. The proposed statistical model is used for tasks like 1) age estimation from new face images, 2) simulating aging effects, and 3) face recognition across age variations. A similar PCA-based statistical face transformation model is used in [63] to obtain a compact parametric representation of an input face image for the task of automatic age estimation. Different classifiers are designed and compared that predict age given the parametric description of the input image.

In [100], Ramanathan and Chellappa study the effect of age progression on facial similarity between a pair of images of the same individual. A Bayesian age difference classifier is proposed to classify images based on age differences and perform face verification across age progression. In [101], they propose a craniofacial growth model to characterize growth related facial shape variations in children. This model makes use of anthropometric evidences to predict appearance across years and to perform face recognition using the synthesized images. Geng *et al.* [40] propose a subspace based approach for automatic age estimation. Given a previously unseen image, its aging pattern is determined by projecting it onto a subspace obtained using training data consisting of several time-separated images of individuals. Suo *et al.* [117] simulate aging effects using a dynamic Markov process on a multi-layer AND-OR graph integrating the effects of global appearance changes in hair style and shape, deformation and aging effects of different facial components, and wrinkle appearance. In [108], Scandrett *et al.* propose linear and piecewise models that rely on average developmental trends, to predict aging effects on human faces. In a recent paper, Ling *et al.* [75] use gradient orientation pyramid in a Support Vector Machine (SVM) based framework to verify images across age progression.

Most existing works address the problem of face verification across age progression from a simulation point of view (other than [75]). Given the difficulty of simulating effects of different factors that can affect the way a person ages, an alternative is to analyze if the difference in two input images can be attributed to aging. In contrast, we propose an approach to perform face verification across age progression based on the coherency of textural variation in the input images.

5.2 Motivation from Feature drifts

Aging brings about a wide variety of changes in the appearance of human faces. While for children, these changes are mainly manifested in the shape, textural variations are more prominent in adults faces [80]. Though the overall shape does not change significantly for adults, there is subtle drift in facial features due to various factors like muscle sagging, weight gain or loss, etc. The features do not drift independent of each other. Depending on the underlying shape and muscle structure of the individual, there is some coherency among these drifts. Figure 5.1 illustrates such coherency in feature drifts for a few images from the FGnet aging database [1]. The drift maps shown in Figure 5.1 are obtained directly using the manually marked feature points available in the dataset. For images of different subjects, due to different shape and muscle structure, the feature drifts computed this way may not be coherent (Figure 5.1). Now, we try to capture this coherency in the facial feature drifts.

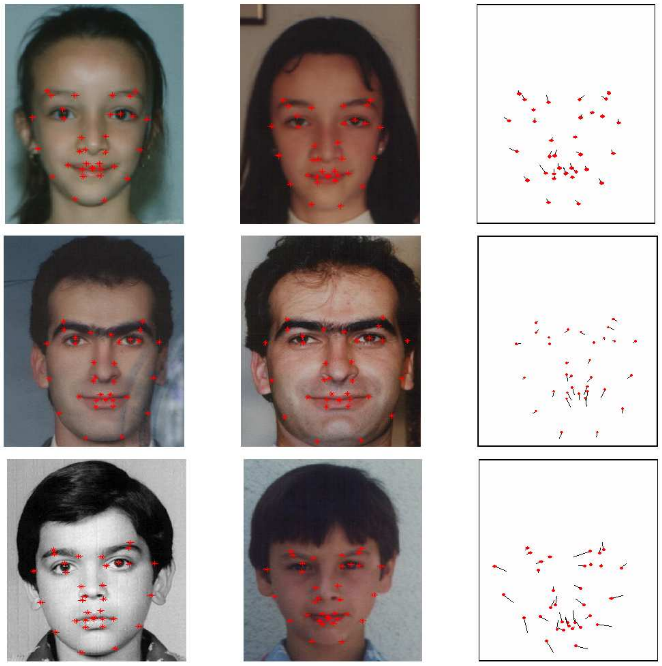


Figure 5.1: Drifts in facial features for a few age-separated face images from the FGnet aging database. The drifts across images of same individuals appear coherent (top two rows) while they are somewhat incoherent (third row) when the images belong to different individuals. (Best viewed in color)

We define a measure of incoherency between two feature drifts as

$$U_{\mathrm{ij}} = \frac{\parallel \boldsymbol{a}_i - \boldsymbol{a}_j \parallel}{r_{ij}} \tag{5.1}$$

Here $\parallel \boldsymbol{a}_i - \boldsymbol{a}_j \parallel$ is the magnitude of the vector difference between the two feature drifts $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ while r_{ij} is the distance between the corresponding feature locations. The drift incoherency is inversely proportional to the distance between the two feature locations in consideration. This follows nicely from the fact that neighboring drifts resulting from the sagging of the same underlying muscle will be coherent. In addition, this allows for far-off regions of the face with different underlying muscles to drift relatively independent of each other, without adding to the incoherency of the drift map. The combined potential energy of the drift map characterized by K feature drifts is given by

$$C_{\text{local}} = \sum_{i=1}^{K} \sum_{j=i+1}^{K} U_{ij} \tag{5.2}$$

5.3 Textural variations with aging

In this section, we focus on the changes in facial appearance due to textural variations. The surface normals, albedo and the intensity image are related by an image formation model. We assume Lambertian reflectance model for the facial surface. For such objects, the diffused component of the surface reflection is modeled using the *Lambert's Cosine Law* given by

$$\boldsymbol{I}_{i,j} = \boldsymbol{\rho}_{i,j} \max(\boldsymbol{n}_{i,j}^T \boldsymbol{s}, 0) \tag{5.3}$$

where $\boldsymbol{I}_{i,j}$ is the pixel intensity, $\boldsymbol{s}$ is the light source direction, $\boldsymbol{\rho}_{i,j}$ is the surface albedo and $\boldsymbol{n}_{i,j}$ is the surface normal of the corresponding surface point. Albedo is an illumination-insensitive characteristic of a surface that represents the fraction of light that a surface point reflects when it is illuminated. The max function accounts for the formation of attached shadows. Taking logarithm on both sides and rearranging the terms, we obtain

$$\log(\boldsymbol{I}_{i,j}) = \log(\boldsymbol{\rho}_{i,j}) + \log(\boldsymbol{n}_{i,j}^T \boldsymbol{s}) \tag{5.4}$$

Here, we omit the explicit max operator by considering only non-zero pixels in the input image. Due to the difference in illumination conditions across images, the variation in texture cannot be analyzed using the input images directly. Instead, one needs to model skin texture variation across aging using albedo maps derived from the input images. If the surface normal and illuminant direction are known, one can obtain the albedo map easily from (5.4).

Similar to the feature drift analysis, given albedo maps corresponding to the two input images, we need a way to identify if the change can be ascribed to aging effects or not. Let us assume that albedo maps of an individual at different ages are related as follows

$$\boldsymbol{\rho}(t_2) = \boldsymbol{f}(\boldsymbol{\rho}(t_1)) \tag{5.5}$$

where $\boldsymbol{f}$ denotes the aging function that maps albedo at base age t_1 to the target age t_2. In reality, such an aging function can be very complicated depending on innumerable factors like individual living conditions, stress-level, habits, etc. that affect how skin texture varies with age. Learning such a function in its true generality is not easy in practice. Here we model the variations in albedo due to aging using a linear function as follows

$$\boldsymbol{\rho}_{i,j}(t_2) = \boldsymbol{f}_{i,j}(t_2, t_1) * \boldsymbol{\rho}_{i,j}(t_1) \tag{5.6}$$

In this form, the aging function depends only on the age difference between the two images. Such a model has also been suggested by Ling *et al.* [75] and Hussein [52]. Considering the problem of face matching across age separation, if the two images are of the same individual, but possibly different illumination conditions, then we have

$$\begin{aligned} \Delta\log(\boldsymbol{\rho}_{i,j}) &= \log(\boldsymbol{\rho}_{i,j}(t_2)) - \log(\boldsymbol{\rho}_{i,j}(t_1)) \\ &= \log(\boldsymbol{f}_{i,j}(t_2, t_1)) \end{aligned} \tag{5.7}$$

In other words, the difference between these logarithm-transformed albedo maps depends only on the aging function if the two albedo maps belong to the same individual. When input images are from different individuals, the term $\Delta\log(\boldsymbol{\rho})$ depends not only on the aging function, but also on the identity of the individuals.

Using this formulation, if we can learn a generic aging function, given a pair of images, we can verify if the corresponding $\Delta\log(\boldsymbol{\rho})$ term can be attributed to textural aging or not.

Most real age-separated images contain variabilities due to differences in pose, expression, image noise, etc. in addition to aging effects. Therefore, we first illustrate the intuition behind the proposed formulation using a few synthetically aged/de-aged images downloaded from the web. Figure 5.2 shows the original images, albedo maps and the corresponding $\Delta\log(\boldsymbol{\rho})$ terms for two such pairs of age-separated images. In most practical scenarios, accurate estimates of shape and illuminant direction are not available, making the albedo map computed using (5.4) erroneous. Here, we use the non-stationary stochastic filtering framework proposed in Chapter 3 to obtain a more accurate albedo map from the input image. The approach utilizes the statistics of errors in erroneous shape and illuminant information to refine the initial albedo map. As desired in our application, the estimated albedo maps and the corresponding textural aging map are able to capture the aging effects like appearance of wrinkles, skin folds, etc.

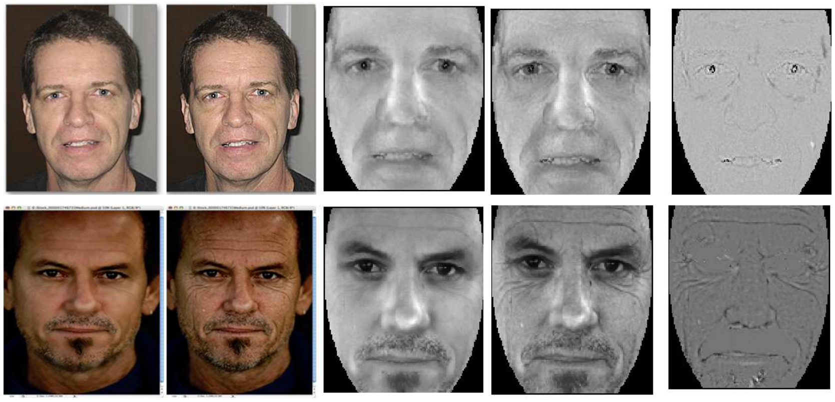


Figure 5.2: Textural variations: (a) Young image; (b) Old image; (c) Albedo of image in (a); (d) Albedo of image in (b); (e) $\Delta\log(\boldsymbol{\rho})$.

Figure 5.3 shows the albedo maps and the corresponding aging maps for a couple of age-separated image pairs from the FGnet dataset. One can observe that these maps preserve the textural variations due to aging. Due to other variations and subtle mis-alignments, wrinkles, skin folds, etc. are not as prominent in these maps as they are in Figure 5.2 with synthetic examples.

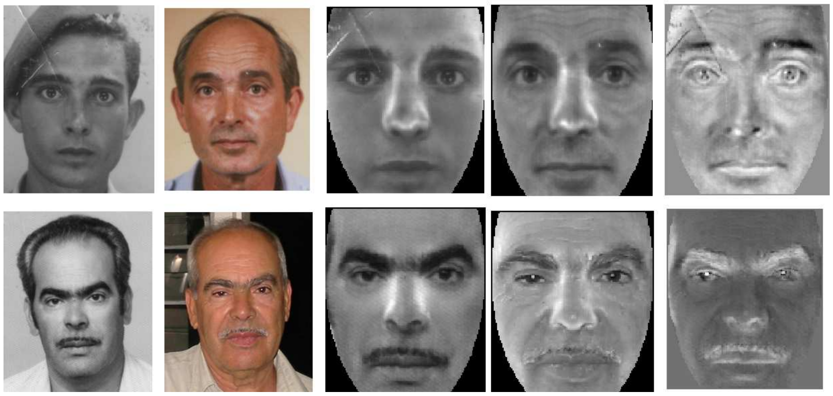


Figure 5.3: Textural variations: (a) Young image; (b) Old image; (c) Albedo of image in (a); (d) Albedo of image in (b); (e) $\Delta\log(\boldsymbol{\rho})$.

5.3.1 Computation of textural coherency

Given $\Delta\log(\boldsymbol{\rho})$ obtained from the input images, one needs to verify if this can be attributed to aging. It is worthwhile to note that quite a few changes in facial skin due to aging, like appearance of wrinkles follow a coherent pattern determined by the underlying muscle structure. For example, contraction of the frontalis muscle that runs vertically on the forehead causes the formation of horizontal wrinkles on the forehead. If the input images belong to the same subject, $\Delta\log(\boldsymbol{\rho})$ being an aging function should have this coherency. In this work, we characterize this coherency using local binary pattern (LBP) based texture features [92, 5].

The LBP operator is known to be a powerful descriptor for textures. Given an input image, the operator assigns a label to each pixel based on its intensity value relative to its neighborhood. The label is the binary string obtained by thresholding the neighborhood of each pixel using the center value. The oper-

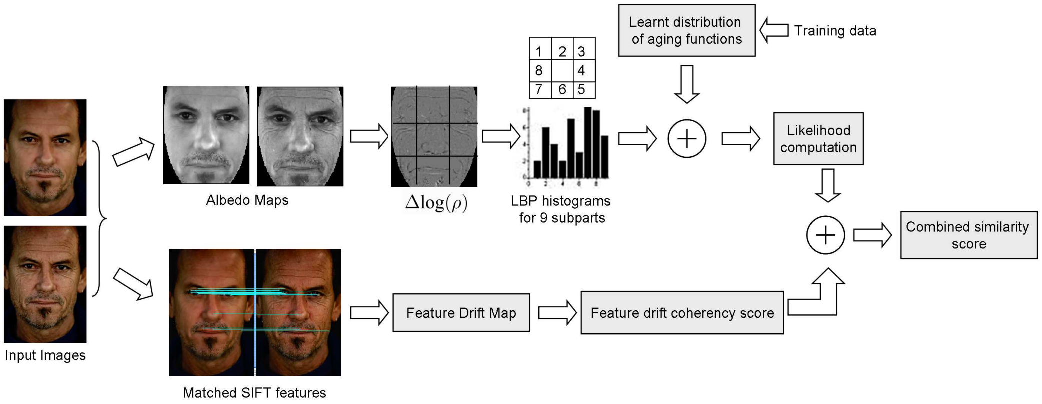


Figure 5.4: A schematic of the proposed approach for face verification across age progression.

ator is invariant to monotonic gray level changes and has proved to be highly discriminative [92, 5]. Given $\Delta\log(\boldsymbol{\rho})$, we first compute 8 bit binary label for each pixel using a 3x3 neighborhood. We further compute region-wise histograms of LBP labels by dividing the $\Delta\log(\boldsymbol{\rho})$ map in 9 regions arranged in a regular square grid. Such region-wise histogram approach has the capability to characterize local variations without being overly sensitive to alignment errors.

Due to the vast variations in aging patterns, it may not be possible to use such representation directly to verify if the two input images are age separated images of the same person or not. So using this LBP based histogram representation, we use the following two approaches to perform verification: 1) train a Support Vector Machine (SVM) to separate genuine and impostor image pairs, and 2) kernel density estimation approach [31] to learn the probability density function (pdf) of the aging functions.

5.4 Experimental evaluation

Figure 5.4 shows a schematic of the proposed non-generative approach for face verification across age progression. In this section, we evaluate our approach for matching age-separated facial images and analyze the results obtained. Here, we focus primarily on aging effects in adults. For this purpose, we use a part of a private Passport dataset which consists of age-separated pairs of adult face images. In our experiments, we use 700 genuine pairs of age separated images with average age separation of slightly over 9 years. Figure 5.5 provides the distribution of age separation for these image pairs. 4200 randomly chosen impostor pairs are used in the verification experiments. For the proposed textural coherency, we perform two-fold cross-validation on the available match and non-match image pairs. Average performance of the two folds is considered as the verification performance. As the proposed feature drift coherency does not involve any training, we use the entire set of match and non-match pairs for testing.

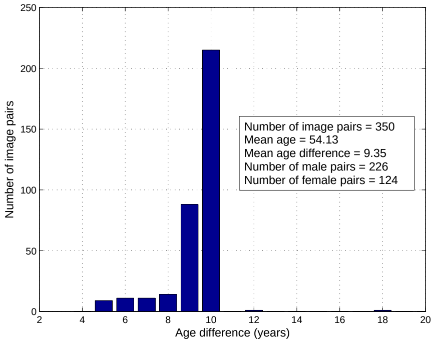


Figure 5.5: Distribution of age difference between the pairs of images used in the experimental evaluation.

Feature Drift Coherency: To compute feature drift coherency, we need to locate the facial features reliably to obtain the drift maps. In our implementation, we use Active Appearance Models (AAM) [27] to detect the facial landmarks.

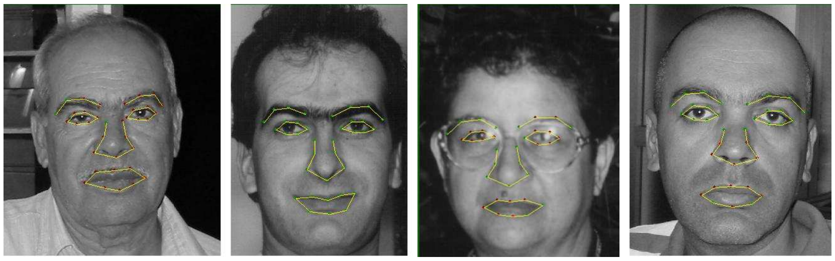


Figure 5.6: Facial landmarks as located automatically using the trained AAM-based face model. The detected landmarks are used to compute feature-drift incoherency.

First a face model is trained using 20 images with manually marked landmark points which is then used to locate features for the remaining images used in the verification experiment. Figure 5.6 shows the facial feature locations detected on a few examples images using the trained face model. Given the facial landmarks, the overall incoherency of the drifts is measured using (5.2). Figure 5.7 shows the Receiver Operator Characteristic (ROC) curve obtained using this incoherency measure as the distance between the input pair of images. The plot compares correct rejection rate against correct acceptance rate. The correct rejection rate is the fraction of correctly rejected impostor pairs while the correct acceptance rate is the fraction of correctly accepted genuine pairs. Ideally, one would want to have both these quantities close to one simultaneously. As desired, despite being extremely simple (the incoherency measure depends on a few corresponding points), the proposed measure is able to reasonably separate the genuine pairs from impostors.

Textural Coherency: Given a pair of input images for verification, texture analysis involves estimation of their respective albedo maps followed by computation of LBP-based histograms as described in Section (5.3). Given the LBP signatures, verification is performed by training a SVM classifier with Radial Basis Function (RBF) kernel to separate genuine and impostor classes. Figure 5.7 shows the verification performance using this feature. The corresponding distributions of match and non-match scores are shown in Figure 5.8.

Comparison with other Approaches: To evaluate the performance of the proposed approaches, we compare their performance with two methods that have recently been used for matching age-separated face images [75]:

1. **SVM+GOP [75]:** Gradient orientation pyramid feature with SVM as classifier, and

2. **SVM+diff [94]:** Differences of normalized images with SVM as classifier.

The images are first aligned with the help of the eye locations and then cropped using an elliptic region as in [75]. The images are resized to 80 × 70 for efficient computation and subsequent training using SVM classifier. For SVM+GOP implementation, we use 3 pyramid layers and a Gaussian kernel with standard deviation of 0.5 as used in [75]. For SVM+diff implementation, each image is first normalized to have zero mean and unit variance before computing the image difference. SVM classifier with RBF kernel is used for both these methods. The performance obtained using the two methods on the Passport dataset is shown in Figure 5.7. We use the same classifier for all methods to be able to better evaluate the usefulness of the proposed feature for capturing the textural variations with aging. The proposed approach performs better than the other two methods which is also evident from the EER values in Table 5.1. Note that the performance of SVM+GOP and SVM+diff is different from the one reported in [75]. As the implementation of the two approaches is quite straight-forward and we use the same parameters as suggested in [75], the difference may be due to the difference in the dataset used. Also among the proposed methods, the textural measure performs better than its feature drift counterpart.

We also evaluate the usefulness of the proposed textural coherency feature using kernel density estimation to learn the probability density function (pdf) of the aging function using the training match pairs. The similarity score of two novel test images is given by the likelihood that the LBP-based histograms

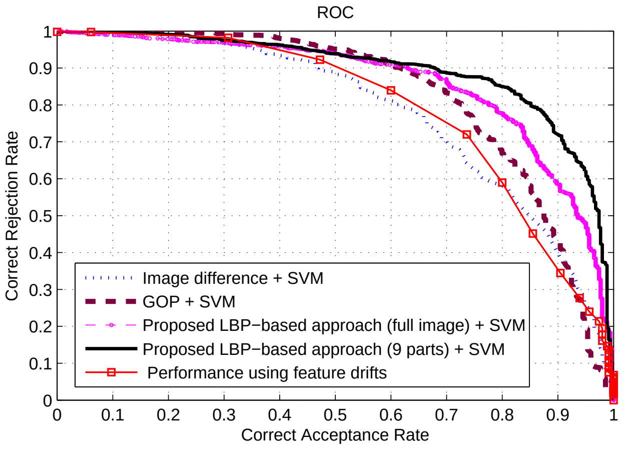


Figure 5.7: Verification performance obtained using the proposed feature drift and texture coherency measure. The performance is compared to that obtained by gradient orientation pyramid [75] and image difference [94] features and SVM as the classifier.

come from the learnt pdfs (one for each image region). We obtain slightly better performance than the one obtained using SVM. This indicates that the proposed feature does well in capturing the different variations due to aging and provides a useful measure for matching age-separated face images.

Table 5.1: Comparison of the Equal Error Rates (EER) obtained using the proposed measures.

SVM+diff	SVM+GOP	Feature Drifts	Texture (full image)	Texture (9 parts)
28.24%	25.00%	27.34%	20.99%	17.13%

5.4.1 Effect of age separation on matching performance

One of the main factors that may affect the performance of matching faces across age-progression is the age separation between the images. Therefore, it is imperative to analyze how the performance of the proposed approach varies across

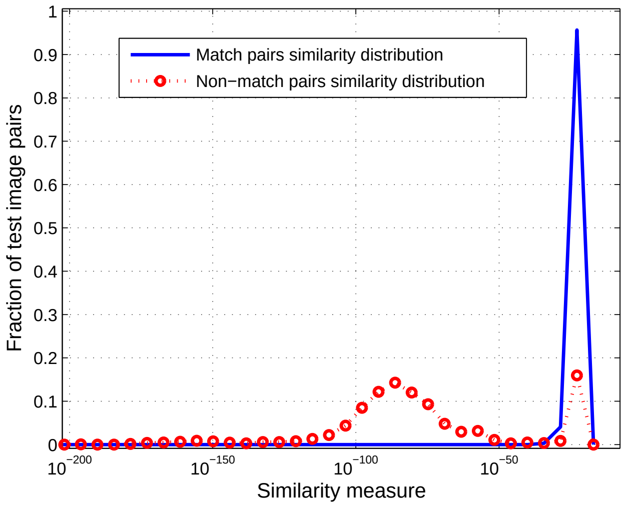


Figure 5.8: Genuine and impostor score distributions obtained using the textural analysis for matching age-separated face images.

different age-separations. Since we use a Passport dataset, the age gaps of the images is mainly concentrated around $9 - 10$ years (thats when people visit passport office for renewal, the very problem we are interested in) with the minimum age gap being 5 years as shown in Figure 5.5. In our analysis, we divide the intra-personal image pairs in the dataset into three groups based on their age gaps: 1) Age gap from $5 - 7$ years consisting of 70 matching image pairs, 2) from $7 - 9$ years having 240 matching pairs and 3) greater then 9 years having 600 matching pairs. For each group, all the available intra-personal image pairs and a large number (around 6-times the number of intra-personal for each age group) of randomly chosen inter-personal image pairs are taken as the testing set. The training set is the same as used in the previous experiment. EERs are used as the criterion to evaluate the performance. Table 5.2 shows the performance of the proposed approach for the different age groups. We see that the performance is more-or-less consistent for all the tested age groups. Such an observation has also been reported by [75]. Ling *et al.* report that the difficulty of matching images across age separation saturates for age gaps larger than four years.

Table 5.2: Comparison of the Equal Error Rates (EER) obtained for different age separations on the Passport dataset.

$5 - 7$ years	$7 - 9$ years	> 9 years
17.04%	15.78%	18.56%

5.5 Discussion and future work

The aging pattern of an individual depends on a variety of different factors that are difficult to model in a computational framework. This makes it extremely difficult to predict the exact appearance of a person as he/she ages. In spite of the large variability, humans are quite good at matching faces across age progression. This may mean that irrespective of the exact manner in which a person ages, there is a coherency in the way facial appearance changes with age. This motivates us to capture and utilize this coherency to recognize age-separated faces. Specifically, we analyze the coherency of the drifts in the various facial features and texture variations to verify whether two age-separated images belong to the same individual or not.

Experimental evaluation presented verify the effectiveness of such a non-generative approach even with simple measures of capturing coherency in aging. Since the drifts of features depends on the underlying facial muscle structure, this information may be used to obtain a better measure of drift coherency. For textural variation, we model variations in facial albedo using a simple linear relation. A more physically driven model of the aging function may be better suited to capture the textural variations that takes place as a person ages.

Chapter 6

Specularity Removal

Lambertian reflectance for the imaged scenes/objects has been commonly assumed in a variety of computer vision algorithms, such as shape reconstruction, image matching, motion detection, as well as photometric and multi-view stereo. However, most real world surfaces exhibit a combination of diffuse and specular components [111] making this assumption very restrictive in practice. Automatic separation of these components would enable these algorithms to be readily applied to a much wider class of non-Lambertian objects. Though the earliest research on reflection component separation using a single image dates back to mid 80's [111], the challenges involved in achieving this automatically and robustly on real images continue to interest researchers.

Most early approaches require explicit color segmentation to handle multicolored surfaces [111] [57] [12]. Not only is automatic color segmentation a research challenge in itself in the presence of specular highlights, it is extremely difficult to perform such a task even manually for complex scenes. A few recent algorithms overcome this limitation by relying on local interactions of pixels [78] [121] to remove specularity in complex textured scenes. While these methods do not require explicit color segmentation, they often involve detection of color boundaries in the input image [121] which might lead to unwanted artifacts around the detected boundaries. Most specularity removal approaches rely on the presence of one or more diffuse pixels of same surface color. The local-interaction based methods cannot take advantage of useful information present elsewhere in the image. In the absence of diffuse pixels in one part of the image, the information of the diffuse pixels present in other non-contiguous parts of the image cannot be utilized for specularity removal. Also, most approaches for reflection component separation assume that the illumination source color is known *a priori* or can be estimated accurately. Errors in the illumination color estimation lead to errors in the recovered diffuse and specular components. To the best of our knowledge, not much effort has gone into analyzing these errors so far.

Here, we propose a global approach which effectively characterizes the errors in the illumination color estimation to robustly estimate diffuse and specular reflection components in complex textured scenes. For dichromatic surfaces, pixels with the same underlying surface color lie on a plane which also contains the illumination source color vector [111]. Error in illumination color information may disturb this co-planarity leading to erroneous separation of diffuse and specular components. The analysis we present here shows how errors in the illumination color can lead to spatially varying uncertainty (in RGB color space) for determining which pixels lie on the same dichromatic plane. The error analysis results in a very simple, effective and robust algorithm which requires just a single parse of the image pixels to separate the diffuse and specular components. The proposed approach offers the following advantages:

- Statistics of errors in the illumination color estimate is effectively utilized for robust recovery of the diffuse and specular reflection components.

- The presented error analysis is very general and may be used with any existing algorithm to make them robust to errors in the illumination color estimate.

- The method is non-iterative and does not require explicit color segmentation or color boundary detection and thus can handle very complicated textures.

In addition, we propose a 4D Hough transform based algorithm to automatically estimate the illumination source color from a single color image. Unlike the state-of-the-art approaches for illumination color estimation for textured dichromatic surfaces [123] [127], the proposed approach does not depend on a candidate list of illumination colors or require detection of specular highlights. In our approach, the illumination color is estimated directly by integrating evidences from image pixels. Results on complex textured surfaces show the usefulness of the proposed algorithms.

6.0.1 Organization of the chapter

The rest of the chapter is organized as follows. Section 6.1 briefly describes the Dichromatic Reflectance Model. The proposed approach for illumination color estimation is detailed in Section 6.2. Section 6.3 presents the error analysis along with the proposed algorithm to separate the reflection components. Results of experimental evaluation are presented in Section 6.4. Section 6.5 concludes the chapter with a brief summary.

6.1 Dichromatic Reflectance Model

According to the dichromatic reflectance model [111], the Bidirectional Reflectance Distribution Function (BRDF) of the surface is a linear combination of two components: the interface (specular) reflectance and the body (diffuse) reflectance. Using this model, the image formation equation (for the k^{th} channel) for a point $\mathbf{x} = \{x, y\}$ with normal $\hat{\mathbf{n}}$ can be written as

$$\mathrm{I}_{\mathrm{k}}(\mathbf{x}) = \big(\bar{m}_d(\mathbf{x})\Lambda_{\mathrm{k}}(\mathbf{x}) + \bar{m}_s(\mathbf{x})\Gamma_{\mathrm{k}}\big)\hat{\mathbf{n}} \cdot \hat{\mathbf{s}} \tag{6.1}$$

where $\hat{\mathbf{s}}$ is the light source direction and functions $\bar{m}_d(\mathbf{x})$ and $\bar{m}_s(\mathbf{x})$ are the diffuse and specular BRDFs, respectively. Γ_{k} represents the effective illumination source strength for the k^{th} channel and Λ_{k} is the effective albedo in that channel. Combining the three color channels as measured using a typical camera, we obtain

$$\mathbf{I}(\mathbf{x}) = m_d(\mathbf{x})\boldsymbol{\Lambda}(\mathbf{x}) + m_s(\mathbf{x})\boldsymbol{\Gamma} \tag{6.2}$$

where $m_d(\mathbf{x}) = \bar{m}_d(\mathbf{x})(\hat{\mathbf{n}} \cdot \hat{\mathbf{s}})$, $m_s(\mathbf{x}) = \bar{m}_s(\mathbf{x})(\hat{\mathbf{n}} \cdot \hat{\mathbf{s}})$, $\mathbf{I} = [\mathrm{I_r}, \mathrm{I_g}, \mathrm{I_b}]$, $\mathbf{\Gamma} = [\Gamma_\mathrm{r}, \Gamma_\mathrm{g}, \Gamma_\mathrm{b}]$ and $\mathbf{\Lambda} = [\Lambda_\mathrm{r}, \Lambda_\mathrm{g}, \Lambda_\mathrm{b}]$. Without loss of generality, we assume $\|\mathbf{\Gamma}\| = \|\mathbf{\Lambda}\| = 1$ as scale can be taken care of in $m_d(\mathbf{x})$ and $m_s(\mathbf{x})$.

6.2 Illumination Color Estimation

For a dichromatic surface, any pixel $\mathbf{I}(\mathbf{x})$ is formed by the linear combination of the two vectors (6.2): the surface color of that pixel denoted by $\mathbf{\Lambda}(\mathbf{x})$ and the illumination color of the image denoted by $\mathbf{\Gamma}$ with $m_d(\mathbf{x})$ and $m_s(\mathbf{x})$ being the combining coefficients. In other words, all pixels with the same surface color $\mathbf{\Lambda}$ lie on a plane (known as the dichromatic plane) formed by $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$.

In this section, we describe the proposed algorithm to estimate the illumination source color from a single image consisting of dichromatic surface(s). Assuming that the image is illuminated by a single light source, the illumination color $\mathbf{\Gamma}$ is same throughout the image. Different dichromatic planes corresponding to different surface colors in the image form a pencil of planes in the RGB color space (Figure 6.1(c)). The axis of the pencil is the unknown $\mathbf{\Gamma}$ we aim to estimate. Here we propose a Hough transform based approach for illumination color estimation that does not require explicit color segmentation. We first briefly describe the basics of Hough transform for ease of understanding.

The classical Hough transform is used to detect straight lines in 2D from a given point set. The main idea is to gather evidence for each straight line by mapping the points from the image space to a parameter space. This is done by first representing the unknown straight lines in terms of their parameters (e.g., slope and intercept) instead of (x, y) co-ordinates. Each input point (x_i, y_i) then votes in the chosen parameter space for all lines that can potentially pass through it. If there exists a set of collinear points in the input space, they all vote for the corresponding line, resulting in a peak in the parameter space. Hence, lines in the input space are detected based on the location of peaks in the parameter

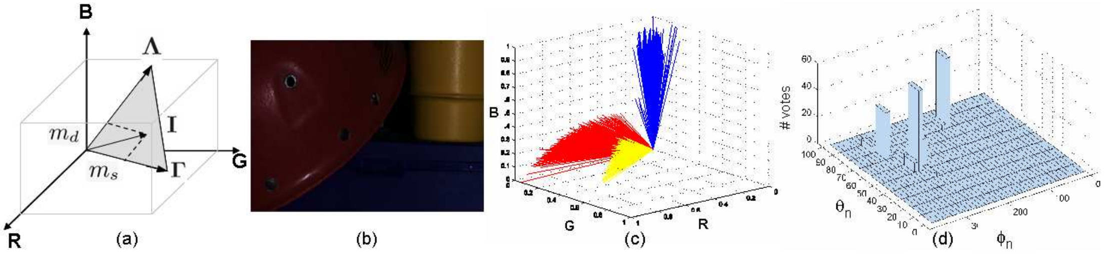


Figure 6.1: (a) Vector diagram for dichromatic surfaces; (b) Input color image; (c) The sheaf of planes formed by three dominant surface colors in the image; (d) The histograms of normals for the correct illumination color.

space. Here we use a modified form of the Hough transform to locate the axis of the pencil directly without requiring to first detect the different planes separately. The proposed method does not require any candidate list of illumination colors or detection of specular highlights.

Suppose the input image $\mathbf{I}$ is composed of K different surface colors (unknown) denoted by $\{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \ldots, \mathbf{\Lambda}_K\}$ and $\mathbf{\Gamma}$ denotes the unknown illumination color. Each surface color $\mathbf{\Lambda}_i$ forms a distinct plane which is characterized by its normal vector $\mathbf{n}_i$, given by

$$\mathbf{n}_i = \mathbf{\Lambda}_i \times \mathbf{\Gamma}, \qquad i \in \{1, 2, \ldots, K\} \tag{6.3}$$

Here we consider a 4D voting space consisting of parameters characterizing the unknown illumination color vector and these normals. Both illumination color vector and normals to the planes are represented using their elevation and azimuth ($\{\theta_\Gamma, \phi_\Gamma\}$ and $\{\theta_\mathrm{n}, \phi_\mathrm{n}\}$ respectively). $\mathbf{\Gamma}$ being a color in the RGB space, $\Gamma_\mathrm{r} \geq 0, \Gamma_\mathrm{g} \geq 0, \Gamma_\mathrm{b} \geq 0$, so $0^\circ \leq \theta_\Gamma, \phi_\Gamma \leq 90^\circ$. On the other hand, the range for the parameters for the normals is $0^\circ \leq \theta_\mathrm{n} \leq 180^\circ$ and $0^\circ \leq \phi_\mathrm{n} \leq 360^\circ$.

Following a Hough transform based approach, a color vector $\mathbf{I}(\mathbf{x})$ with surface color $\mathbf{\Lambda}_i$ votes for all the planes that can pass through it and the corresponding illumination color vectors. Note that among all the planes that pass through $\mathbf{I}(\mathbf{x})$, there is one and only one plane that passes through the vector representing the true unknown $\mathbf{\Gamma}$ which is given by

$$\mathbf{I}(\mathbf{x}) \times \mathbf{\Gamma} = \big(m_d(\mathbf{x})\mathbf{\Lambda}_{\mathrm{i}} + m_s(\mathbf{x})\mathbf{\Gamma}\big) \times \mathbf{\Gamma} = m_d(\mathbf{x})\hat{\mathbf{n}}_i \tag{6.4}$$

Quite clearly, all pixels with surface color $\mathbf{\Lambda}_{\mathrm{i}}$ will form the same plane (with normal $\hat{\mathbf{n}}_i$) with the true $\mathbf{\Gamma}$, thereby resulting in a peak at $\{\theta_\Gamma, \phi_\Gamma, \theta_{\mathrm{n}_i}, \phi_{\mathrm{n}_i}\}$ in the 4D parameter space. Similarly, pixels corresponding to other surface colors $\mathbf{\Lambda}_{\mathrm{j}}, \forall j \in \{1, 2, \dots, K\}, j \neq i$ form respective peaks at the locations $\{\theta_\Gamma, \phi_\Gamma, \theta_{\mathrm{n}_j}, \phi_{\mathrm{n}_j}\}$. Therefore, the voting in 4D parameter space results in K peaks for the correct $\{\theta_\Gamma, \phi_\Gamma\}$ pair, one corresponding to each of the K dichromatic planes as illustrated in Figure 6.1(d).

Given an input color image, the number of dichromatic planes forming the pencil and their orientation in the RGB space is not known. Therefore, the unknown illumination color cannot be determined by locating $\{\theta_\Gamma, \phi_\Gamma\}$ which forms peaks corresponding to each of the surface colors in the image. If the voting space is appropriately thresholded and marginalized over the two dimensions corresponding to normals to the plane $\{\theta_{\mathrm{n}}, \phi_{\mathrm{n}}\}$, the true illumination color $\{\theta_\Gamma, \phi_\Gamma\}$ should get the highest number of votes assuming the surfaces in the image follow the dichromatic model strictly.

For real data, due to factors like camera noise and deviations from the model, all pixels of one surface color do not always form a nice plane with the true illumination color vector. Also, if there are very few pixels of a certain surface color in the image, the peak formed even at the correct illumination color is very small. This makes automatic selection of the threshold quite difficult. In our implementation, we use the sum of the two highest peaks as the measure for determining the correct illumination color vector. We find this measure to be quite robust for real data as illustrated in our experimental evaluation. This kind of measure implicitly assumes that the image has at least two surface colors containing significant number of pixels.

6.2.1 Illumination color estimation for surfaces with single surface color

The approach for estimating the illumination source color presented here assumes that the image has $K \geq 2$ different surface colors. For a uniformly colored surface, all pixels lie on the same plane defined by the corresponding surface color $\mathbf{\Lambda}$ and the illumination color $\mathbf{\Gamma}$. Therefore, the correct $\{\theta_\Gamma, \phi_\Gamma\}$ will have just one peak corresponding to this plane in the 4D parameter space. This peak also exists for other prospective $\{\theta_\Gamma, \phi_\Gamma\}$ lying in the same dichromatic plane, making it impossible to identify the correct illumination color.

Uniformly colored surfaces can be identified beforehand by verifying if a plane can be fit through the input color pixels in the RGB space. In our implementation, we use Eigen-analysis for this purpose. To estimate the illumination color for such images, we first transform the input color vectors in the 3D RGB space to the local co-ordinate system of the dichromatic plane. Since $\mathbf{\Lambda}(\mathbf{x})$ is constant throughout the image, different pixels differ only due to the combining coefficients $m_d(\mathbf{x})$ and $m_s(\mathbf{x})$. Interestingly, if there exists a set of pixels with same $m_d(\mathbf{x})$, the corresponding points will form a straight line on the dichromatic plane as shown in Figure 6.2. If there are considerable number of pixels for each m_d value in the image, they form a collection of straight lines with different intercepts on the $\mathbf{\Lambda}$-axis but slope same as that of the illumination color vector (Figure 6.2). These lines can easily be detected using the classical 2D Hough transform by transforming the points on the dichromatic plane to the slope-intercept parameter space. This results in several peaks (one for each $m_d(\mathbf{x})$ value) for the slope corresponding to the illumination vector. This can be detected using appropriate thresholding and marginalization of the 2D voting space of intercept and slope.

The value of m_d depends on the diffuse albedo of the pixel, the intensity of the incident light, and the angle between the lighting direction and the surface normal. So, in general the value of m_d is different for different pixels in the image. Often

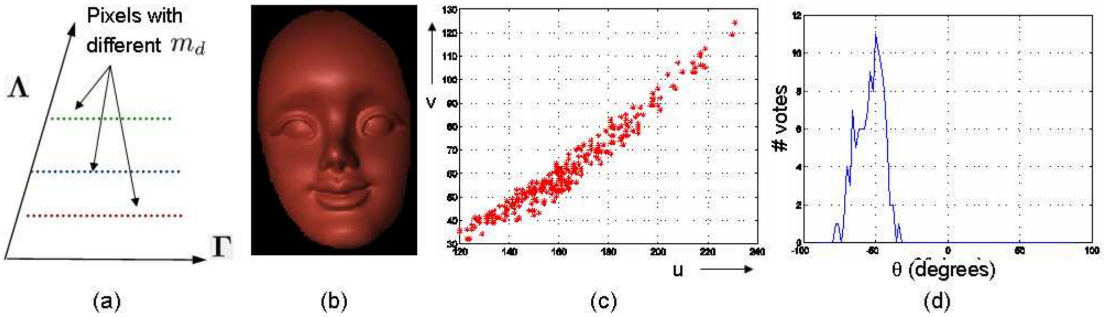


Figure 6.2: Illumination color estimation for uniformly colored surfaces; (a) A sketch illustrating the underlying intuition; (b) Example image; (c) Color vectors in the local co-ordinate system of the dichromatic plane; (d) The thresholded and marginalized votes in the 2D Hough space.

for neighboring pixels, the variation of m_d is small and so it can be approximated as constant. This is a reasonable assumption for ordinary digital cameras because of their accuracy limitation.

6.3 Separation of Reflection Components

Most of the existing methods for reflectance components separation assume accurate knowledge of illumination color which is a restrictive assumption. But in most practical cases, the illumination color is not known *a priori* and it has to be estimated which introduces errors in the illumination color estimate. In this section, we propose an approach for separation of the diffuse and specular components which takes into account errors in the illumination color estimate.

6.3.1 Analysis of Errors in Illumination Color $\mathbf{\Gamma}$

For a dichromatic surface, any pixel $\mathbf{I}(\mathbf{x})$ is formed by the linear combination of the two vectors (6.2): the surface color of that pixel denoted by $\mathbf{\Lambda}(\mathbf{x})$ and the illumination color of the image denoted by $\mathbf{\Gamma}$ with $m_d(\mathbf{x})$ and $m_s(\mathbf{x})$ being the combining coefficients. In other words, all pixels with the same surface color $\mathbf{\Lambda}$ lie on a plane (known as the dichromatic plane) formed by $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$. Hence, for any two pixels $\mathbf{I}(\mathbf{x}_i)$ and $\mathbf{I}(\mathbf{x}_j)$ of the same surface color,

$$\mathbf{I}(\mathbf{x}_j) \cdot \big(\hat{\mathbf{I}}(\mathbf{x}_i) \times \mathbf{\Gamma}\big) = 0 \tag{6.5}$$

where $\hat{\mathbf{I}}(\mathbf{x}_i)$ is the normalized color vector. The operators $\cdot$ and $\times$ denote the vector dot and cross product respectively. Equation (6.5) assumes that the illumination color is accurately known. We now analyze how errors in $\mathbf{\Gamma}$ affect this relation.

Let the estimated illumination color be denoted as $\mathbf{\Gamma}^{\text{est}}$. Note that the left hand side of (6.5) essentially represents the perpendicular distance of color vector $\mathbf{I}(\mathbf{x}_j)$ from the plane formed by $\hat{\mathbf{I}}(\mathbf{x}_i)$ and $\mathbf{\Gamma}$. Using the erroneous estimate $\mathbf{\Gamma}^{\text{est}}$ of the illumination color, this distance (say, d) can be written as

$$d = \mathbf{I}(\mathbf{x}_j) \cdot \big(\hat{\mathbf{I}}(\mathbf{x}_i) \times \mathbf{\Gamma}^{\text{est}}\big) = \mathbf{\Gamma}^{\text{est}} \cdot \big(\mathbf{I}(\mathbf{x}_j) \times \hat{\mathbf{I}}(\mathbf{x}_i)\big) \tag{6.6}$$

Expressing the estimated illumination color in terms of its true value and an error term, we get

$$d = \big(\mathbf{\Gamma} + \mathbf{\Gamma}^{\text{err}}\big) \cdot \big(\mathbf{I}(\mathbf{x}_j) \times \hat{\mathbf{I}}(\mathbf{x}_i)\big) \tag{6.7}$$

Since the two color vectors lie in the same plane as the true unknown illumination color, $\mathbf{\Gamma} \cdot \big(\mathbf{I}(\mathbf{x}_j) \times \hat{\mathbf{I}}(\mathbf{x}_i)\big) = 0$. Therefore, the expression for d simplifies to

$$d = \mathbf{\Gamma}^{\text{err}} \cdot \big(\mathbf{I}(\mathbf{x}_j) \times \hat{\mathbf{I}}(\mathbf{x}_i)\big) = \mathbf{\Gamma}^{\text{err}} \cdot \big(\sin(\theta) \parallel \mathbf{I}(\mathbf{x}_j) \parallel \hat{\mathbf{n}}\big) \tag{6.8}$$

where θ is the angle between the two vectors and $\hat{\mathbf{n}} = [\hat{\mathbf{n}}_{\text{r}},\ \hat{\mathbf{n}}_{\text{g}},\ \hat{\mathbf{n}}_{\text{b}}]$ is the unit normal to the plane formed by them. One can express the error in illumination color in terms of its components along the three color channels. Assuming that the error in illumination color is unbiased along the three channels, we have $E(\mathbf{\Gamma}^{\text{err}}_{\text{r}}) = E(\mathbf{\Gamma}^{\text{err}}_{\text{g}}) = E(\mathbf{\Gamma}^{\text{err}}_{\text{b}}) = 0$. Thus from (6.8), the expected value of the perpendicular distance is zero, i.e., $E(d) = 0$. The expression for the variance $\sigma^2(d)$ is given by

$$\sigma^2(d) = K^2 E\big(\boldsymbol{\Gamma}_{\mathrm{r}}^{\mathrm{err}}\hat{\mathbf{n}}_{\mathrm{r}} + \boldsymbol{\Gamma}_{\mathrm{g}}^{\mathrm{err}}\hat{\mathbf{n}}_{\mathrm{g}} + \boldsymbol{\Gamma}_{\mathrm{b}}^{\mathrm{err}}\hat{\mathbf{n}}_{\mathrm{b}}\big)^2 \qquad (6.9)$$
$$\text{where,} \qquad K = \sin(\theta) \parallel \mathbf{I}(\mathbf{x}_j) \parallel$$

Assuming that the error in illumination color estimate is uncorrelated in the three color channels and the variances are same i.e., $E(\boldsymbol{\Gamma}_{\mathrm{r}}^{\mathrm{err}})^2 = E(\boldsymbol{\Gamma}_{\mathrm{g}}^{\mathrm{err}})^2 = E(\boldsymbol{\Gamma}_{\mathrm{b}}^{\mathrm{err}})^2 = \epsilon^2$, the expression for the variance $\sigma^2(d)$ simplifies to

$$\sigma^2(d) = K^2\epsilon^2\big(\hat{\mathbf{n}}_{\mathrm{r}}^2 + \hat{\mathbf{n}}_{\mathrm{g}}^2 + \hat{\mathbf{n}}_{\mathrm{b}}^2\big) = K^2\epsilon^2 \qquad (6.10)$$

since $\hat{\mathbf{n}} = [\hat{\mathbf{n}}_{\mathrm{r}},\ \hat{\mathbf{n}}_{\mathrm{g}},\ \hat{\mathbf{n}}_{\mathrm{b}}]$ is a unit normal vector. Appropriately, the variance $\sigma^2(d)$ is proportional to the error variance of illumination color vector and $\parallel \mathbf{I}(\mathbf{x}_j) \parallel^2$. Interestingly, the variance is also proportional to $\sin^2(\theta)$ where θ is the angle between the two considered color vectors.

We investigate the validity of such a relation using a synthetic image [122] (Figure 6.3(a)). The color vectors are normalized to unit norm to analyze solely the effect of angle between color vectors. The angular error between the true and erroneous illumination color vectors is 4.4° in this experiment. The maps in Figure 6.3 (b) and (d) show the perpendicular distances obtained using the top-left corner and center pixels as reference $\hat{\mathbf{I}}(\mathbf{x}_i)$ respectively. The corresponding $\sin(\theta)$ maps are shown in Figure 6.3 (c) and (e). As can be seen, the corresponding perpendicular distance and $\sin(\theta)$ maps are similar justifying the analysis. Most existing approaches do not account for any error in illumination color, thereby implicitly assuming such maps to be uniform.

The above analysis considers two color vectors of the same surface color. Now let us consider the case where the two color vectors have different surface colors and thus lie on different dichromatic planes. Following a similar analysis, the perpendicular distance (d) of color vector $\mathbf{I}(\mathbf{x}_j)$ from the plane formed by $\hat{\mathbf{I}}(\mathbf{x}_i)$

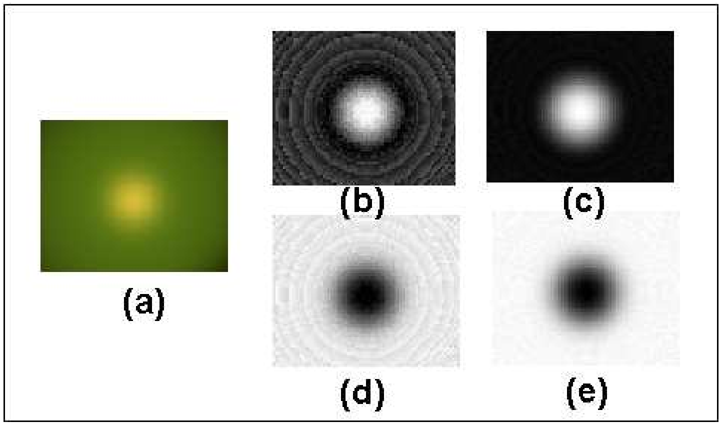


Figure 6.3: An illustration to validate the dependence of $\sigma^2(d)$ on the sine of the angle between the two color vectors as given by (6.10). (a) Input synthetic image; (b) and (d) show the perpendicular distance maps for two different choices of reference pixel $\hat{\mathbf{I}}(\mathbf{x}_i)$; (c) and (e) are the corresponding $\sin(\theta)$ maps.

and the erroneous illumination color estimate $\mathbf{\Gamma}^{\text{est}}$ can be written as

$$d = \parallel \mathbf{I}(\mathbf{x}_j) \parallel \sin(\alpha)\sin(\beta) + \mathbf{\Gamma}^{\text{err}} \cdot \big(\sin(\theta) \parallel \mathbf{I}(\mathbf{x}_j) \parallel \hat{\mathbf{n}}\big) \tag{6.11}$$

Here α is the angle subtended by the color vector $\mathbf{I}(\mathbf{x}_j)$ with the true illumination color vector $\mathbf{\Gamma}$. β is the angle between the normals to the two dichromatic planes. Following similar analysis as in the same surface color case, we get the expression of the mean $E(d)$ and variance $\sigma^2(d)$ of the distance as follows

$$\begin{aligned} E(d) &= \parallel \mathbf{I}(\mathbf{x}_j) \parallel \sin(\alpha)\sin(\beta) \\ \sigma^2(d) &= \sin^2(\theta) \parallel \mathbf{I}(\mathbf{x}_j) \parallel^2 \epsilon^2 \end{aligned} \tag{6.12}$$

From (6.10) and (6.12), errors in illumination color information result in uncertainty in determining if two pixels lie on the same dichromatic plane or not. Since such a task is the underlying theme in all dichromatic model based separation algorithms, errors in source color lead to errors in estimation of diffuse and specular components. These expressions for $\sigma^2(d)$ indicate how to account for errors in source color as long as the errors are not large enough to overcome the $E(d)$ factor in (6.12).

6.3.2 Proposed algorithm

Based on the error analysis given in the previous section, we propose a simple and effective algorithm to robustly separate diffuse and specular reflection components given a single image. The different steps of the proposed algorithm are described in this section.

Determination of dichromatic planes: In the absence of any error in the illumination color estimate (and any other modeling error), one can directly use (6.5) to determine all pixels whose color vectors lie on the same dichromatic plane. Such an approach will not give the desired result in the presence of any noise. Based on the error analysis, the proposed approach uses the dichormatic reflectance model to robustly separate input pixels based on their underlying surface color (not input pixel intensities which may show large variations due to specular highlights).

If the error in the illumination color in the r, g, b color channels has a mean 0 and variance ϵ^2, the the perpendicular distance d of a color vector $\mathbf{I}(\mathbf{x}_j)$ from a dichromatic plane formed by $\mathbf{I}(\mathbf{x}_i)$ (of same surface color) and $\mathbf{\Gamma}$ has mean 0 and variance $\sigma^2(d)$ respectively (6.10). Thus, if a test color vector has Euclidean distance of x from the plane, the corresponding Mahalanobis distance is given by

$$D_M(x) = \frac{x}{\sin(\theta) \parallel \mathbf{I}(\mathbf{x}_j) \parallel \epsilon} \tag{6.13}$$

If we further assume that the distribution is normal, then the probability of the test point lying in the same dichromatic plane is given by

$$P(x) = \frac{1}{\sin(\theta) \parallel \mathbf{I}(\mathbf{x}_j) \parallel \epsilon\sqrt{2\pi}} e^{-D_M{}^2(x)} \tag{6.14}$$

Refinement of illumination color estimate: The normals to the dichromatic planes are obtained in a Least Squares (LS) fashion by solving the following homogeneous system of linear equations

$$\begin{pmatrix} \boldsymbol{I}_{\boldsymbol{i}1}^{(R)} & \boldsymbol{I}_{\boldsymbol{i}1}^{(G)} & \boldsymbol{I}_{\boldsymbol{i}1}^{(B)} \\ \boldsymbol{I}_{\boldsymbol{i}2}^{(R)} & \boldsymbol{I}_{\boldsymbol{i}2}^{(G)} & \boldsymbol{I}_{\boldsymbol{i}2}^{(B)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}_{\boldsymbol{i}N_i}^{(R)} & \boldsymbol{I}_{\boldsymbol{i}N_i}^{(G)} & \boldsymbol{I}_{\boldsymbol{i}N_i}^{(B)} \end{pmatrix} \hat{\mathbf{n}}_i^{\mathrm{LS}} = \mathbf{0} \tag{6.15}$$

where $[\boldsymbol{I}_{\boldsymbol{i}1}, \boldsymbol{I}_{\boldsymbol{i}2}, \ldots, \boldsymbol{I}_{\boldsymbol{i}N_i}]$ are the N_i pixels that are determined to lie on i^{th} dichromatic plane in the previous step. Since the true illumination color vector is perpendicular to these dichormatic normals, a new estimate ($\mathbf{\Gamma}^{\mathrm{LS}}$) of the illumination color vector is obtained in a similar LS method.

Separation of diffuse and specular reflection components: Suppose $\{S_i\}$ denotes the set of pixels in the given image which lie on the same dichromatic plane $\hat{\mathbf{n}}_i$. The plane and thus all pixels in $\{S_i\}$ are bounded by the illumination color vector on one side and the surface color $\mathbf{\Lambda}_i$ on the other. Now, if the set contains one or more purely diffuse pixels, (for which the specular coefficient $m_s = 0$), the corresponding vectors $\mathbf{I}(\mathbf{x}) = m_d(\mathbf{x})\mathbf{\Lambda}_i$ will have the same direction (say $\hat{\mathbf{l}}_i$) as the surface color $\mathbf{\Lambda}_i$. Since the vector corresponding to surface color (and thus the diffuse pixels) forms the largest angle with the direction of the illumination color vector, the desired direction $\hat{\mathbf{l}}_i$ is given by

$$\hat{\mathbf{l}}_i = \arg \max_{\mathbf{I}(\mathbf{x}_i) \in \{S_i\}} \left\{ \cos^{-1} \left(\frac{\mathbf{I}(\mathbf{x}_i) \cdot \mathbf{\Gamma}^{\mathrm{LS}}}{\parallel \mathbf{I}(\mathbf{x}_i) \parallel} \right) \right\} \tag{6.16}$$

The diffuse component of a pixel $\mathbf{I}(\mathbf{x}_1)$ is thus given by its projection on the determined surface color direction

$$\mathbf{I}_d(\mathbf{x}_1) = m_d(\mathbf{x}_1)\hat{\mathbf{I}}_i, \qquad m_d(\mathbf{x}_1) = \frac{\| \mathbf{I}(\mathbf{x}_1) \times \mathbf{\Gamma}^{\mathrm{LS}} \|}{\| \hat{\mathbf{I}}_i \times \mathbf{\Gamma}^{\mathrm{LS}} \|} \tag{6.17}$$

Similarly, the specular reflection component is given by the projection along the illumination color vector

$$\mathbf{I}_s(\mathbf{x}_1) = m_s(\mathbf{x}_1)\mathbf{\Gamma}^{\mathrm{LS}}, \qquad m_s(\mathbf{x}_1) = \frac{\| \mathbf{I}(\mathbf{x}_1) \times \hat{\mathbf{I}}_i \|}{\| \mathbf{\Gamma}^{\mathrm{LS}} \times \hat{\mathbf{I}}_i \|} \tag{6.18}$$

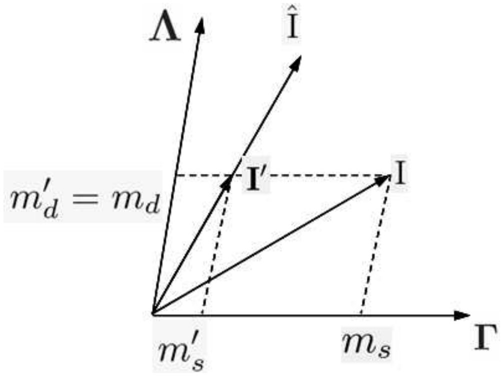


Figure 6.4: A sketch illustrating the separation of reflectance components in the absence of purely diffuse pixels.

The above analysis assumes that the set $\{S_i\}$ contains one or more purely diffuse pixels for which $m_s = 0$. In the absence of any purely diffuse pixels, no pixel in the set has the same direction as the surface color $\mathbf{\Lambda}_i$, and thus the surface color cannot be determined by (6.16). In this case, $\hat{\mathbf{I}}_i$ denotes the *most diffuse direction* and projection along this direction gives a more diffuse version of the pixel given by

$$\mathbf{I}'(\mathbf{x}_1) = m'_d(\mathbf{x}_1)\mathbf{\Lambda}_i + m'_s(\mathbf{x}_1)\mathbf{\Gamma}^{\mathrm{LS}} \tag{6.19}$$

Using (6.17) and (6.18) , it is straightforward to prove that

$$m'_d(\mathbf{x}_1) = m_d(\mathbf{x}_1) \qquad \text{and} \qquad m'_s(\mathbf{x}_1) < m_s(\mathbf{x}_1) \tag{6.20}$$

where $m_d(\mathbf{x}_1)$ and $m_s(\mathbf{x}_1)$ are the true unknown diffuse and specular coefficients of $\mathbf{I}(\mathbf{x}_1)$ respectively. The image $\mathbf{I}'(\mathbf{x}_1)$ has the same diffuse component as the original image, but is less specular. This is illustrated in Figure 6.4.

6.4 Experimental Evaluation

In this section, we report the results of experimental evaluation of the proposed approaches.

6.4.1 Robustness to errors in source color

One of the main advantages of the proposed approach for specularity removal is its robustness to errors in illumination color estimate. Therefore, it is important to evaluate the performance of the method for varying degrees of errors in the illumination color estimate. Figure 6.5 shows the result for varying degrees of angular errors in the illumination color estimate. As desired, the recovered diffuse component seems quite robust to the errors and degrades very gracefully with the increase in error. Both the input images have been taken from the dataset provided by Simon Fraser University for evaluation of computational color constancy algorithms [16] which also contains the true illumination color information.

6.4.2 Importance of global information

For complex textured surfaces, pixels with same surface color may not always occur in a single contiguous region and may be spread throughout the input image as shown by the highlighted regions in Figure 6.6. Even if some parts of the image (shown by oval) are completely specular with no diffuse pixels, the

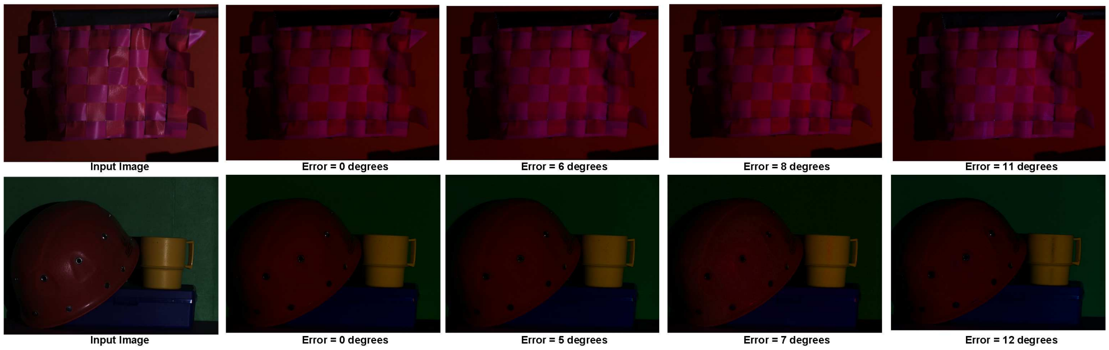


Figure 6.5: Robustness of the proposed approach to errors in illumination color. First column: Original images; Other columns show the recovered diffuse reflectance component with varying errors in the illumination color (the numbers under each image indicate the angular error with the true illumination color).

proposed algorithm can effectively utilize the diffuse pixels of the same surface color present elsewhere in the image (shown by square) to remove the specularity. This is due to the fact that irrespective of the spatial location in the image, all pixels of the same surface color will lie on the same dichromatic plane.

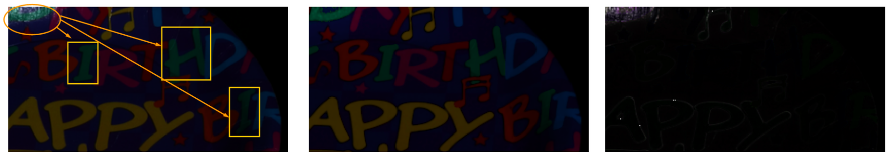


Figure 6.6: Dealing with spatial discontinuity of pixels with same surface color. Left: Original image; Middle: Diffuse component; Right: Specular component.

Also, the proposed method gives a more diffused image in the absence of purely diffused pixels. Figure 6.7 shows another result on the same image in which a component along the direction of illumination color is added to each pixel. This ensures that the image does not contain any purely diffuse pixels. As desired, the method is successful in removing a significant amount of specularity.

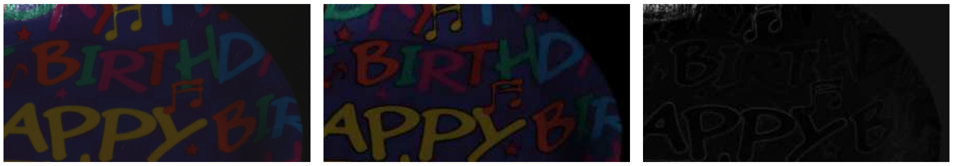


Figure 6.7: Reflectance component separation in the absence of purely diffuse pixels. Left: Original image; Middle: Diffuse component; Right: Specular component.

6.4.3 Robust color segmentation in the presence of specular highlights

The surface color information for each pixel (obtained in an intermediate step of the proposed approach) can also be used to obtain surface color segmentation on textured surfaces in the presence of specularity. Figure 6.8 shows the result of such a surface color labeling. Not that this is different from color segmentation which is based on pixel intensities which may show large variations due to specular highlights. Color segmentation results on the same images using Mean Shift algorithm are provided for comparison (http://www.caip.rutgers.edu/riul/research/code.html). Unlike Mean Shift algorithm, the labeling obtained using the proposed approach is able to account for specularity in the input image.

More results on complex textured surfaces are shown in Figure 6.9. As shown, the proposed approach does well in separating the diffuse and specular reflectance components. All the images used in this experiment are taken from the datasets used in [78] and [119]. The illumination colors for these images have been estimated using the proposed Hough transform based approach.

6.4.4 Illumination color estimation

We test the proposed illumination color estimation approach on the dataset provided by Simon Fraser University [16]. The database contains images of different scenes taken under 11 different lights. As the proposed algorithm is meant for dichromatic surfaces (with non-trivial specular component), we select a subset of over 60 images from the database which have significant amount of specularity.

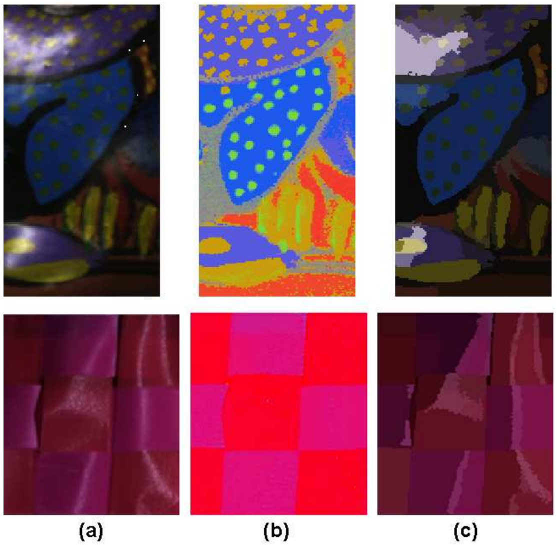


Figure 6.8: (a) Input image; (b) Surface color labeling obtained using our algorithm; (c) Image color segmentation obtained using the Mean Shift algorithm.

The number of selected images in each category along with the mean angular error of the estimated illumination color is given in Table 6.1. The angular errors obtained compare favorably to the ones reported in [127].

Table 6.1: Evaluation of the proposed illumination color estimation approach

Data category [16]	books-4	books-5	fruit-1	plastic-1	plastic-2	tape-1	apples
Number of images	3	5	11	11	11	11	11
Mean angular error	4.9°	6.2°	6.3°	7.0°	3.4°	3.7°	2.7°

6.5 Summary

The main contribution of this work is the analysis of errors in source color information to perform robust separation of diffuse and specular reflectance components from a single image. The error analysis is very general and should be useful even for other algorithms to account for source color errors. The analysis leads to a very

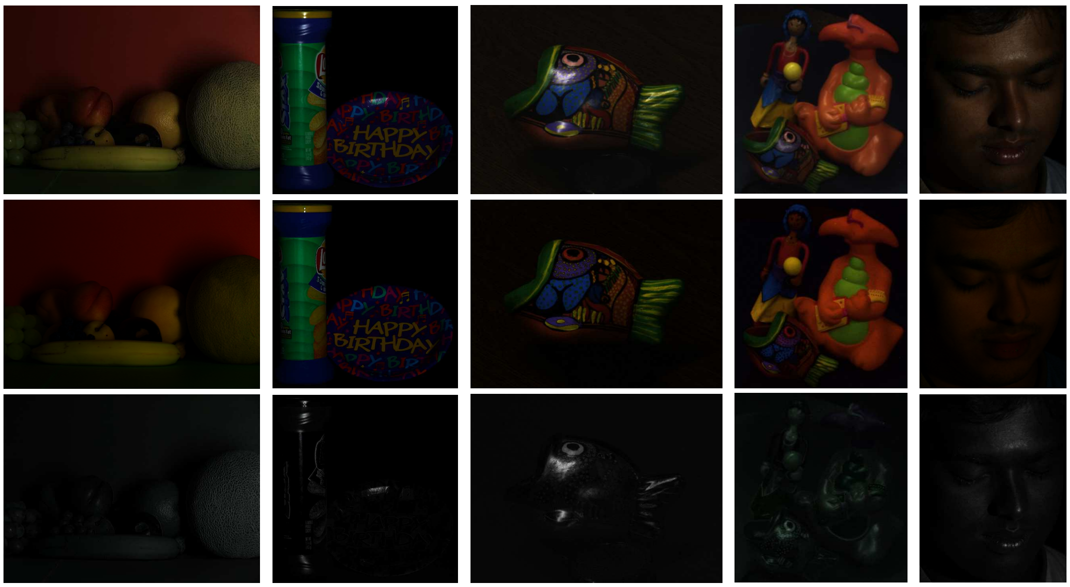


Figure 6.9: Separation results; First Row: Original images; Second Row: Diffuse component; Third Row: Specular component. Estimated illumination colors for these images (from left to right) are [0.66 0.56 0.50], [0.55 0.57 0.61], [0.59 0.61 0.53], [0.56 0.66 0.48] and [0.54 0.62 0.57].

simple and robust algorithm to separate the two reflection components. Unlike many recent techniques which deal with complex textures, our method is global in nature and does not rely solely on the local information provided by the neighboring pixels. In addition, we also presented a Hough transform based approach for source color estimation that does not require color segmentation or candidate list of source colors. Illustrations and results of experimental evaluations show the usefulness of the proposed algorithm.

Chapter 7

Efficient Shape Representation and Matching

Numerous applications of shape matching and recognition have made it a very important area of research in the field of computer vision (see Figure 7.1). Character recognition, trademark logo retrieval, activity recognition, object recognition, human pose estimation and matching range data are a few of the challenging applications that can benefit from accurate and efficient shape matching techniques. Different applications require different representations and hence different matching algorithms to handle the large variations in shapes. Also with the recent advancement in technology and the availability of different kinds of sensors, the amount of data to be handled has increased tremendously over the last few decades. So even though research in the area of shape matching has matured, the challenges involved in achieving high performance in terms of both accuracy and computational complexity continues to interest researchers.

Matching shapes across complex deformations has been the main focus of most works in recent times. Many existing shape matching algorithms require computationally demanding matching schemes to be able to handle the different variabilities making them not so effective for large databases. In contrast, we propose an indexing system for fast and robust matching and retrieval of shapes. We envisage a shape matching system which can efficiently scale to large databases without compromising on the retrieval performance obtained by the state-of-the-art shape matching algorithms.

We model a shape as a collection of landmark points arranged in a plane (2D) or in 3D space. In our approach, each shape is characterized by features that are used to index it to a table. The table is analogous to the inverted page table

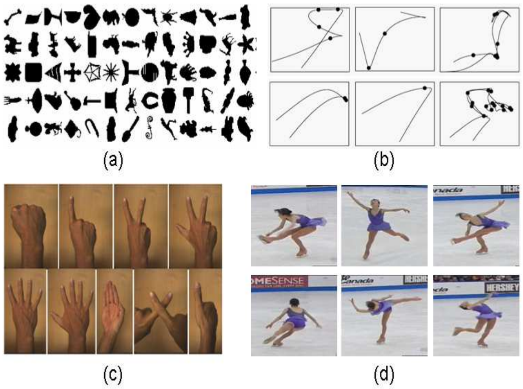


Figure 7.1: A few applications that can benefit from robust and efficient shape matching. (a) Matching and retrieval of 2D shapes [66], like digit recognition [68], trademark retrieval [18], leaf recognition [74]; (b) Activity Classification [102]; (c) Gesture recognition; (d) Pose estimation in sports clips [133].

used to index web pages using words/phrases. Given a test shape, similar ones from a pre-indexed collection are determined based on its characterizing features. The computational overhead (of establishing point-wise correspondence) involved in the traditional way of matching the query with each shape in the dataset is thereby avoided. As we deal with shapes, the only information usually available is the underlying geometry. Appropriate features are chosen to encode this geometry as richly as possible, without compromising on robustness. Quite clearly, the set of useful features vary depending on the particular application at hand. For example, invariance to articulations of part structures is very important in applications like gait-based human identification whereas the same feature is not desired for applications like retrieval based on human pose. Similarly, scale invariance may be critical for some application but detrimental to another. Our goal here is to develop a system that supports fast retrieval of shapes without needing any costly correspondence step during matching. To this end, we use (or propose) features that address most challenges faced by shape matching tasks including invariance to object translation, rotation, scale, articulations, etc. Depending on the demands of the application at hand, all or a subset of the proposed features are used for indexing and retrieval. A given shape is then represented using a collection of feature vectors, each characterizing a geometrical relationship between a pair of landmark points. The features should be easily computable for the matching algorithm to be efficient and to be able to scale up to large database sizes. The feature vectors are suitably quantized for indexing. The fact that feature vectors depend only on a few points and are quantized, provides the necessary robustness to the representation which is required to generalize across large intra-class variability. Since all the desired characteristics of a shape matching algorithm like invariance to rotation, articulation, etc., are incorporated in the feature vectors themselves, this kind of representation allows the proposed system to have a very simple and efficient retrieval scheme. Experimental results are provided to show the usefulness of the proposed approach.

The rest of the chapter is organized as follows: Section 7.1 introduces the indexing framework proposed in the chapter. Section 7.2 describes the feature based 2D shape representation along with the different features used. A detailed description of the indexing and retrieval algorithms is given in Section 7.3. Section 7.4 presents the results of extensive evaluations done to compare the proposed algorithm with others. Some real applications of shape matching are shown in Section 7.5. Finally we also discuss a implicit surface based approach for representation and matching of 3D range data in Section 7.6. The chapter concludes with a summary and discussion.

7.1 Indexing Framework - A Glance

Our focus here is to come up with a fast and efficient framework for shape indexing and retrieval that performs robust shape matching. In most approaches, given a query, it needs to be compared with every shape in the dataset to return the most similar ones. Comparisons often involve computationally demanding operations like registration, establishing correspondence, etc., which are repeated for each shape in the dataset. Such approaches are usually not scalable since the computational load can become prohibitively high as the size of the database increases.

In contrast, we propose a scalable and efficient shape matching and retrieval scheme. Figure 7.2 illustrates a prototype of our indexing framework. Here, a shape is represented using a set of indexable feature vectors which are appropriately mapped to a *hash table*. For a shape s_k, a bin i in the hash table stores an entry $\langle s_k, n_{ki} \rangle$, $n_{ki} > 0$ where n_{ki} is the number of feature vectors from shape s_k that get hashed to bin i. The hash table is populated by performing the operation for each shape in the database. The resulting table typically has several 2-tuples from different shapes in each bin. The quantization scheme determines how uniformly the entries are distributed across the hash table.

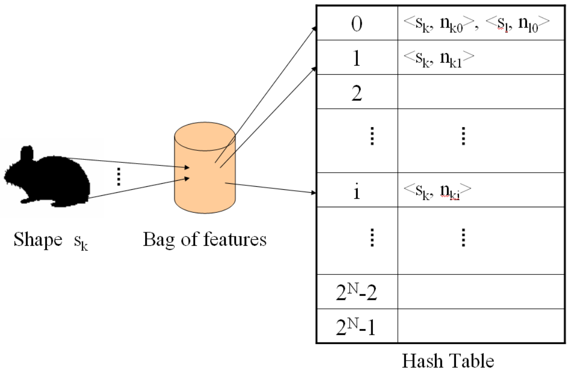


Figure 7.2: A prototype of the proposed shape indexing framework. Each shape in the database is indexed to a hash table using a set of indexable feature vectors extracted from the shape.

Given a test shape s_t, its feature vectors are extracted and its hash table entries $\langle s_t, n_{ti} \rangle$, $\forall n_{ti} > 0$ are determined by mapping the feature vectors to the table. Once this is done, its similarity with the shapes in the database can be estimated using a single parse through the matching bins. Parsing through the bins that contain a 2-tuple $\langle s_t, n_{ti} \rangle$, one can simultaneously compute the similarity of the query with all the shapes in the database. In such a retrieval scheme, the processing time depends only on the number of 2-tuples $\langle s_t, n_{ti} \rangle$ and the number of database entries in the matching bins. Quite clearly, the more uniformly distributed the hash table is, less is the average time required to process a query. Typically, the processing time increases much slowly as compared to the database size.

7.2 Feature based Shape Representation

In this section, we describe features which are invariant to different deformations like rigid transformations and articulations as required by the application at hand. The choice of features affects both the generalizability and discriminability of the approach. Therefore, we look for features that depend only on a few points on the shape and also take the global shape into account. The dependence on only a few points ensures robustness while their relative configuration with respect to the global shape provides discriminability.

Complexity of a typical matching algorithm depends on the complexity of the type of transformations that need to be handled which in turn depends on the application. In the proposed approach, we choose the features to be invariant to these transformations to make the retrieval process fast and efficient. Articulation of part structures being one of the most difficult kind of deformations addressed by several recent shape matching techniques, here we describe representative features that are invariant to articulations in addition to rigid transformations.

7.2.1 Pairwise Geometrical Features

Following these guidelines, each shape is characterized by a set of feature vectors where each vector encodes pairwise geometrical relationships on the shape. Each vector consists of the following features that are robust to different deformations.

7.2.1.1 Inner distance between the two points

The Euclidean distance between two interest points is invariant to rigid transformations of the shapes and is useful for applications where it is required to preserve articulation-dependent discriminability. But even small articulations can change the Euclidean distance significantly for several point-pairs on the shape. Therefore, for applications requiring invariance to articulations, we use the inner distance (ID) [74] which is robust to articulations of part structures. The inner distance between two points is the length of the shortest path within the silhouette of the shape. Figure 7.3 (Left) illustrates the difference of inner distance over the standard Euclidean one.

Computation of inner distance involves forming a graph with landmark points on the shape forming the nodes. Two nodes in this graph are connected if there is a straight line path between the corresponding points which is completely inside the shape contour. The corresponding edge weight is the Euclidean distance between the two. From this graph, any standard shortest path algorithm can be used to compute the inner-distance for all the unconnected nodes.

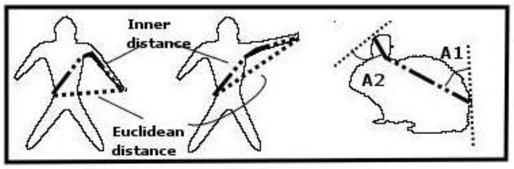


Figure 7.3: Inner distance and Relative angles. The two human silhouettes on the left show the insensitivity of inner distance with articulation of part structures.

7.2.1.2 Relative Angles

Relative angles (A1 and A2) encode the angular relationship between a pair of points. Absolute orientation of the line segment connecting the points is not invariant to rotations. Therefore, relative orientation of the connecting line segment with respect to the incident tangents at each end point is used. When using the inner distance, this becomes the relative orientation of the first segment of the path corresponding to the inner distance (Figure 7.3) (Right). The angles can be computed easily during inner distance computation.

7.2.1.3 Contour Distance

The contour distance (CD) is analogous to geodesic distance for 3D shapes. For 2D silhouettes, the contour distance between two points is simply the length of the contour between the two points. It captures the relative positions of the two points with respect to the entire shape contour. The distance is robust to both articulations and contour length preserving deformations. It complements inner distance in characterizing the relative location of the point pair with respect to the entire shape. Figure 7.4 shows the contour distance between two points of an object across several deformations. Though the contour distance may seem sensitive to missing points and outliers, we observe that quantization during indexing phase makes it reasonably robust.

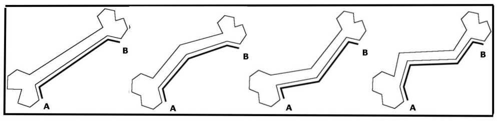


Figure 7.4: Contour Distance. The shown shapes illustrate the insensitivity of contour distance to length-preserving deformations.

7.2.1.4 Articulation-invariant Center of Mass

The features described so far depend on the entire shape, but none of them capture much information about the relative placement of various point pairs in the shape. Though robust, such a representation may not be able to provide the desired level of discriminability. To encode the relative placement, one can use the distance of the points and the line segment joining them from the center of mass as additional features where shapes need to be matched across rigid deformations. But clearly, these features are not invariant to articulations as the center of mass can change appreciably with articulations. Therefore, we propose an articulation-insensitive alternative to the traditional center of mass if invariance to articulation is required.

Here, we first describe how the location of articulation invariant center of mass is determined followed by a description of the features derived from it. Determining such a point directly is not easy. The proposed approach first transforms a given shape to an *articulation-invariant space*. All objects related by articulations of their part structures get transformed to the same shape in the new space. This essentially means that the distances between the transformed points are invariant to articulations. In other words, the Euclidean distances between transformed points should be the same as the inner distances in the original space.

The transformation is done using multi-dimensional scaling (MDS) [32]. MDS essentially places the points in a new Euclidean space such that the inter-point distances are as close as possible to the given inner distances in a collective manner. We use the classical MDS as opposed to other more accurate but iterative algorithms for efficiency. The transformation computation involves spectral decomposition of inner product matrix B, which is related to the (squared) inner-distance matrix $D_{n\times n}$ as follows

$$\begin{aligned} B &= -\frac{1}{2}JDJ \\ J &= I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \qquad \text{where, } \mathbf{1}_{1\times n} = [1, 1, \cdots, 1]^T \end{aligned} \tag{7.1}$$

The matrix B is symmetric, positive semidefinite and can be expressed as

$$B = V\Lambda V^T \qquad \text{where,} \qquad \Lambda = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_n) \tag{7.2}$$

The required transformed coordinates in an m-dimensional output space can be obtained by

$$X_{n\times m} = V_{n\times m}\Lambda^{\frac{1}{2}}_{m\times m} \tag{7.3}$$

Figure 7.5 shows the result of performing MDS on a few shapes. Here m is taken to be two for visualization. The approximation improves with the dimensionality of the output space. As expected, the transformed shapes look quite similar across articulations. The desired articulation-invariant center of mass is the center of mass of the transformed shape.

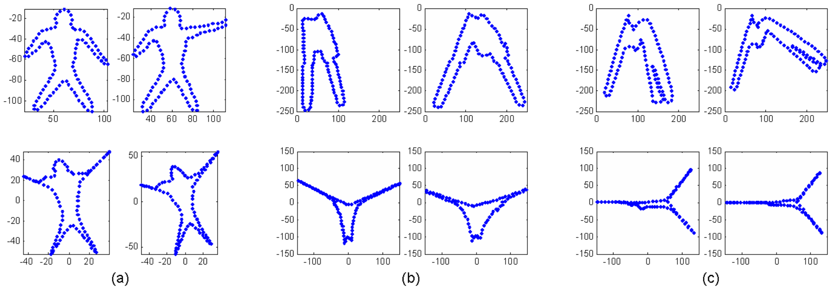


Figure 7.5: Articulation-invariant center of mass. Row 1: Original shapes, Row 2: Transformed shapes after MDS.

Given the articulation-invariant center of mass of a shape, we derive features which capture the relative positioning of the point pairs. For each point pair, distances (D1, D2, D3) of the points and the line segment joining them from the estimated center of mass are computed. This is done in the transformed space itself as the distances in the transformed space are insensitive to articulations.

7.2.2 Bag of Features

Given a shape, the pairwise geometrical features are computed for each pair of landmark points on the shape. Here, each point pair is characterized by a 7-dimensional feature vector (or less, depending on the invariant properties required for the application), comprising of the features described above. The distance based features in the vector are made robust to variations in scale by normalizing each with their medians. Note that here we provide a basic set of features that are robust to rigid transformations and articulations of part structures. The exact choice of the set of features may depend upon the semantics of the application at hand. The collection of such feature vectors for all pairs of landmark points characterize the shape.

It is worthwhile to note that though the various features are not entirely uncorrelated, they capture different characteristics of the shape. Even experiments show that each one of them contributes to the good performance of the system.

7.3 Indexing and Retrieval of Shapes

In this section, we describe how the proposed representation is used for shape indexing and retrieval. A shape is indexed by hashing each of its feature vectors to the index table. This requires discretization of the space of feature vectors. Here, we quantize each dimension of the vector independently using a suitably chosen number of levels for each. Suppose $\{f_1, f_2, \ldots, f_7\}$ denotes the 7-dimensional feature vector. The number of levels assigned to each feature is empirically chosen based on the robustness of the feature. If the number of quantization levels for feature f_i is given by 2^{N_i}, then N_i bits are required to represent the feature. So each feature vector consisting of 7 features is represented using $N = N_1 + \cdots + N_7$ number of bits. There are 2^N possible combinations of the feature vectors and hence any vector belongs to one of the $0, 1, 2, \ldots, (2^N - 1)$ bins in the hash table. Though the appropriate number of bits assigned to each feature may vary depending on the application, Table 7.1 shows the typical number of bits assigned to each feature in our system.

Table 7.1: Number of quantization bits for the used features.

ID	A1	A2	CD	D1	D2	D3
4	2	2	4	2	2	2

The quantization boundaries for each feature are chosen such that there are almost same number of entries in each level. This is done by using a set of training shapes which are representative of the database. In addition to being the basic requirement of an indexing system, quantization provides robustness to the variations in the actual values of the features across different instances of the same shape.

7.3.1 Indexing

Figure 7.2 illustrates the overall indexing procedure. The steps in the indexing are described below in detail.

1. For each shape in the database, landmark points are extracted from the shape contour. Though one can choose these points judicially, we simply pick points uniformly on the shape contours in all our experiments.

2. For each pair of landmark points, features are computed as described in the previous chapter. This results in a collection of feature vectors for each shape. If there are n landmark points, we have $\binom{n}{2}$ feature vectors.

3. Each feature vector is quantized using the proposed quantization scheme.

4. The quantized feature vectors are mapped on to the appropriate bins in the hash table. The i^{th} bin contains 2-tuples of the form $\langle s_k, n_{ki} \rangle \, \forall n_{ki} > 0$, where s_k is the k^{th} shape in the database and n_{ki} denotes the number of feature vectors of shape s_k that hash to bin i.

7.3.2 Retrieval

Given a query shape, the aim is to retrieve the similar shapes in the database as efficiently as possible. Figure 7.6 illustrates the retrieval phase using a flow chart. The different steps involved in the retrieval phase are enumerated below.

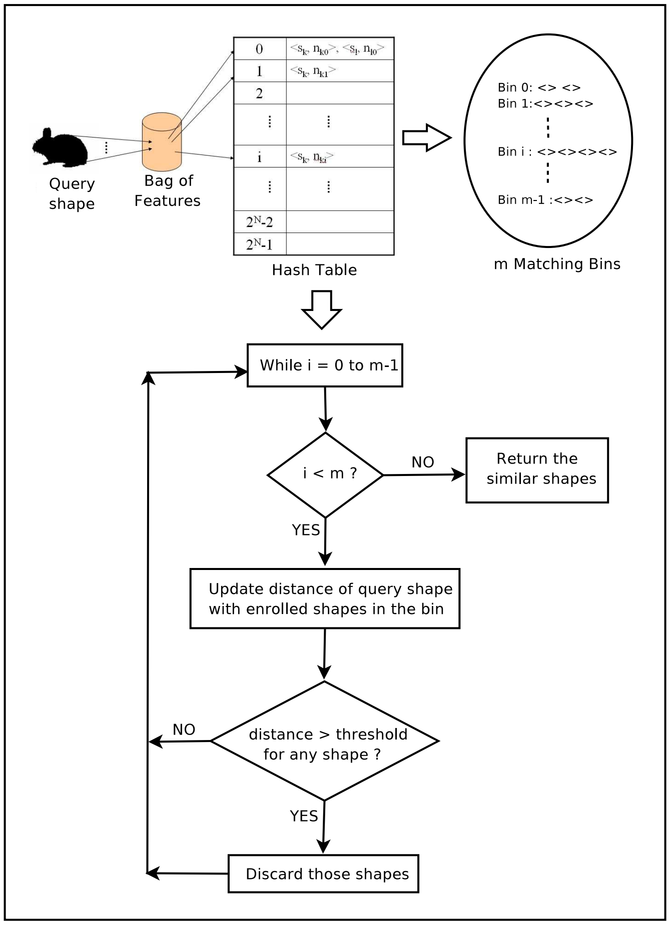


Figure 7.6: Retrieval Algorithm.

1. Feature vectors for the query shape s_t are extracted in a manner similar to the one used for indexing.

2. Each vector is quantized using the same quantization steps as used for the shapes enrolled in the database.

3. Hashing each feature vector to the index table results in a list of matching bins $M = \{i | n_{ti} > 0\}$, where n_{ti} is the number of query feature vectors which hash to bin i. In general, the number of matching bins is much less than the total number of bins in the hash table.

4. The distance $D(t,k)$ of the query s_t with each shape s_k in the database is initialized to zero.

5. Now we parse through the list M and update the distance of the query with each enrolled shape at every step using the following distance metric

$$D(t,k) = D(t,k) + \frac{1}{2}\frac{(n_{ti} - n_{ki})^2}{n_{ti} + n_{ki}} \tag{7.4}$$

   where the shape s_k has an entry $\langle s_k, n_{ki} \rangle$ in the i^{th} matching bin. If there is no such entry for a shape s_p in the bin, n_{pi} is taken to be zero. The choice of distance metric is inspired by the standard χ^2 statistic.

6. If during parsing, the distance for any particular shape in the database exceeds a pre-specified threshold, then that shape is discarded from further computation.

7. At the end of the parse, we get a list of shapes from the database which are most similar to the query shape.

7.3.3 Computational Complexity

The computational complexity of the indexing phase depends on the complexity of feature extraction. For a shape with n landmarks, the inner distance computation is of complexity $O(n^3)$. Computation of relative angles and contour distances take $O(n)$. The complexity of calculating the articulation invariant center of mass is $O(n^2)$ while deriving features based on it take $O(n)$. Therefore, indexing a shape takes $O(n^3)$. Note that indexing can be done off-line so that query processing time is not affected. To ensure fairness, all running times reported here include the time spent in indexing.

As in the indexing phase, for a query shape with n landmarks, feature extraction and hashing is $O(n^3)$. Hashing results in $m \ll \binom{n}{2}$ matching bins. Suppose each bin has $p \ll N$ entries, where N is the number of shapes in the database, we need to perform $O(pm)$ distance updates (Equation 7.4). This does not take into account the fact that a lot of shapes are discarded during retrieval which would further reduce the query processing time. It is difficult to put a bound on how large m and p can be. In the worst case, m can be as large as $\binom{n}{2}$ and p as large as N, but that does not happen in practice. With suitable quantization, p increases much slower than N. Moreover, if elimination of dissimilar shapes during retrieval process is taken into account, the complexity of the process depends on the number of those database shapes which are somewhat similar to the query. These attributes make the system quite scalable.

7.4 Experiments

In this section, we report the results of empirical evaluation of the proposed system. The performance of the system is compared with many state-of-the art matching algorithms on standard datasets. In addition, we highlight the computational advantages of our indexing approach. In the next section, we also perform experiments on human pose estimation and activity classification to further highlight the usefulness of the proposed framework for real world problems that involve large size databases. In all the experiments, we take 100 uniformly sampled points on the shape contour as landmarks.

7.4.1 MPEG7 Shape Dataset

As our focus is to show the efficiency of the proposed system along with its accuracy, we first test it on the MPEG7 CE-Shape-1 [66] dataset, which is the probably the largest benchmark used for evaluating shape matching algorithms. The dataset consists of 1400 silhouettes with 20 images each for 70 different objects. Figure 7.7 (Left) shows a few images from the dataset. The standard test for this dataset is the Bullseye test. It is a leave-one-out kind of test where 40 most similar shapes are determined for every query shape. The final score is given by the ratio of the number of correct hits to the best possible number of hits (20×1400).

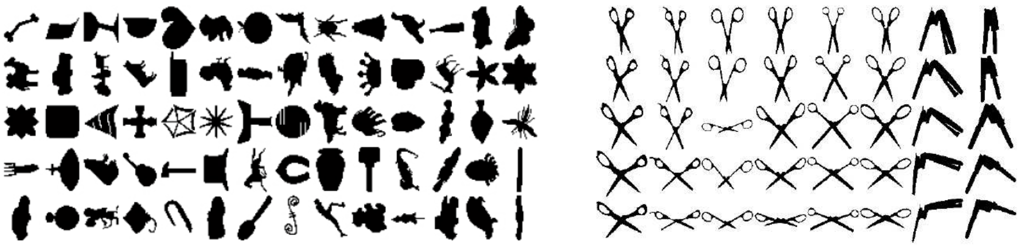


Figure 7.7: (Left) Example shapes from MPEG7 CE Shape 1 dataset [66]; (Right) Articulation database [74]: There are 8 objects with 5 shapes each. Each column in the figure shows the different articulated state of an object from the database.

Table 7.2 compares the performance and computation time of the proposed approach with many algorithms reported in the literature. In terms of accuracy, the proposed algorithm performs quite well, though the performance is not exactly at par with some of the very recently published approaches. On the other hand, as can be seen from Table 7.2, the proposed approach takes several order of magnitudes less time than other approaches. The system runs on a regular desktop and is implemented in MATLAB. The run-times reported for other algorithms are directly taken from the respective references and may vary slightly due to differences in machine configurations. The accuracy and computational time comparisons show that the proposed system achieves the original goal of developing a fast and efficient shape matching system that is scalable for large size datasets without comprising much on the accuracy.

7.4.2 Articulation Database

The features used in our framework were chosen so as to support articulation-invariant matching. Therefore, it is important to evaluate the performance of the system on a dataset which explicitly deals with large articulations. Here we

Table 7.2: Performance comparison on MPEG7 dataset. D_{sc}: shape context distance. DP: dynamic programming based matching.

Algorithm	Score	Computation Time
CSS [83]	75.44%	
Visual Parts [65] [66]	76.45%	
Curve Edit [110]	78.17%	$1s \times {}^{1400}C_2$ (50 segments)
Gen. Models [131]	80.03%	$0.2s \times {}^{1400}C_2$
SC + D_{sc} [74]	64.59%	
SC + TPS [18]	76.51%	$0.2s \times {}^{1400}C_2$
IDSC + D_{sc} [74]	68.83%	
IDSC + DP [74]	85.40%	$0.31s \times {}^{1400}C_2$
HPM-Fn [82]	86.35%	$0.1 - 0.2s \times {}^{1400}C_2$
Shape Tree [34]	87.70%	$0.5s \times {}^{1400}C_2$
Proposed	**81.8%**	**10 minutes**

use the articulation dataset introduced in [74] which consists of 8 objects with 5 shapes each as shown in Figure 7.7 (Right).

We use the same test scheme as in [74]. For each shape, 4 most similar shapes are selected and the number of correct hits for ranks 1, 2, 3 and 4 are calculated. Clearly, the best performance of any system possible is to get 40 correct matches at all the four ranks. Table 7.3 summarizes the results obtained. The proposed approach favorably compares with other approaches. It is noteworthy that unlike other approaches, our system does not require any alignment or costly matching for computing similarity with each shape in the dataset.

Since the proposed set of features are meant to be insensitive to articulations, we perform an analysis of the features on the articulation dataset. For this analysis, we divide the features into three sets namely, inner distance + relative angles, contour distance and articulation-invariant center of mass (AICM) based features. Table 7.4 summarizes the performance of these feature sets on the articulation dataset. Interestingly, even individually all three feature sets outperform the performance obtained using Shape Context based approach (Table 7.3).

Table 7.3: Retrieval result on the articulation dataset.

Algorithm	Rank 1	Rank 2	Rank 3	Rank 4
SC + DP [74]	20/40	10/40	11/40	5/40
IDSC + DP [74]	40/40	34/40	35/40	27/40
Proposed	**40/40**	**38/40**	**33/40**	**20/40**

Table 7.4: Analysis of the various features used on the articulation dataset.

Feature	Rank 1	Rank 2	Rank 3	Rank 4
Inner distance + Angles	33/40	33/40	21/40	11/40
Contour distance	36/40	31/40	28/40	23/40
AICM-based	25/40	12/40	9/40	11/40
All combined	**40/40**	**38/40**	**33/40**	**20/40**

7.4.3 Kimia Dataset 1 and 2

Kimia dataset 1 [112] (Figure 7.8 (a)) consists of 25 shapes from 5 categories. The experiment is run in a leave-one-out pattern. Similar to the articulation dataset, the performance is measured by accumulating the correct matches at ranks 1, 2 and 3. The best one can get at any rank is 25. Table 7.5 compares the results obtained with other approaches. The proposed indexing approach compares well with other approaches.

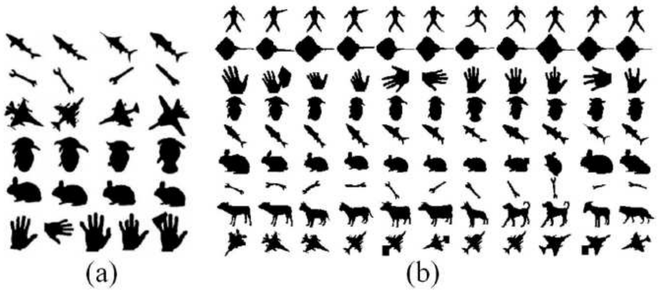


Figure 7.8: Kimia database. (a) Kimia dataset 1 [112] consisting of 25 shapes from 5 categoris, (b) Kimia dataset 2 [109] consisting of 99 silhouettes from 9 categories.

Kimia dataset 2 [109] (Figure 7.8 (b)) is a larger version of dataset 1. It consists of 99 silhouettes from 9 categories. The performance is measured by examining the correct matches at top 10 ranks for each query. The best one can get for each rank is 99. Table 7.6 summarizes the results obtained. In addition to being extremely efficient, the proposed approach compares favorably with many existing algorithms.

Table 7.5: Retrieval result on Kimia 1 dataset.

Algorithm	Rank 1	Rank 2	Rank 3
Sharvit *et al.* [112]	23/25	21/25	20/25
Gdalyahu *et al.* [39]	25/25	21/25	19/25
Belongie *et al.* [18]	25/25	24/25	22/25
IDSC + DP [74]	25/25	24/25	25/25
Proposed	**25/25**	**25/25**	**23/25**

Table 7.6: Retrieval result on Kimia 2 dataset.

	1	2	3	4	5	6	7	8	9	10
SC [18]	97	91	88	85	84	77	75	66	56	37
Gen. Models [131]	99	97	99	98	96	96	94	83	75	48
Shock Edit [109]	99	99	99	98	98	97	96	95	93	82
IDSC + DP [74]	99	99	99	98	98	97	97	98	94	79
Our Method	**99**	**97**	**98**	**96**	**97**	**97**	**96**	**91**	**83**	**75**

7.4.4 ETH-80 database

The ETH-80 database [72] contains a total of 80 objects, 10 each from 8 different categories (Figure 7.9). Each object is represented by 41 images taken from viewpoints spaced equally over the upper viewing hemisphere resulting in a total of 3280 images. We follow the intended testing protocol for the database which is leave-one-object-out cross-validation. Each image in the database is compared with all the images (all 41 views) from the other 79 objects and if the correct category label is assigned, the recognition is considered successful. The recognition rate is averaged over all the objects.

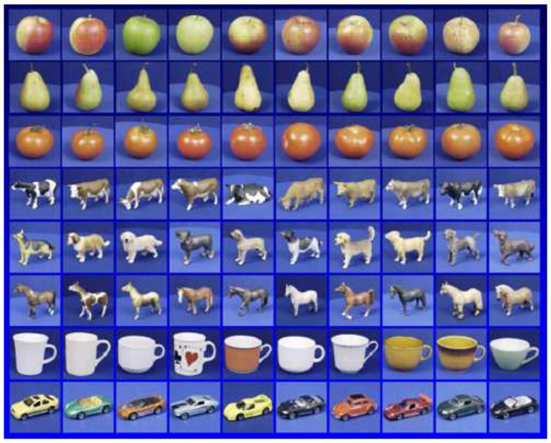


Figure 7.9: The 8 object categories of the ETH-80 database [72]. Each category contains 10 objects with 41 views per object.

Table 7.7 summarizes the results obtained. The approaches listed in the table use a single cue (either appearance or shape) for performing object recognition [72]. It is worthwhile to note that the best reported result on this dataset (to the best of our knowledge) is 93.02% which is obtained using a decision trees [72] based approach that combines the first seven approaches (i.e., combines multiple cues of shape, color, etc.) for better performance. We see that among the approaches which use only a single cue (as our approach uses only shape information), the proposed approach performs quite well and is only next to IDSC+DP.

Table 7.7: Recognition result on the ETH-80 image dataset. The first row show sthe different algorithms and the second row shows the recognition rate. All results other the proposed one are obtained from [72].

Color Hist	DxDy	Mag-Lap	PCA Masks	PCA Gray	SC Greedy	SC +DP	IDSC +DP	Proposed
64.85%	79.79%	82.23%	83.41%	82.29%	86.40%	86.40%	88.11%	87.48%

7.5 Applications

Efficient shape matching and retrieval is useful for many practical applications. Here, we describe two such applications, namely human pose estimation and activity classification.

7.5.1 Human Pose Estimation

Due to the easy availability of capturing and storage devices, large amounts of visual data is being captured. Data retrieval based on content rather than human annotation which might be absent or erroneous, has received much attention recently. The ability to automatically describe human activities in long video sequences is very useful for automatic video archiving, browsing and retrieval. Though motion is a very important cue, human activities in videos can often be described by the body pose in still frames [133]. In our context, human pose estimation implies matching the corresponding human silhouettes in the 2D images based on their body posture and not explicitly estimating the 3D pose. For efficient handling of large database sizes, the shape matching algorithms need to be able to scale up to large database sizes.

7.5.1.1 Evaluation Protocol

As the underlying pose space is continuous, so exemplars cannot be easily classified into positive and negative samples. Here, we use the same evaluation protocol as followed by Tresadern and Reid [128]. If the body joint locations are known, then for each query image I_q, the sum of squared errors between corresponding joint center projections in the image between the query image and each image I_t in the database are calculated. Let this distance in the pose space be denoted by $d(I_t, I_q)$. The database poses are then ranked in order of similarity to the query as determined by the shape descriptor. Let the index of the closest training example be $r(1)$ and the furthest be $r(N)$ where N is the number of images in the database.

The curve $f(k)$, given by

$$f(k) = \frac{\sum_{j=1}^{k} d(I_{r(j)}, I_q)}{k}. \tag{7.5}$$

represents the mean distance of the k highest ranking database examples to the query for $k = 1, \cdots, N$. Intuitively speaking, the function determines how well the ranking obtained using the shape descriptors correlates with the one given by joint locations. Clearly, for a good shape matching approach, the best ranking poses are closer to the query in joint locations space also. The exact shape of the ideal curve (perfect correlation) is determined by the discriminability across various poses in the joint locations space.

7.5.1.2 Experiments on MOCAP Data

We first evaluate the proposed shape indexing method using binary silhouettes of a human body model generated from motion capture data which contains information about the joint centers (http://mocap.cs.umd.edu). Figure 7.10 shows a few examples of binary silhouettes.

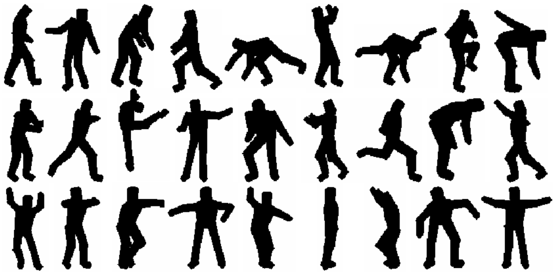


Figure 7.10: Example silhouettes from the CMU MOCAP dataset.

The training data consists of 1500 binary silhouettes of size 128×128 from different motions. The evaluation is performed on over 400 synthetically generated test silhouettes. The silhouettes generated from the synthetic data were automatically labeled with the image projections of the joint centers for evaluation.

Figure 7.11 shows the normalized curve of k/N against $f(k)/f(N)$ where N is the total number of training images. As mentioned earlier, the lower the curve is, the better is the performance.

To evaluate the effectiveness of the proposed approach for human pose estimation, we compare the results with two different approaches for shape representation which are briefly discussed below.

Lipschitz embeddings: Often computing the distance of the query image with all the images in the database is very costly. One solution is to embed the data objects in a vector space so that the distances of the embedded objects approximates the actual distances [47]. In Lipschitz embeddings, every image is represented by the vector of distances from a chosen set of *exemplars*. Intuitively, similar images will have similar distances to the chosen *exemplars* and thus have similar feature vectors. This global shape representation has been successfully used for estimating the 3D hand pose given an input image [10].

Histogram of Shape Contexts: Given an input silhouette, every contour point is represented by its shape context which characterizes the distribution of the other neighborhood contour points. The shape contexts of all the contour points are then clustered using k-means and the cluster centers are used as a vector quantization codebook for assigning each contour point on a given silhouette to a cluster. A histogram over cluster assignments forms the feature vector for a given silhouette. Recently this shape representation has been used for estimating the 3D pose of a human body from an input image [3].

Comparison of these approaches with the proposed approach for the task of human pose estimation is shown in Figure 7.11. We see that the performance of the proposed approach compares favorably with other shape descriptors. The dashed line at unity indicates the average curve produced by random ordering while the dash-dot curve indicates the best possible ranking where distance in image space correlates perfectly with distance in pose space.

Table 7.8 compares the time (in seconds) taken by the different methods for

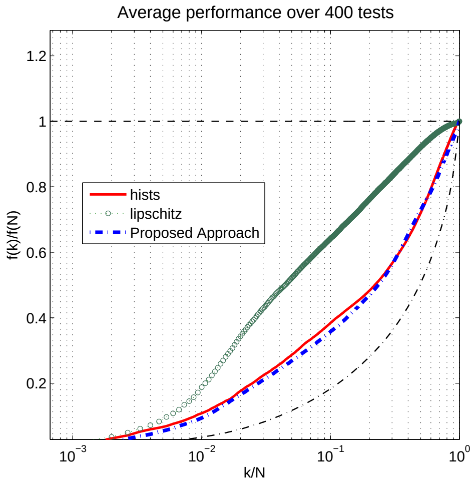


Figure 7.11: Evaluation of the proposed method for human pose estimation. Comparisons with (a) Lipschitz embeddings (lipschitz) and (b) Histogram of Shape Contexts (hists) are also shown. The dashed line at unity indicates the average curve produced by random ordering while the dash-dot curve indicates the best possible ranking where distance in image space correlates perfectly with distance in pose space.

the task of pose estimation in this experiment. We see that the proposed approach is much faster than Histogram of Shape Contexts but gives similar/better performance. Though our approach takes longer time than Lipschitz embeddings, the performance is significantly better.

Table 7.8: Time taken by different algorithms (in seconds) for human pose estimation.

Lipschitz	HoSC	Proposed Approach
101	985	393

7.5.1.3 Experiment on Figure Skating Data [133]

We also perform human pose estimation on a real figure skating dataset [133]. The data consists of figure skating videos. As expected, it is unconstrained and involves swift motion of the skater and real-world motion of the camera including pan, tilt and zoom making it very challenging. Figure 7.12 shows some representative frames from the dataset.

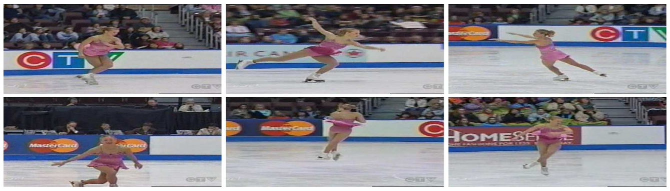


Figure 7.12: Sample frames from the Figure Skating Data [133].

In [133], Wang *et al.* address the problem of discovering actions performed by humans from still images in an unsupervised manner based on the body pose and they perform experiments on figure skating data. They propose a deformable template matching algorithm which tries to find an optimal assignment between sets of points sampled from the two images using a linear programming-based relaxation technique. This being a computationally expensive process, they employ a fast pruning method using shape contexts to handle large collection of images. But in such a two stage process, if similar images are falsely discarded in the initial pruning, those mistakes cannot be rectified in the later stage. Since the proposed indexing framework is meant for efficient matching in large databases, we evaluate it for the task of pose classification in this challenging dataset.

Low-level pre-processing of data: We first perform simple pre-processing of the raw video data to obtain the binary silhouettes of the skater. First the foreground pixels are separated from the background by building color models for both which is followed by median filtering to reject small isolated blobs. A bounding box is then fit to the foreground pixels by estimating the 2D mean and second order moments along x and y directions and the binary image of the pixels inside the bounding box is used as the input to our algorithm. Though the described pre-processing steps work reasonably well, the unconstrained nature of

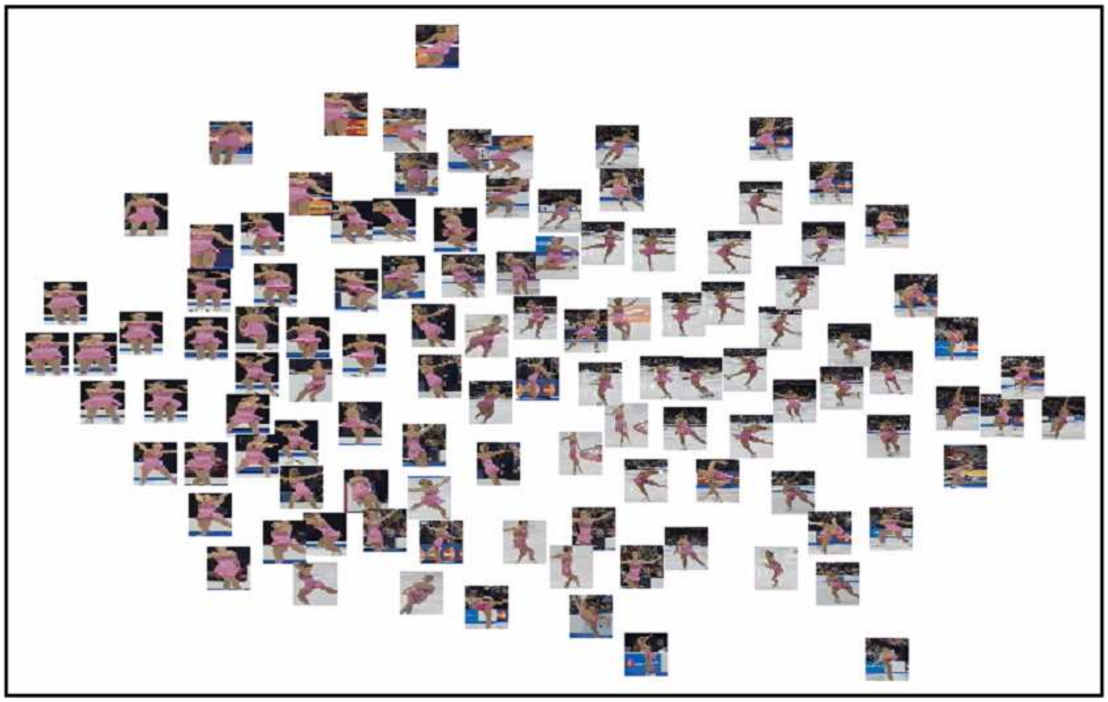


Figure 7.13: Visualization of similarity of the different poses of the skater using MDS. MDS places the input silhouettes in a new Euclidean space such that the inter-silhouette distances in the transformed space are as close to the distances obtained using the proposed shape matching approach. We see that similar poses appear closer to each other even after the dimensionality of the transformed space is reduced to two.

the input skating video presents numerous challenges like motion blur that cannot be handled using such simple heuristics. Therefore, the extracted silhouettes are noisy and present quite a challenge for any shape matching algorithm.

Matching result: We now evaluate our shape matching framework for the extracted noisy skating silhouettes. Since the pose space here is continuous, it is not straightforward to divide the data into separate classes and perform quantitative evaluation of the retrieval results. Here, we use MDS to analyze the effectiveness of the proposed method for representing the different poses of the skater. MDS places the input binary silhouettes in a new Euclidean space such that the inter-point distances (here each point represents an input silhouette) in the new space are as close to the inter-silhouette distances obtained using the proposed shape matching approach.

Note that to facilitate such a transformation, we need to have the inter-silhouette shape distances across all the used shapes. Therefore, the proposed retrieval process is performed without discarding any silhouette. Figure 7.13 shows the result of performing MDS on a subset of the figure skating data. Here the output space is taken to be two-dimensional for visualization purposes. The approximation obviously improves with the dimensionality of the output space. As desired, similar poses appear closer to one another and different poses appear farther apart in the transformed space.

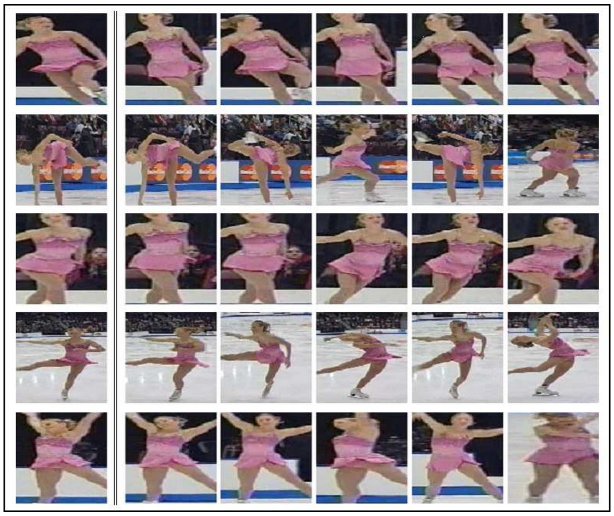


Figure 7.14: Image retrieval based on pose. First column shows query image. Second to sixth columns show the top 5 matches.

We also perform a retrieval experiment to retrieve similar poses from the database for qualitative evaluation. Figure 7.14 shows the top five matches for a few query images (shown in the first column). In the figure, other than for the second query, the algorithm successfully returns images having similar pose as in the query. These examples show the ability of the proposed framework to effectively match complicated shapes using noisy silhouettes extracted from real data.

7.5.2 Activity Classification

Though the proposed framework works quite well for matching human poses using silhouettes from a still image (or a video frame), depending on the application it may be necessary to analyze a sequence of contiguous frames jointly to infer the action/activity taking place in those frames. For example, considering a single frame of human walking sequence may not be sufficient to discriminate it against other similar activities like running, jumping, etc.

The goal of activity classification is to classify the content of human activity sequences in an unsupervised manner without any prior knowledge of the type of actions being performed. Many activity classification methods have addressed this task from a shape matching perspective [24] [21] [90] [137] [45]. Approaches based on key frames [24] or eigenshapes [44] of foreground silhouettes perform classification without taking the temporal information into account. Bobick and Davis [21] incorporate the temporal information in the form of temporal templates representing the motion properties at different spatial locations of an image sequence. Braided patterns extracted from a person walking in a direction orthogonal to the optical axis of the camera has been used by Niyogi and Adelson [90] for analyzing and recognizing walking figures. Recently, there has been tremendous interest to consider space-time shapes generated in the space-time volume by the performed activity [137] [45].

Here, we present a very simple approach to show the usefulness of the proposed indexing approach for the task of activity classification. In addition to analyzing the sequence of silhouettes to characterize the spatial information, we propose a novel *temporal shape* representation to capture temporal characteristics of the observed activity. Given an input silhouette sequence, a certain number of landmark frames containing both the spatial and temporal information is automatically extracted and analyzed using the indexing framework to recognize the activity in the sequence. Note that any method which transforms the activity classification task into a shape matching problem can benefit from the computational efficiency provided by our framework, irrespective of the exact form of representation. The following discussion provides the details of the approach and the results of the experiments performed for its evaluation.

Spatial characterization: Depending on the input video sequence, the foreground silhouettes are obtained using low-level image processing techniques. Temporal clustering is performed on these silhouettes to obtain N number of clusters based on the pose ($N = 5$ in our experiments). We use the distance transform to do the clustering. But they can be taken as key frames or any shape representations from the approaches which view activity classification as a shape matching problem. Temporal clustering results in N silhouettes which provide the spatial characterization of the sequence of foreground silhouettes.

Temporal characterization: The indexing approach presented is useful for efficient matching of shapes. In order to efficiently utilize the temporal information for activity classification, we transform it to another shape matching problem. As described before, an activity sequence can be represented using a 3D space-time volume. The silhouettes are essentially slices of this volume taken at different instances along the temporal axis. In a similar manner, one can slice the space-time 3D volume along one of the spatial axis (here y-axis) to obtain 2D space-time shapes which we call as *temporal shapes.* Similar to the temporal clustering of the silhouettes, spatial clustering is performed on these temporal shapes to obtain K ($K = 5$ in our experiments) key temporal shapes. Figure 7.15 shows the landmark silhouettes and temporal shapes for a few activities. From the figure, we see that this representation seems to contain discriminative information which can be utilized for classifying different activities.

Using such silhouette and temporal shape representation, each video sequence

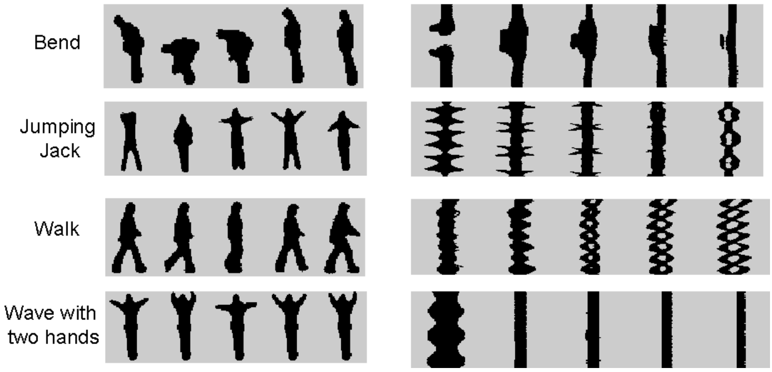


Figure 7.15: The silhouettes (first column) and temporal shapes (second column) for a few activities as chosen by our algorithm.

is represented with $(N + K)$ 2D shapes (N silhouettes and K temporal shapes). Note that these 2D shapes are ordered (in time and space, respectively). Each shape is then indexed based on the computed features, resulting in separate $(N + K)$ hash tables. During retrieval, each shape of the query video is used to retrieve similar shapes from the corresponding hash table in a manner similar to the one described in the previous sections. The similarity scores of the retrieved shapes are then fused in an additive manner to obtain the final similarity scores.

7.5.2.1 Experimental Evaluation

We evaluate the proposed approach on the activity dataset introduced in [45]. The dataset consists of 90 video sequences of nine different persons performing ten different activities, namely, run, walk, skip, jumping jack (or jack in short) jump forward on two legs (or jump in short), jump in place with two legs (pjump), gallop sideways (side), wave with two hands (wave2), wave with one hand (wave1) and bend. We follow a leave-one-out protocol as suggested in [45], i.e., for each query sequence , we remove the entire sequence from the database and compare it against the remaining 89 sequences. Table 7.9 shows the performance obtained in this experiment using the proposed spatial and temporal characterization of activity sequences. The performance is measured by verifying if the best match for each query sequence is from the same category or not. Clearly, the best performance possible is to get 9 correct matches in all the diagonal entries (as there are 9 instances per category that act as queries in a leave-one-out fashion). The performance is comparable to the approach in [45] which computes features from the complete space-time volume for classification.

Table 7.9: Activity classification performance obtained from silhouettes-based spatial and temporal characterization. The two numbers in each table entry shows the performance obtained using the proposed spatial and temporal characterizations, respectively.

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
bend (a1)	9/9	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
jack (a2)	0/0	8/9	0/0	0/0	0/0	0/0	0/0	0/0	0/0	1/0
jump (a3)	0/0	0/0	9/9	0/0	0/0	0/0	0/0	0/0	0/0	0/0
pjump (a4)	0/0	0/0	0/0	8/9	0/0	1/0	0/0	0/0	0/0	0/0
run (a5)	0/0	0/0	0/0	0/0	9/9	0/0	0/0	0/0	0/0	0/0
side (a6)	0/0	0/0	0/0	1/0	0/0	8/9	0/0	0/0	0/0	0/0
skip (a7)	0/0	0/0	1/0	0/0	0/1	0/0	8/7	0/1	0/0	0/0
walk (a8)	0/0	0/0	0/0	0/0	0/0	0/0	0/0	9/9	0/0	0/0
wave1 (a9)	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0	8/8	1/1
wave2 (a10)	0/0	0/1	0/0	0/0	0/0	0/0	0/0	0/0	1/1	8/7

7.6 Representation of 3D range data

In this section, we address the problem of representation of 3D range data. The representation should not only describe the object well but also generalize and extend seamlessly to perform robust recognition and registration. In addition, we also address the problem of registration and recognition of 3D point clouds i.e., given a pair of 3D point clouds we should be able to estimate the similarity of the two and also find the rigid transformation that relates the two in case they are reasonably similar. The aim is to develop an approach which does not rely on either knowing or estimating the point-to-point correspondence between the two sets. Also, the approach should be able to match and register point clouds separated by all possible rigid transformations and should degrade gracefully in the presence of outliers or in the event of missing data.

Shape histograms [8], shape distributions [93], Extended Gaussian Images [49], wavelets [38], higher order moments [33] etc. are a few of the descriptors which have been explored to describe 3D shapes. Other than a few histogram based features which are invariant to geometric transformations, the descriptors are normalized by using the center of mass for translation, standard deviation for scale and principal axes based alignment for rotation. Though translation and scale normalizations perform reasonably well, PCA-normalization falls short of providing a robust alignment [56]. To this end, Kazhdan et al. [56] propose a spherical harmonic representation of such descriptors to achieve rotation invariance. Here, we also use spherical harmonics for rotation invariance but we apply it on implicit values based feature vector which, as we show, is a more complete and robust representation.

The alternative approach involves explicitly solving for optimal transformation using registration methods like Iterative Closest Point Matching (ICP) [19] [141], Generalized Hough Transform [13], Geometric Hashing [61], etc. before computing the similarity of the models. Such approaches can be quite inefficient in a database retrieval kind of application as one will need to register every query model with all the models in the database (assuming the algorithm is able to register models correctly across large transformations).

Given the advantages implicit representation provides, implicit surface generation has been an important area of research in Computer Graphics. The book by Bloomenthal et al. [54] provides an excellent overview of the area. Most of the methods define implicit surfaces in the form of quadrics, blobs or radial basis functions around the input 3D points. Almost all of them assume that a polygonal mesh connecting the 3D point cloud is given as input. In contrast, we generate implicit surface using only 3D point clouds. The parameters of the implicit surface can be obtained by solving a set of linear equations which can be used to perform robust recognition and registration. To perform object matching across affine transformations, spherical harmonic decomposition is used to achieve rotation invariance. Non-linear optimization is required to explicitly solve for the transformation parameters.

7.6.1 Implicit representation of a surface

We use implicit surfaces based on a variational interpolation technique which is a generalization of thin-plate interpolation. The method is similar to the one proposed by Yngve et al. [37] with two main differences:

1. We do not use the polygonal mesh information and generate implicit surfaces using just the 3D point cloud.

2. We use uniformly sampled points on concentric spheres as the pivot points instead of choosing them adaptively in an iterative fashion as done in [37].

This provides us with a globally unique representation of the object. The drawback is that this can prevent us from getting a very precise representation of the object but that is not the goal here as opposed to [37] where accurate reconstruction is the main objective. The approximate isosurface, we obtain, helps in generalization while not losing discriminable characteristics. These are the kind of properties one desires from a representation for the task of matching.

Generation of variational implicit surfaces involves solving a scattered data interpolation problem [132]. To create a variational implicit function, one needs to choose a certain number of constraint points $\{x_1, x_2, \ldots, x_n\}$, along with a set of implicit function values $\{h_1, h_2, \ldots, h_n\}$ at the given constraint positions. Typically, there are three types of constraints. *Boundary constraints* are those constraint points which lie on the surface and take the value zero. The *interior constraints* lie inside the surface represented by the point cloud and are given positive values. The *exterior constraints* lie outside the surface and are assigned negative values. We use an implicit function of the form:

$$f(x) = \sum_{j=1}^{k} d_j \phi(x - c_j) + P(x) \tag{7.6}$$

Here c_j are the locations of the pivots, d_j are the weights which we need to estimate, $P(x)$ is a first degree polynomial to account for the linear and constant portions of the implicit function f.

As we deal with point sets which represent object surfaces, the implicit functions should be chosen to make the surfaces reasonably smooth. The smoothness is also useful in making the representation fairly robust to outliers. Therefore, we take $\phi(x) =\parallel x \parallel^3$ as this function minimizes the curvature functional $\int_{x\epsilon\Omega} \sum_{i,j} \left(\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right)^2 dx$ [30]. Given a set of 3D points, we solve for the weights d_j and coefficients of the polynomial $P(x)$ using the following linear constraints:

$$h_i = \sum_{j=1}^{k} d_j \phi(x_i - c_j) + P(x_i) \tag{7.7}$$

These equations being linear with respect to d_j and the coefficients of $P(x)$, can be formulated as

$$\begin{pmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1k} & 1 & x_1^x & x_1^y & x_1^z \\ \phi_{21} & \phi_{22} & \dots & \phi_{2k} & 1 & x_2^x & x_2^y & x_2^z \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{n1} & \phi_{n2} & \dots & \phi_{nk} & 1 & x_n^x & x_n^y & x_n^z \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ c_1^x & c_2^x & \dots & c_k^x & 0 & 0 & 0 & 0 \\ c_1^y & c_2^y & \dots & c_k^y & 0 & 0 & 0 & 0 \\ c_1^z & c_2^z & \dots & c_k^z & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \\ p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{7.8}$$

The pivot points c_j are sampled uniformly on a bunch of concentric spheres around the center of mass of the object. As they are fixed irrespective of the object, we get a globally unique and compact representation of the 3D point cloud in the form of the parameter vector (containing d's and p's) obtained by solving the system in (7.8).

We use all the input 3D points to generate linear constraints of the form (7.7) with zero as the implicit function value. As we do not use the polygon information, it is not easy to identify the points which lie inside the object with certainty (if the object is not convex). In comparison, choosing exterior points is much easier even without any polygonal information. We envelop the point cloud with a tight fitting ellipsoid with the axes of the ellipsoid aligned in the direction of the principal components of the distribution of the 3D points. Points are sampled on the enveloping ellipsoid to get the exterior constraints. The points on the ellipsoid which lie inside the convex hull of the 3D point cloud are not considered. The negative of the distance of each exterior point from the closest point in the point cloud is assigned as the implicit function value to get the linear constraints of the form (7.7).

Though one can come up with more complex and iterative strategies [37] to get a better reconstruction but as mentioned before that is not our goal. We aim at generating a smooth and approximate isosurface for a given object which is representative of its class. Figure 7.16 shows a few isosurfaces generated using this approach. It is worthwhile to note that only point cloud information is used to generate these surfaces. In contrast, most state-of-the-art graphics approaches use polygonal or volumetric information to generate isosurfaces.

7.6.2 Rotation invariant representation

Though the choice of the same set of pivots to model any point cloud provides us with a unique representation, the estimated parameters are not invariant to similarity transformations of the object. Thus two point clouds cannot be directly compared based on their estimated parameter values.

Spherical harmonic decomposition of a spherical function provides a very sim-

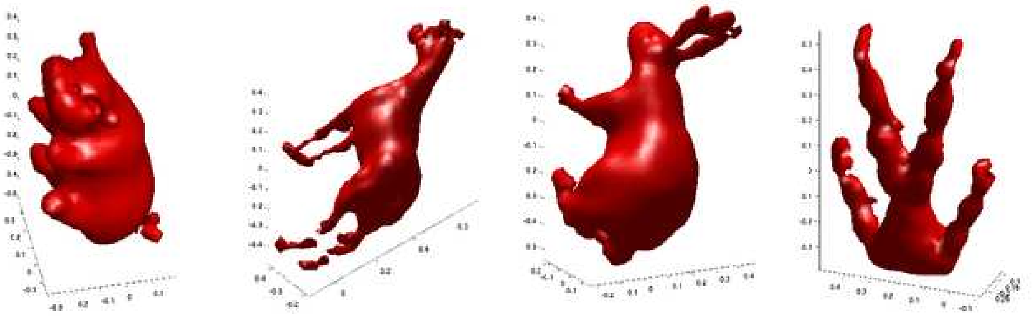


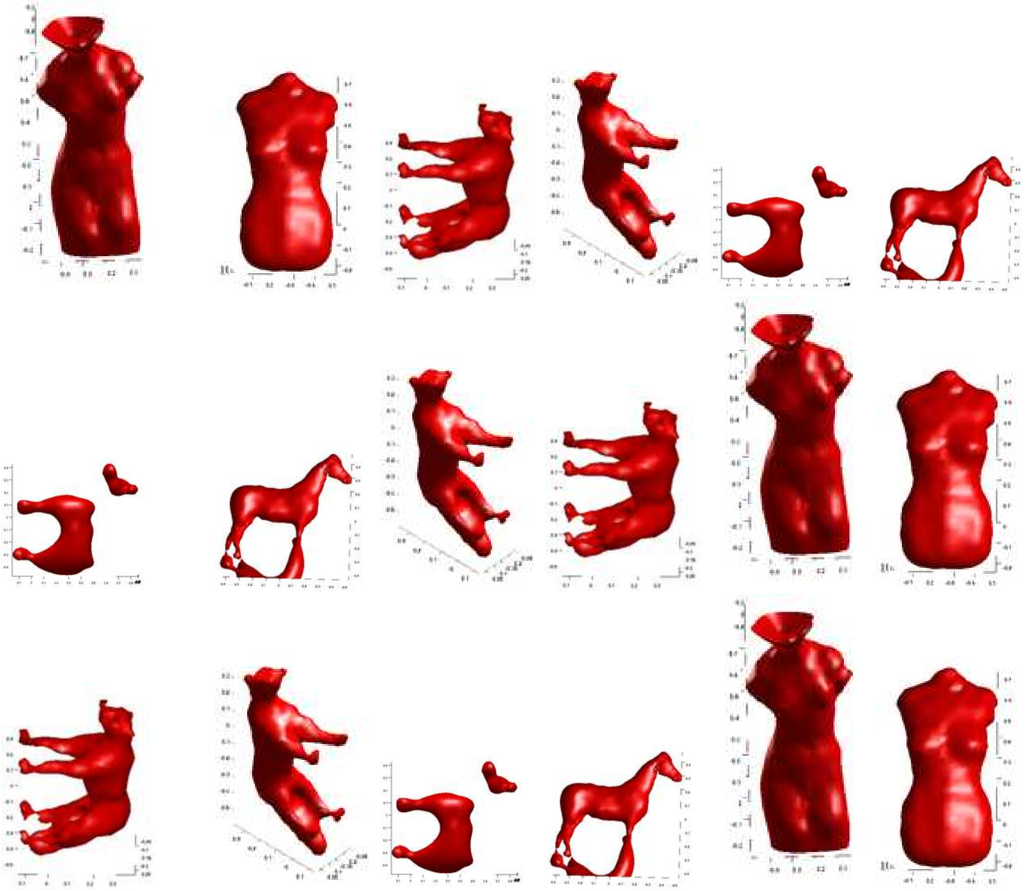
Figure 7.16: A few examples of the generated isosurfaces. All the surfaces were generated using just 500 pivot points.

ple and efficient way to obtain a rotation invariant representation (though not lossless) of the function. The proposed approach provides a very efficient method to generate approximate smooth isosurfaces of given 3D point clouds which as shown in Figure 7.16, can help generalize without losing discriminability. Therefore, instead of extracting some generic feature and using spherical harmonic transformation, we intend to use the isosurfaces generated from 3D point clouds to obtain the rotation-invariant descriptor. To this end, we propose using implicit function values as the spherical feature. We compute implicit function values using (7.6) at uniformly sampled points on a set of concentric spheres around the object. The spherical harmonic decomposition of these implicit values based spherical functions is then computed for each sphere to get the rotation invariant signature as described below.

If the implicit values based function can be represented in terms of spherical harmonics as,

$$f(\theta,\phi) = \sum_{l=0}^{k} \sum_{m=-l}^{l} a_{lm} Y_l^m(\theta,\phi) \tag{7.9}$$

then the norms, $\sqrt{\sum_{-l \le m \le l} |a_{lm}|^2} \quad \forall \quad 0 \le l < k$, are invariant to rotation. These rotation invariant energies at different harmonic levels for all the concentric spheres are used to form the desired rotation invariant feature vector. Quite clearly, larger the value of k is, the better the function can be represented using the spherical harmonics. In our experiments, we set k to 64. Figure 7.17 shows the ability of the proposed feature vector to measure similarity of various models (from The Princeton Shape Benchmark [113]) reliably across changes in scale and rotation.



The proposed rotation invariant descriptor helps in matching 3D point clouds across arbitrary rotations.

Figure 7.17: The figure displays the ability of the proposed feature vector to match objects across geometric transformations. In each row, the models are arranged in the order of decreasing estimated similarity w.r.t. the leftmost model in the row.

7.6.3 Estimation of the rigid transformation using isosurfaces

In this section, we will show how to estimate the underlying rotation between two reasonably similar objects using their generated isosurfaces. Spherical harmonic decomposition for handling rotation in 3D is analogous to that of Fourier decomposition for handling translation. Though the phase information in Fourier decomposition is useful in estimating the translation, spherical harmonics can not directly be used for estimating the underlying rotation.

The approach makes use of the intuition that correct rotation will make a point cloud satisfy the implicit function (Equation (7.6)) of the other at most points and vice-versa i.e., $R' = R$ minimizes the following implicit function value for all points if the correct underlying rotation is R:

$$f(x_i^R) = \sum_{j=1}^{k} d_j \phi(x_i^R - R^{'}(c_j)) + P((R^{'})^{-1}(x_i^R)) \tag{7.10}$$

where x_i^R are the points of the rotated object, c_j are the pivots of the base object while d_j and P are the estimated isosurface parameters of the base object and R' is the rotation matrix corresponding to the hypothesized rotation. The optimization to obtain the optimal R is done using *lsqnonlin* function in MATLAB. Translation and scale variations are taken care of using the traditional normalizations. Figure 7.18 shows an example of registering two animals using this approach. It is worthwhile to note that the method is able to cope with small mis-alignment of the center-of-mass of the objects as can be seen in the shown example. In addition, unlike ICP [19], the proposed approach performs registration without explicitly solving for correspondence.

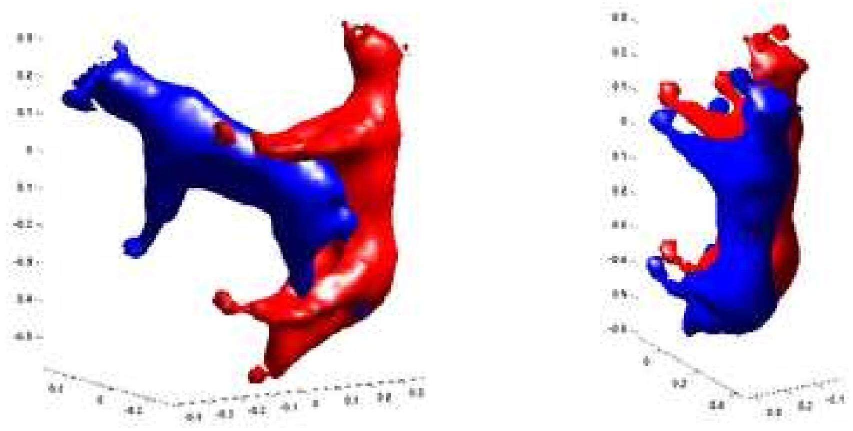


Figure 7.18: Registration result: The figure shows two models before and after registration using the proposed approach.

7.7 Summary and Discussion

We presented an efficient and robust approach for fast matching and retrieval of shapes. In most existing techniques, the alignment process has to be repeated for every shape in the database for retrieval, making them much slower than the proposed scheme. As dissimilar shapes are eliminated very early during our retrieval process, little effort is wasted in comparing a query to the database shapes which are very different, making the system scalable. The extensive experimental evaluations performed illustrate the effectiveness of the proposed indexing framework. Due to increase in the amount of data to be handled, most real-life applications require efficient algorithms which can scale upto large size databases. The results obtained are extremely promising and make a strong case for such an efficient indexing based framework for shape matching. We also presented an isosurface-based approach for representation and matching of 3D range data without requiring correspondence establishment.

Chapter 8

Directions for Future Research

Though the proposed methods work quite well for the problems they address, a lot more needs to be done to apply the algorithms for more general objects and scenes. The problems addressed in this dissertation and the different approaches proposed leads to many interesting future directions of research. We will discuss some of these research directions in this chapter.

8.1 Face recognition in challenging scenarios

In this dissertation, we have proposed a model based approach to estimate the albedo of an object for matching objects across illumination and pose variations. Other than variations in illumination and view-point, depending on the imaging environment, there can be other external factors which result in deterioration of the image quality. Blur, low resolution, occlusion in face images are often encountered in many scenarios, specially for images captured from a large distance. The proposed approach uses an image estimation formulation to obtain the albedo estimate. The formulation can be extended to account for factors such as blur in the input image. To account for large occlusions, recently Wright *et al* [134] proposed a sparse representation for faces. In their approach, each test face image is written as a linear combination of the face images of the same person in the gallery and illumination variations is handled by including multiple images under different illumination conditions in the gallery. It has been shown [70] that an image of an arbitrarily illuminated face can be approximated by a linear combination of the images of the same face, illuminated by nine different light sources placed at pre-selected positions. Since the estimated albedo can be used along with average facial surface normal to generate images under different illuminations, these relighted images can be used along with the sparse representation to handle large occlusions in addition to illumination variations with one a single image in the gallery.

8.2 Illumination invariant tracking and recognition in video

In the dissertation, a single image was used for estimating the albedo of the object. Since the accuracy of the initial albedo estimate depends on the angle between the surface normal and the light source direction, if multiple images under different illumination conditions are present, they can be suitably combined to obtain a more accurate albedo estimate than what can be obtained using either one of them. This is particularly important for shadow pixels and also finds natural use in video based recognition where multiple frames are present, usually under varying illumination conditions.

Many existing video-based face recognition methods [146] use very simple intensity normalization techniques to account for the illumination variations across frames. Though these measures work for simple cases, such simple techniques may not be adequate in more complicated realistic scenarios when there may be large illumination variation across the frames. Failure to handle these variations can lead to poor tracking performance which in turn will lead to poor recognition results. Using the albedo estimation technique from multiple images, we propose an algorithm for illumination-invariant tracking and recognition of faces for video.

- The input to the algorithm is the video sequence, average surface normal and the pose in the first frame.

- The albedo is estimated from the first frame of the video sequence.

- The face is tracked to the next frame. The albedo from the current frame is estimated and the correctness of the tracking result is evaluated by comparing the albedo obtained with that of the previous frames.

- The albedo estimate is updated using the multiple image based formulation and the sequence is repeated for each frame in the video.

The proposed method will probably be capable of dealing with large illumination changes since it is based on the albedo which is a intrinsic property of the object. Also use of 3D information for tracking will make it possible for tracking across large pose changes.

8.3 Hashing 3D shapes for fast matching and retrieval

We have described an approach for efficient articulation-invariant indexing and retrieval of 2D contours. This can be directly extended for 3D range data. Using line segments to model geometry of 3D shapes may not be rich enough for efficient and discriminative characterization. One way of making the representation rich is to use planes or tetrahedron as basis geometric entity.

Another way of characterizing 3D shapes is by representing them as a union of 2D curves on the shape. Samir *et al.* [107] represent 3D surfaces using union of level curves for the task of three-dimensional face recognition. Such an approach essentially reduces the problem of matching 3D shapes to one of matching multiple 2D contours which can easily be modeled using the approach presented in the chapter. The ideas from 2D shape matching and implicit surface matching can also probably be combined for fast object matching under articulations which is scalable for large databases.

Bibliography

[1] "Fg-net aging database: Face and gesture recognition working group," http://www-prima.inrialpes.fr/FGnet/.

[2] J. F. Abramatic and L. M. Silverman, "Nonlinear restoration of noisy images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 141–149, March 1982.

[3] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.

[4] G. Aggarwal and R. Chellappa, "Face recognition in the presence of multiple light sources," in *International Conference on Computer Vision*, 2005, pp. 1169–1176.

[5] T. Ahonen, A. Hadid, and M. Pietikinen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[6] G. L. Anderson and A. N. Netravali, "Image restoration based on subjective criterion," *IEEE Transactions on Systems, Man and Cybernatics*, vol. SMC-6, pp. 845–853, Dec 1976.

[7] H. C. Andrews and B. R. Hunt, *Digital Image Restoration.* Prentice-Hall signal processing series, 1977.

[8] M. Ankerst, G. Kastenmuller, H. P. Kriegel, and T. Seidl, "3d shape histograms for similarity search and classification in spatial databases." in *SSD*, 1999.

[9] A. Ashraf, S. Lucey, and T. Chen, "Learning patch correspondences for improved viewpoint invariant face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[10] V. Athitsos and S. Sclaroff, "Estimating 3d hand pose from a cluttered image," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 432–439.

[11] J. J. Atick, P. A. Griffin, and A. N. Redlich, "Statistical approach to sfs: Reconstruction of 3d face surfaces from single 2d images," *Neural Computation*, vol. 8, pp. 1321–1340, 1996.

[12] R. Bajcsy, S. Lee, and A. Leonardis, "Detection of diffuse and specular interface reflections and inter-reflections by color image segmentation," *IJCV*, vol. 17, no. 3, pp. 241–272, March 1996.

[13] D. Ballard, "Generalized hough transform to detect arbitrary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 111–122, 1981.

[14] K. Barnard, B. Funt, and C. Vlad, "A comparison of computational color constancy algorithms, part one; theory and experiments with synthetic data," *IP*, vol. 11, no. 9, pp. 972–984, September 2002.

[15] K. Barnard, L. Martin, A. Coath, and B. Funt, "A comparison of computational color constancy algorithms, part 2; experiments with images," *IP*, vol. 11, no. 9, pp. 985–996, September 2002.

[16] K. Barnard, L. Martin, B. Funt, and A. Coath, "A data set for color research," *Color Research and Application*, vol. 27, no. 3, pp. 148–152, 2002.

[17] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 218–233, 2003.

[18] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[19] P. Besl and N. McKay, “A method for registration of 3d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 239–256, 1992.

[20] V. Blanz and T. Vetter, “Face recognition based on fitting a 3d morphable model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, Sep 2003.

[21] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[22] M. J. Brooks and B. K. P. Horn, “Shape and source from shading,” in *Proceedings of International Joint Conference on Artificial Intelligence*, Aug 1985, pp. 932–936.

[23] D. Burt and D. Perrett, “Perception of age in adult caucasian male faces: computer graphic manipulation of shape and colour information,” *Proceedings of the Royal Society of London B*, vol. B-259, pp. 137–143, 1995.

[24] S. Carlsson and J. Sullivan, “Action recognition by shape matching to key frames,” in *Proceedings of the IEEE*, 2001.

[25] C. Castillo and D. Jacobs, “Using stereo matching for 2-d face recognition across pose,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[26] H. Chui and A. Rangarajan, “A new point matching algorithm for non-rigid registration,” *Computer Vision and Image Understanding*, vol. 89, pp. 114–141, 2003.

[27] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[28] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. MIT Press, Cambridge, Mass., 2001.

[29] R. Dovgard and R. Basri, "Statistical symmetric shape from shading for 3d structure recovery of faces," in *European Conference on Computer Vision*, 2004.

[30] J. Duchon, "Spline minimizing rotation-invariant semi-norms in sobolev spaces," *Lecture Notes in Math*, vol. 571, pp. 85–100, 1976.

[31] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley-Interscience Publication, 2000.

[32] A. Elad and R. Kimmel, "On bending invariant signatures for surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1285–1295, 2003.

[33] M. Elad, A. Tal, and S. Ar, "Content based retrieval of vrml objects: an iterative and interactive approach," in *Proceedings of the sixth Eurographics workshop on Multimedia 2001*, 2002, pp. 97–108.

[34] P. Felzenszwalb and J. Schwartz, "Hierarchical matching of deformable shapes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[35] G. D. Finlayson and G. Schaefer, "Convex and non-convex illuminant constraints for dichromatic colour constancy," in *CVPR*, 2001.

[36] ——, "Solving for colour constancy using a constrained dichromatic reflection model," *IJCV*, vol. 42, no. 3, pp. 127–144, 2001.

[37] Y. G. and G. Turk, "Robust creation of implicit surfaces from polygonal meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 4, pp. 346–359, October-December 2002.

[38] J. Gain and J. Scott, "Fast polygon mesh querying by example," in *SIGGRAPH '99: ACM SIGGRAPH 99 Conference abstracts and applications*, 1999.

[39] Y. Gdalyahu and D. Weinshall, "Flexible syntactic matching of curves and its applications to automatic hierarchical classification of silhouettes," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 12, pp. 1312–1328, 1999.

[40] X. Geng, Z. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2234–2240, December 2007.

[41] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, June 2001.

[42] R. S. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameter clustering," *Computational Science and Engineering*, vol. 4, no. 4, pp. 42–49, 1997.

[43] J. Geusebroek, G. Burghouts, and A. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, January 2005.

[44] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Behavior classification by eigendecomposition of periodic motions," *Pattern Recognition*, vol. 38, no. 7, pp. 1033–1043, 2005.

[45] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[46] A. B. Hamza and H. Krim, "Geodesic object representation and recognition," in *DGCI, LNCS 2886*, 2003, pp. 378–387.

[47] G. R. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 530–549, 2003.

[48] B. K. P. Horn and M. J. Brooks, *Shape from Shading.* Cambridge Massachusetts: MIT Press, 1989.

[49] B. Horn, “Extended gaussian images,” *PIEEE*, vol. 72, no. 12, pp. 1656–1678, December 1984.

[50] B. R. Hunt and T. M. Cannon, “Nonstationary assumptions for gaussian models of images,” *IEEE Transactions on Systems, Man and Cybernatics*, vol. SMC-6, pp. 876–881, Dec 1976.

[51] A. Hurlbert, “Formal connections between lightness algorithms,” *Journal of the Optical Society of America A*, vol. 3, no. 10, pp. 1684–1693, October 1986.

[52] H. K. Hussein, “Towards realistic facial modeling and re-rendering of human skin aging animation,” in *Proceedings of IEEE International Conference on Shape Modeling International*, 2002, pp. 205–212.

[53] C. Y. Ip, D. Lapadat, L. Sieger, and W. C. Regli, “Using shape distributions to compare solid models,” in *ACM symposium on Solid modeling and applications*, 2002, pp. 273–280.

[54] J. Bloomenthal, ed. Morgan Kauffman, “Introduction to implicit surfaces,” 1997.

[55] T. Kanade and A. Yamada, “Multi-subregion based probabilistic approach toward pose-invariant face recognition,” in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2003, pp. 954–959.

[56] M. Kazhdan, T. Funkhouser, and R. S., “Rotation invariant spherical harmonic representation of 3d shape descriptors,” in *Eurographics Symposium on Geometry Processing*, 2003.

[57] G. Klinker, S. Shafer, and T. Kanade, “The measurement of highlights in color images,” *IJCV*, vol. 2, no. 1, pp. 7–32, 1988.

[58] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel, “Adaptive noise smoothing filter for images with signal-dependent noise,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 165–177, March 1985.

[59] Y. H. Kwon and N. da Vitoria Lobo, "Age classification from facial images," *Computer Vision and Image Understanding*, vol. 74, no. 1, pp. 1–21, 1999.

[60] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of textured-mapped 3d models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, April 2000.

[61] Y. Lamdan and H. Wolfson, "Geometric hashing: A general and efficient model-based recognition scheme," 1988, pp. 238–249.

[62] Y. Lamdan and H. J. Wolfson, "Geometric hashing:a general and efficient model-based recognition scheme," in *Proceedings of International Conference on Computer Vision*, 1988, pp. 238–249.

[63] A. Lanitis, C. Draganova, and C. Christodoulou, "Comparing different classifiers for automatic age estimation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 1, pp. 621–628, February 2004.

[64] A. Lanitis, C. Taylor, and T. Cootes, "Toward automatic simulation of aging effects on face images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 442–455, April 2002.

[65] L. J. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1185–1190, 2000.

[66] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 424–429.

[67] D. S. Lebedev and L. I. Mirkin, "Smoothing of two-dimensional images using the 'composite' model of a fragment," *Iconics-Digital Holography-Image Processing*, pp. 57–62, 1975.

[68] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[69] H. Lee, "Method for computing the scene-illuminant chromaticity from specular highlights," *Journal of the Optical Society of America A*, vol. 3, no. 10, pp. 1694–1699, 1986.

[70] K. C. Lee, J. Ho, and D. Kriegman, "Nine points of light: acquiring subspaces for face recognition under variable lighting," in *IEEE Conference on Computer Vision and Pattern Recognition*, December 2001, pp. 519–526.

[71] K. C. Lee and B. Moghaddam, "A practical face relighting method for directional lighting normalization," in *International Workshop on Analysis and Modeling of Faces and Gestures*, Oct 2005.

[72] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[73] A. Li, S. Shan, X. Chen, and W. Gao, "Maximizing intra-individual correlations for face recognition across pose differences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[74] H. Ling and D. W. Jacobs, "Using the inner-distance for classification of articulated shapes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 719–726.

[75] H. Ling, S. Soatto, N. Ramanathan, and D. Jacobs, "A study of face recognition as people age," in *Proceedings of IEEE International Conference on Computer Vision*, 2007.

[76] X. Liu and T. Chen, "Pose-robust face recognition using geometry assisted probabilistic modeling," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 502–509.

[77] Y. Liu, H. Zha, and H. Qin, "The generalized shape distributions for shape matching and analysis," in *International Conference on Shape Modeling and Applications*, 2002.

[78] S. P. Mallick, T. Zickler, K. D., and P. N. Belhumeur, "Specularity removal in images and videos: A pde approach," in *ECCV*, May 2006.

[79] S. P. Mallick, T. Zickler, D. Kriegman, and P. N. Belhumeur, “Beyond lambert: reconstructing specular surfaces using color,” in *CVPR*, 2005.

[80] L. S. Mark, J. B. Pittenger, H. Hines, C. Carello, R. E. Shaw, and J. T. Todd, “Wrinkling and head shape as coordinated sources of age level information,” *Journal of Perception and Psychophysics*, vol. 27, no. 2, pp. 117–124, 1980.

[81] L. S. Mark and J. T. Todd, “The perception of growth in three dimensions,” *Journal of Perception and Psychophysics*, vol. 33, no. 2, pp. 193–196, 1983.

[82] G. McNeill and S. Vijayakumar, “Hierarchical procrustes matching for shape retrieval,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 885–894.

[83] F. Mokhtarian, F. Abbasi, and J. Kittler, “Efficient and robust retrieval by shape content through curvature scale space,” *Image Databases and Multimedia Search*, pp. 51–58, 1997.

[84] G. Mori, S. Belongie, and J. Malik, “Shape contexts enable efficient retrieval of similar shapes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 723–730.

[85] ——, “Efficient shape matching using shape contexts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1832–1837, 2005.

[86] G. Mori and J. Malik, “Recognizing objects in adversarial clutter: breaking a visual captcha,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 134–141.

[87] E. Murphy-Chutorian and M. Trivedi, “Head pose estimation in computer vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607–626, April 2009.

[88] F. Naderi and A. A. Sawchuk, “Estimation of images degraded by film-grain noise,” *Appl. Optics*, vol. 17, pp. 1228–1237, Apr 1978.

[89] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2161– 2168.

[90] S. A. Niyogi and E. H. Adelson, "Analyzing and recognizing walking figures in xyt," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 469–474.

[91] R. Ohbuchi, T. Minamitani, and T. Takei, "Shape-similarity search of 3d models by using enhanced shape functions," in *Theory and Practice of Computer Graphics*, 2003, pp. 97–104.

[92] T. Ojala, M. Pietikinen, and T. Menp, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[93] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, no. 4, 2002.

[94] P. J. Phillips, "Support vector machines applied to face recognition," in *Proceedings of the Conference on Advances in Neural Information Processing Systems II*, 1999, pp. 803–809.

[95] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[96] J. B. Pittenger and R. E. Shaw, "Aging faces as viscal-elastic events: Implications for a theory of nonrigid shape perception," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 1, no. 4, pp. 374–382, 1975.

[97] J. B. Pittenger, R. E. Shaw, and L. S. Mark, "Perceptual information for the age level of faces as a higher order invariant of growth," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 5, no. 3, pp. 478–493, 1979.

[98] S. Prince, J. Warrell, J. Elder, and F. Felisberti, “Tied factor analysis for face recognition across large pose differences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 970–984, June 2008.

[99] R. Ramamoorthi and P. Hanrahan, “On the relationship between radiance and irradiance: determining the illumination from images of convex Lambertian object,” *JOSA-A*, pp. 2448–2459, Oct 2001.

[100] N. Ramanathan and R. Chellappa, “Face verification across age progression,” *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3349–3361, November 2006.

[101] ——, “Modeling age progression in young faces,” in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2006, pp. 387–394.

[102] C. Rao, A. Yilmaz, and M. Shah, “View-invariant representation and recognition of actions,” *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.

[103] S. Romdhani, V. Blanz, and T. Vetter, “Face identification by fitting a 3d morphable model using linear shape and texture error functions,” in *European Conference on Computer Vision*, 2002, pp. 3–19.

[104] S. Romdhani, J. Ho, T. Vetter, and D. Kriegman, “Face recognition using 3-d models: Pose and illumination,” *Proceedings of the IEEE*, vol. 94, no. 11, November 2006.

[105] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Comm. and Control*. McGraw-Hill, 1971.

[106] D. Samaras and D. Metaxas, “Illumination constraints in deformable models for shape and light direction estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 247–264, Feb 2003.

[107] C. Samir, A. Srivastava, and M. Daoudi, “Three-dimensional face recognition using shapes of facial curves,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1858–1863, 2006.

[108] C. Scandrett, C. Solomon, and S. Gibson, “A person-specific, rigorous aging model of the human face,” *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1776–1787, November 2006.

[109] T. B. Sebastian, P. N. Klein, and B. B. Kimia, “Recognition of shapes by editing their shock graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 550–571, 2004.

[110] T. B. Sebastian, P. N. Klien, and B. B. Kimia, “On aligning curves,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 116–125, 2003.

[111] S. Shafer, “Using color to separate reflection components,” *Color Research and Applications*, vol. 10, no. 4, pp. 210–218, 1985.

[112] D. Sharvit, J. Chan, H. Tek, and B. B. Kimia, “Symmetry-based indexing of image databases,” *Journal of Visual Communication and Image Representation*, vol. 9, no. 4, pp. 366–380, 1998.

[113] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, “The princeton shape benchmark,” *Shape Modeling International, Genova, Italy*, June 2004.

[114] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, “Shock graphs and shape matching,” *International Journal on Computer Vision*, vol. 35, no. 1, pp. 13–32, 1999.

[115] T. Sim, S. Baker, and M. Bsat, “The CMU pose, illumination, and expression database,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, 2003.

[116] W. A. P. Smith and E. R. Hancock, “Recovering facial shape using a statistical model of surface normal direction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1914–1930, Dec 2006.

[117] J. Suo, F. Min, S. Zhu, S. Shan, and X. Chen, “A multi-resolution dynamic model for face aging simulation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[118] P. Tan, S. Lin, L. Quan, and H.-Y. Shum, "Highlight removal by illumination-constrained inpainting," in *ICCV*, 2003, pp. 164–169.

[119] P. Tan, Q. Long, and S. Lin, "Separation of highlight reflections on textured surfaces," in *CVPR*, 2006.

[120] R. Tan and K. Ikeuchi, "Reflection components decomposition of textured surfaces using linear basis functions," in *CVPR*, 2005.

[121] ——, "Separating reflection components of textured surfaces using a single image," *PAMI*, vol. 27, no. 2, pp. 178–193, February 2005.

[122] R. Tan, K. Nishino, and K. Ikeuchi, "Illumination chromaticity estimation using inverse-intensity chromaticity space," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[123] ——, "Color constancy through inverse-intensity chromaticity space," *JOSA-A*, vol. 21, no. 3, pp. 321–334, 2004.

[124] ——, "Separating reflection components based on chromaticity and noise analysis," *PAMI*, vol. 26, no. 10, pp. 1373–1379, October 2004.

[125] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 127–133.

[126] B. Tiddeman, M. Burt, and D. Perrett, "Prototyping and transforming facial textures for perception research," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 42–50, 2001.

[127] J. Toro and B. Funt, "A multilinear constraint on dichromatic planes for illumination estimation," *IP*, vol. 16, no. 1, pp. 92–97, January 2007.

[128] P. Tresadern and I. Reid, "An evaluation of shape descriptors for image retrieval in human pose estimation," in *British Machine Vision Conference*, 2007.

[129] H. T. Trussell and B. R. Hunt, "Sectioned methods for image restoration," *IEEE Transactions on Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 157–164, Apr 1978.

[130] P. S. Tsai and M. Shah, "Shape from shading using linear approximation," *Image and Vision Computing Journal*, vol. 12, no. 8, pp. 487–498, 1994.

[131] Z. Tu and A. L. Yuille, "Shape matching and recognition: Using generative models and informative features," in *Proceedings of European Conference on Computer Vision*, 2004, pp. 195–209.

[132] G. Turk and J. F. O'Brien, "Variational implicit surfaces," *Technical Report 15, Georgia Institute of Technology*, 1999.

[133] Y. Wang, H. Jiang, M. Drew, L. Ze-Nian, and G. Mori, "Unsupervised discovery of action classes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1654–1661.

[134] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, February 2009.

[135] Y. Wu, N. Thalmann, and D. Thalmann, "A dynamic wrinkle model in facial animation and skin aging," *Journal of Visualization and Computer Animation*, vol. 6, pp. 195–205, 1995.

[136] X. Yilei and A. Roy-Chowdhury, "Integrating motion, illumination, and structure in video sequences with applications in illumination-invariant tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 793–806, May 2007.

[137] A. Yilmaz and M. Shah, "Actions sketch: a novel action representation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 984–989.

[138] Z. Yue, W. Zhao, and R. Chellappa, "Pose-encoded spherical harmonics for face recognition and synthesis using a single image," *EURASIP Journal on Advances in Signal Processing*.

[139] L. Zhang and D. Samaras, "Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 351–363, March 2006.

[140] R. Zhang, P. Tsai, J. Cryer, and M. Shah, "Shape from shading: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, 1999.

[141] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *IJCV*, pp. 119–152, 1994.

[142] S. Zhao and Y. Gao, "Textural hausdorff distance for wider-range tolerance to pose variation and misalignment in 2d face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[143] W. Zhao and R. Chellappa, "Symmetric shape from shading using self-ratio image," *International Journal of Computer Vision*, vol. 45, no. 1, pp. 55–75, Oct 2001.

[144] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[145] S. Zhou, R. Chellappa, and D. Jacobs, "Characterization of human faces under illumination variations using rank, integrability, and symmetry constraints," in *European Conference on Computer Vision*, 2004.

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