ABSTRACT

Title of Dissertation: View Synthesis from Image and Video
for Object Recognition Applications

Zhanfeng Yue, Doctor of Philosophy, 2007

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

Object recognition is one of the most important and successful applications in computer vision community. The varying appearances of the test object due to different poses or illumination conditions can make the object recognition problem very challenging. Using view synthesis techniques to generate pose-invariant or illumination-invariant images or videos of the test object is an appealing approach to alleviate the degrading recognition performance due to non-canonical views or lighting conditions.

In this thesis, we first present a complete framework for better synthesis and understanding of the human pose from a limited number of available silhouette images. Pose-normalized silhouette images are generated using an active virtual camera and an image based visual hull technique, with the silhouette turning function distance being used as the pose similarity measurement. In order to overcome the inability of the shape from silhouettes method to reconstruct concave regions for human postures, a view synthesis algorithm is proposed for articulating humans using visual hull and contour-based body part segmentation. These two components improve each other for better performance through the correspondence across viewpoints built via the inner distance shape context measurement.

Face recognition under varying pose is a challenging problem, especially when illumination variations are also present. We propose two algorithms to address this scenario. For a single light source, we demonstrate a pose-normalized face synthesis approach on a pixel-by-pixel basis from a single view by exploiting the bilateral symmetry of the human face. For more complicated illumination condition, the spherical harmonic representation is extended to encode pose information. An efficient method is proposed for robust face synthesis and recognition with a very compact training set.

Finally, we present an end-to-end moving object verification system for airborne video, wherein a homography based view synthesis algorithm is used to simultaneously handle the object's changes in aspect angle, depression angle, and resolution. Efficient integration of spatial and temporal model matching assures the robustness of the verification step. As a byproduct, a robust two camera tracking method using homography is also proposed and demonstrated using challenging surveillance video sequences.

View Synthesis from Image and Video for Object Recognition Applications

by

Zhanfeng Yue

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
2007

Advisory Committee:

Professor Rama Chellappa, Chairman
Professor P. S. Krishnaprasad
Professor K. J. Ray Liu
Professor David Jacobs
Professor Min Wu

DEDICATION

To my parents: Kuizheng Yue and Guijie Wang.

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor, Prof. Rama Chellappa, for his guidance and support during my study in University of Maryland. He always encouraged me to pursue my goal and work hard to achieve excellence. I especially appreciate his effort to help his students and give them advice whenever they need. He has played a significant role in both my professional and personal development in Maryland, and his vision, energy and desire for excellence have influenced me with lifetime benefits.

I am also indebt to Prof. David Jacobs, Dr. Qinfen Zheng, Dr. Wenyi Zhao and Prof. Gang Qian for their help and support during my early stage of research. Working with them has tremendously helped me to understand the broad area of computer vision and image processing, and most importantly, become a professional researcher. I am also grateful to Dr. Liang Zhao, Dr. Shaohua Kevin Zhou, Mr. David Guarino at SAIC Corp. and Dr. Haibin Ling for the helpful discussions and collaborations.

I would like to take this chance to thank members in the Computer Vision Lab for their friendship, encouragement and help. I always feel lucky to be in such an energetic and excellent group, and their companionship during my stay in Maryland has helped me to enjoy my graduate studies. Special thanks to my officemates: Dr. Amit Kale, Prof. Namrata Vasawani, Dr. Jian Li, Jie Shao, Aswin Sankaranarayanan and Mahesh Ramachandran. Thank you all for the time spent together, and the happy time in our office will be always in my memory.

I give my heartfelt gratitude to Hong, my wife, for her love and support during our study in University of Maryland. It is because of her help and encouragement that my life in the past years is much more joyful and colorful. I would also like to thank my sister, my brother-in-law and my parents-in-law for their care and support.

Finally, I owe my deepest thanks to my parents. Without their love, unconditional support and countless sacrifices, I could never accomplish so much and reach this milestone in my life. I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Object recognition is one of the most important and successful applications in computer vision. It is usually stated in the following form: Given a database of training images (sometimes called a gallery set, or gallery images), the task of object recognition is to recognize the object(s) in an incoming test image. Typically the training images in the database are obtained under controllable environments, under standard pose and illumination. In contrast, the test image may be acquired in uncontrolled environments with different poses and illumination conditions from the training images. The varying appearances of the test object can make the recognition very challenging, and significantly degrade the recognition performance. Therefore, a canonical view (e.g., the frontal view for face recognition, or the side view for gait recognition) or a standard illumination condition (e.g., the frontal point light source) for the images or videos of the test object is often required in existing recognition algorithms. However, these images or videos are usually not available in practical applications.

The most direct method to handle this scenario is to build a 3D model of the test object and generate the novel image at the same pose or under the same illumination condition as the training images. The problem of building 3D representations from a video sequence (or several images from different viewpoints), known as structure from motion problem, has been studied for more than twenty years. Methods using flows of various kinds (optical, normal and image), discrete features (points, lines and curves) have been considered. When frames from a single camera are used, one obtains a relative depth map from which novel views can be generated; estimates of absolute depth values can be obtained when multiple cameras are used. Reviews and comparisons of different Structure from Motion (SfM) methods can be found in [23, 46, 52]. Although many algorithms have been developed, few give satisfactory performance in real applications. To develop acceptable estimates of 3-D structure, the following issues have to be considered: observation noise (noise present in token correspondence or in computing optical flow), feature occlusion, motion/structure recovery ambiguities, mixed domain sequences having both small and large baselines and mismatched tokens and/or independently moving objects in the observed image frames. Being able to handle these issues is critical for producing practical structure recovery algorithms. Although recently, elegant methods have been reported in [32, 93], much more needs to be done in addressing these issues. Another critical issue in developing practical SfM algorithms is accurate camera calibration, which itself poses some challenging issues [101, 117].

For some applications, it is not always necessary to explicitly reconstruct the 3D shape of the object being observed. Alternatively, we are more interested in synthesizing the pose-invariant or illumination-invariant images/videos of the test object using image rendering techniques. It provides an appealing approach to alleviate the degrading recognition performance due to test images acquired in non-canonical views or lighting conditions.

1.2 Background on View Synthesis and Image based Rendering

View synthesis is the technique of visualizing and manipulating the appearance of an object for a given viewing direction from several existing viewpoints. The traditional approach for generating virtual views of an object or a scene is to render directly from an appropriately constructed 3D model. The 3D model can be produced using a CAD modeler or from real data. More recently, image-based rendering (IBR) has become an emerging and competing rendering paradigm. In contrast to the traditional geometry-based rendering, IBR techniques rely on interpolation using the original set of input images, or pixel re-projection from source images onto the target image in order to produce a novel view. A significant advantage of IBR is that the speed of rendering is independent of the scene complexity. Given an observing direction, the IBR technique is able to synthesize the corresponding view of the object without recovering its 3D structure.

IBR techniques are classified into four distinct categories in [55]: *non-physically based image mapping, mosaicing, interpolation from dense sample, and geometrically-valid pixel re-projection*, wherein *mosaicing* and *interpolation from dense sample* are not our goal in interactive rendering. *Non-physically based image mapping* uses a training set of specific kinds of images to produce novel views, without considering 3D geometry in the pixel location computation. It was shown in [99, 100] that for linear object classes, linear transformations can be learned exactly from a basis set of 2D prototypical views. *Geometrically-valid pixel re-projection* is a more attractive method since it uses a relatively small number of input images and does not need a training set. With multi view geometry constraints, the change of each pixel location from the reference view to the desired view is determined in a predictable way, which can be described by a 3D warping equation [84, 85], a homography [54], or a trilinear tensor [5]. The trilinearities, which can be represented by a *trilinear tensor*, provide a general warping function from reference images to novel synthesized images governed directly by the parameters of the virtual camera. In [5], Avidan et al. derived a tensor operator that describes the transformation from a given tensor of three views to a novel tensor of a new configuration of three views. The desired virtual view can then be created using this new trilinear tensor. The illumination-based image synthesis method in [58], which does not require the determination of point or line correspondences, can synthesize not only novel viewpoints, but also novel illuminations conditions. These approaches require that views must often be close enough so that correspondences across these views are easy to establish. Also correspondences must be maintained over many views which spans large changes. An alternative approach is based on constructing the volumes or surfaces in 3D space that are consistent with input images. The most common method to represent this volume is voxels which can be encoded with a space-efficient octrees technique [94]. Given a set of silhouette images, a generalized 3D cone within which the object must lie can be reconstructed using shape from silhouette techniques [3, 73]. When the input images have additional photometric information other than the silhouettes, shape from photo-consistency methods [8, 57] can be used to improve the 3D reconstruction process. The space carving algorithm [60] uses a multi-plane-sweep approach to remove the non-photo-consistent voxels to guarantee that the remaining shape is the photo hull. These methods generally depend on calibrated cameras. A detailed review of volumetric scene reconstruction from multiple views may be found in [29].

1.3 Thesis Overview and Contribution

In this thesis, we study how view synthesis technique can be used to boost the performance of various object recognition applications.

Human activity (walking, carrying, throwing, etc.) carries much information which can be used for recognition or (suspicious) activity analysis applications. In order to achieve good performance for these applications, a monocular video sequence is usually not enough for recognizing arbitrary human activities due to possible acquisition in non-canonical view or self-occlusion. For example, face and gait are often used as biometric signature for human identification. Usually face recognition needs the frontal view of the human face, while gait recognition requires the side view of the human silhouette. If the person does not walk parallel to the image plane, the gait recognition rate will degrade seriously. Similarly, if there is no frontal face images in the test video, the face recognition algorithm will also have poor performance. A well controlled multi camera environment not only has a larger coverage range and provides more information than a single camera environment, but also makes it possible to render a novel image (video) for a desired viewpoint, or even reconstruct the 3D shape. In Chapter 2, we propose a complete framework which processes the images/video from a multi-camera environment, and produces a pose-invariant video sequence for human recognition applications and body part segmentation results for a better understanding of the human posture. It combines the active image based visual hull (IBVH) algorithm and a contour-based human body part segmentation technique. They improve each other for better performance by establishing the correspondence across viewpoints built via the inner distance shape context (IDSC) measurement proposed in [62].

For decades, face recognition has been one of the most important applications of image analysis and understanding. Face recognition under varying pose is a challenging problem, especially when illumination variations are also present. We propose two approaches in Chapter 3 and Chapter 4 in order to improve the recognition performance degradation caused by the pose and illumination variations. Specifically, when the test face image is taken under a single light source, we propose in Chapter 3 a pose-normalized face synthesis approach from a single view by exploiting the bilateral symmetry of the human face. We show that given illumination and pose estimation and the required correspondences, the mirror view under the same illumination as the original view can be determined on a pixel-by-pixel basis using the original view and its mirror image. Consequently the pose-normalized view under the given illumination can be generated using view morphing techniques.

For more complicated illumination conditions, we propose to address one of the most challenging scenarios in face recognition. That is, to identify a subject from a test image that is acquired under different pose and illumination condition from the only one training sample of this subject in the database. For example, the test image could be semi-frontal and under multiple lighting sources while the corresponding training image is frontal under a single lighting source. Under the assumption of Lambertian reflectance, the spherical harmonics representation has proved to be effective in modeling illumination variations for a fixed pose. In

Chapter 4, we extend the spherical harmonics representation to encode pose information. More specifically, we utilize the fact that 2D harmonic basis images at different poses are related by close-form linear transformations, and give a more convenient transformation matrix to be directly used for basis images. An immediate application is that we can easily synthesize a different view of a subject under arbitrary lighting conditions by changing the coefficients of the spherical harmonics representation. A more important application of this algorithm is an efficient face recognition method, based on the orthonormality of the linear transformations, for addressing the above-mentioned challenging scenario. Thus we directly project a non-frontal view test image onto the space of frontal view harmonic basis images. The impact of some empirical factors due to the projection is embedded in a sparse warping matrix, and we prove that the recognition performance does not deteriorate after warping the test image to the front view. Very good recognition results are obtained using this method for both synthetic and challenging real images.

Recently, analysis of airborne surveillance videos has drawn extensive attention for both military and civilian applications, e.g. UAVs and police video. Target tracking and object verification are two important problems for most of the airborne surveillance video. When the object of interest is at a great distance to the camera (e.g., the vehicle in the airborne video sequence), its depth-relief is small compared to the distance between the sensor and the object. Therefore, it is difficult to reliably build the 3D structure of the object and perform tracking or verification. However, in this case, it is reasonable to assume that the observed object moves on a dominant plane (the ground plane) which induces a homography relation between two views. In Chapter 5, we first present a robust two camera tracking method which handles occlusions using the homography between the two views. An adaptive appearance model is incorporated in Sequential Monte Carlo (SMC) framework to accomplish the single view tracking. Correct transformation of the target in the occluded view can be inferred from the homography and the tracking result of the un-occluded view. We then present an end-to-end verification system for moving objects in airborne video. Lacking prior training data, the object information is collected on the fly from a short real-time learning sequence. Using a sample selection module, the system selects samples from the learning sequence and stores them in an exemplar database. To handle appearance change due to potentially large aspect angle variations, a homography-based view synthesis method is used to generate a novel view of each image in the exemplar database at the same pose as the query object in each frame of a query sequence. A spatial match score is obtained using a Distance Transform to compare the novel view and query object. After looping over all query frames, the set of match scores is passed to a temporal analysis module to examine the behavior of the query object, and calculate a final likelihood.

Finally, we draw conclusions and discuss some possible future directions in Chapter 6.

Chapter 2

View Synthesis for Articulating Human Using Image-based Visual Hull

Silhouette images from multiple views provide much information on the pose and activity of a person being observed, and can be used in various applications. In the absence of required number of cameras, the articulated human pose analysis from silhouettes can be very ambiguous. In this chapter, we propose a complete framework for a better synthesis and understanding of the human pose from a limited number of available silhouette images [107]. It combines an active IBVH algorithm and a contour based body part segmentation technique, and does not reconstruct the 3D shape of the subject. Instead of solving a non-linear optimization problem, we derive a simple, approximate algorithm to decide the extrinsic parameters of a virtual camera. By doing so, we are able to synthesize the turntable image collection of the person using the IBVH algorithm by actively moving the virtual camera on a properly computed circular trajectory around the person. Using the turning function distance as the silhouette similarity measurement, this approach can be used to generate the desired pose-normalized images for recognition applications. In order to overcome the inability of the visual hull (VH) method to reconstruct concave regions, we propose a contour-based human body part localization algorithm to segment the available and synthesized silhouette images into convex body parts. The body parts in the virtual view are separately generated from the corresponding body parts in the input views and then assembled together for a more accurate VH reconstruction. Furthermore, as the turntable image collection is obtained, it helps to improve the body part segmentation and identification process. By using the inner distance shape context (IDSC) measurement, we are able to build the correspondence between the contours taken from two different viewpoints which are not too far from each other, and therefore estimate the body part locations more accurately from a synthesized view where we can localize the body part more confidently. Experiments show that the proposed algorithm can greatly improve the body part segmentation and hence the shape reconstruction results. Fig **??** shows the relationship among the components of the system.

2.1 Background and Previous Work

A significant body of work on human pose analysis from the 2D projections exists in the literature. In one type of approach, 3D pose can be efficiently recovered from the 2D video sequence [79, 95] by assuming some specific features, e.g. the image locations of the center of each body joint, can be reliably detected and a generic model of the human body articulation is available. Based on a training set of synthesized motion capture data, Howe et al [44] recover the 3D pose from the detected feature locations using a Bayesian learning framework. In [70], a

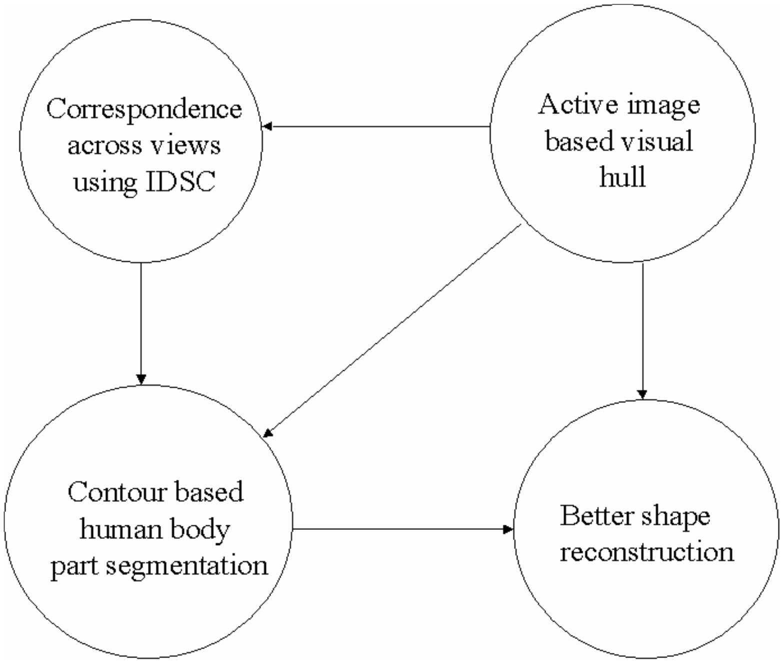

Correspondence across views using IDSC
Active image based visual hull
Contour based human body part segmentation
Better shape reconstruction

Figure 2.1: The relationship among the components of the proposed system.

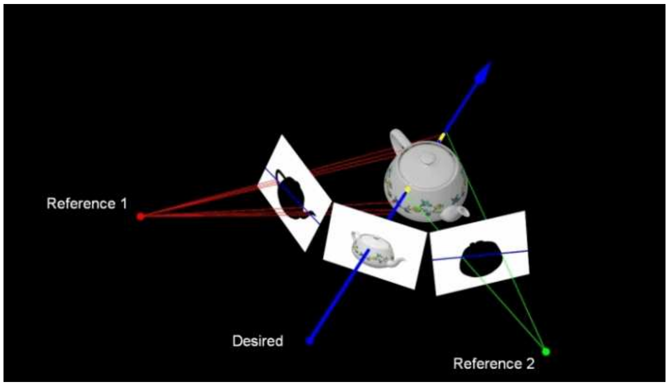
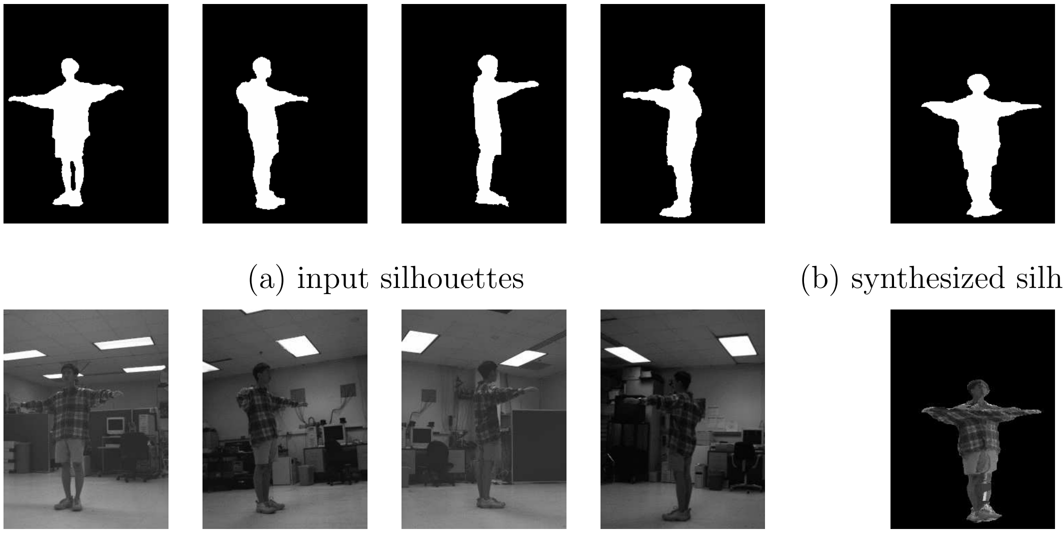


Figure 2.2: The formulation of image based visual hull.



(a) input silhouettes

(b) synthesized silhouette

(c) input images

(d) synthesized image

Figure 2.3: An example of IBVH:(a) the silhouette images observed from four static cameras. (b) The rendered silhouette image for a novel view obtained with IBVH. (c) The original images captured from the four static cameras. (d) The corresponding synthesized texture-mapped image for the novel view.

shape context descriptor is used to estimate the feature locations against a set of training images with pre-marked features. The 3D pose is then reconstructed using the algorithm proposed in [79]. In [80], the mapping of a silhouette to 3D pose is learned using multi-view training data. These techniques were successful, but they mainly depend on reliable detection of feature (joint) locations. Another type of approach directly learns the 3D pose from image measurements. In [17], a dynamical manifold of human body configurations represented by a Hidden Markov Model is learned using entropy minimization. Shakhnarovich et al [88] propose Parameter-Sensitive Hashing, which finds approximate neighbors in time sublinear in the number of examples, to rapidly find relevant examples in a large database of training images and estimate the articulated human body pose using a local model learned from those examples.

Shape from Contours (SFC) technique, which approximates the shape of an object using silhouette images, has been an important and active research topic in computer vision for over two decades. Estimating 3D shape using SFC has many advantages. Silhouettes are readily and easily obtainable and the implementation of the SFC algorithms is generally straightforward. As one of the most important methods in SFC, Visual Hull (VH) [61] construction provides an upper bound on the shape of the object. The VH of an object is the intersection of all the extruded cone-like shapes that result from back-projecting the silhouettes in all views. Hence, VH can be obtained by volume carving. It is possible to reduce the computation of VH to 2D operations since it contains only points that project onto the silhouettes. Image based visual hull (IBVH) [68] is a an effective and fast method to compute the VH and view synthesis. It is shown that for each pixel in the desired view, the epipolar line in each input view is intersected with the contour approximation, then the intersected 2D line segment is projected back to 3D space to form the VH. IBVH is a view dependent algorithm. It ensures the correctness of the generated image for the desired viewpoint (with the epipolar constraints), with no need to explicitly build the VH in 3D space. Figure 2.2 shows how IBVH is formulated. The algorithm is able to render a desired view of n^2 pixels in $O(kn^2)$ where k is the number of input views. After the VH is constructed, its surface is texture mapped using the weighted sum of intensity values in the input images [67]. Considering the visibility during the texture mapping process, an occlusion-compatible warping ordering scheme [69] is used to solve the object occlusion problem. An advantage of the IBVH technique is its tradeoff between accuracy and efficiency. With the widely-positioned views as inputs, IBVH allows us to produce the virtual view without finding the wide baseline correspondence. It also provides information about the object's 3D shape and location. Besides, since the VH is formed by volume carving, the noise from input images is greatly reduced in the intersecting process.

Researchers have proposed various methods to accomplish 3D reconstruction from silhouettes [21,22] by utilizing the fact that the intersection of the generalized cones associated with a set of cameras define a volume of scene space containing the object. However, most silhouette-based reconstruction encloses the true volume and only approximates the true 3D shape, depending on the number of views, the positions of the viewpoints, and the complexity of the object. In particular, the concave patches are not observable in any silhouette. Fig 2.3 shows an example of view synthesis with IBVH. We can observe from Fig 2.3 that the person stands with a 3D concave posture which is formed by the stretching arms and the torso. Although the rendered silhouette image shown in Fig 2.3 (b) is correct due to the fact that human eyes can be fooled into perceiving convex and concave regions with only silhouette images, the error coming from the concave regions can be easily observed on the texture-mapped chest part in Fig 2.3 (d). [61] stated that the VH of an object depends not only on the object itself but also on the region allowed to the viewpoint. The *external visual hull* is related to the convex hull, and the *internal visual hull* can not be observed from any viewpoint outside the convex hull.

2.2 Pose Normalized View Synthesis from Silhouettes

The varying appearances due to different poses can make the human recognition problem very challenging. Some promising results have been reported for integrated gait and face recognition from multiple views [87]. A strong assumption they made is that the person is moving forward. Under this assumption, the person's motion trajectory is easy to estimate and the virtual camera can be placed accordingly. This approach will not work if the motion trajectory is hard to estimate, or not available (e.g., turning around).

With IBVH technique and an active virtual camera, images from different viewpoints can be generated to give us a better understanding of the object. In this section, we show how to generate a collection of the object's images (named the turntable image collection) which are captured by a camera moving around the object, with the optical axis parallel to the plane that the object stands on. Using a small number of widely-placed views as input, the turntable image collection can be rendered quickly and efficiently with the IBVH technique [104]. Using this syn-

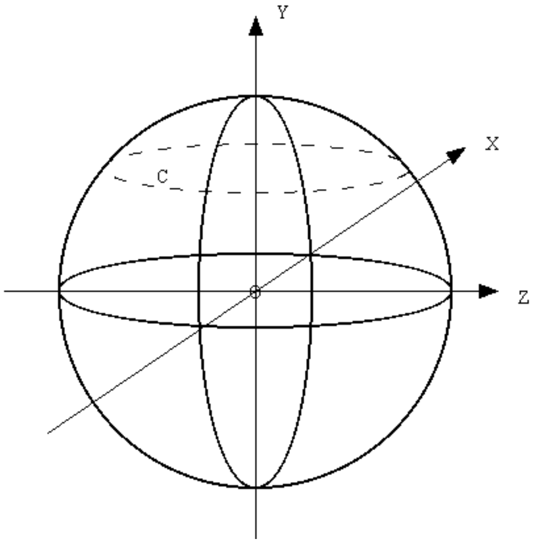


Figure 2.4: The coordinate system being used coincides with the world coordinate system, so the trajectory of the virtual camera should be the dotted circle C which is parallel to the X-Z plane.

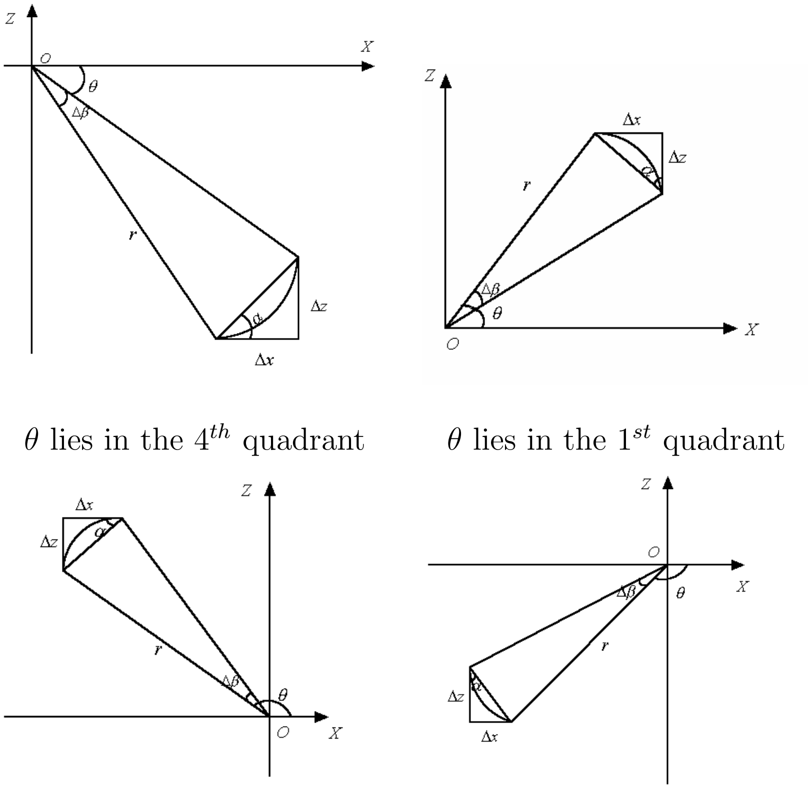
thesized image collection, we are able to produce a pose-invariant video sequence by using the turning function distance [43] as the similarity measurement of the silhouette images [105].

2.2.1 Turntable Image Collection Rendering With IBVH

We propose to render the turntable image collection captured by a virtual camera moving around the person, with the optical axis parallel to the plane on which the person is in a standing position. We derive a method to align the camera calibration coordinate system and the world coordinate system if they do not coincide with each other, with which the virtual camera's position on the trajectory can be accurately decided.

In order to generate the turntable image collection, we have to move a virtual camera along a properly computed circular trajectory on the view sphere, where the view sphere of an object is a sphere which is centered at the object and has a fixed radius [6]. Assuming that the virtual camera's intrinsic parameters are known (they can be assumed same as the available real cameras' parameters), its extrinsic parameters at each position on the circular trajectory needs to be determined. We use the same coordinate system as the one in which the real cameras are calibrated, except that the origin of the coordinate system is set as the 3D centroid O of the computed VH. Apparently, this is not a static coordinate system because the origin changes with the centroid of the VH from frame to frame. All the world coordinates are mapped to this coordinate system. Here the world coordinate system refers to the one with Y-axis perpendicular to the ground plane. Suppose we start from the initial position of the virtual camera at $P(X_p, Y_p, Z_p)$, the view sphere is set up around O with radius $R = \|\vec{P} - \vec{O}\|$. The extrinsic parameters to be determined include the translation vector $[T_x, T_y, T_z]$ and the rotation angles $[\phi, \theta, \psi]$ (pitch, yaw and roll, respectively) around the X, Y, and Z-axes respectively.

There are two cases to be considered. If the real cameras are calibrated in a coordinate system which coincides with the world coordinate system, resulting in the X-Z plane being parallel to the ground plane, then the virtual camera's motion trajectory is a circle parallel to the X-Z plane, as the dotted circle C shown in Fig 2.4. This case is trivial. Starting from the initial position P, the virtual camera's circular trajectory is centered at $(0, Y_p, 0)$, and with radius $r = \sqrt{R^2 - Y_p^2}$. Since the circle is parallel to the X-Z plane, Y_p, ϕ and ψ do not change at each position along the circle. Given θ, we can uniquely determine X and Z coordinates if r is fixed. Hence, θ is the only parameter we need to control. Let $\Delta\theta$ be the step size for θ. Fig 2.5 shows the example of deriving the changes in X and Z coordinates



θ lies in the 4^{th} quadrant θ lies in the 1^{st} quadrant

θ lies in the 2^{nd} quadrant θ lies in the 3^{rd} quadrant

Figure 2.5: The example of deriving the changes in X and Z coordinates from θ and $\Delta\theta$ if the virtual camera's motion is clockwise.

from θ and $\Delta\theta$ if the virtual camera's motion is clockwise, where X_n and Z_n denote the X and Z coordinates for the n-th position of the virtual camera, and α is an auxiliary angle. The counter clockwise motion case is similar.

If θ lies in the 4^{th} quadrant, $\Delta\theta/2 - \pi/2 \leq \theta \leq \Delta\theta/2$,

$$\alpha = \begin{cases} \pi/2 - |\theta| - \Delta\theta/2 & \text{if } \theta \leq 0 \\ \pi/2 + |\theta| - \Delta\theta/2 & \text{if } \theta \geq 0 \end{cases} \tag{2.1}$$

$$X_{n+1} = X_n - 2r\sin(\Delta\theta/2)\cos\alpha \tag{2.2}$$

$$Z_{n+1} = Z_n - 2r\sin(\Delta\theta/2)\sin\alpha \tag{2.3}$$

If θ lies in the 1^{st} quadrant, $\Delta\theta \leq \theta \leq \pi/2 + \Delta\theta/2$,

$$\alpha = |\theta| - \Delta\theta/2 \tag{2.4}$$

$$X_{n+1} = X_n + 2r\text{sin}(\Delta\theta/2)\sin\alpha \tag{2.5}$$

$$Z_{n+1} = Z_n - 2r\text{sin}(\Delta\theta/2)\cos\alpha \tag{2.6}$$

If θ lies in the 2^{nd} quadrant, $\Delta\theta + \pi/2 \leq \theta \leq \pi$ or $\theta \leq -\pi + \Delta\theta/2$,

$$\alpha = \begin{cases} 3\pi/2 - |\theta| - \Delta\theta/2 & \text{if } \theta \leq 0 \\ -\pi/2 + |\theta| - \Delta\theta/2 & \text{if } \theta \geq 0 \end{cases} \tag{2.7}$$

$$X_{n+1} = X_n + 2r\text{sin}(\Delta\theta/2)\cos\alpha \tag{2.8}$$

$$Z_{n+1} = Z_n + 2r\text{sin}(\Delta\theta/2)\sin\alpha \tag{2.9}$$

If θ lies in the 3^{rd} quadrant, $\Delta\theta - \pi/2 \leq \theta \leq -\pi + \Delta\theta/2$,

$$\alpha = \pi - |\theta| - \Delta\theta/2 \tag{2.10}$$

$$X_{n+1} = X_n - 2r\text{sin}(\Delta\theta/2)\sin\alpha \tag{2.11}$$

$$Z_{n+1} = Z_n + 2r\text{sin}(\Delta\theta/2)\cos\alpha \tag{2.12}$$

If the real cameras are calibrated in a coordinate system whose X-Z plane is not parallel to the ground plane, then the virtual camera's motion trajectory should be a circle perpendicular to the person's principal axis Y', as the shaded circle C' shown in Fig 2.6. This case often happens when some accurate calibration hardware is utilized to facilitate strong calibration of the camera system, such as the Peak Performance calibration frame [26] shown in Fig 2.7. In this case, the person's vertical principal axes is along the direction of Y' which has an unknown angle ω with the Y-axis. If the turntable image collection of the object is obtained

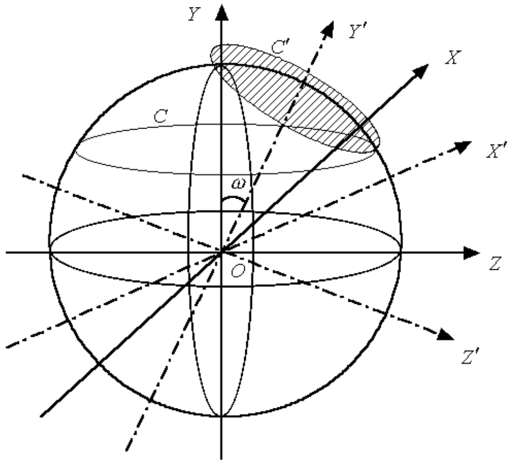


Figure 2.6: The coordinate system being used does not coincide with the world coordinate system, so the circular trajectory of the virtual camera should be the shaded C' which is not parallel to the X-Z plane.

along the circle C parallel to the X-Z plane, we can observe that the object keeps moving upward in the first half circle and downward in the second half. Also the x-coordinate of the object's 2D centroid does not remain fixed. To solve this problem, we need to align the coordinate system X'-Y'-Z' with the world coordinate system X-Y-Z.

Assuming that the person being observed stands upright on the ground plane, we can use his/her vertical principal axis in each input image as the corresponding 2D line and estimate ω by solving an optimization problem. Here we propose another feasible solution with which neither the solution to the optimization problem nor extra computational cost are needed.

Let $p_n = (x_n, y_n)$ be the 2D centroid of the image observed by the virtual camera at position n, and Δy_n and Δx_n the change of y_n and x_n from position n to position $n+1$ respectively. Then we have $\Delta y_n = y_{n+1} - y_n$ and $\Delta x_n = x_{n+1} - x_n$.

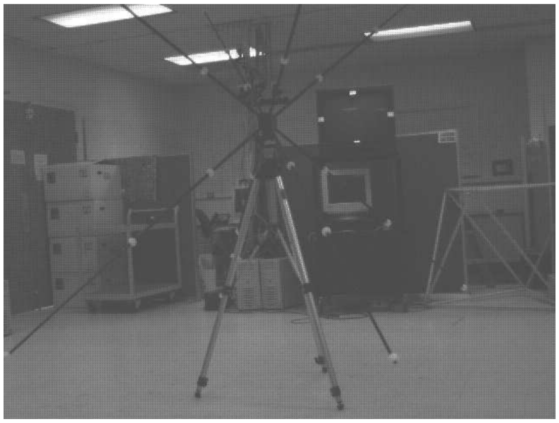


Figure 2.7: The Peak Performance calibration frame used in [26] for more accurate camera calibrations.

We try to approximate the circle C' with the observed Δy_n and Δx_n, as shown in Fig 2.8.

Let ΔY_n be the change along the Y direction from position n to position $n+1$. From the theorem on triangle similarity we have

$$\frac{\Delta y_n dp_y}{\Delta Y_n} = \frac{f}{R} \tag{2.13}$$

where dp_y is the size of each pixel along the y direction and f is the camera's focal length. Similarly,

$$\frac{\Delta x_n dp_x}{\Delta D_n} = \frac{f}{R} \tag{2.14}$$

where dp_x is the size of each pixel along the x direction and ΔD_n is the translation adjustment on the circle C in order to keep $x_{n+1} = x_n$. ΔD_n can be compensated by adjusting θ accordingly. Since ΔD_n is very small compared to the sphere radius R, the adjusting angle $\Delta\theta$ can be approximated as $\Delta\theta = 2\arctan\dfrac{\Delta D_n}{2R}$.

In order to keep the object's principal axes perpendicular and parallel to the ground plane respectively, the roll angel ψ also has to be modified. At position n, the eigenvector $[\vec{e_x}, \vec{e_y}]$ of the silhouette image is computed and we have $\psi_{n+1} =$

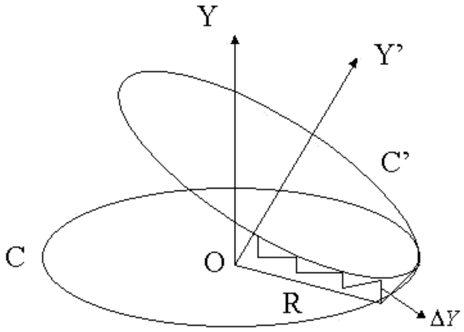


Figure 2.8: With the silhouette centroid observed for the previous position, the circle C' can be approximated by adjusting the Y coordinate of the virtual camera.

$\psi_n + \arctan e_{x1}/e_{x2}$, where (e_{x1}, e_{x2}) is the element of $\vec{e_x}$.

Since the view sphere radius R has to remain constant, X and Z coordinates have to be further adjusted based on ΔY_n, as shown in Fig 2.9. Denote $r' = \sqrt{R^2 - (Y_n + \Delta Y_n)^2}$ and $\Delta r = |r - r'|$, then $\Delta X_n = \Delta r \cos|\theta|$ and $\Delta Z_n = \Delta r \sin|\theta|$. We consider the virtual camera's clockwise motion again as mentioned before.

If θ lies in the 4^{th} quadrant,

$$\begin{cases} X_{n+1} = X_n - \Delta X,\ Z_{n+1} = Z_n + \Delta Z,\ \text{if } Y_n \cdot \Delta Y_n \geq 0 \\ X_{n+1} = X_n + \Delta X,\ Z_{n+1} = Z_n - \Delta Z,\ \text{otherwise} \end{cases} \tag{2.15}$$

If θ lies in the 1^{st} quadrant,

$$\begin{cases} X_{n+1} = X_n - \Delta X,\ Z_{n+1} = Z_n - \Delta Z,\ \text{if } Y_n \cdot \Delta Y_n \geq 0 \\ X_{n+1} = X_n + \Delta X,\ Z_{n+1} = Z_n + \Delta Z,\ \text{otherwise} \end{cases} \tag{2.16}$$

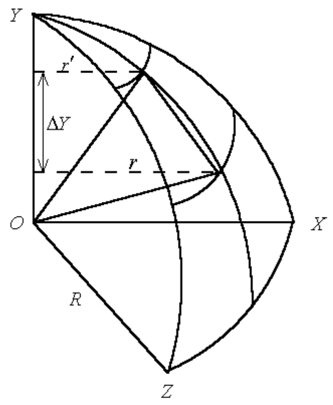


Figure 2.9: Adjust the X and Z coordinates according to the value of ΔY.

If θ lies in the 2^{nd} quadrant,

$$\begin{cases} X_{n+1} = X_n + \Delta X,\ Z_{n+1} = Z_n - \Delta Z, \text{ if } Y_n \cdot \Delta Y_n \geq 0 \\ X_{n+1} = X_n - \Delta X,\ Z_{n+1} = Z_n + \Delta Z, \text{ otherwise} \end{cases} \tag{2.17}$$

If θ lies in the 3^{rd} quadrant,

$$\begin{cases} X_{n+1} = X_n + \Delta X,\ Z_{n+1} = Z_n + \Delta Z, \text{ if } Y_n \cdot \Delta Y_n \geq 0 \\ X_{n+1} = X_n - \Delta X,\ Z_{n+1} = Z_n - \Delta Z, \text{ otherwise} \end{cases} \tag{2.18}$$

The active virtual camera positioning algorithm is summarized as follows:

1. Choose the extrinsic parameters of the virtual camera as the average of any two real cameras' parameters. Usually this is a good position to start with. Select the step size $\Delta\theta$ for θ.

2. Get the silhouette image at the current position n, and compute the 2D centroid changes Δx_n and Δy_n from the silhouette image at the previous position. Compute the eigenvector $[\vec{e_x}, \vec{e_y}]$ of the current silhouette image.

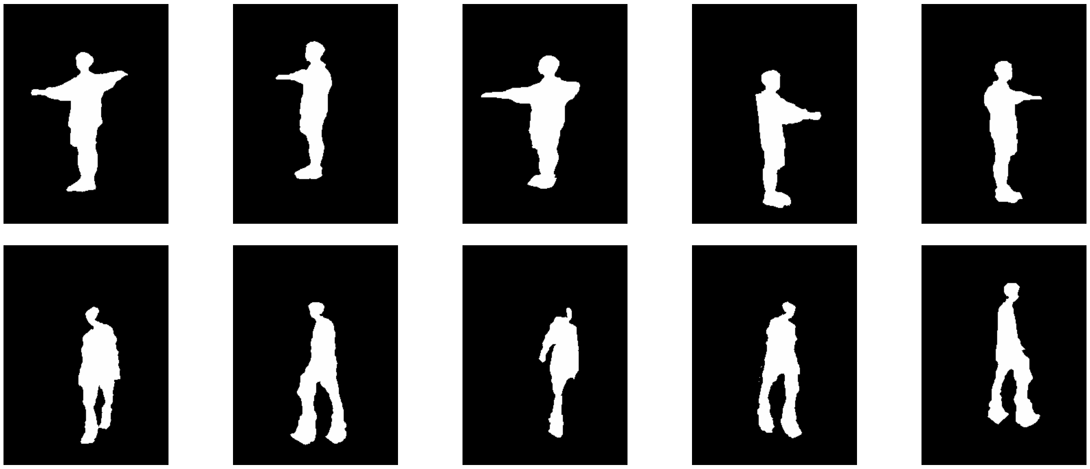


Figure 2.10: The synthesized turntable silhouette image collection. Top: the turning and pointing sequence taken at the Keck Lab in UMD. Bottom: walking sequence collected at MIT AI lab.

3. Compute ΔY_n and ΔD_n as in (13) and (14), then compute $\Delta\theta$ with ΔD_n. Let $\psi_{n+1} = \psi_n + \arctan(e_{x1}/e_{x2})$.

4. Modify X and Z coordinates through (15)-(18) with the ΔY_n obtained in the previous step.

5. Move the virtual camera to the next position as in (1)-(12), and let $\theta_{n+1} = \theta_n + \Delta\theta$.

6. Repeat steps 2 through 5 until the virtual camera comes back to the original position.

This algorithm was implemented and tested on several sequences. The input is the synchronized perspective 4-view silhouette sequences for a person, with the cameras fully calibrated. The output is the rendered turntable image collection of the person for each frame. The turning and pointing sequence was taken at the Keck Lab at University of Maryland. The person's motion is mainly turning

Table 2.1: Virtual camera position and rotation angles for the turning and pointing sequence.

position	4	8	10	16	19
$C_x(m)$	3.827	4.881	3.288	-2.585	-2.502
$C_y(m)$	0.086	1.208	1.990	1.752	0.479
$C_z(m)$	3.866	-0.608	-2.333	-0.618	2.884
$\phi(rad)$	2.825	2.825	2.825	2.825	2.825
$\theta(rad)$	0.624	1.860	2.479	-2.082	-1.134
$\psi(rad)$	-0.213	-0.228	-0.136	0.230	0.234

motion, so the trajectory information is hard to extract from the sequence. The top row of Fig 2.10 is the result for the pointing and turning sequence, with Table 1 showing the virtual camera's 3D position (C_x, C_y, C_z) and its rotation angles (ϕ, θ, ψ). The normal walking sequence was collected at MIT AI lab. The trajectory information can be estimated from the 3D centroid coordinates of the VH as mentioned in [87]. Our algorithm also works well as shown in the bottom row of Fig 2.10, with Table 2 showing the virtual camera's 3D position (C_x, C_y, C_z) and its rotation angles (ϕ, θ, ψ). In both sequences, $\Delta\theta = 0.3$ rad, so there are 21 positions on the whole circle around the person. We can observe from (C_x, C_y, C_z) in Table 1 and Table 2 that for both sequences the virtual cameras' trajectories are circles not parallel to the X-Z planes. Fig 2.10 shows that the circle approximation approach gives satisfactory results.

Table 2.2: Virtual camera position and rotation angles for the normal walking sequence.

position	1	4	10	13	16
$C_x(m)$	2.481	4.800	-1.138	-4.350	-4.219
$C_y(m)$	2.927	0.918	-1.591	-0.543	1.851
$C_z(m)$	3.617	-0.145	-4.084	-1.831	2.275
$\phi(rad)$	3.682	3.682	3.682	3.682	3.682
$\theta(rad)$	0.669	1.636	-2.870	-2.032	-1.003
$\psi(rad)$	1.867	2.068	1.574	1.160	1.093

2.2.2 Desired Viewpoint Selection

Computation of good viewpoints is important in computational geometry, visual servoing, robot motion, graph drawing, etc. It is rapidly becoming a key issue in image based rendering. Much work has been done on this topic. In [66], the image-based virtual camera motion approach is presented. The method is based on the *visual servoing* approach and consists of positioning a camera according to the information perceived in the image, with the model of the scene being fully known. To be able to react automatically to modifications of the environment, the introduction of constraints into the control is also considered. A method for visual understanding of a scene by efficient automatic movement of a camera is presented in [6]. The purpose is to choose a trajectory for a virtual camera, allowing the user to have a good knowledge of the scene at the end of minimal exploration. Starting from a good view point, the virtual camera moves on the surface of a sphere surrounding the scene, combining good views, a smooth camera movement and distance from the starting point based heuristics. Similarly, [78] presents an approach to selecting a minimal number of views that allow each object face to be adequately viewed according to specified constraints on viewpoints and other features. With the CAD boundary representation model of the object of interest, and a description of the visibility of each of the object faces, the planner can select viewpoints suitable for a variety of machine vision tasks in two stages: *viewpoint planning* and *viewpoint selection*. In [98], the quality of a viewpoint is measured with the information it gives about the scene, and the authors designed an algorithm to automatically explore objects or scenes with this viewpoint entropy measure. *shape from silhouette* with equally distributed viewpoints is an often used reconstruction technique for computer animation applications, but is not suitable for arbitrary shaped objects. For this reason, a camera viewpoint control is introduced in [71], which purposefully rotates a turntable with the 3D object depending on the trace of the silhouette contour points over the rotation angle. It is reported that the remaining 3D reconstruction error is greatly reduced. Kutulakos and Dyer [59] present an approach for recovering surface shape from the occluding contour using an active observer, which is based on a relation between the geometries of a surface in a scene and its occluding contour. They have shown that there is a simple and efficient viewing strategy, depending on only curvature measurements on the occluding contour, that allows the observer to align the viewing direction with one of the two principal directions for a point on the surface.

In order to select the desired view from the turntable image collection, we need to compare the turntable images with the knowledge base of silhouettes associated with known poses. In [39] a template matching method is proposed to estimate the human pose from silhouettes, where a body posture is represented by the normalized horizontal and vertical projection histograms, the median coordinate, and the major axis of its silhouette. The extracted silhouette is compared with the projection templates using the sum of absolute difference method to estimate the main posture. While this method is simple and fast, it produces some ambiguities. A distance which can measure the similarity of two silhouettes more robustly is needed. According to [4], this distance should satisfy a number of properties, including that 1) it should be a metric, 2) it should be invariant under translation, rotation and change-of-scale, 3) it should be reasonably easy to compute, and 4) it should match our intuition. To compare a shape A, which is stored as a model (in our case, the knowledge base of silhouettes associated with known pose s), with a shape B, which is found to exist in an image (in our case, the turntable images), the distance between the turning functions $\Theta_A(s)$ and $\Theta_B(s)$ is an efficient measure of similarity.

The turning function $\Theta_A(s)$ measures the angle of the counterclockwise tangent as a function of the arc-length s measured from some reference point O on A's boundary. $\Theta_A(s)$ keeps track of the turning that takes place, increasing with left-hand turns and decreasing with right-hand turns. The turning function measures the turning that takes place as we move along the perimeter. Mathematically, if $\kappa(s)$ is the curvature function of a curve then $\Theta(s) = \int \kappa(s)$.

In [4], the distance function between two polygons A and B is formally defined as the L_p distance between their two turning functions $\Theta_A(s)$ and $\Theta_B(s)$, minimized with respect to rotation and choice of reference points,

$$\begin{aligned} d_p(A,B) &= (\min_{\theta\in\Re, t\in[0,1]} \int_0^1 |\Theta_A(s+t) - \Theta_B(s) + \theta|^p ds)^{\frac{1}{p}} \\ &= (\min_{\theta\in\Re, t\in[0,1]} D_p^{A,B}(t,\theta))^{\frac{1}{p}}, \end{aligned}$$

where $D_p^{A,B}(t,\theta) = \int_0^1 |\Theta_A(s+t) - \Theta_B(s) + \theta|^p ds$. If the L_2 metric is used, the authors proved that the distance $d_2(A, B)$ between two polygons A and B (with m and n vertices) can be computed exactly in time $O(mn \log mn)$.

The turning function metric has been shown to correlate well with the human notion of shape similarity [83]. Howe [43] used both the turning function and the Chamfer distance for silhouette lookup for automatic tracking of poses. In implementing the turning function distance, we use the method mentioned in [83], where dynamic programming is used to account for warpings that may exist between the query object and database object that result in stretching and compression. It is quite possible that the matching between the points along the border of shape A and the points along the border of shape B is not one-to-one, but one-to-many or many-to-one. It computes the global best match between $\Theta_A(s)$ and $\Theta_B(s)$ in the sense that it pairs up each element of $\Theta_A(s)$ with an element of $\Theta_B(s)$(and vice versa), but the matching must proceed monotonically through both sets. Thus it computes two sequences $i_1, i_2, \ldots, i_k$ and $j_1, j_2, \ldots, j_k$ such that either $i_{t+1} = i_t$ or $i_{t+1} = i_t + 1$ (similarly for j), by normalizing the distance between matched turning angle points: $D = \sum_{t=1,2,\ldots,k} |i_t - j_t|$.

The knowledge base of silhouettes consists of the turning functions $\Theta_{A_i}(s)$ of the silhouettes for some canonical poses, e.g. the 5 standard stances for the human walking activity. By definition, the turning function is invariant under translation and scaling of the polygon A_i. Therefore the normalization is not necessary in building the knowledge base. The turning function $\Theta_{B_j}(s)$ of the silhouette at current viewing direction is calculated and the distance functions $d_2(A_i, B_j)$ between $\Theta_{A_i}(s)$ and $\Theta_{B_j}(s)$ are computed. In addition to the silhouette B_j at the current viewing direction, we can get an auxiliary silhouette C_j by placing the virtual camera at the position where the angle around the Y-axis has

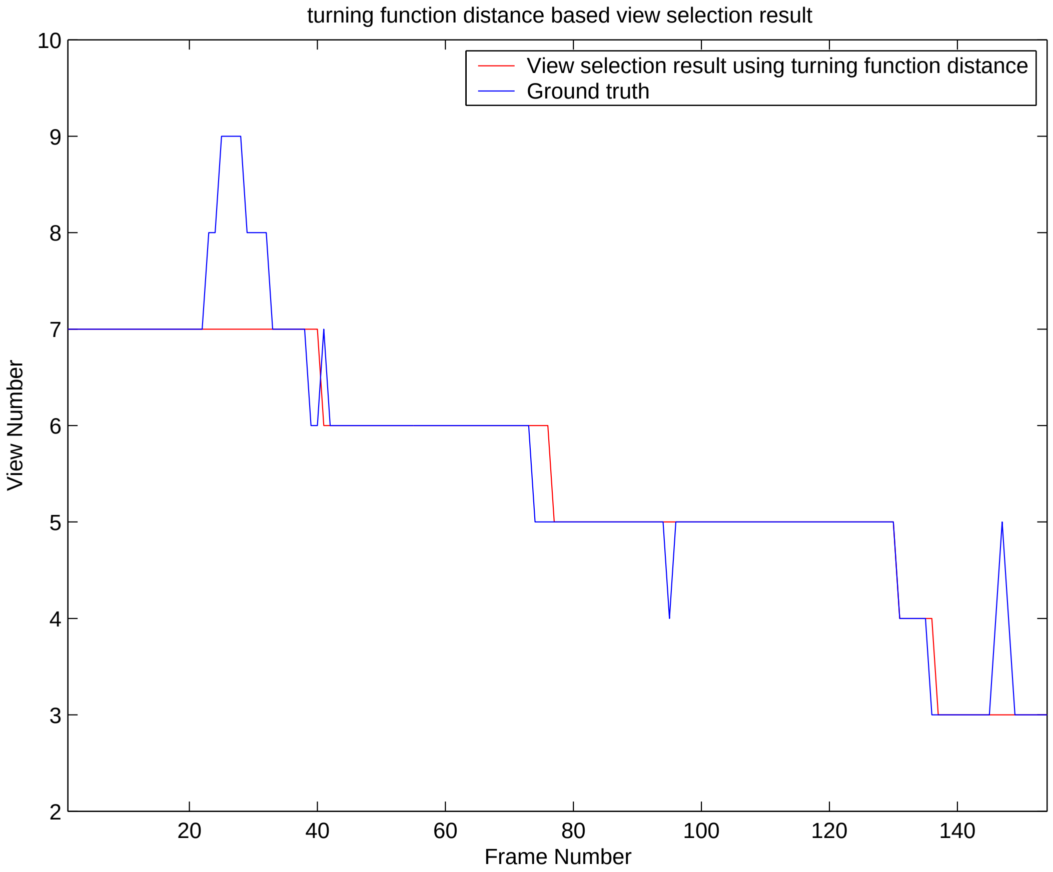
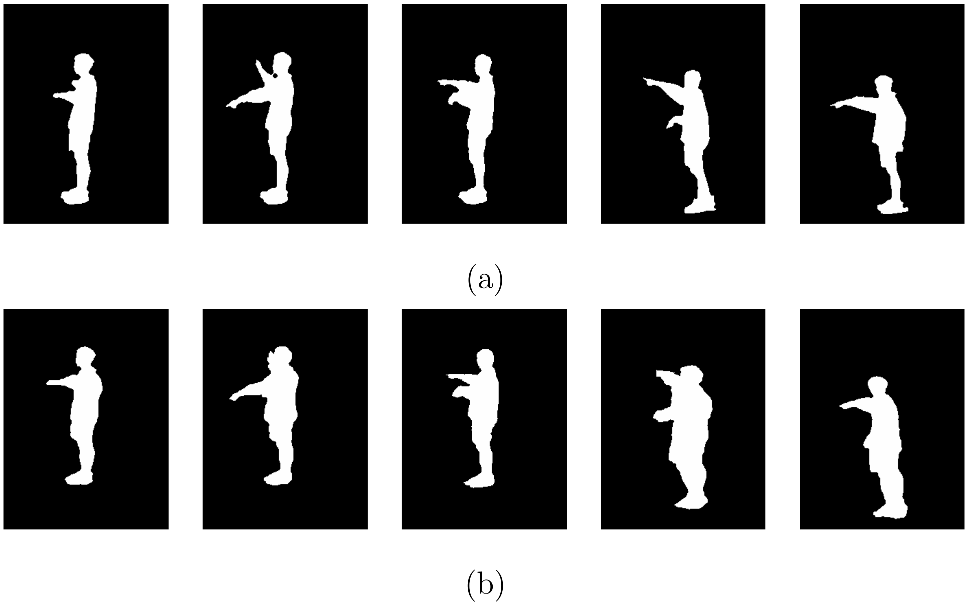


Figure 2.11: The view selection result comparison for the turning and pointing sequence, with the blue curve indicating the view selection result using the turning function distance, and the red curve indicating the ground truth.

$\pi/2$ difference with the current position. Consider the example that the desired view is a side view, let $d_2(SA_i, B_j)$ be the turning function distance between B_j and the standard stances SA_i for the side view, and $d_2(FA_i, C_j)$ be the distance between C_j and the standard stances FA_i for the frontal view, then the final decision measurement is $S(i, j) = d_2(SA_i, B_j) + d_2(FA_i, C_j)$. The view with the minimal distance not only gives the side view, but also gives the stance at which the person stands for the current frame. With this method, the desired view is



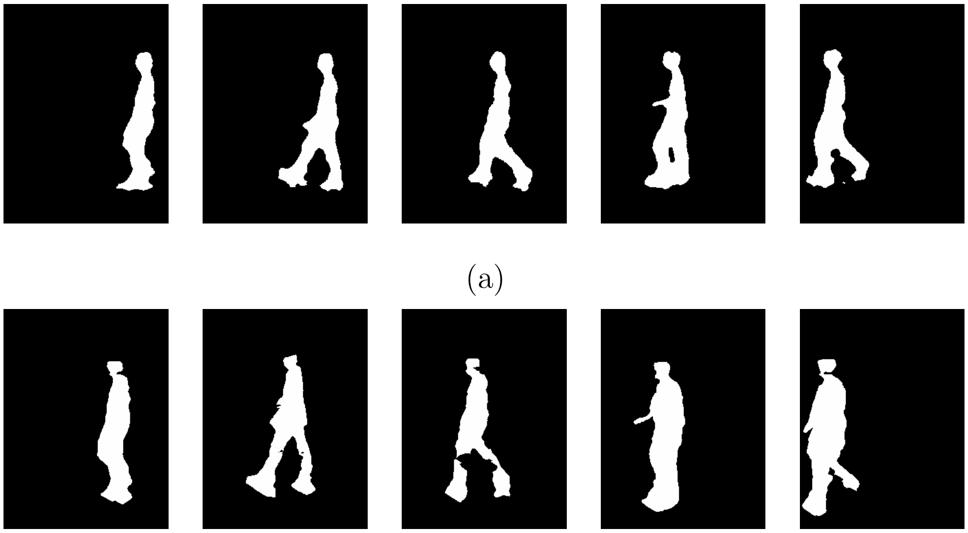
(a)

(b)

Figure 2.12: (a) The side view ground truth for the Keck Lab sequence. (b) The rendered side view for the Keck Lab sequence.

selected only when it has a small distance in matching the side view stance and its auxiliary silhouette has a small distance in matching the frontal view stance at the same time. This greatly reduces the possible ambiguities when considering the side view stance alone.

Usually there will be no abrupt change from one frame to the next, so we do not need to generate all the virtual views around the person for each frame. Only a small number of neighboring positions of the selected view in the previous frame are synthesized and compared. Experiments show that the results are good enough while the speed is much faster compared to generating all the virtual views around the person for each frame. As can be seen from Fig 2.11, which shows the view selection result for the turning and pointing sequence using the turning function distance, the selected view follows the ground truth quite well for most of the frames. Although some error still exists for several frames, it disappears in



(a)

(b)

Figure 2.13: (a) The side view ground truth for the MIT sequence. (b) The rendered side view for the MIT sequence.

the next 2-3 frames. Fig. 2.12 and Fig. 2.13 show the virtual side views for the turning and pointing sequence and the the normal walking sequence respectively, which demonstrates the efficiency of the proposed pose-normalized view synthesis algorithm from silhouettes.

2.3 Articulating Object Synthesis Using Visual Hull

Although the VH technique is a fast and efficient volumetric scene reconstruction method, like all the SFC algorithms, it still suffers from the inability of reconstructing the concave region for human postures. In order to overcome the inability of the SFC method to reconstruct concave regions for human postures, in this section we propose a simple and robust contour-based body part segmentation algorithm [106].

Observing that in many cases the concave human posture is formed due to the position of arms, we are inspired to explore the possibility of body part based view synthesis with IBVH. Several methods have been proposed for human body part segmentation from silhouette (contour). The work in [39] gives a silhouette-based human body labeling template by using topological order-constraints of body parts for different postures. A contour-based body part localization method was presented in [118] with a probabilistic similarity measure which combines the local shape and global relationship constraints to guide body part identification. More recently, a hierarchical model fitting method to estimate the 3D shape with density fields was proposed in [16]. The body parts of the human can be described accurately with the estimated parameters. We use the work in [118] for body part segmentation because of its simplicity and robustness, where the short-cut rule and the saliency requirement are combined to constrain the other end of a cut, and several computationally efficient strategies are used to reduce the effects of noise. Using this method, the silhouette image in each input view is partitioned into arms and torso (with legs) so that each human part is a convex object. All the parts are separately processed with IBVH, and assembled together to get the final result. It is possible that the final view has some disconnected or squeezed regions since it is obtained by assembling the separately processed body parts. To prevent this problem, a silhouette image for the desired viewing direction is first generated without segmenting the body parts.

2.3.1 Contour-Based Body Part Segmentation

Human body parts segmentation and identification are important and challenging problems in computer vision. Contours are the common features used to overcome inconsistent texture; parts based approaches can effectively handle occlusion and articulated motion. We segment a human body into parts at *negative minima of curvature* so that the decomposed parts are convex regions. Hoffman and Singh [41] noted that when boundary points can be joined in more than one way to decompose a silhouette, human vision prefers the partitioning scheme which uses the shortest cuts (A cut is the boundary between a part and the rest of the silhouette). They further restrict a cut to cross a symmetry axis in order to avoid short but undesirable cuts. However, most symmetry axes are very sensitive to noise and are expensive to compute. In contrast, we use the constraint on the saliency of a part to avoid short but undesirable cuts. According to Hoffman and Singh's [41] study, there are three factors that affect the saliency of a part: the size of the part relative to the whole object, the degree to which the part protrudes, and the strength of its boundaries. Among these three factors, the computation of a part's protrusion (the ratio of the perimeter of the part (excluding the cut) to the length of the cut) is more efficient and robust to noise and partial occlusion of the object. Thus, we employ the protrusion of a part to evaluate its saliency; the saliency of a part increases as its protrusion increases.

In summary, we combine the short-cut rule and the saliency requirement to constrain the other end of a cut. For example in Fig 2.14, let S be a silhouette, C be the boundary of S, P be a point on C with negative minima of curvature, and P_m be a point on C so that P and P_m divide the boundary C into two curves C_l, C_r of equal arc length. Then two cuts are formed passing through point P: $\overline{PP_l}$,

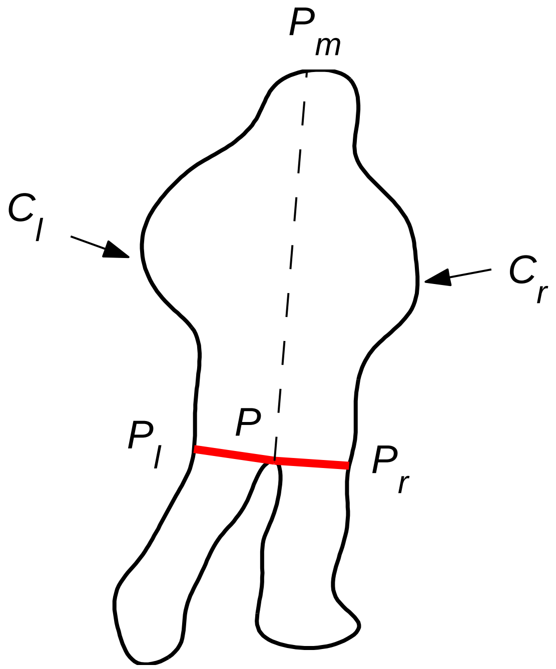


Figure 2.14: Computing the cuts passing through point P.

$\overline{PP_r}$ such that points P_l and P_r lies on C_l and C_r, respectively. The ends P_l and P_r of the two cuts are located as follows:

$$P_l = arg\min_{P'} \|\overline{PP'}\| \text{s.t.} \quad \frac{\|\widehat{PP'}\|}{\|\overline{PP'}\|} > T_p, P' \in C_l, \overline{PP'} \in S \qquad (2.19)$$

$$P_r = arg\min_{P'} \|\overline{PP'}\| \text{s.t.} \quad \frac{\|\widehat{PP'}\|}{\|\overline{PP'}\|} > T_p, P' \in C_r, \overline{PP'} \in S \qquad (2.20)$$

where $\widehat{PP'}$ is the smaller part of boundary C between P and P', $\|\widehat{PP'}\|$ is the arc length of $\widehat{PP'}$, and $\frac{\|\widehat{PP'}\|}{\|\overline{PP'}\|}$ is the saliency of the part bounded by curve $\widehat{PP_l}$ and cut $\overline{PP_l}$.

Eq. (2.19) means that point P_l is located so that the cut $\overline{PP_l}$ is the shortest one among all cuts sharing the same end P, lying within the silhouette with the other end lying on contour C_l, and resulting in a significant part whose salience is above a threshold T_p. The other point P_r is located in the same way using Eq. (2.20). It is possible that only one cut is selected if the other cut does not satisfy the saliency requirement.

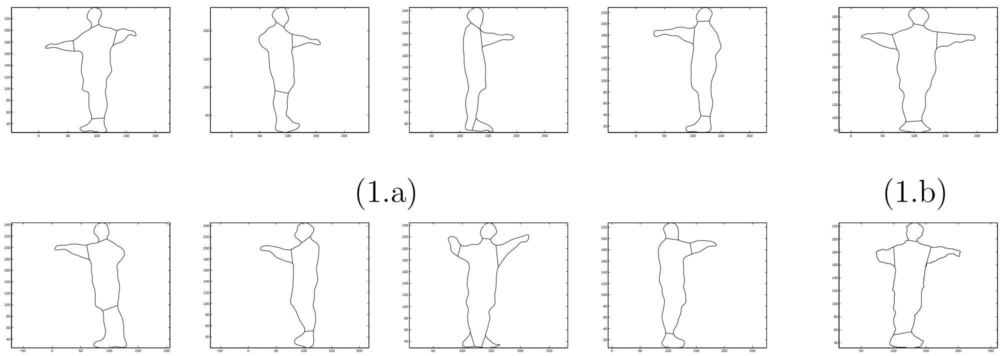
Since negative minima of curvature are obtained by local computation, their computation is not robust in real digital images. We take several computationally efficient strategies to reduce the effects of noise. First, a B-spline approximation is used to moderately smooth the boundary of a silhouette, since the B-spline representation is stable and easy to manipulate locally without affecting the remaining part of the silhouette. Second, the negative minima of curvatures with small magnitudes are removed to avoid parts due to noise or small local deformations. However, the curvature is not scale invariant (e.g. its value doubles if the silhouette shrinks by half). One way to transform curvature into a scale-invariant quantity is to first find the chord joining the two closest inflections which bound the point, then multiply the curvature at the point by the length of this chord. The resulting normalized curvature does not change with scale — if the silhouette shrinks to half size, the curvature doubles but the chord halves, so the product remains a constant.

2.3.2 View Synthesis of Articulating Humans

Having segmented each input image into convex body parts, we need to render the image for each body part in the given viewing direction and assemble them together. In order to generate each body part separately for the desired view, we have to use the corresponding body part in each input image. Since the body part localization method in previous section does not give such corresponding relationship between views, we can not tell which body part is left arm and which one is right arm from the input silhouette images. In the assembling process, it is possible that the "stitched" final view has unconnected or squeezed regions because the separately-generated virtual parts are not guaranteed to match each other.

To solve these two problems, a virtual silhouette image corresponding to the given viewing direction is first generated using the image based visual hull computed from the input silhouette images. In this process, we only need to decide whether each pixel in the virtual view belongs to the foreground or the background. If a pixel's corresponding 3D ray intersection in the visual hull formulation process is not null, the pixel is marked as a foreground pixel and the intersection coordinates are stored in a table for later use. Each input image is segmented into left arm, right arm and torso (with legs). The rendered silhouette image can also be segmented into body parts in the same way. Since the visual hull of the person has been built, the 3D centroid for each body part can be roughly approximated with the center of gravity of the body part's visual hull. By projecting the 3D centroid to each input image, we are able to locate the corresponding body part in each input image for the rendered body part in the synthetic image.

To map the texture for the foreground pixels in the desired view, a nearest neighbor scheme is used [68]. For each foreground pixel, the 3D closet frontal point is retrieved from the stored table and projected onto each input view. The intensity value P for the desired view pixel is a weighted sum of intensity values P_i of the corresponding pixels in the input views, $P = \sum P_i \cos\theta_i$, where θ_i is the angle between the 3D ray from the desired view foreground pixel and the 3D ray from the corresponding pixel in input view i if the closet frontal point is visible in this view. If the concave regions are not considered in the formulation of the visual hull, the pixels in the desired view projected by the points inside the

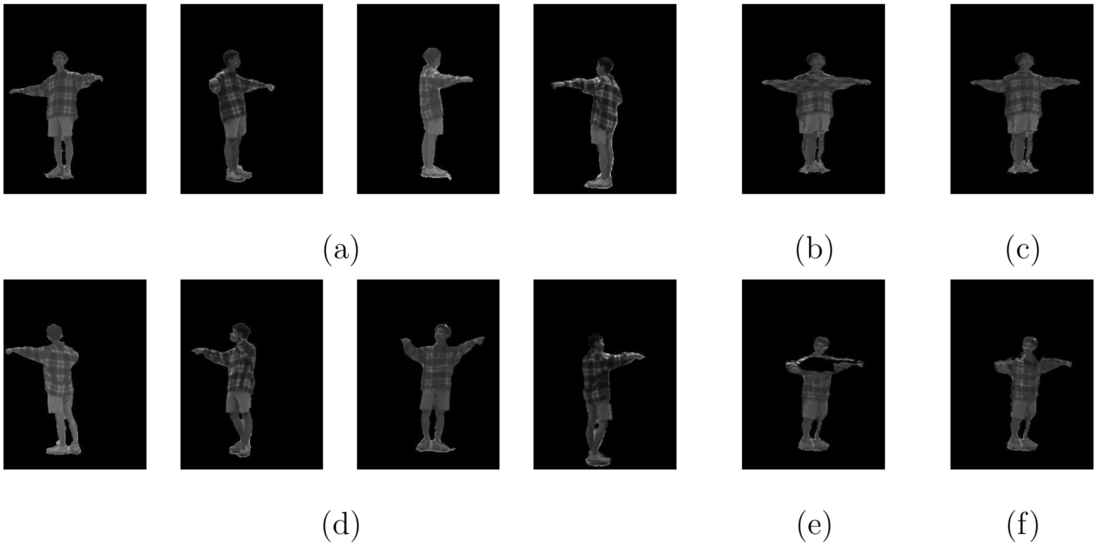


(1.a) (1.b)

(2.a) (2.b)

Figure 2.15: Two examples of human body part segmentation results: (1.a) and (2.a) are the body part segmentation results for input views. (1.b) and (2.b) are the body part segmentation result for the rendered silhouette images.

concavities will have erroneous 3D closet frontal points and their intensity values will be wrong. In order to obtain correct visual hull and texture mapping results, the human body part segmentation method is used in the reconstruction process. For the desired view, each foreground pixel in a segmented body part will have its epipolar line intersected with the corresponding body part contour in each input view. These 2D line intersections are projected back into 3D space and intersect with the retrieved 3D ray starting from the pixel in the desired view. If the pixel is the projection of a 3D point which lies on the concave region, the new 3D ray intersection will be shorter compared to the previously-stored intersection because the epipolar line only intersects with the corresponding body part instead of the whole body contour. Hence, the 3D closet frontal points for these pixels are closer to their correct positions so that their intensity values can be decided with the corresponding pixels in the input views. For the pixels corresponding to the 3D points which do not lie on concave regions, the 3D ray intersections are same as the stored ones. In this way, even if the epipolar line of a pixel in a desired view body


(a) (b) (c)

(d) (e) (f)

Figure 2.16: Two examples of view synthesis of articulating humans with visual hull: (a) and (d) are the input views. (b) and (e) are the texture mapping results without using body part segmentation method. (c) and (f) are the texture mapping results using body part segmentation method.

part has no intersection with the corresponding body part contour in the input views, this pixel is still marked as a foreground pixel and has its intensity value decided using the nearest neighboring scheme. Therefore, no unconnected region will be observed in the assembled view. Since the independently processed body parts are segmented from the previously generated silhouette image, no region will be squeezed together in the assembled view.

The body part segmentation results for four input views and the rendered silhouette image are shown in Fig 2.15. The texture mapping results obtained with and without using the body part segmentation method are compared in Fig 2.16. The hole on the chest part of Fig 2.16 (b) is because the concave region formed by the arms and the torso is treated as a convex region. Since the desired viewing direction is from above the concave region while the input viewing directions are either from below the concave region or make the concave region occluded, so the front-most points corresponding to these pixels are not visible in any of the input views and marked as invisible. From Fig 2.16 (c) and Fig 2.16 (f) we observe that the texture mapping results are greatly improved when body part based method is used. It should be mentioned that if the desired viewing direction makes the rendered image have self occlusion between the limbs and the torso, the rendered image has no obvious improvement compared to the result obtained without using the body part based method.

2.4 Improved Body Part Segmentation For Turntable Image Collection

As mentioned in Section 2.3, the body part segmentation result depends both on the contour and the viewing direction. When the person is not observed from a good viewpoint, the body part segmentation result can be very unreliable due to self-occlusion or the difficulty in detecting the negative minima of curvature, as shown in Fig. 2.17(a) and Fig. 2.18(a). It is not unreasonable to conclude that if we have the images of the person observed from all the viewing directions around him or her (the turntable image collection), we can improve the body part segmentation results effectively as in some views we may better localize the body part positions than in other views. Although the actual available views (the input views) are beyond our control, we are still able to generate the turntable image collection using IBVH technique as we have shown in Section 2.2.1. After obtaining the more reliable body part segmentation points from some virtual views with the algorithm in Section 2.3.1, we have to link them back to the actual available views.

This requires us to find the correspondence of the segmentation points across views, which is not a trivial problem especially for the contour images. This is due to the fact that some points on the contour in one view may be occluded in another view and thus not lie on the contour (boundary). Therefore, the re-projection from the foremost point on the already-computed VH from the available views to the new view may result in the line segment which does not have intersection with the contour.

In order to find the correspondence on the contour points across views, we reexamine the fact that human body is an articulation of rigid body parts. In [62], Ling and Jacobs propose to use the inner distance to build shape descriptors that are robust to articulation and capture part structure. The inner distance is defined as the length of the shortest path between sample points within the shape silhouette. It can be computed using the shortest path algorithm with $O(n^3)$ time complexity for n sample points along the contour. Since the human body can be decomposed into rigid body parts connected by the joints which are assumed very small compared to the parts they connect, the shortest path sample points can be divided into segments within each part. The authors show that the inner distance is articulation insensitive and more effective at capturing part structures than the Euclidean distance. This suggests that the inner-distance can be used as a replacement for the Euclidean distance to build more accurate descriptors for complex shapes, especially for those with articulated parts like a human body.

We need to find the correspondence of the segmentation points from the "good" virtual view to the actual available views. Although only these sparse segmentation points are of our interest, we have to consider all the sample points along the whole contour considering the point ordering constraint. Therefore, it is actually a contour matching problem which is usually stated as follows: Two given contours O and S are described by the sample point sequences, $o_1, o_2, ...,_n$ for O with n points, and $s_1, s_2, ..., s_m$ for D with m points. We seek to match O to S through the mapping M which is from $1, 2, ..., n$ to $0, 1, 2, ..., m$, where o_i is matched to $s_{M(i)}$ if $M(i) \neq 0$ and otherwise left unmatched. M must minimize the match score defined as $C(M) = \sum_{1 \leq i \leq n} c(i, M(i))$ where $c(i, 0)$ is the penalty for unmatched p_i, and $c(i, j)$ is the cost of matching o_i to s_j for $1 \leq j \leq m$. Let $h_{O,i}$ and $h_{S,j}$ be the shape context histograms of o_i and s_j respectively, the cost of matching is measured using the χ^2 statistics $c(i, j) \equiv \frac{1}{2} \sum_{1 \leq k \leq K} \frac{[h_{O,i}(k) - h_{S,j}(k)]^2}{h_{O,i}(k) + h_{S,j}(k)}$, where K is the number of histogram bins. By replacing the Euclidean distance with the inner distance in the definition of shape context [11], the inner distance shape context can be used to accomplish contour mapping through dynamic programming (DP) which is widely used to solve the contour matching problem.

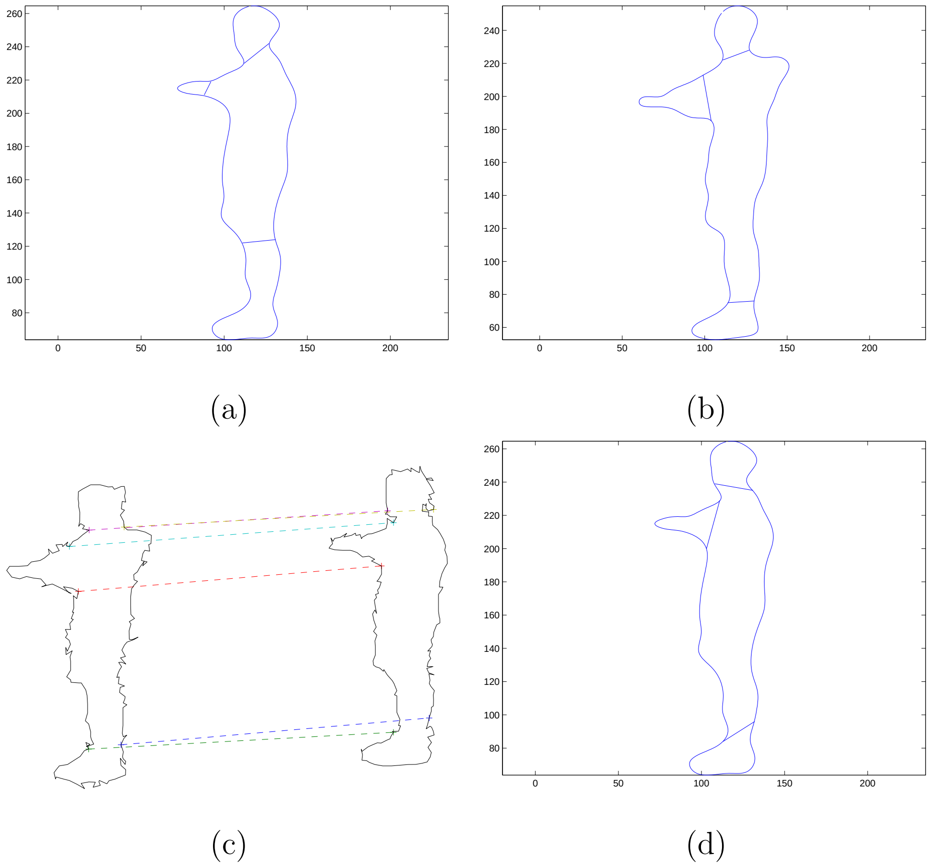
This algorithm works the best for matching contours of the same articulated object at different postures but observed from the same viewing direction. Some matching examples for the contours of a person taken at different time instants from the same video sequence were shown in [62] to demonstrate the effectiveness of the proposed algorithm. It is yet to explore how the algorithm works for matching the contours across viewpoints. We conducted experiments on the very challenging turning and pointing sequence, the results of which are shown in Fig. 2.17 and Fig. 2.18. Fig. 2.17(a) shows the contour image that has poor body part segmentation, and (b) shows the contour image with the good body part segmentation of the same person observed from another viewpoint, which is actually generated using the algorithm shown in Section 2.2.1. We seek to improve the body part segmentation results on (a) by finding the corresponding points of the marked body part segmentation points on the contour image in (b). In order to match these two contours, we use the IDSC+DP technique described in [62]. From the experiments, we found that within a certain range of viewing angles (usually in the neighborhood of 20 degrees from the actual available views), we are able to match the contours quite well. The reason behind this is that to some extent, the structure of the shape does not change dramatically following the change of the viewing direction. Therefore the IDSC descriptor can still capture the main posture of the articulated object (the person in our case). Fig. 2.17(c) shows the contour matching result at body part segmentation points, and Fig. 2.17 (d) shows the improved body part segmentation result for the same contour image as in Fig. 2.17 (a). Another example is shown in Fig. 2.18 with similar performance. As we can observe from the figures, the body part segmentation results are greatly improved.

For the contours observed from the viewing directions beyond this range, the matching errors become large. The main error comes from self-occlusion and foreshortening effects because of different viewing directions. It is still an open topic to match two contours across large viewpoint changes without building a 3D shape or using a generic 3D model.

2.5 Summary and Future Work

We have presented a complete framework combining the active image based visual hull algorithm and a contour based body part segmentation technique for a better synthesis and understanding of the human pose from a limited number of available silhouette images. No 3D body model is explicitly reconstructed. The turntable image collection of the object can be obtained by properly moving a virtual camera on a circular trajectory. We showed how to derive the virtual camera's translation and rotation at each position on the trajectory. The silhouette turning function distance is measured against the pre-stored silhouette images with known poses to get the pose-normalized views for recognition applications. In order to overcome the inability of the visual hull method to reconstruct concave regions, a contour-based human body part localization algorithm is proposed to segment the input silhouette images and the rendered virtual silhouette image into convex body parts. The body parts in the virtual view are generated separately from the corresponding body parts in the input views and then assembled together for a more accurate VH reconstruction. Furthermore, the two components mentioned above improve each other for better performance through the correspondence across viewpoints built via the inner distance shape context measurement.

The original SFC formulation assumes that all of the silhouette images are captured either at the same time or while the object is static. This assumption is violated when the object moves or changes shape. Hence the use of SFC with moving objects has been restricted to treating each time instant sequentially and independently. The temporal continuity of the input video stream has not been fully utilized to reduce the computation burden or capture the motion information. For a moving object, since the motion between the nearby frames is usually small, it is possible to improve the shape approximation by combining multiple silhouette images captured across time. Recall in Section 2.3 we have segmented the input silhouettes into convex body parts. By tracking these body parts in each input view, their motion in the desired view can be inferred under the multi view geometry constraints. In [89] a stochastic body part tracking method is proposed in a Bayesian framework. For initialization, a simple generic 3D human body model can be aligned to be consistent with the given views. For the desired view, after deriving the motion for each body part, we need to estimate the motion for the articulation points. Since these points lie on both the body parts they connect to, they must satisfy the motion equations for both the body parts. With this proposed approach, we should be able to not only dynamically synthesize the desired view, but also catch the motion information for each body part and hence analyze the human activity.



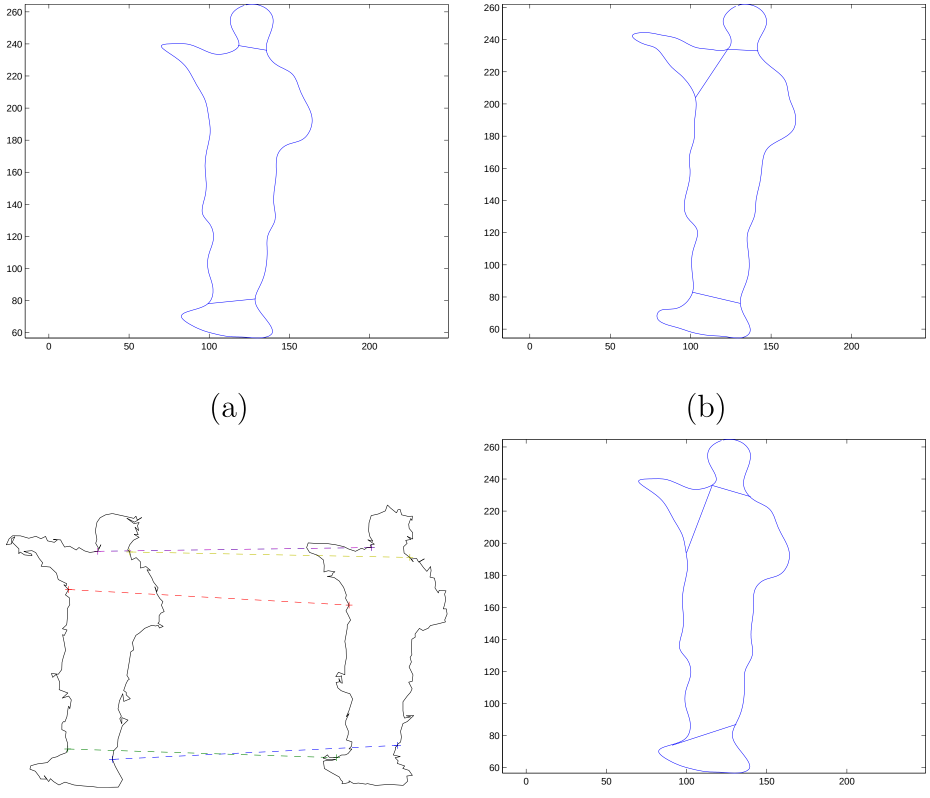
(a)

(b)

(c)

(d)

Figure 2.17: (a) The contour image in the example with poor body part segmentation. (b) the contour image with a good body part segmentation of the same person observed from another viewpoint. (c) the contour matching result using the IDSC+DP technique, with the body part segmentation points explicitly marked. (d) the improved body part segmentation result for the same contour image as in (a).



(a)

(b)

(c)

(d)

Figure 2.18: (a) The contour image for the case of bad body part segmentation. (b) the contour image for the case of good body part segmentation of the same person observed from another viewpoint. (c) the contour matching result using the IDSC+DP technique, with the body part segmentation points explicitly marked. (d) the improved body part segmentation result for the same contour image as in (a).

Chapter 3

Pose-Normalized View Synthesis for Face Recognition Using a Single Image

3.1 Challenges and Prior Art On Face Recognition

Face recognition is one of the most successful applications of image analysis and understanding [121]. Given a database of training images (sometimes called a gallery set, or gallery images), the task of face recognition is to determine the facial ID of an incoming test image. Face recognition under varying pose is a challenging problem, especially when illumination variations are also present. Built upon the success of earlier efforts, recent research has focused on robust face recognition to handle the issue of significant difference between a test image and its corresponding training images (i.e., they belong to the same subject). As pointed out in [121] and many references cited therein, pose and/or illumination variations can cause serious performance degradation to many existing face recognition systems. A review of these two problems and proposed solutions can be found in [121]. Despite significant progress, robust face recognition under varying lighting and different pose conditions remains to be a challenging problem. The problem becomes even more difficult when only one training image per subject is available.

Most earlier methods focused on either illumination or pose alone. For example, an early effort to handle illumination variations is to discard the first few principal components that are assumed to pack most of the energy caused by illumination variations [9]. To handle complex illumination variations more efficiently, spherical harmonics representation has been proposed independently by Basri et al. [7] and Ramamoorthi [75]. It has been shown that the set of images of a convex Lambertian face object obtained under a wide variety of lighting conditions can be approximated by a low-dimensional linear subspace. The basis images spanning the illumination space for each face can be rendered from a 3D scan of the face [7]. Following the statistical learning scheme in [91], Zhang et al. [115] showed that the basis images spanning this space can be recovered from just one image taken under arbitrary illumination conditions for a fixed pose.

To handle the pose problem, a template matching scheme was proposed in [12] that needs many different views per person and no lighting variations are allowed. Approaches for face recognition under pose variations [72] [33] avoid the strict correspondence problem by storing multiple normalized images at different poses for each person. View-based eigenface methods [72] explicitly code the pose information by constructing an individual eigenface for each pose. [33] treats face recognition across poses as a bilinear factorization problem and facial identity and head pose are the two factors.

To handle the combined pose and illumination variations, researchers have proposed several methods. The synthesis method in [34] can handle both illumination and pose variations by reconstructing the face surface using the illumination cone method under a fixed pose and rotating it to the desired pose. The proposed method essentially builds illumination cones at each pose for each person. [120] presented a symmetric shape-from-shading (SFS) approach to recover both shape and albedo for symmetric objects. This work was extended in [28] to recover the 3D shape of a human face using a single image. In [119], a unified approach was proposed to solve the pose and illumination problem. A generic 3D model was used to establish the correspondence and estimate the pose and illumination direction. [123] extended the photometric stereo algorithms to recover albedos and surface normals from one image illuminated by an unknown single distant illumination source.

Building upon the highly successful statistical modeling of 2D face images [24], the authors in [103] propose a 2D + 3D AAM scheme to enhance AAM in handling 3D effects to some extent. A sequence of face images (900 frames) is tracked using AAM and a 3D shape model is constructed using Structure-From-Motion (SFM) algorithms. As camera calibration and 3D reconstruction accuracy can be severely affected when the camera is far away from the subjects, the authors imposed these 3D models as soft constraints for the 2D AAM fitting procedure and showed convincing tracking and image synthesis results on a set of five subjects. However, this is not a true 3D approach with accurate shape recovery and does not handle occlusion.

A 3D morphable face model has been proposed in [15] to handle both pose and illumination variations, where the shape and texture of each face is represented as a linear combination of a set of 3D face exemplars and the parameters are estimated by fitting a morphable model to the input image. By far the most impressive face synthesis results were reported in [15] accompanied by very high recognition rates. One drawback of this approach is that it does not handle complex illumination problem since a single light source is assumed. In order to effectively handle both illumination and pose, a recent work [116] combines spherical harmonics and the morphable model. It works by assuming that shape and pose can be first solved by applying the morphable model and illumination can then be handled by building spherical harmonic basis images at the resolved pose. Most of the 3D morphable model approaches are computationally intense because of the large number of parameters that need to be estimated.

3.2 Pose-Normalized Face Synthesis from a Single Image

The bilateral symmetry of human face has been used by some researchers [120] for 3D modeling and subsequent novel view synthesis. We propose a pose-normalized face synthesis approach from a single view by exploiting the bilateral symmetry of the human face [108]. Given a test image, with different pose and illumination from the training images, we suppose that the pose is obtained by rotating the head about the Y-axis by θ. The mirror image of the original view can be thought of as the head rotated about the Y-axis by $-\theta$ and is under the opposite lighting direction in the X-direction. We show that given pose, illumination parameters and the required correspondence, the mirror view under the same illumination as the original view can be determined on a pixel-by-pixel basis using the original view and its mirror image. Consequently the pose-normalized view under the given illumination can be generated using view morphing techniques [84].

3.2.1 Derivations of Key Equations

Let (p, q) be the partial derivatives of the depth map $z[x, y]$ for the frontal view of the given image and $-\vec{L} = (Ps, Qs, 1)$ be the opposite direction of the single light source. The light source can also be represented by two angles, slant α (the angle between the negative $\vec{L}$ and the positive Z-axis) and tilt τ (the angle between the negative $\vec{L}$ and the $x - z$ plane), and the following expression holds:

$$Ps = k \sin\alpha \cos\tau,\ Qs = k \sin\alpha \sin\tau \tag{3.1}$$

where k is the length of vector $\vec{L}$. The Lambertian model we use here is commonly used in the shape from shading literature, with the standard equation [120]

$$I(x, y) = \rho \frac{1 + pP_s + qQ_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + P_s^2 + Q_s^2}} \tag{3.2}$$

where ρ is the composite albedo.

The partial derivatives $(p[x, y], q[x, y])$ become $(p_\theta[x', y'], q_\theta[x', y'])$ after rotating θ about the Y-axis and they are related by [119]

$$\begin{cases} p_\theta[x', y'] = \tan(\theta + \theta_0) \\ q_\theta[x', y'] = \frac{q[x,y]\cos\theta_0}{\cos(\theta+\theta_0)} \end{cases} \tag{3.3}$$

where $\tan\theta_0 = p[x, y]$.

By reversing the pixel order on each row of the original view I, we obtain the *mirror image* M. M can be thought as the head rotated about the Y-axis by $-\theta$

under illumination direction $(-P_s, Q_s, 1)$. So the partial derivatives $(p[x,y], q[x,y])$ become $(p_{-\theta}[x',y'], q_{-\theta}[x',y'])$ after the rotation and they are related by

$$\begin{cases} p_{-\theta}[x',y'] = \tan(-\theta+\theta_0) \\ q_{-\theta}[x',y'] = \frac{q[x,y]\cos\theta_0}{\cos(-\theta+\theta_0)} \end{cases} \tag{3.4}$$

The *mirror view*, denoted as S, is the view with the same pose as M but under the same lighting condition as I. We are interested in finding out how S relates to I and M.

If the pose θ and the illumination direction (α, τ) have been estimated and the correspondence between I and M has been established, then we have

$$I(x,y) = \rho \frac{1 + p_\theta P_s + q_\theta Q_s}{\sqrt{1+p_\theta^2+q_\theta^2}\sqrt{1+P_s^2+Q_s^2}} \tag{3.5}$$

and

$$M(x,y_m) = \rho \frac{1 - p_{-\theta} P_s + q_{-\theta} Q_s}{\sqrt{1+p_{-\theta}^2+q_{-\theta}^2}\sqrt{1+P_s^2+Q_s^2}} \tag{3.6}$$

where (x, y_m) is the corresponding point in M for pixel (x,y) in I.

By substituting (3.3) and (3.4) into (3.5) and (3.6) we have

$$I(x,y) = \rho \frac{\cos(\theta+\theta_0) + \sin(\theta+\theta_0)P_s + q\cos\theta_0 Q_s}{\sqrt{1+q^2\cos^2\theta_0}\sqrt{1+P_s^2+Q_s^2}} \tag{3.7}$$

and

$$M(x,y_m) = \rho \frac{\cos(-\theta+\theta_0) - \sin(-\theta+\theta_0)P_s + q\cos\theta_0 Q_s}{\sqrt{1+q^2\cos^2\theta_0}\sqrt{1+P_s^2+Q_s^2}} \tag{3.8}$$

From (3.7) and (3.8) we obtain

$$I(x,y) + M(x,y_m) = 2\rho \frac{\cos\theta_0(\cos\theta + P_s\sin\theta) + q\cos\theta_0 Q_s}{\sqrt{1+q^2\cos^2\theta_0}\sqrt{1+P_s^2+Q_s^2}} \tag{3.9}$$

$$I(x,y) - M(x,y_m) = 2\rho \frac{\sin\theta_0(P_s\cos\theta - \sin\theta)}{\sqrt{1+q^2\cos^2\theta_0}\sqrt{1+P_s^2+Q_s^2}} \tag{3.10}$$

Since the mirror view S has the same pose as M, it has the same correspondences with I as M has. S only differs from M in illumination direction $(P_s, Q_s, 1)$. So we have

$$\begin{aligned} S(x, y_m) &= \rho \frac{1 + p_{-\theta} P_s + q_{-\theta} Q_s}{\sqrt{1 + p_{-\theta}^2 + q_{-\theta}^2}\sqrt{1 + P_s^2 + Q_s^2}} \\ &= \rho \frac{\cos(-\theta + \theta_0) + \sin(-\theta + \theta_0) P_s + q \cos\theta_0 Q_s}{\sqrt{1 + q^2 \cos^2\theta_0}\sqrt{1 + P_s^2 + Q_s^2}} \\ &= \rho \frac{\cos\theta_0 (\cos\theta - P_s \sin\theta + q Q_s)}{\sqrt{1 + q^2 \cos^2\theta_0}\sqrt{1 + P_s^2 + Q_s^2}} + \rho \frac{\sin\theta_0 (\sin\theta + P_s \cos\theta)}{\sqrt{1 + q^2 \cos^2\theta_0}\sqrt{1 + P_s^2 + Q_s^2}} \end{aligned} \tag{3.11}$$

From (3.9) and (3.10) we obtain

$$\begin{cases} \frac{\cos\theta_0}{\sqrt{1+q^2\cos^2\theta_0}\sqrt{1+P_s^2+Q_s^2}} = \frac{I(x,y)+M(x,y_m)}{2\rho(\cos\theta + P_s \sin\theta + qQ_s)} \\ \frac{\sin\theta_0}{\sqrt{1+q^2\cos^2\theta_0}\sqrt{1+P_s^2+Q_s^2}} = \frac{I(x,y)-M(x,y_m)}{2\rho(P_s \cos\theta - \sin\theta)}. \end{cases} \tag{3.12}$$

Substitution of (3.12) into (3.11) yields

$$\begin{aligned} S(x, y_m) = & \frac{(I(x,y) + M(x,y_m))(\cos\theta - P_s \sin\theta + qQ_s)}{2(\cos\theta + P_s \sin\theta + qQ_s)} \\ & + \frac{(I(x,y) - M(x,y_m))(P_s \cos\theta + \sin\theta)}{2(P_s \cos\theta - \sin\theta)} \end{aligned} \tag{3.13}$$

In order to estimate the illumination direction $(Ps, Qs, 1)$, i.e., (α, τ), the illumination estimation method in [122] is used. We present a method for estimating the head pose and the correspondence between I and M in Section 3.3. Having the pose, the illumination and the correspondence between I and M, we are able to synthesize the mirror view S pixelwise using the intensities of I and M. The frontal view of the given probe image under the same lighting condition can be easily rendered with view morphing techniques [84]. And the recognition task can be accomplished using the methods in [10, 50] since the pose is now fixed.

3.2.2 Some Discussions

(3.13) is composed of two terms. The first term is $(I(x,y) + M(x,y_m))/2$ times a coefficient in which the unknown surface normal component q plays a limited role. Since P_s is usually less than 1 if the length of $\vec{L}$ is normalized to 1, and $\sin\theta$ is very small when θ is small, $P_s \sin\theta$ is small compared to $\cos\theta + qQ_s$ and so can be neglected. Actually for each pixel (x,y), $q(x,y)$ is the partial derivative of the depth in the y direction and usually in a very small range for most of the human face points. So the coefficient of $(I(x,y)+M(x,y_m))/2$ is close to 1, or we can use a reasonable constant to replace $q(x,y)$ for every pixel (x,y). Experimental results show that different constants selected for $q(x,y)$ do not make much difference as long as they are in reasonable range of values for a human face. This first term actually captures the common part in S and I, which is not significantly affected by the pose but by the illumination, especially by Q_s.

The second term in (3.13) is $(I(x,y) - M(x,y_m))/2$ times a coefficient which depends only on θ and P_s and is a constant for every pixel (x,y_m) in S. This term actually captures the difference caused by the pose and opposing lighting conditions. One thing to mention here is that this method will be very sensitive to noise if P_s is close to $\tan\theta$. From (18) we can see that the coefficient of this term has the denominator $(P_s\cos\theta - \sin\theta)$. If P_s is close to $\tan\theta$, this denominator will be close to zero. So any small noise in $I(x,y) - M(x,y_m)$ will be enlarged by the denominator lowering the quality of the synthesized view S. When P_s is close to $\tan\theta$, it means that the illumination is frontal with respect to the rotated pose θ. Thus when the mirror image M is obtained, the opposite lighting source in X direction is also frontal to the mirror pose. Therefore the intensity for the corresponding pixels in the two views are same for all the pixels, which means in

(3.13) the second term is zero for all the pixels and does not contribute to the synthesis.

Note for each pixel pair in I and M, there are only two observations $I(x, y)$ and $M(x, y_m)$ but three unknowns $\rho(x, y)$, $[p(x, y), q(x, y)]$. It is impossible to solve them explicitly. In (3.13), $q(x, y)$ is approximated with a reasonable constant because it varies least among the three unknowns. Compared with the existing single view based synthesis methods [14,33,77,119], the one proposed here is view based and only uses the information from the given image. Neither a 3D model nor a linear combination of other faces is needed.

3.3 Finding Correspondence and Pose Estimation

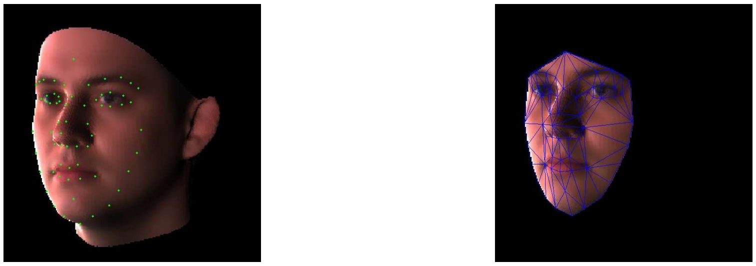
3.3.1 Finding Correspondence

For the pose normalized view synthesis method, we need to build the correspondence between I and M. Establishing automatic correspondence is always a challenging problem. Recently, promising results have been shown by using the 4 planes, 4 transitions stereo matching algorithm described in [25]. The disparity map can be reliably built for a pair of images of the same person taken under the same lighting conditions, even with some occlusions. We conducted some experiments using this technique on both synthetic and real images. Reasonably good correspondence maps were achieved, even for cross-subject images. This technique has been used for 2D face recognition across pose [18]. However, like all the other stereo methods, the pixel intensities across views are assumed to be same, which does not hold if the images are taken under different lighting conditions. For arbitrary face recognition application, the lighting condition of the test image is uncontrollable. Therefore, currently this stereo method can not be directly used to build the correspondence between I and M. Further investigations are being done for dense stereo with illumination variations compensated.

Although a necessary component of the algorithm, finding correspondence is not the main focus of our research. Like most approaches that handle pose variations, we use sparse main facial features to build the dense cross-pose or cross-subject correspondence [115]. Although automatic facial feature detection/selection techniques are available, but most of them are not robust enough to reliably detect the facial features from images at arbitrary poses and taken under arbitrary lighting conditions. For now we manually pick sixty three designated feature points (eyebrows, eyes, nose, mouth and the face contour) on I at the arbitrary pose. These feature points are selected in a bilateral symmetric manner so that no extra work needs to be done on the mirror image M. Triangular meshes on both faces were constructed and barycentric interpolation inside each triangle was used to find the dense correspondence. Using this method, the corresponding point (x, y_m) in M for pixel (x, y) in I is easily built. The number of feature points needed in our approach is comparable to the 56 manually picked feature points in [115] to deform the 3D model. Fig. 3.1 illustrates the selected facial feature points and the constructed triangular meshes to build the dense correspondence map.

3.3.2 Head Pose Estimation

Estimating head pose from a single face image is an active research topic in computer vision. Either a generic 3D face model or several main facial features are utilized to estimate the head pose. Since we already have the feature points to



(a) The selected feature points (b) The constructed triangular mesh

Figure 3.1: Designated facial feature points are selected to build the correspondence between I and M. (a) The selected feature points. (b) The constructed triangular mesh.

build the correspondence across views, it is natural to use these feature points for pose estimation. In [42], five main facial feature points (four eye corners and the tip of the nose) are used to estimate the 3D head orientation. The approach employs the projective invariance of the cross-ratios of the eye corners and anthropometric statistics to determine the head yaw, roll and pitch angles. The focal length f is assumed to be known, which is not always available for the uncontrollable test image. In order to remove this requirement, we first calculate an average face at the frontal pose using images generated from Vetter's 3D face database [1], with the main facial feature points selected. We then re-size this frontal view average face and the facial features to the same scale as I and M. Next we estimate the head pose without knowing f. All notations follow those in [42].

Let (u_2, u_1, v_1, v_2) be the image coordinates of the four eye corners, and D and D_1 denote the width of the eyes and half of the distance between the two inner eye corners respectively. From the well known projective invariance of the cross ratios we have $I_1 = \dfrac{(u_2 - u_1)(v_1 - v_2)}{(u_2 - v_1)(u_1 - v_2)} = \dfrac{D^2}{(2D_1 + D)^2}$ which yields $D_1 = \dfrac{DQ}{2}$

where $Q = \dfrac{1}{\sqrt{I_1} - 1}$. In order to recover the yaw angle θ (around the Y-axis), it is easy to have, as shown in [42], that $\theta = arctan\dfrac{f}{(S+1)u_1}$, where f is the focal length and S is the solution to the equation $\dfrac{\Delta u}{\Delta v} = -\dfrac{(S-1)(S-(1+2/Q))}{(S+1)(S+1+2/Q)}$ where $\Delta u = u_2 - u_1$ and $\Delta v = v_1 - v_2$. Assume that u_1^f is the inner corner of one of the eyes for the front-view mean face. With perspective projection, we have $u_1^f = \dfrac{fD_1}{Z}$ and $u_1 = \dfrac{fX_1}{Z+Z_1} = \dfrac{fD_1 \cos\theta}{Z + D_1 \sin\theta}$. Thus,

$$f = (S+1)u_1 \tan\theta \tag{3.14}$$

Then we have $S = \dfrac{u_1}{u_1^f}\dfrac{(S+1)}{\cos\theta}$, which gives

$$\theta = arccos\frac{(S+1)}{S}\frac{u_1}{u_1^f} \tag{3.15}$$

In [42], the pitch β (around the X-axis) is shown to be $\beta = arcsin(E)$ with $E = \dfrac{f}{p_0(p_1^2+f^2)}[p_1^2 \pm \sqrt{(p_0^2p_1^2 - f^2p_1^2 + f^2p_0^2)}]$, where p_0 denotes the projected length of the bridge of the nose when it is parallel to the image plane, and p_1 denotes the observed length of the bridge of the nose at the unknown pitch β. Anthropometric statistics is employed in [42] to get p_0. With the facial features on the mean face at the front-view available, we do not need the anthropometric statistics. p_0 is just the length between the upper mid-point of the nose and the tip of the nose for the front-view mean face. So we can directly use this value and the estimated focal length f in (3.14) to get the pitch angle β.

The head pose estimation algorithm is tested on both synthetic and real images. For synthetic images, we use Vetter's 3D face database. The 3D face model for each subject is rotated to the desired angle and projected onto the 2D image plane. Four eye corners and the tip of the nose are used to estimate the head pose. The mean and standard deviation of the estimated poses are listed in Table 3.1. For

Table 3.1: The mean and standard deviation (Std) of the estimated pose for images from Vetter's database.

Rotation angles	($\theta = 30^o$,$\beta = 0^o$)	($\theta = 30^o$,$\beta = -20^o$)	($\theta = -30^o$,$\beta = 0^o$)	($\theta = -30^o$,$\beta = 20^o$)
Mean of the estimated pose	($\theta = 28^o$,$\beta = 2^o$)	($\theta = 31^o$,$\beta = -23^o$)	($\theta = -32^o$,$\beta = 1^o$)	($\theta = -33^o$,$\beta = 22^o$)
Std of the estimated pose	(3.2^o,3.1^o)	(3.9^o,4.2^o)	(3.4^o,2.7^o)	(4.2^o,4.5^o)

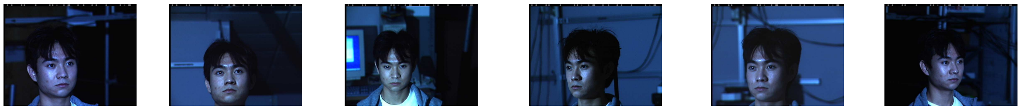
real images, we use the CMU-PIE [90] database which contains face images of 68 subjects at 13 different poses and under 43 different illumination conditions. The ground truth of the head pose can be obtained from the available 3D locations of the head and the cameras. The experiments are conducted for all 68 subjects in the CMU-PIE database at six different poses, illustrated in Fig 3.2 with the ground truth of the pose shown beside each pose index. The mean and standard deviation of the estimated poses are listed in Table 3.2. Overall the pose estimation results are satisfactory and we believe that the relatively large standard deviation is due to some unavoidable error in selecting the facial features.

3.4 Experimental Results

The pose normalized view synthesis algorithm has been implemented and tested on the face images of 50 different subjects from Vetter's database, each under various illuminations and with pose $-10°$. The unknown surface normal component $q(x, y)$ is set to -0.5 for all pixels. Good synthesis results have been observed as shown in

Table 3.2: The mean and standard deviation (Std) of the estimated pose for images from the CMU-PIE database.

Pose index	c05	c07	c09	c11	c29	c37
Mean of the estimated pose	$\theta = 15^o$	$\beta = 11^o$	$\beta = -15^o$	$\theta = -36^o$	$\theta = -17^o$	$\theta = 35^o$
Std of the estimated pose	4.1^o	3.8^o	4.0^o	6.2^o	3.3^o	5.4^o



c05	c07	c09	c11	c29	c37
$(\theta = 16^o)$	$(\beta = 13^o)$	$(\beta = -13^o)$	$(\theta = -32^o)$	$(\theta = -17^o)$	$(\theta = 31^o)$

Figure 3.2: An illustration of the pose variation in part of the CMU-PIE database, with the ground truth of the pose shown beside each pose index. Four of the cameras (c05, c11, c29, and c37) sweep horizontally, and the other two are above (c09) and below (c07) the central camera respectively.

Fig. 3.3 and Fig. 3.4. In both figures, the first row shows the given probe image, under the illumination of ($\alpha = 30, \tau = 30$), ($\alpha = 60, \tau = 30$), ($\alpha = 60, \tau = 60$), ($\alpha = 30, \tau = 150$), ($\alpha = 60, \tau = 120$), and ($\alpha = 60, \tau = 165$) for each column respectively. The second row is the ground truth for the mirror view of the given image, to be compared with the third row which shows the synthesized mirror view. The fourth and fifth rows present the ground truth and the synthesized frontal view under the given illumination respectively. In Table 3.3 we give the average PSNR of the synthesis results, with the first column showing the illumination condition, the second and third column showing the average PSNR of the synthesized mirror view and frontal view respectively. We can see from Table 3.3 that the proposed approach gives decent synthesis results, considering only one probe image is given for each illumination condition. The synthesized frontal views have higher PSNRs than the synthesized mirror views because the frontal views are obtained using view morphing techniques by linearly interpolating the given images and the synthesized mirror views. As another example, Fig. 3.5 and Fig. 3.6 show the synthesis results for images taken under various illuminations and with pose -30°. Better synthesis results are expected if more accurate self-correspondences can be established.

3.5 Summary and Future Work

We have described a pose-normalized object synthesis method, which handles both non-frontal pose and non-frontal illumination, from a single image. It is a pixelwise view-based synthesis scheme and easy to implement. Experimental results show that the proposed method is good under various illuminations.

The quality of the synthesized view using the proposed method highly depends on the accuracy of pose estimation result. Four eye corners have to be extracted

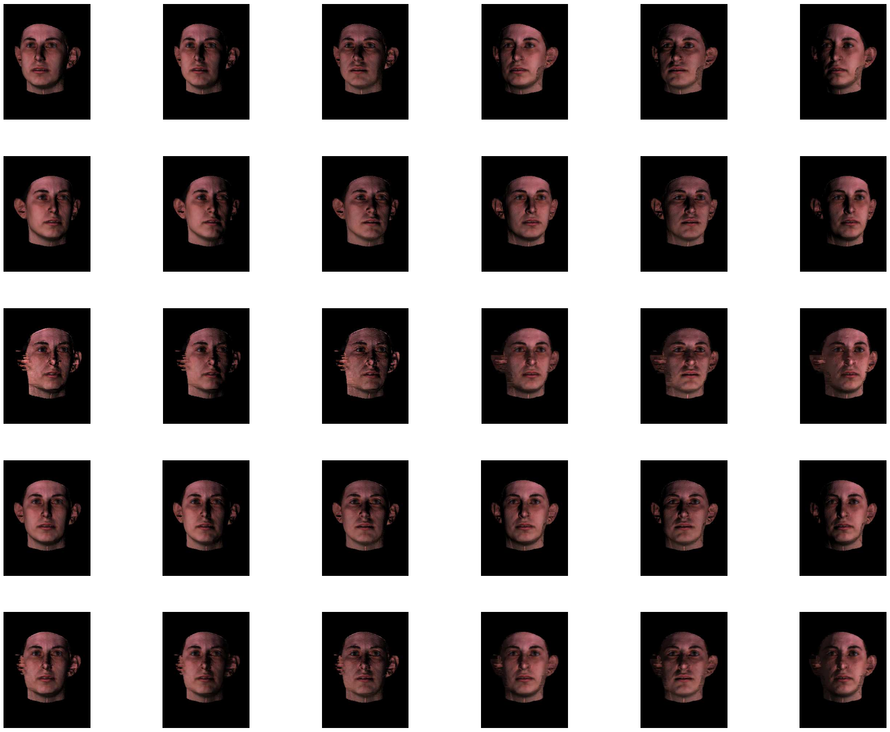


Figure 3.3: The pose-normalized view synthesis results. First row: the given probe image with pose 10°, under the illumination of ($\alpha = 30, \tau = 30$), ($\alpha = 60, \tau = 30$), ($\alpha = 60, \tau = 60$), ($\alpha = 30, \tau = 150$), ($\alpha = 60, \tau = 120$), and ($\alpha = 60, \tau = 165$) for each column respectively. Second and third row: the ground truth and the synthesized mirror view respectively. Fourth and fifth row: the ground truth and the synthesized frontal view under the given illumination.

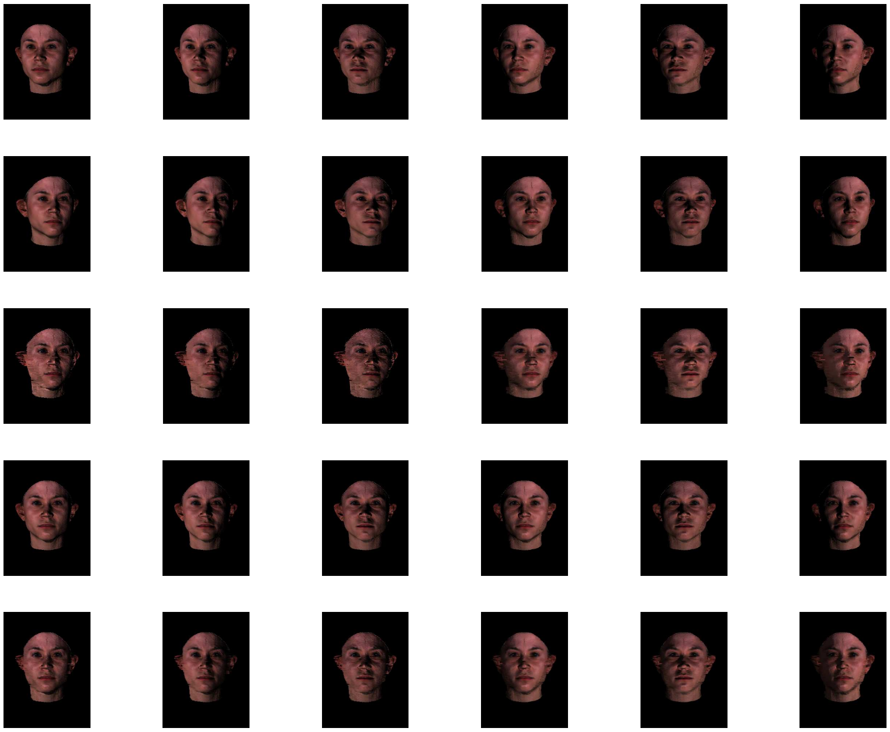


Figure 3.4: The pose-normalized view synthesis results. First row: the given probe image with pose 10°, under the illumination of ($\alpha = 30, \tau = 30$), ($\alpha = 60, \tau = 30$), ($\alpha = 60, \tau = 60$), ($\alpha = 30, \tau = 150$), ($\alpha = 60, \tau = 120$), and ($\alpha = 60, \tau = 165$) for each column respectively. Second and third row: the ground truth and the synthesized mirror view respectively. Fourth and fifth row: the ground truth and the synthesized frontal view under the given illumination.

Table 3.3: The average PSNR of the synthesis results, with the first column showing the illumination condition, the second and third column showing the average PSNR of the synthesized mirror view and frontal view respectively.

Illumination	Average PSNR (mirror view)	Average PSNR (frontal view)
$(\alpha = 30, \tau = 30)$	24.881 db	27.612 db
$(\alpha = 30, \tau = 45)$	24.908 db	27.669 db
$(\alpha = 30, \tau = 60)$	25.380 db	27.975 db
$(\alpha = 30, \tau = 75)$	26.016 db	28.362 db
$(\alpha = 60, \tau = 30)$	26.359 db	28.391 db
$(\alpha = 60, \tau = 45)$	26.098 db	28.408 db
$(\alpha = 60, \tau = 60)$	25.216 db	28.156 db
$(\alpha = 60, \tau = 75)$	26.321 db	28.901 db
$(\alpha = 30, \tau = 120)$	29.170 db	29.799 db
$(\alpha = 30, \tau = 135)$	28.839 db	29.644 db
$(\alpha = 30, \tau = 150)$	28.560 db	29.473 db
$(\alpha = 30, \tau = 165)$	28.415 db	29.338 db
$(\alpha = 60, \tau = 120)$	28.024 db	29.924 db
$(\alpha = 60, \tau = 135)$	27.006 db	29.390 db
$(\alpha = 60, \tau = 150)$	26.489 db	28.978 db
$(\alpha = 60, \tau = 165)$	26.430 db	28.783 db

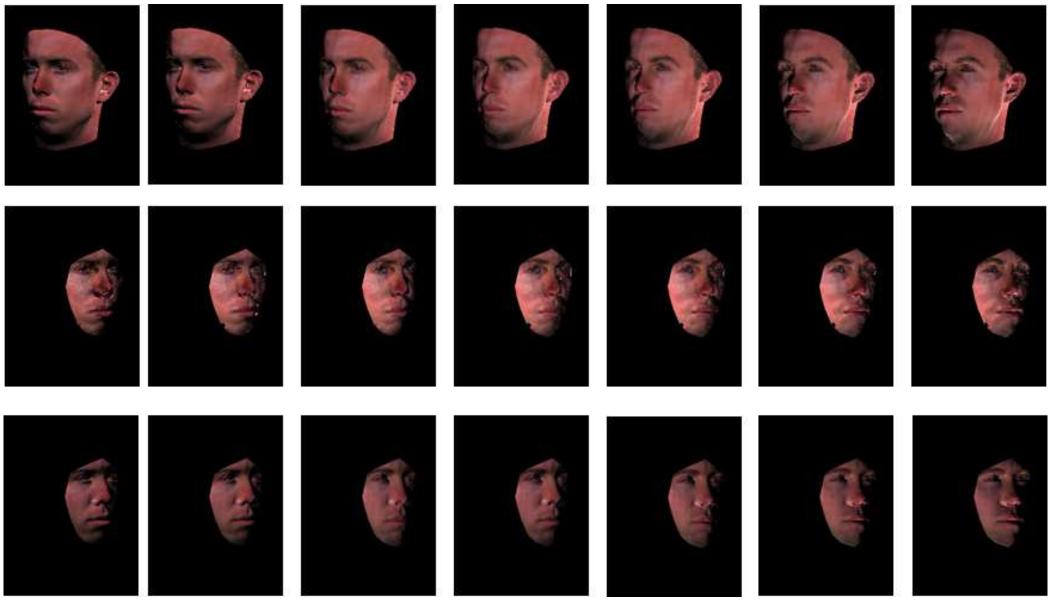


Figure 3.5: The pose-normalized view synthesis results. First row: the given probe image with pose 30°, under various illumination conditions. Second and third rows: the ground truth and the synthesized mirror view respectively.

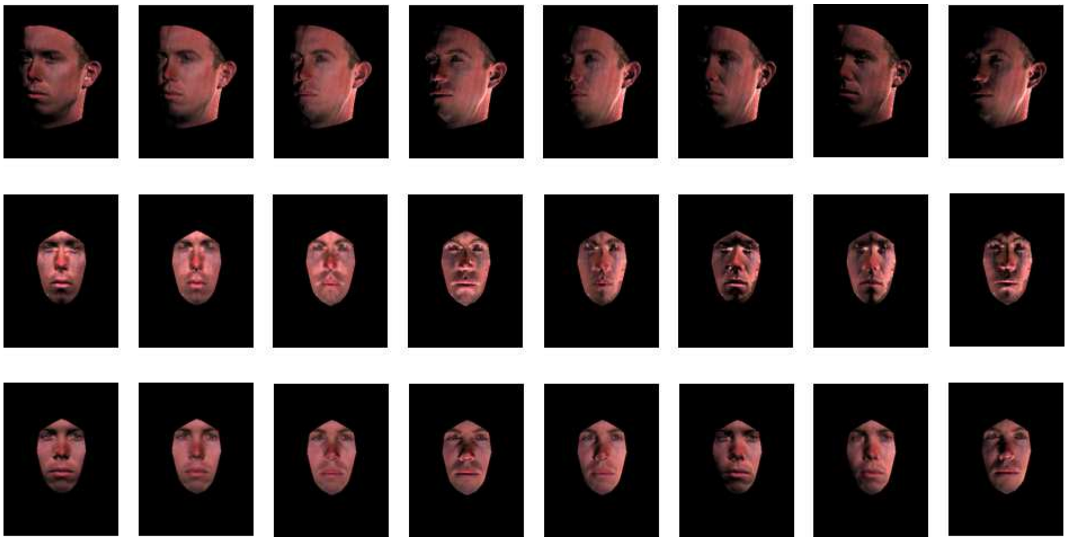


Figure 3.6: The pose-normalized view synthesis results. First row: the given probe image with pose 30°, under various illumination conditions. Second and third rows: the ground truth and the synthesized frontal view under the given illumination.

accurately to estimate the head pose, which is not easy to achieve especially for images taken in poorly illuminated environments. Robust facial feature detection algorithms are being sought for a better pose estimation. In order to build the dense correspondence between the given image and its mirror image more accurately, we will further investigate the dense stereo algorithm in [25] to compensate for illumination variations for images under arbitrary lighting condition.

Chapter 4

Pose-Encoded Spherical Harmonics for Robust Face Recognition Using a Single Image

Recently, methods have been proposed to handle the combined pose and illumination problem when only one training image is available, for example, the methods based on morphable models [15] and their extensions [116] that propose to handle complex illumination problem by integrating spherical harmonics representation [7, 75]. In these methods, either arbitrary illumination conditions cannot be handled [15] or expensive computations of harmonics basis images is required for each pose per subject [116].

Under the assumption of Lambertian reflectance, the spherical harmonics representation has proved to be effective in modeling illumination variations for a fixed pose. In this chapter, we propose to extend the harmonics representation to encode pose information. We utilize the fact that all the harmonic basis images of a subject at various poses are related to each other via closed-form linear transformations [48, 76], and give a more convenient transformation matrix to analytically synthesize basis images of a subject at various poses from just one set of basis images at a fixed pose, say, the frontal view. We prove that the given transformation matrix is consistent with the general rotation matrix of spherical harmonics. According to the theory of spherical harmonics representation [7, 75], this implies that we can easily synthesize from one image under a fixed pose and lighting to any images under different poses and arbitrary lightings. Moreover, these linear transformations are orthonormal. This suggests that recognition methods based on projection onto fixed-pose harmonic basis images [7] for test images under the same pose can be easily extended to handle test images under various poses and illuminations. In other words, our method does not require the time-consuming procedure of building a new set of basis images at the same pose as that of the test image. Instead, we can warp the test image to the same pose as that of the existing basis images and perform recognition. The impact of some empirical factors (i.e., correspondence and interpolation) due to the warping is embedded in a sparse transformation matrix, and we prove that the recognition performance is not adversely affected after warping to the front view [109, 110].

Briefly, we propose an efficient face synthesis and recognition method that needs only one single training image per subject for novel view synthesis and robust recognition of faces under variable illuminations and poses. The flow chart of our face recognition system is shown in Fig. 4.1. We have a single training image at the frontal pose for each subject in the training set. The basis images for each training subject are recovered using a statistical learning algorithm [115] with the aid of a bootstrap set consisting of 3D face scans. For a test image at a rotated pose and under an arbitrary illumination condition, we first establish the image correspondence between the test image and a mean face image at the frontal pose. The frontal view image is then synthesized from the test image. A face is identified for which there exists a linear reconstruction based on basis images that is the closest to the test image. Furthermore, the user is given the option to visualize the recognition result by viewing the images of the chosen subject at the same pose as the test image. Specifically, we can generate novel images of the chosen subject at the same pose as the test image by using the close-form linear transformation between the harmonic basis images of the subject across poses. The pose of the test image is estimated from a few manually selected facial features. The novel image of the chosen subject is then easily synthesized for any given transformation coefficients.

We present results of our face recognition method on both synthetic and real images. For synthetic images, we generate the training images at the frontal pose and under various illumination conditions, and the test images at different poses, under arbitrary lighting conditions, all using Vetter's 3D face database [1]. For real images, we use the CMU-PIE [90] database. The test images are at six different poses and under twenty one different lighting sources. High recognition rates are achieved on both synthetic and real test images using the proposed algorithm.

4.1 Pose-Encoded Spherical Harmonics

The spherical harmonics are a set of functions that form an orthonormal basis for the set of all square-integrable functions defined on the unit sphere [7]. Let L denote the distant lighting distribution. By neglecting the cast shadows and near-field illumination, the irradiance E is then a function of the surface normal n only, and is given by an integral over the upper hemisphere $\Omega : E(n) = \int L(\omega)(n \cdot \omega)d\omega$.

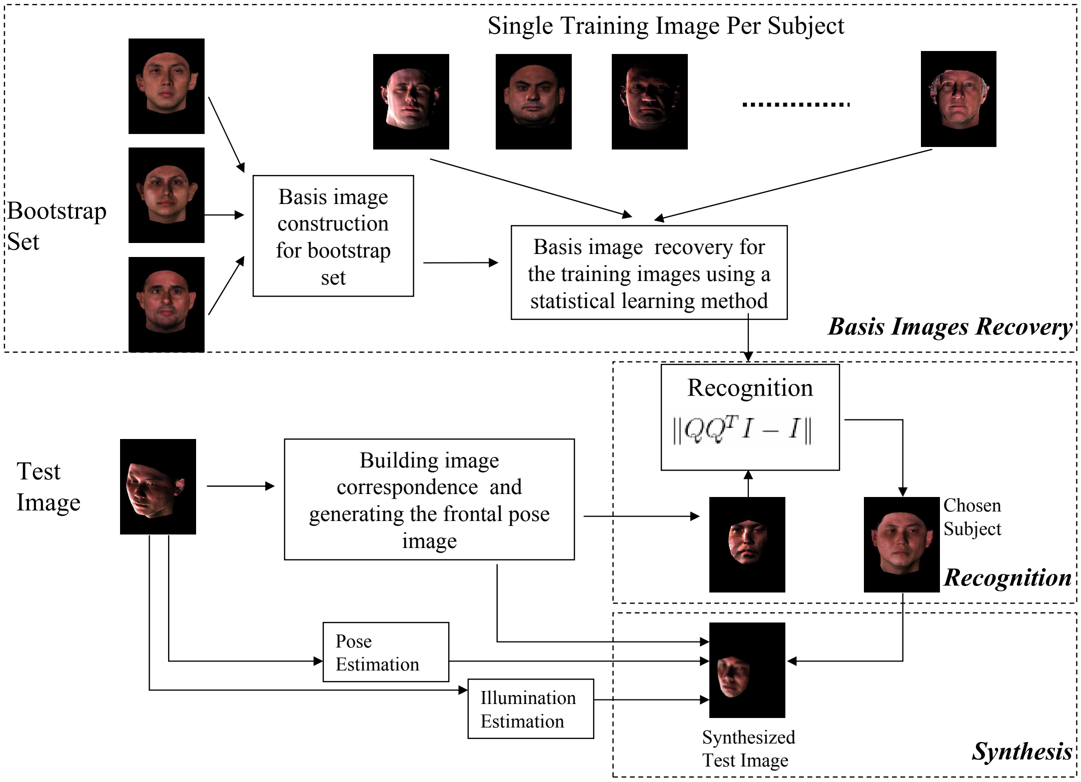

Single Training Image Per Subject
Bootstrap Set
Basis image construction for bootstrap set
Basis image recovery for the training images using a statistical learning method
Basis Images Recovery
Recognition
$\|QQ^T I - I\|$
Test Image
Building image correspondence and generating the frontal pose image
Chosen Subject
Recognition
Pose Estimation
Illumination Estimation
Synthesized Test Image
Synthesis

Figure 4.1: The flow chart of the proposed face recognition system.

We then scale E by the surface albedo λ to find the radiosity I, which corresponds to the image intensity directly: $I(p;n) = \lambda(p)E(n)$.

Any image of a Lambertian object under certain illumination conditions is a linear combination of a series of spherical harmonic basis images $\{b_{lm}\}$. In order to generate the basis images for the object, 3D information is required. The harmonic basis image intensity of a point p with surface normal $n = (n_x, n_y, n_z)$ and albedo λ can be computed as the combination of the first nine spherical harmonics, shown in (1), where $n_{x^2} = n_x n_x$. n_{y^2}, n_{z^2}, n_{xy}, n_{xz}, n_{yz} are defined similarly. $\lambda. * t$ denotes the component-wise product of λ with any vector t. The superscripts e and o denote the even and the odd components of the harmonics respectively.

$$b_{00} = \frac{1}{\sqrt{4\pi}}\lambda,\ b_{10} = \sqrt{\frac{3}{4\pi}}\lambda. * n_z,\ b_{11}^e = \sqrt{\frac{3}{4\pi}}\lambda. * n_x,\ b_{11}^o = \sqrt{\frac{3}{4\pi}}\lambda. * n_y,$$

$$b_{20} = \frac{1}{2}\sqrt{\frac{5}{4\pi}}\lambda. * (2n_{z^2} - n_{x^2} - n_{y^2}),\ b^e_{21} = 3\sqrt{\frac{5}{12\pi}}\lambda. * n_{xz},\ b^o_{21} = 3\sqrt{\frac{5}{12\pi}}\lambda. * n_{yz},$$
$$b^e_{22} = \frac{3}{2}\sqrt{\frac{5}{12\pi}}\lambda. * (n_{x^2} - n_{y^2}),\ b^o_{22} = 3\sqrt{\frac{5}{12\pi}}\lambda. * n_{xy} \tag{4.1}$$

Given a bootstrap set of 3D models, the spherical harmonics representation has proved to be effective in modeling illumination variations for a fixed pose, even when only one training image per subject is available [115]. In the presence of both illumination and pose variations, two possible approaches can be taken. One is to use a 3D morphable model to reconstruct the 3D model from a single training image and then build spherical harmonic basis images at the pose of the test image [116]. Another approach is to require multiple training images at various poses in order to recover the new set of basis images at each pose. However, multiple training images are not always available and a 3D morphable model-based method could be computationally expensive. As for efficient recognition of a rotated test image, a natural question to ask is: can we represent the basis images at different poses using one set of basis images at a given pose, say, the frontal view? The answer is yes, as the 2D harmonic basis images at different poses are related by close-form linear transformations. This enables an analytic method for generating new basis images at poses different from that of the existing basis images.

Rotations of spherical harmonics have been studied by researchers [48, 76] and it can be shown that rotations of spherical harmonic with order l are linearly composed entirely of other spherical harmonics of the same order. In terms of group theory, the transformation matrix is the $(2l + 1)$-dimensional representation of the rotation group $SO(3)$ [76]. Let $Y_{l,m}(\gamma, \phi)$ be the spherical harmonic, the general rotation formula of spherical harmonic can be written as $Y_{l,m}(R_{\theta,\alpha,\beta}(\gamma, \phi)) = \sum_{m'=-l}^{l} D^l_{mm'}(\theta, \alpha, \beta) Y_{l,m'}(\gamma, \phi)$. This means that for each order l, D^l is a matrix that tells us how a spherical harmonic transforms under rotation. The transformation is found to have the following block diagonal sparse form:

$$\left\{\begin{array}{ccl} Y'_{0,0} & = & D^0_{00}Y_{0,0} \\ \begin{bmatrix} Y'_{1,-1} \\ Y'_{1,0} \\ Y'_{1,1} \end{bmatrix} & = & \begin{bmatrix} D^1_{-1,-1} & D^1-1,0 & D^1-1,1 \\ D^1_{0,-1} & D^1_{0,0} & D^1_{0,1} \\ D^1_{1,-1} & D^1_{1,0} & D^1_{1,1} \end{bmatrix} \begin{bmatrix} Y_{1,-1} \\ Y_{1,0} \\ Y_{1,1} \end{bmatrix} \\ \begin{bmatrix} Y'_{2,-2} \\ Y'_{2,-1} \\ Y'_{2,0} \\ Y'_{2,1} \\ Y'_{2,2} \end{bmatrix} & = & \begin{bmatrix} D^2_{-2,-2} & D^2_{-2,-1} & D^2_{-2,0} & D^2_{-2,1} & D^2_{-2,2} \\ D^2_{-1,-2} & D^2_{-1,-1} & D^2_{-1,0} & D^2_{-1,1} & D^2_{-1,2} \\ D^2_{0,-2} & D^2_{0,-1} & D^2_{0,0} & D^2_{0,1} & D^2_{0,2} \\ D^2_{1,-2} & D^2_{1,-1} & D^2_{1,0} & D^2_{1,1} & D^2_{1,2} \\ D^2_{2,-2} & D^2_{2,-1} & D^2_{2,0} & D^2_{2,1} & D^2_{2,2} \end{bmatrix} \begin{bmatrix} Y_{2,-2} \\ Y_{2,-1} \\ Y_{2,0} \\ Y_{2,1} \\ Y_{2,2} \end{bmatrix} \end{array}\right. \tag{4.2}$$

The analytic formula is rather complicated, and is presented as equation 7.48 in [48].

Assuming that the test image I_{test} is at a different pose (e.g., a rotated view) from the training images (usually at the frontal view), we look for the basis images at the rotated pose from the basis images at the frontal pose. It will be more convenient to use the basis image form as in (4.1), rather than the spherical harmonics form $Y_{l,m}(\gamma, \phi)$. The general rotation can be decomposed into three concatenated Euler angles around the X, Y and Z axes, namely elevation, azimuth and roll, respectively. Roll is an in-plane rotation that can be handled much easily and will not be discussed here. The following proposition gives the linear transformation matrix from the basis images at the frontal pose to the basis images at the rotated pose for orders $l = 0, 1, 2$, which capture 98% of the energy [7].

Proposition 1 Assume that a rotated view is obtained by rotating a front-view head with an azimuth angle $-\theta$. Having the correspondence between the frontal view and the rotated view built, the basis images B' at the rotated pose are related to the basis images B at the frontal pose as,

$$\left\{\begin{array}{l}
b'_{00} = b_{00} \\
\begin{bmatrix} b'_{10} \\ b_{11}^{\prime e} \\ b_{11}^{\prime o} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_{10} \\ b_{11}^{e} \\ b_{11}^{o} \end{bmatrix} \\
\begin{bmatrix} b'_{20} \\ b_{21}^{\prime e} \\ b_{21}^{\prime o} \\ b_{22}^{\prime e} \\ b_{22}^{\prime o} \end{bmatrix} = \begin{bmatrix} 1-\frac{3}{2}\sin^2\theta & -\sqrt{3}\sin\theta\cos\theta & 0 & \frac{\sqrt{3}}{2}\sin^2\theta & 0 \\ \sqrt{3}\sin\theta\cos\theta & \cos^2\theta-\sin^2\theta & 0 & -\cos\theta\sin\theta & 0 \\ 0 & 0 & \cos\theta & 0 & -\sin\theta \\ \frac{\sqrt{3}}{2}\sin^2\theta & \cos\theta\sin\theta & 0 & 1-\frac{1}{2}\sin^2\theta & 0 \\ 0 & 0 & \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} b_{20} \\ b_{21}^{e} \\ b_{21}^{o} \\ b_{22}^{e} \\ b_{22}^{o} \end{bmatrix}
\end{array}\right. \tag{4.3}$$

Further, if there is also an elevation angle $-\beta$, the basis images B'' for the newly rotated view are related to B' in the following linear form:

$$\left\{\begin{array}{l} b''_{00} \quad = \quad b'_{00} \\ \begin{bmatrix} b''_{10} \\ b''^{e}_{11} \\ b''^{o}_{11} \end{bmatrix} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & \\ -\sin\beta & 0 & \cos\beta & \end{bmatrix} \begin{bmatrix} b'_{10} \\ b'^{e}_{11} \\ b'^{o}_{11} \end{bmatrix} \\ \begin{bmatrix} b''_{20} \\ b''^{e}_{21} \\ b''^{o}_{21} \\ b''^{e}_{22} \\ b''^{o}_{22} \end{bmatrix} = \begin{bmatrix} 1-\frac{3}{2}\sin^2\beta & 0 & \sqrt{3}\sin\beta\cos\beta & \frac{-\sqrt{3}}{2}\sin^2\beta & 0 \\ 0 & \cos\beta & 0 & 0 & \sin\beta \\ -\sqrt{3}\sin\beta\cos\beta & 0 & \cos^2\beta-\sin^2\beta & -\cos\beta\sin\beta & 0 \\ \frac{-\sqrt{3}}{2}\sin^2\beta & 0 & \cos\beta\sin\beta & 1-\frac{1}{2}\sin^2\beta & 0 \\ 0 & -\sin\beta & 0 & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} b'_{20} \\ b'^{e}_{21} \\ b'^{o}_{21} \\ b'^{e}_{22} \\ b'^{o}_{22} \end{bmatrix} \end{array}\right. \tag{4.4}$$

A direct proof (rather than deriving from the general rotation equations) of this proposition is shown below.

Proof:

Assume that (n_x, n_y, n_z) and (n'_x, n'_y, n'_z) are the surface normals of point p at the frontal pose and the rotated view respectively. (n'_x, n'_y, n'_z) is related to (n_x, n_y, n_z) as

$$\begin{bmatrix} n'_x \\ n'_y \\ n'_z \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \tag{4.5}$$

where $-\theta$ is the azimuth angle.

By replacing (n'_x, n'_y, n'_z) in (4.5) with $(n_z\sin\theta + n_x\cos\theta, n_y, n_z\cos\theta - n_x\sin\theta)$, and assuming that the correspondence between the rotated view and the frontal view has been built, we have

$$\begin{aligned}
b'_{00} &= \frac{1}{\sqrt{4\pi}}\lambda,\ b'_{10} = \sqrt{\frac{3}{4\pi}}\lambda. * (n_z\cos\theta - n_x\sin\theta),\\
b_{11}^{\prime e} &= \sqrt{\frac{3}{4\pi}}\lambda. * (n_z\sin\theta + n_x\cos\theta),\ b_{11}^{\prime o} = \sqrt{\frac{3}{4\pi}}\lambda. * n_y,\\
b'_{20} &= \frac{1}{2}\sqrt{\frac{5}{4\pi}}\lambda. * (2(z\cos\theta - n_x\sin\theta)^2 - (n_z\sin\theta + n_x\cos\theta)^2 - n_y^2),\\
b_{21}^{\prime e} &= 3\sqrt{\frac{5}{12\pi}}\lambda. * (n_z\sin\theta + n_x\cos\theta) * (n_z\cos\theta - n_x\sin\theta),\\
b_{21}^{\prime o} &= 3\sqrt{\frac{5}{12\pi}}\lambda. * n_y(n_z\cos\theta - n_x\sin\theta),\\
b_{22}^{\prime e} &= \frac{3}{2}\sqrt{\frac{5}{12\pi}}\lambda. * ((n_z\sin\theta + n_x\cos\theta)^2 - n_y^2),\\
b_{22}^{\prime o} &= 3\sqrt{\frac{5}{12\pi}}\lambda. * (n_z\sin\theta + n_x\cos\theta)n_y \qquad (4.6)
\end{aligned}$$

Rearranging, we get

$$\begin{aligned}
b'_{00} &= b_{00},\ b'_{10} = b_{10}\cos\theta - b_{11}^e\sin\theta,\ b_{11}^{\prime e} = b_{11}^e\cos\theta + b_{10}\sin\theta,\ b_{11}^{\prime o} = b_{11},\\
b'_{20} &= b_{20} - \sqrt{3}\sin\theta\cos\theta b_{21}^e - \sqrt{\frac{5}{4\pi}}\frac{3}{2}\sin^2\theta(n_z^2 - n_x^2),\\
b_{21}^{\prime e} &= (\cos^2\theta - \sin^2\theta)b_{21}^e + 3\sqrt{\frac{5}{12\pi}}\sin\theta\cos\theta(n_z^2 - n_x^2),\\
b_{21}^{\prime o} &= b_{21}^o\cos\theta - b_{22}^o\sin\theta,\\
b_{22}^{\prime e} &= b_{22}^e + \cos\theta\sin\theta b_{21}^e + \sqrt{\frac{5}{12\pi}}\frac{3}{2}\sin^2\theta(n_z^2 - n_x^2),\\
b_{22}^{\prime o} &= b_{22}^o\cos\theta + b_{21}^o\sin\theta. \qquad (4.7)
\end{aligned}$$

As shown in (4.7), $b'_{00}, b'_{10}, b_{10}^{\prime e}, b_{11}^{\prime o}, b_{21}^{\prime o}$ and $b_{22}^{\prime o}$ are linear combinations of the basis images at the frontal pose. For b'_{20},$b_{21}^{\prime e}$ and $b_{22}^{\prime e}$, we need to have $(n_z^2 - n_x^2)$ which is not known. From [7], we know that if the sphere is illuminated by a single directional source in a direction other than the z direction, the reflectance obtained would be identical to the kernel, but shifted in phase. Shifting the phase of a function distributes its energy between the harmonics of the same order n (varying m), but the overall energy in each order n is maintained. The quality of the approximation, therefore, remains the same. This can be verified by noting that $b_{10}'^2+b_{11}'^{e2}+b_{11}'^{o2} = b_{10}^2+b_{11}^{e2}+b_{11}^{o2}$ for the order $n = 1$. Noticing that $b_{21}'^{o2}+b_{22}'^{o2} = b_{21}^{o2}+b_{22}^{o2}$, we still need $b_{20}'^2 + b_{21}'^{e2} + b_{22}'^{e2} = b_{20}^2 + b_{21}^{e2} + b_{22}^{e2}$ to preserve the energy for the order $n = 2$.

Let $G = 3\sqrt{\frac{5}{12\pi}}\sin^2\theta(n_z^2 - n_x^2)$ and $H = 3\sqrt{\frac{5}{12\pi}}\sin\theta\cos\theta(n_z^2 - n_x^2)$, we have

$$\begin{aligned} b_{20}' &= b_{20} - \sqrt{3}\sin\theta\cos\theta b_{21}^e - \frac{\sqrt{3}}{2}G, \\ b_{21}'^e &= (\cos^2\theta - \sin^2\theta)b_{21}^e + H, \\ b_{22}'^e &= b_{22}^e + \cos\theta\sin\theta b_{21}^e + \frac{1}{2}G. \end{aligned} \tag{4.8}$$

Then

$$\begin{aligned} & b_{20}'^2 + b_{21}'^{e2} + b_{22}'^{e2} \\ = \;& b_{20}^2 + b_{21}^{e2} + b_{22}^{e2} + \frac{3G^2}{4} - 2\sqrt{3}\sin\theta\cos\theta b_{20}b_{21}^e - \sqrt{3}b_{20}G + 3\sin\theta\cos\theta G + H^2 \\ & +2(\cos^2\theta - \sin^2\theta)b_{21}^e H + \frac{G^2}{4} + 2\sin\theta\cos\theta b_{22}^e b_{21}^e + b_{22}^e G + \sin\theta\cos\theta G \\ = \;& b_{20}^2 + b_{21}^{e2} + b_{22}^{e2} + G^2 + 4\sin\theta\cos\theta b_{21}^e G + (b_{22}^e - \sqrt{3}b_{20})(G + 2\sin\theta\cos\theta b_{21}^e) \\ & +H^2 + 2(\cos^2\theta - \sin^2\theta)b_{21}^e H \end{aligned}$$

Having $b_{20}'^2 + b_{21}'^{e2} + b_{22}'^{e2} = b_{20}^2 + b_{21}^{e2} + b_{22}^{e2}$ and $H = G\frac{\cos\theta}{\sin\theta}$, we get

$$G^2 + 2\sin\theta\cos\theta b_{21}^e G + (b_{22}^e - \sqrt{3}b_{20})(G\sin^2\theta + 2\sin\theta\cos\theta b_{21}^e) = 0$$

and then $(G + 2\sin\theta\cos\theta b_{21}^e)(G + \sin^2\theta(b_{22}^e - \sqrt{3}b_{20})) = 0$.

The two possible roots of the polynomial are $G = -2\sin\theta\cos\theta b_{21}^e$ or $G = -\sin^2\theta(b_{22}^e - \sqrt{3}b_{20})$. Taking $G = -2\sin\theta\cos\theta b_{21}^e$ into (4.9) gives $b_{20}' = b_{20}$, $b_{21}'^e = -b_{21}^e$, $b_{22}'^e = b_{22}^e$, which is incorrect. Therefore, we have $G = -\sin^2\theta(b_{22}^e - \sqrt{3}b_{20})$ and $H = -\cos\theta\sin\theta(b_{22}^e - \sqrt{3}b_{20})$. Substituting them in (4.9) we get

$$b_{20}' \;=\; b_{20} - \sqrt{3}\sin\theta\cos\theta b_{21}^e + \frac{\sqrt{3}}{2}\sin^2\theta(b_{22}^e - \sqrt{3}b_{20}),$$

$$
\begin{aligned}
b'^{e}_{21} &= (\cos^2\theta - \sin^2\theta) b^e_{21} - \cos\theta\sin\theta(b^e_{22} - \sqrt{3}b_{20}),\\
b'^{e}_{22} &= b^e_{22} + \cos\theta\sin\theta b^e_{21} - \frac{1}{2}\sin^2\theta(b^e_{22} - \sqrt{3}b_{20}).
\end{aligned} \tag{4.9}
$$

Using (4.7) and (4.9), we can write the basis images at the rotated pose in the matrix form of the basis images at the frontal pose, as shown in (4.3).

Assuming that there is an elevation angle $-\beta$ after the azimuth angle $-\theta$ and denoting by (n''_x, n''_y, n''_z) the surface normal for the new rotated view, we have

$$
\begin{bmatrix} n''_x \\ n''_y \\ n''_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} n'_x \\ n'_y \\ n'_z \end{bmatrix} \tag{4.10}
$$

Repeating the above derivation easily leads to the linear equations in (4.4) which relates the basis images at the new rotated pose to the basis images at the old rotated pose.

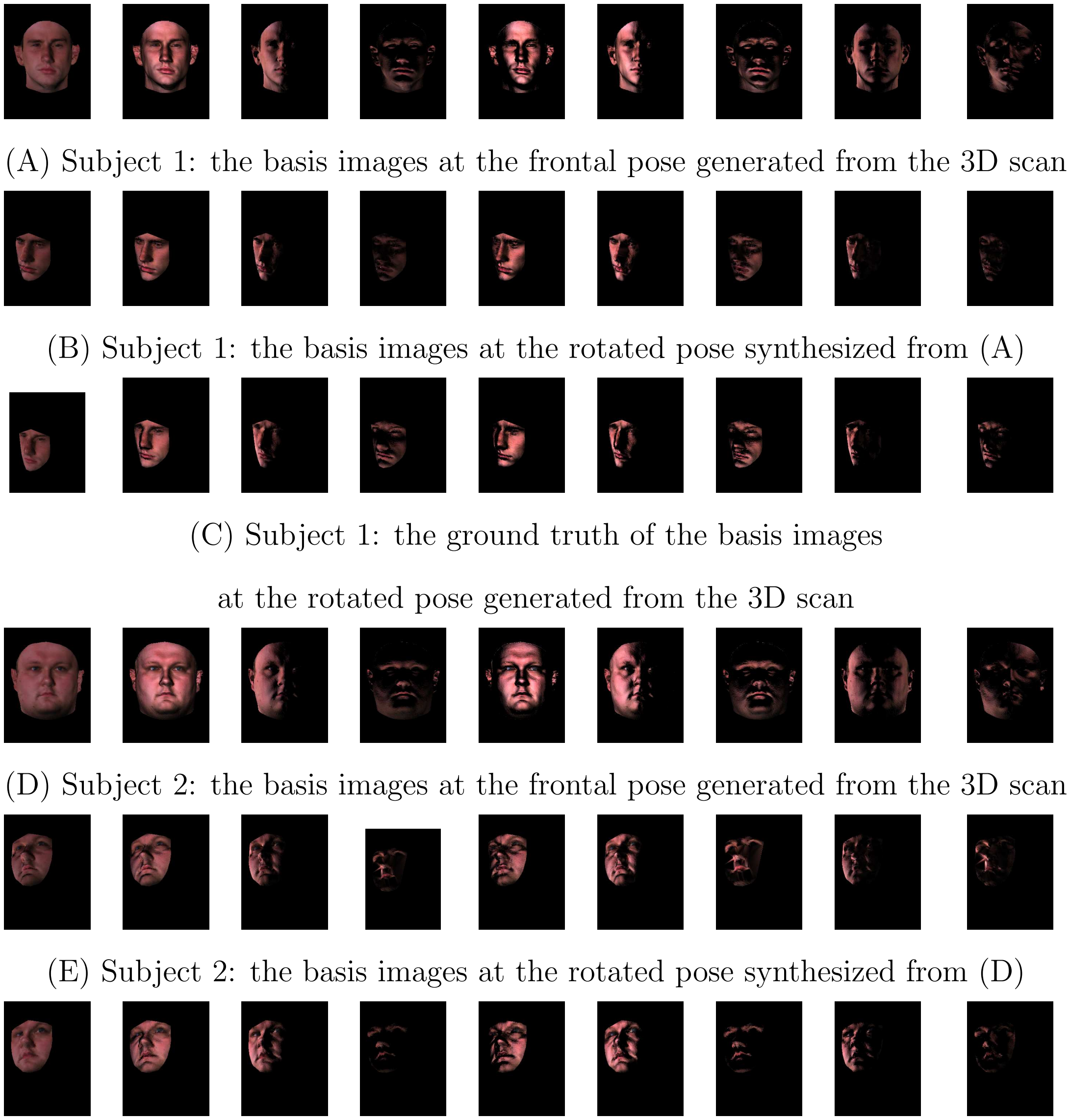
The above proposition can be shown to be consistent with the general rotation matrix of spherical harmonics. If we use a ZYZ formulation for the general rotation, we have $R_{\theta,\alpha,\beta} = R_z(\alpha)R_y(\theta)R_z(\beta)$, the dependence of D_l on α and β is simple, $D^l_{m,m'}(\theta,\alpha,\beta) = d^l_{m,m'}(\theta)e^{im\alpha}e^{im'\beta}$ where d^l is a matrix that defines how a spherical harmonic transforms under rotation about the Y-axis. We can further decompose it into a rotation of 90^o about the X-axis, a general rotation θ about the Z-axis followed finally by a rotation of -90^o about the X-axis [35]. Since

$$X_{\mp 90} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mp 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \pm 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/2 & 0 & -\sqrt{3}/2 \\ 0 & 0 & 0 & 0 & \mp 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\sqrt{3}/2 & 0 & 1/2 \end{bmatrix}$$

and

$$Z_\theta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & \sin\theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\sin\theta & 0 & \cos\theta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos 2\theta & 0 & 0 & 0 & \sin 2\theta \\ 0 & 0 & 0 & 0 & 0 & \cos\theta & 0 & \sin\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & -\sin 2\theta & 0 & 0 & 0 & \cos 2\theta \end{bmatrix},$$

it is easy to show that $R_Y(\theta)$ is exactly the same as shown in (4.3) by taking the above equations into $R_Y(\theta) = X_{-90} Z_\theta X_{+90}$ and re-organizing the order of the spherical harmonics $Y_{l,m}$. Since (4.4) is derived similarly as (4.3), the rotation around X axis can be proved to be the same as (4.4). This can also be verified by taking the rotation angle $\beta = \mp 90^o$ into (4.4) which gives the same $X_{\mp 90^o}$ as shown above.



(A) Subject 1: the basis images at the frontal pose generated from the 3D scan

(B) Subject 1: the basis images at the rotated pose synthesized from (A)

(C) Subject 1: the ground truth of the basis images

at the rotated pose generated from the 3D scan

(D) Subject 2: the basis images at the frontal pose generated from the 3D scan

(E) Subject 2: the basis images at the rotated pose synthesized from (D)

(F) Subject 2: the ground truth of the basis images

at the rotated pose generated from the 3D scan

Figure 4.2: (A)-(C) present the results of the synthesized basis images for subject 1, where (A) shows the basis images at the frontal pose generated from the 3D scan, (B) the basis images at a rotated pose synthesized from (A), and (C) the ground truth of the basis images at the rotated pose. (D)-(E) present the results of the synthesized basis images for subject 2, with (D) showing the basis images at the frontal pose generated from the 3D scan, (E) the basis images at a rotated pose synthesized from (D), and (F) the ground truth of the basis images at the rotated pose.

We synthesized the basis images at arbitrary rotated pose from those at the frontal pose using (4.3) and (4.4), and compared them with the ground truth generated from the 3D scan in Fig. 4.2. The first three rows present the results for subject 1, with the first row showing the basis images at the frontal pose generated from the 3D scan, the second row showing the basis images at the rotated pose (azimuth angle $\theta = -30^o$, elevation angle $\beta = 20^o$) synthesized from the images at the first row, and the third row, the ground truth of the basis images at the rotated pose generated from the 3D scan. Rows four through six present the results for subject 2, with the fourth row showing the basis images at the frontal pose generated from the 3D scan, the fifth row, the basis images for another rotated view (azimuth angle $\theta = -30^o$, elevation angle $\beta = -20^o$) synthesized from the images at the fourth row, and the last row the ground truth of the basis images at the rotated pose generated from the 3D scan. As we can see from Fig. 4.2, the synthesized basis images at the rotated poses are quite close to the ground truth. Note in Fig. 4.2 and the figures in the sequel, the dark regions represent the negative values of the basis images.

Given that the correspondence between the rotated-pose image and the frontal-pose image is available, a consequence of the existence of such linear transformation is that the procedure of first rotating objects and then recomputing basis images at the desired pose can be avoided. The block diagonal form of the transformation matrices preserves the energy on each order $l = 0, 1, 2$. Moreover, the orthonormality of the transformation matrices helps to further simplify the computation required for recognition of the rotated test image as shown in Section 4.2.2. Although in theory new basis images can be generated from a rotated 3D model inferred by the existing basis images (since basis images actually capture the albedo (b_{00}) and the 3D surface normal $(b_{10}, b^e_{11}, b^o_{11})$ of a given human face), the procedure of such 3D recovery is not trivial in practice, even if computational cost is taken out of consideration.

4.2 Face Recognition Using Pose-Encoded Spherical Harmonics

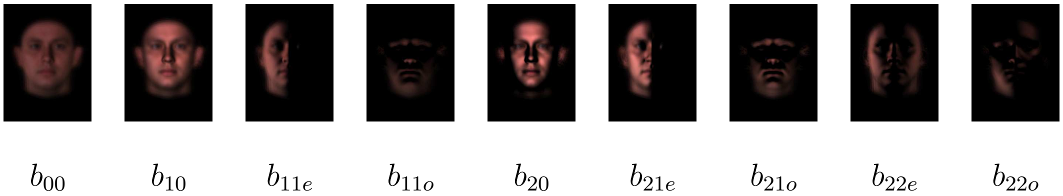
In this section we present an efficient face recognition method using pose-encoded spherical harmonics. Only one training image is needed per subject and high recognition performance is achieved even when the test image is at a different pose from the training image and under an arbitrary illumination condition.

4.2.1 Statistical Models of Basis Images

We briefly summarize a statistical learning method to recover the harmonic basis images from only one image taken under arbitrary illumination conditions, as shown in [115].

We build a bootstrap set with fifty 3D face scans and corresponding texture maps from Vetter's 3D face database [1], and generate nine basis images for each face model. For a novel d-dimensional vectorized image I, let B be the $d \times 9$ matrix of basis images, α a 9 dimensional vector and E a d-dimensional error term. We have $I = B\alpha + E$. It is assumed that the probability density function (pdf)'s of B are Gaussian distributions. The sample mean vectors $\mu_b(x)$ and covariance matrixes $C_b(x)$ are estimated from the basis images in the bootstrap set. Figure 4.4 shows the sample mean of the basis images estimated from the bootstrap set.

The problem of estimating the basis images B and the illumination coefficients



b_{00} b_{10} b_{11e} b_{11o} b_{20} b_{21e} b_{21o} b_{22e} b_{22o}

Figure 4.3: The sample mean basis images estimated from the bootstrap set.

α is a coupled estimation problem because of its bilinear form. It is simplified by estimating α in a prior step with kernel regression and using it consistently across all pixels to recover B. K bootstrap images $\{J_k\}_{k=1}^K$ with known coefficients $\{\alpha_k\}_{k=1}^K$ are generated from the 3D face scans in the bootstrap set. Given a new image i_{tra}, the coefficients α_{tra} can be estimated as

$$\alpha_{tra} = \frac{\sum_{k=1}^K w_k \alpha_k}{\sum_{k=1}^K w_k} \tag{4.11}$$

where $w_k = exp[-\frac{1}{2}(D(i, J_k)/\sigma_k)^2]$ and $D(i, J_k) = \|i - J_k\|_2$, σ_k is the width of the k-th Gaussian kernel which controls the influence of J_k on the estimation of α_{tra}. All $\{\sigma_k\}_{k=1}^K$ are pre-computed in a way such that ten percent of the bootstrap images are within $1 \times \sigma_k$ at each σ_k. The sample mean $\mu_e(x, \alpha)$ and the sample variance $\sigma_e^2(x, \alpha)$ of the error term $E(\alpha)$ are also estimated using kernel regression, similar to (4.11).

Given a novel face image $i(x)$, with the estimated coefficients α, the corresponding basis images $b(x)$ at each pixel x are recovered by computing the maximum a posteriori (MAP) estimate, $b_{MAP}(x) = argmax_{b(x)}(P(b(x)|i(x)))$. Using the Bayes rule:

$$\begin{aligned} b_{MAP}(x) &= argmax_{b(x)} P(i(x)|b(x))P(b(x)) \\ &= argmax_{b(x)} \left\{ \mathcal{N}\left(b(x)^T\alpha + \mu_e, \sigma_e^2\right) \times \mathcal{N}\left(\mu_b(x), C_b(x)\right) \right\} \qquad (4.12) \end{aligned}$$

Taking logarithm, and setting the derivatives of the right hand side of (4.12)

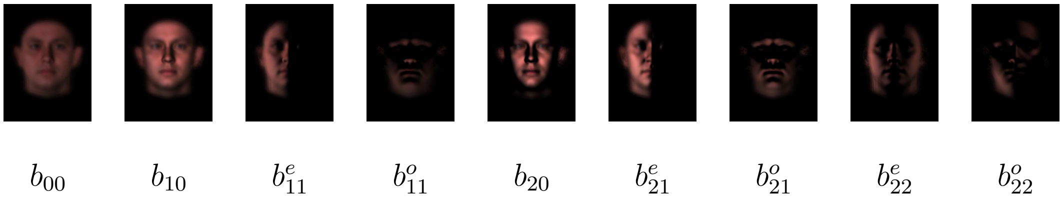
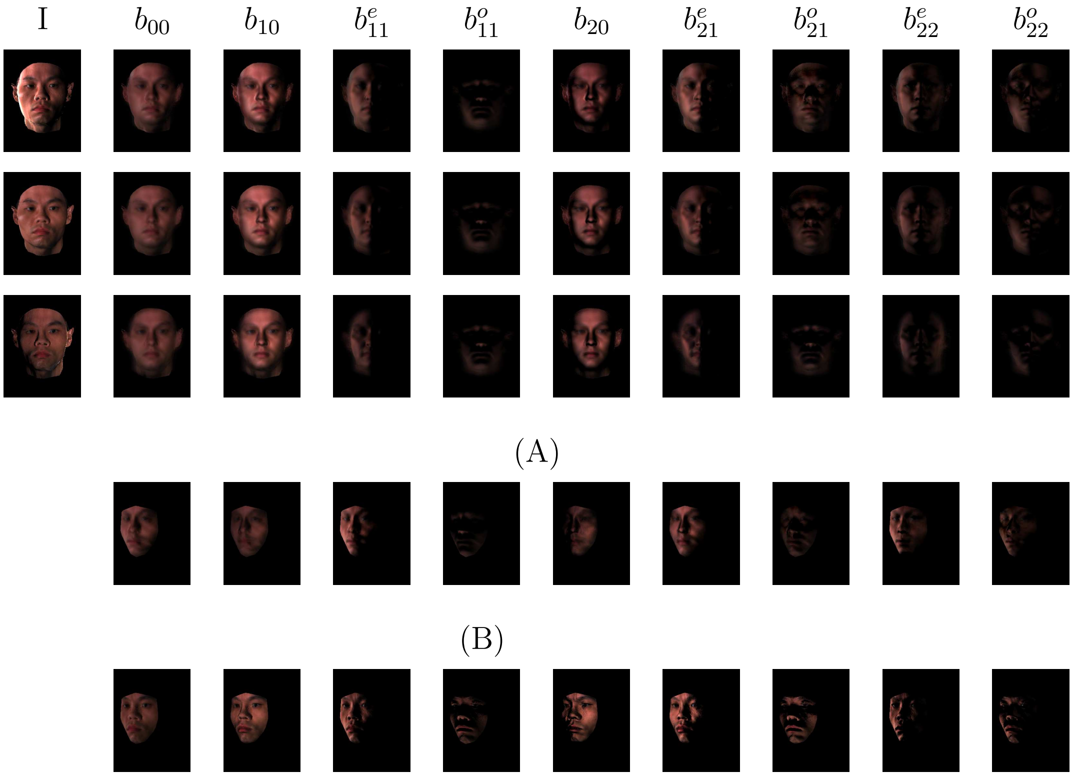


Figure 4.4: The sample mean of the basis images estimated from the bootstrap set [1].

(w.r.t $b(x)$) to 0, we get $A * b_{MAP} = T$, where $A = \frac{1}{\sigma_e^2}\alpha\alpha^T + C_b^{-1}$ and $T = \frac{(i-\mu_e)}{\sigma_e^2}\alpha + C_b^{-1}\mu_b$. By solving this linear equation, $b(x)$ of the subject can be recovered.

In Fig. 4.5 we illustrate the procedure for generating the basis images at a rotated pose (azimuth angle $\theta = -30^o$) from a single training image at the frontal pose. In Fig. 4.5, rows one through three show the results of the recovered basis images from a single training image, with the first column showing different training images I under arbitrary illumination conditions for the same subject and the remaining nine columns showing the recovered basis images. We can observe from the figure that the basis images recovered from different training images of the same subject look very similar. Using the basis images recovered from any training image in row one through three, we can synthesize basis images at the rotated pose, as shown in row four. As a comparison, the fifth row shows the ground truth of the basis images at the rotated pose generated from the 3D scan.

Our experiments on the CMU-PIE [90] database used the images of each subject at the frontal pose (c27) as the training set. One hundred 3D face models from Vetter's database [1] were used as the bootstrap set. The training images were first re-scaled to the size of the images in the bootstrap set. The statistics of the harmonic basis images was then learnt from the bootstrap set and the basis images B for each training subject were recovered. Fig. 4.6 shows two examples of the recovered basis images from the single training image, with the first



(A)

(B)

(C)

Figure 4.5: The first column in (A) shows different training images I under arbitrary illumination conditions for the same subject and the remaining nine columns in (A) show the recovered basis images from I. We can observe that the basis images recovered from different training images of the same subject look very similar. Using the basis images recovered from any training image I in (A), we can synthesize basis images at the rotated pose, as shown in (B). As a comparison, (C) shows the ground truth of the basis images at the rotated pose generated from the 3D scan.

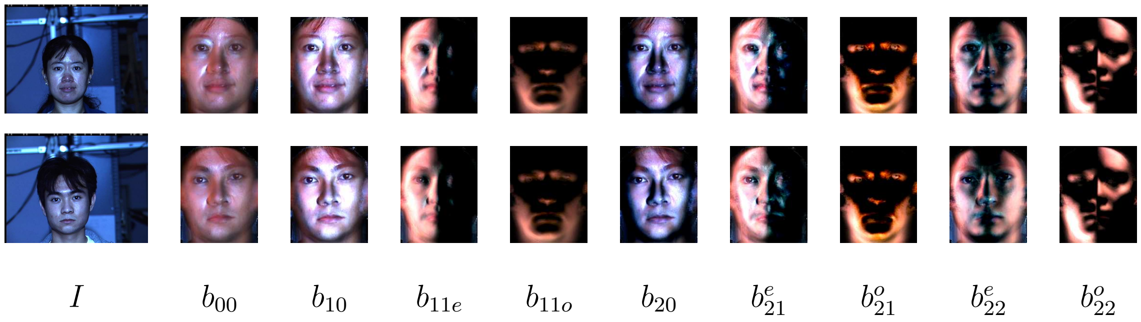


Figure 4.6: The first column shows the training images I for two subjects in the CMU-PIE database and the remaining nine columns show the reconstructed basis images.

column showing the training images I and the remaining 9 columns showing the reconstructed basis images.

4.2.2 Recognition

For recognition, we follow a simple yet effective algorithm given in [7]. A face is identified for which there exists a weighted combination of basis images that is the closest to the test image. Let B be the set of basis images at the frontal pose, with size $N \times r$, where N is the number of pixels in the image and $r = 9$ is the number of basis images used. Every column of B contains one spherical harmonic image. These images form a basis for the linear subspace, though not an orthonormal one. A QR decomposition is applied to compute Q, a $N \times r$ matrix with orthonormal columns, such that $B = QR$ where R is an $r \times r$ upper triangular matrix.

For a vectorized test image I_{test} at an arbitrary pose, let B_{test} be the set of basis images at that pose. The orthonormal basis Q_{test} of the space spanned by B_{test} can be computed by QR decomposition. The matching score is defined as the distance from I_{test} to the space spanned by B_{test}: $s_{test} = \|Q_{test}Q_{test}^T I_{test} - I_{test}\|$.

However, this algorithm is not efficient overall because the set of basis images B_{test} has to be generated for each training subject at the pose of an arbitrarily rotated test image.

We propose to warp the test image I_{test} at the arbitrary (rotated) pose to its front-view image I_f to perform the recognition. In order to warp I_{test} to I_f, we have to find the point correspondence between these two images, which can be embedded in a sparse $N \times N$ warping matrix K, i.e., $I_f = KI_{test}$. The positions of the non-zero elements in K encode the 1-to-1 and many-to-1 correspondence cases (the 1-to-many case is same as 1-to-1 case for pixels in I_f) between $_{test}$ and I_f, and the positions of 0's on the diagonal line of K encode the no-correspondence case. More specifically, if pixel $I_f(i)$ (the $i-th$ element in vector I_f) corresponds to pixel $I_{test}(j)$ (the $j-th$ element in vector I_{test}), then $K(i,j) = 1$. There might be cases that there are more than one pixels in I_{test} corresponding to the same pixel $I_f(i)$, i.e., there are more than one 1's in the i-th row of K and the column indices of these 1's are the corresponding pixel indices in I_{test}. For this case, although there are several pixels in I_{test} mapping to the same pixel $I_f(i)$, it can only have one reasonable intensity value. We compute a single "virtual" corresponding pixel in I_{test} for $I_f(i)$ as the centroid of $I_f(i)$'s real corresponding pixels in I_{test}, and assign it the average intensity. The weight for each real corresponding pixel $I_{test}(j)$ is proportional to the inverse of its distance to the centroid, and this weight is assigned as the value of $K(i,j)$. If there is no correspondence in I_{test} for $I_f(i)$ which is in the valid facial area and should have a corresponding point in I_{test}, it means that $K(i,i) = 0$. This is often the case that the corresponding "pixel" of $I_f(i)$ falls in the sub-pixel region. Thus interpolation is needed to fill the intensity for $I_f(i)$. Barycentric coordinates are calculated with the pixels which have real corresponding integer pixels in I_{test} as the triangle vertices. These Barycentric coordinates are assigned as the values of $K(i,j)$ where j is the column index for each vertex of the triangle.

We now have the warping matrix K which encodes the correspondence and interpolation information in order to generate I_f from I_{test}. It provides a very convenient tool to analyze the impact of some empirical factors in image warping. Note that due to the self-occlusion, I_f does not cover the whole area, but only a sub-region, of the full frontal face of the subject it belongs to. The missing facial region due to the rotated pose is filled with zeros in I_f. Assume that B_f is the basis images for the full front-view training images and Q_f is its orthonormal basis, and let b be the corresponding basis images of I_f and q its orthonormal basis. In b, the rows corresponding to the valid facial pixels in I_f is a submatrix of the rows in B_f corresponding to the valid facial pixels in the full frontal face images. For recognition, we can not directly use the orthonormal columns in Q_f because it is not guaranteed that all the columns in q are still orthonormal.

We study the relationship between the matching score for the rotated view $s_{test} = \|Q_{test}Q_{test}^T I_{test} - I_{test}\|$ and the matching score for the frontal view $s_f = \|qq^T I_f - I_f\|$. Assume subject a is the one that has the minimum matching score at the rotated pose, i.e., $s_{test}^a = \|Q_{test^a}Q_{test^a}^T I_{test} - I_{test}\| \leq s_{test}^l = \|Q_{test^l}Q_{test^l}^T I_{test} - I_{test}\|, \forall l \in [1, 2, ...L]$ where L is the number of training subjects. If a is the correct subject for the test image I_{test}, warping Q_{test^a} to q^a undertakes the same warping matrix K as warping I_{test} to I_f, i.e., the matching score for the frontal view $s_f^a = \|q^a q^{aT} I_f - I_f\| = \|KQ_{test^a}Q_{test^a}^T K^T K I_{test} - K I_{test}\|$. Note here we only consider the correspondence and interpolation issues. Due to the orthonormality of the transformation matrices as shown in (4.3) and (4.4), the linear transformation from B_{test} to b does not affect the matching score. For all the other subjects $l \in [1, 2, ...L], l \neq a$, the warping matrix K^l for Q^l_{test} is different from that for I_{test}, i.e., $s^l_f = \|K^l Q^l_{test} Q^{lT}_{test} K^l T K I_{test} - K I_{test}\|$. We will show that warping I_{test} to I_f does not deteriorate the recognition performance, i.e., given $s^a_{test} \leq s^l_{test}$, we have $s^a_f \leq s^l_f$.

In terms of K, we consider the following cases: Case 1: $K = \begin{pmatrix} E_k & 0 \\ 0 & 0 \end{pmatrix}$, where E_k is the k-rank identity matrix. It means that K is a diagonal matrix and the first k elements on the diagonal line are 1, all the rest are zero. This is the case that I_{test} is at the frontal pose. The difference between I_{test} and I_f is that there are some missing (non-valid) facial pixels in I_f than in I_{test}, and all the valid facial pixels in I_f are packed in the first k elements. Since I_{test} and I_f are at the same pose, Q_{test} and q are also at the same pose. In this case, for subject a, the missing (non-valid) facial pixels in q are at the same locations as in I_f since they have the same warping matrix K. On the other hand, for any other subject l, the missing (non-valid) facial pixels in q are not at the same locations as in I_f since $K^l \neq K$. Apparently the 0's and 1's on the diagonal line of K^l has different positions from that of K, thus $K^l K$ has more 0's on the diagonal line than K.

Assume $K = \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix}$ and $V = Q_{test} Q^T_{test} = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$ where V_{11} is a $(k \times k)$ matrix. Similarly, let $I_{test} = \binom{I_1}{I_2}$ where I_1 is a $(k \times 1)$ vector. Then $K Q_{test} Q^T_{test} K^T = \begin{pmatrix} V_{11} & 0 \\ 0 & 0 \end{pmatrix}$, $K I_{test} = \binom{I_1}{0}$ and $K Q_{test} Q^T_{test} K^T K I_{test} - K I_{test} = \binom{V_{11} I_1}{0} - \binom{I_1}{0} = \binom{(V_{11} - E_k) I_1}{0}$. Therefore, $s^a_f = \|(V_{11} - E_k)\, I_1\|$. Similarly, $K^l Q_{test} Q^T_{test} K^{lT} = \begin{pmatrix} V^l_{11} & 0 \\ 0 & 0 \end{pmatrix}$ where V^l_{11} is also a $(k \times k)$ matrix that might contain rows with all 0's, depending on the locations of the 0's on the diagonal line of K^l. We have $K^l Q_{test} Q_{test}^T K^{l^T} K I_{test} - K I_{test} = \binom{V_{11}^l I_1}{0} - \binom{I_1}{0} = \binom{\left(V_{11}^l - E_k\right) I_1}{0}$. Thus $s_f^l = \left\| \left(V_{11}^l - E_k\right) I_1 \right\|$.

If V_{11}^l has rows with all 0's in the first k rows, these rows will have -1's at the diagonal positions for $V_{11}^l - E_k$, which will increase the matching score s_f^l. Therefore, $s_f^a \leq s_f^l$.

Case 2: K is a diagonal matrix with rank k, however, the k 1's are not necessarily the first k elements on the diagonal line.

We can use some elementary transformation to reduce this case to the previous case. That is, there exists a orthonormal matrix P, such that, $\hat{K} = PKP^T = \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix}$.

Let $\hat{Q}_{test} = PQ_{test}P^T$ and $\hat{I}_{test} = PI_{test}$. Then $s_f^a = \left\| P \left(K Q_{test} Q_{test}^T K^T K I_{test} - K I_{test} \right) \right\|$ $= \left\| \hat{K} \hat{Q}_{test} \hat{Q}_{test}^T \hat{K}^T \hat{K} \hat{I}_{test} - \hat{K} \hat{I}_{test} \right\|$. Note that elementary transformation does not change the norm. Hence, it reduces to the previous case. Similarly, we have s_f^l stays the same as in case 1. Therefore, $s_f^a \leq s_f^l$ still holds.

In the general case, 1's in K can be off-diagonal. This means that I_{test} and I_f are at different poses. There are three sub-cases we need to discuss for a general K.

Case 3.1: 1-to-1 correspondence between I_{test} and I_f. If pixel $I_{test}(j)$ has only one corresponding point in I_f, denoted as $I_f(i)$, then $K(i,j) = 1$ and there are no 1's in both the i-th row and the j-th column in K. Suppose there are only k columns of the matrix K contains 1. Then, by some elementary transformation again, we can left-multiply and right-multiply K by an orthonormal transformation matrix P^1 and P^2 respectively, such that $\widetilde{K} = P^1 K P^2$. If we define $Q_{test}^1 = P^{2T} Q_{test} P^1$ and $I_1 = P^{2T} I_{test}$, then $s_f^a = \left\| K Q_{test} Q_{test}^T K^T K I_{test} - K I_{test} \right\| =$

$\left\| P^1 \left(KQ_{test}Q_{test}^T K^T K I_{test} - K I_{test} \right) \right\|$
$= \left\| P^1 K P^2 P^{2T} Q_{test} P^1 P^{1T} Q_{test}^T P^2 P^{2T} K^T P^{1T} P^1 K P^2 \left(P^{2T} I_{test} \right) - P^1 K P^2 \left(P^{2T} I_{test} \right) \right\|$
$= \left\| \widetilde{K} Q_{test}^1 Q_{test}^{1T} \widetilde{K}^T \widetilde{K} X I_1 - \widetilde{K} I_1 \right\|$ Under $\widetilde{K}$, it reduces to case 2, which can be further reduced to case 1 by the aforementioned technique. Similarly, we have s_f^l stays the same as in case 2. Therefore, $s_f^a \leq s_f^l$ still holds.

In all the cases discussed up to now, the correspondence between I_{test} and I_f is 1-to-1 mapping. For such cases, the following lemma shows that the matching score stays the same before and after the warping.

Lemma 1 Given the correspondence between a rotated test image I_{test} and its geometrically synthesized front-view image I_f is 1-to-1 mapping, the matching score s_{test} of I_{test} based on the basis images B_{test} at that pose is the same as the matching score s_f of I_f based on the basis images b.

Let C be the transpose of the combined coefficient matrices in (4.3) and (4.4), we have $b = B_{test}C = Q_{test}RC$ by QR decomposition. Applying QR decomposition again to RC, we have $RC = \tilde{q}\tilde{r}$ where $\tilde{q}_{r \times r}$ is an orthonormal matrix and $\tilde{r}$ is an upper triangular matrix. We now have $b = Q_{test}\tilde{q}\tilde{r} = q\tilde{r}$ by assuming $q = Q_{test}\tilde{q}$. Since $Q_{test}\tilde{q}$ is the product of two orthonormal matrices, it forms a valid orthnormal basis for b. Hence the matching score is $s_f = \|Q_{test}\tilde{q}\tilde{q}^T Q_{test}^T I_{test} - I_{test}\|$. Now $qq^T = Q_{test}\tilde{q}\tilde{q}^T Q_{test}^T = Q_{test}Q_{test}^T$ since $\tilde{q}$ is orthonormal. Hence the final matching score is $\|Q_{test}Q_{test}^T I_{test} - I_{test}\| = s_{test}$.

Case 3.2: many-to-1 correspondence between I_{test} and I_f.

Case 3.3: There is no correspondence for $I_f(i)$ in I_{test}.

For case 3.2 and 3.3, since the 1-to-1 correspondence assumption does not hold any more, it becomes more complicated to analytically discuss the relationship between s_{test} and s_f. This is due to the effects of fortshortening and interpolation.

Consider the same actual 3D facial area, it may contribute more in the rotated view recognition but contribute less in the frontal view recognition (or vice versa) because of the fortshortening. The increased (or decreased) information comes from the interpolation, and the assigned weight for each interpolated pixel is not guaranteed to be the same as that before the warping. Therefore, the relationship between s_{test} and s_f relies on each specific K, which may vary significantly depending on the variation of the head pose. Instead of theoretical analysis, the empirical error bound between s_{test} and s_f is sought to give a general idea of how the warping affects the matching scores. We conducted experiments using the Vetter's database. For the fifty subjects which are not used in the bootstrap set, we generate images at various poses and get their basis images at each pose. For each pose, s_{test} and s_f are compared, and the mean of the relative error and the relative standard deviation for some poses are listed in the following table.

Pose	$(\theta = 30^o,\beta = 0^o)$	$(\theta = 30^o,\beta = -20^o)$	$(\theta = -30^o,\beta = 0^o)$	$(\theta = -30^o,\beta = 20^o)$
$mean(\frac{s_f - s_r}{s_r})$	3.4%	3.9%	3.5%	4.1%
$std(\frac{s_f - s_r}{s_r})$	5.0%	5.2%	4.9%	5.1%

We can see from the experimental results that although s_r and s_f are not exactly the same, the difference between s_r and s_f is very small. We examined the ranking of the matching scores before and after warping. The following table shows the percentage that the top one pick before warping still remains as the top one after warping.

Pose	$(\theta = 30^o,\beta = 0^o)$	$(\theta = 30^o,\beta = -20^o)$	$(\theta = -30^o,\beta = 0^o)$	$(\theta = -30^o,\beta = 20^o)$
percentage of the top one pick keeps its position	98.4%	97.6%	99.2%	97.9%

Thus warping the test image I_{test} to its front-view image I_f does not adversely affect the recognition performance. We now have a very efficient solution for face recognition to handle both pose and illumination variations as only one image I_f needs to be synthesized. We also wish to point out that the experimental results also verified Lemma 1: The matching scores before and after warping is exactly the same if we only consider the pixels with 1-to-1 correspondence.

Now the only remaining problem is that the correspondence between I_{test} and I_f has to be built. An average face calculated from training images at the frontal pose and the corresponding feature points were used to build the correspondence between I_{test} and I_f. Then we can use the same method as presented in 3.3.1. Fig. 4.7 shows an example of building dense correspondence between the rotated view and the frontal view using sparse features.

4.2.3 View Synthesis

To verify the recognition results, the user is given the option to visually compare the chosen subject and the test image I_{test} by generating the face image of the chosen subject at the same pose and under the same illumination condition as I_{test}. The desired N-dimensional vectorized image I_{des} can be synthesized easily as long as we can generate the basis images B_{des} of the chosen subject at that pose by using

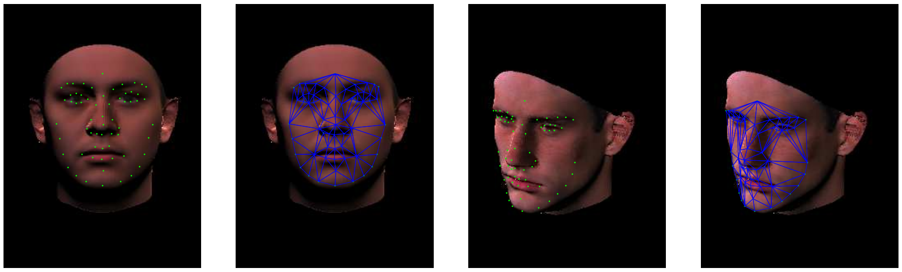
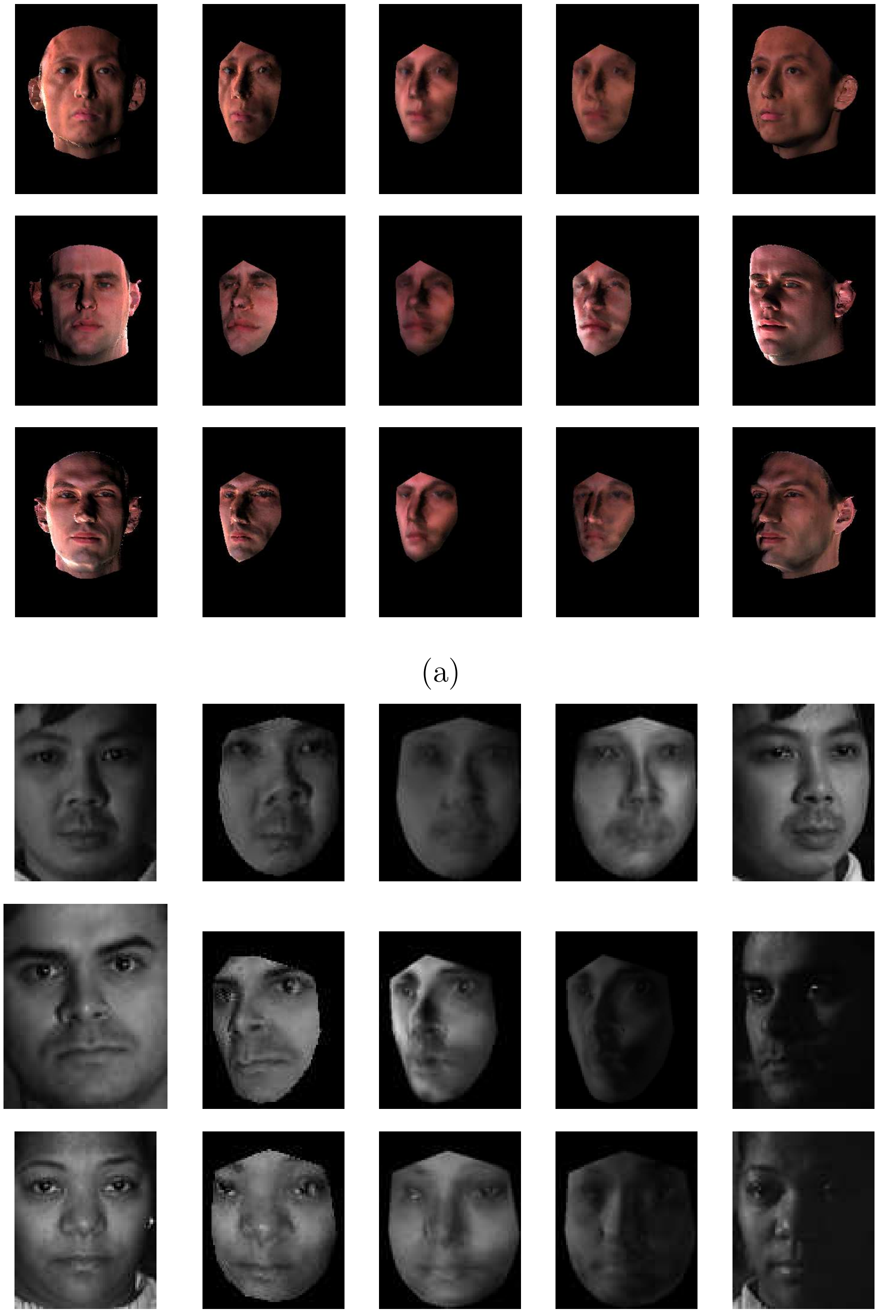


Figure 4.7: Building dense correspondence between the rotated view and the frontal view using sparse features. The first and second images show sparse features and the constructed meshes on the mean face at the frontal pose. The third and fourth images show the picked features and the constructed meshes on the given test image at the rotated pose.

$I_{des} = B_{des}\alpha_{test}$. Assuming that the correspondence between I_{test} and the frontal pose image has been built as described in Section 3.3.1, B_{des} can be generated from the basis images B of the chosen subject using (4.3) and (4.4) given that the pose (θ, β) of I_{test} can be estimated as described in Section 3.3.2. We also need to estimate the 9 dimensional lighting coefficient vector α_{test}. Assuming that the chosen subject is the correct one, and thus $B_{test} = B_{des}$, we have $I_{test} = B_{des}\alpha_{test}$ by substituting $B_{test} = B_{des}$ into $I_{test} = B_{test}\alpha_{test}$. Recall that $B_{des} = Q_{des}R_{des}$, we have $I_{test} = Q_{des}R_{des}\alpha_{test}$ and then $Q_{des}^T I_{test} = Q_{des}^T Q_{des}R_{des}\alpha_{test} = R_{des}\alpha_{test}$ due to the orthonormality of Q_{des}. Therefore, $\alpha_{test} = R_{des}^{-1}Q_{des}^T I_{test}$.

Having both B_{des} and α_{test} available, we are ready to generate the face image of the chosen subject at the same pose and under the same illumination condition as I_{test} using $I_{des} = B_{des}\alpha_{test}$. The only unknown is the pose (θ, β) of I_{test}, which can be estimated as described in 3.3.2.

Having the head pose estimated, we can now perform face synthesis. Fig. 4.8 shows the comparison of the given test image I_{test} and some synthesized face



(a)

(b)

Figure 4.8: View synthesis results for (a) synthetic images from the Vetter's 3D database and (b) real images in the CMU-PIE database. Columns from left to right show: the training images, the synthesized images at the same pose as the test images using direct warping, the synthesized images at the same pose as the test images from B_{des} and α_{tr}, the synthesized images at the same pose as the test images from B_{des} and α_{test}, and the given test images I_{test}.

images at the same pose as I_{test} from the chosen subject, where (a) is for the synthetic images in Vetter's 3D database and (b) is for real images in the CMU-PIE database. Column one shows the training images. Column two shows the synthesized images at the same pose as I_{test} by direct warping. Column three shows the synthesized images using the basis images B_{des} from the chosen subject and the illumination coefficients α_{tr} of the training images. A noticeable difference between column two and three is the lighting change. By direct warping, we obtain the synthesized images by not only rotating the head pose, but also rotating the lighting direction at the same time. By using α_{tr}, we only rotate the head pose to get the synthesized images, while the lighting condition stays same as the training images. Column four shows the synthesized images using the basis images B_{des} from the chosen subject and the same illumination coefficients α_{test} of I_{test}. As a comparison, column five shows the given test image I_{test}. Overall, the columns from left to right in Fig. 4.8 show the procedure migrating from the training images to the given test images.

4.2.4 Recognition Results

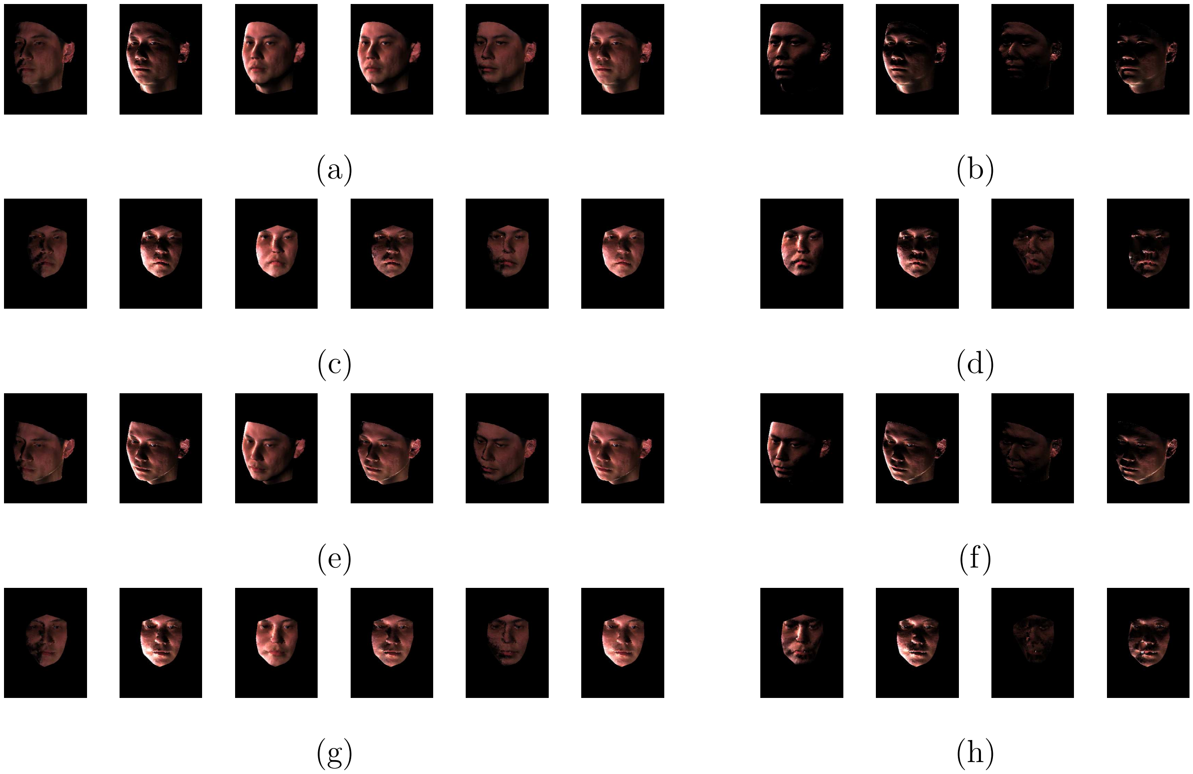
We first conducted recognition experiments on Vetter's 3D face model database. There are totally one hundred 3D face models in the database, from which fifty were used as the bootstrap set and the other fifty were used to generate training images. We synthesized the training images under a wide variety of illumination conditions using the 3D scans of the subjects. For each subject, only one frontal view image was stored as a training image and used to recover the basis images B using the algorithm presented in Section 4.2.1. We generated the test images at different poses from the training images by rotating the 3D scans and illuminated

Table 4.1: The correct recognition rates at two rotated pose under various lighting conditions for synthetic images generated from the Vetter's 3D face model database.

lighting\pose	f2f	Pose $\theta = -30^o$, $\beta = 0^o$		Pose $\theta = -30^o$, $\beta = 20^o$	
		r2f	r2r	r2f	r2r
$(\gamma = 90^o, \tau = 10^o)$	100	100	96	84	80
$(\gamma = 30^o, \tau = 50^o)$	100	100	100	100	100
$(\gamma = 40^o, \tau = -10^o)$	100	100	100	100	100
$(\gamma = 70^o, \tau = 40^o)$	100	100	100	94	88
$(\gamma = 80^o, \tau = -20^o)$	100	100	98	88	84
$(\gamma = 50^o, \tau = 30^o)$	100	100	100	100	96
$(\gamma = 20^o, \tau = -70^o)$	94	86	64	80	68
$(\gamma = 20^o, \tau = 70^o)$	100	100	80	96	76
$(\gamma = 120^o, \tau = -70^o)$	92	84	74	74	64
$(\gamma = 120^o, \tau = 70^o)$	96	90	64	82	70
mean	98	96	88	90	83
std	3	6.6	15	9.5	13

them with various lighting conditions (represented by the slant angle γ and tilt angle τ). Some examples are shown in Fig. 4.9(a)-(b) and (c)-(d). For a test image I_{test} at an arbitrary pose, the frontal pose image I_f was synthesized by warping I_{test}, as shown in Fig. 4.9 (e)-(f) and (g)-(h).

The recognition score was computed as $\|qq^T I_f - I_f\|$ where q is the orthonormal basis of the space spanned by B. The first column (f2f) of Table 4.1 lists the recognition rates when both the testing images and the training images are from the frontal view. The correct recognition rates using the proposed method are



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.9: (a) shows the test images of a subject at azimuth $\theta = -30^o$ under different lighting conditions (($\gamma = 90^o, \tau = 10^o$), ($\gamma = 30^o, \tau = 50^o$), ($\gamma = 40^o, \tau = -10$), ($\gamma = 20^o, \tau = 70^o$), ($\gamma = 80^o, \tau = -20^o$) and ($\gamma = 50^o, \tau = 30^o$) from left to right). The test images of the same subject under some extreme lighting conditions (($\gamma = 20^o, \tau = -70^o$), ($\gamma = 20^o, \tau = 70^o$), ($\gamma = 120^o, \tau = -70^o$) and ($\gamma = 120^o, \tau = -70^o$) from left to right) are shown in (b). (c) and (d) show the generated frontal pose images from the test images in (a) and (b) respectively. The test images at another pose (with $\theta = -30^o$ and $\beta = 20^o$) of the same subject are shown in (e) and (f), with the generated frontal pose images shown in (g) and (h) respectively.

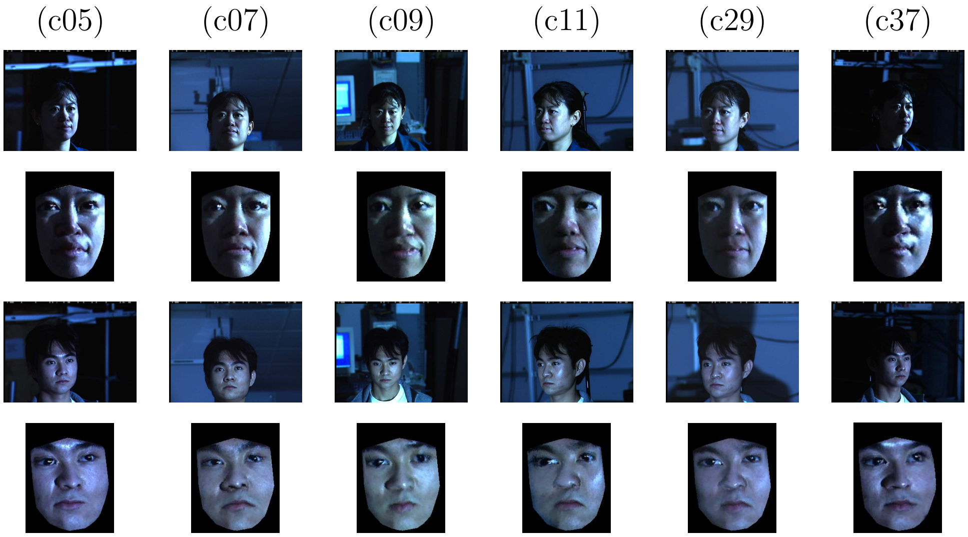


Figure 4.10: The first and third rows show the test images of two subjects in the CMU-PIE database at six different poses, with the pose numbers shown above each column. The second and fourth rows show the corresponding frontal view images generated by directly warping the given test images.

listed in columns (r2f) of Table 4.1. As a comparison, we also conducted the recognition experiment on the same test images assuming that the training images at the same pose are available. By recovering the basis images B at that pose using the algorithm in Section 4.2.1 and computing $\|\widetilde{Q}\widetilde{Q}^T I_{test} - I_{test}\|$, we achieved the recognition rates as shown in columns (r2r) of Table 4.1. As we can see, the recognition rates using our approach are comparable to those when the training images at the rotated pose are available. The last two rows of Table 4.1 show the mean and standard deviation of the recognition rates for each pose under various illumination conditions. We believe that relatively larger standard deviation is due to the images under some extreme lighting conditions, as shown in Fig. 4.9 (b) and (f).

We also conducted experiments on real images from the CMU-PIE database. For testing, we used images at six different poses, as shown in the first and third rows in Fig. 4.10, and under twenty one different illuminations. Examples of the generated frontal view images are shown in the second and fourth rows of Fig. 4.10.

Similar to Table 4.1, Table 4.2 lists the correct recognition rates under all these poses and illumination conditions, where column (f2f) is the front-view testing image against front-view training images, columns (r2r) are the rotated testing image against the same pose training images, and columns (r2f) are the rotated testing image against the front-view training images. The last two rows of 4.2 show the mean and standard deviation of the recognition rates for each pose under various illumination conditions. As we can see, the recognition rates using our approach are comparable to those when the training images at the rotated pose are available. The reason is that the training images of different subjects at the *same* rotated pose are actually at *slightly different* poses. Therefore, the 2D-3D registration of the training images and the bootstrap 3D face models are not perfect, producing *slightly inferior* basis images recovery than the frontal pose case.

For the Lambertian model, spherical harmonics representation has proved to be effective in modelling illumination variations, even when the Lambertian object is under multiple light sources. In order to verify the effectiveness of the proposed method to handle complex illumination conditions, we randomly generated the test images under multiple light sources by adding the face images at the same pose but under different single lighting conditions (we call them the component images) and taking the average. The recognition performance remains almost same as that for cases when the test images are under a single light source, as shown in Table 4.3, which demonstrates the effectiveness of the proposed method to handle complex illumination conditions.

We have to mention that although colored basis images are recovered for visualization purpose, all the recognition experiments are performed on grayscale images for faster speed. We are now investigating how color information affects the recognition performance.

4.3 Summary and Future Work

We have presented an efficient face synthesis and recognition method to handle arbitrary pose and illumination from a single training image per subject using pose-encoded spherical harmonics. With a pre-built 3D face bootstrap set, we use a statistical learning method to obtain the spherical harmonic basis images from a single training image. For a test image at a different pose from the training images, recognition is accomplished by comparing the distance from a warped version of

Table 4.2: The correct recognition rates under various poses and illuminations for 68 subjects in the CMU-PIE database, with L being the illumination and P the pose index.

$L \setminus P$	f2f	c05		c07		c09		c11		c29		c37	
		(r2f)	(r2r)	(r2f)	(r2r)	(r2f)	(r2r)	(r2f)	(r2r)	(r2f)	(r2r)	(r2f)	(r2r)
$f02$	86	84	80	84	82	82	80	82	76	82	80	80	76
$f03$	95	94	90	95	92	94	92	92	84	92	88	90	84
$f04$	97	96	94	97	95	97	94	94	90	97	94	92	88
$f05$	98	98	94	98	96	96	96	94	90	96	94	92	90
$f06$	100	100	99	100	100	100	100	98	96	100	99	98	94
$f07$	98	98	96	100	100	100	98	94	94	97	95	92	92
$f08$	97	96	94	97	95	97	94	92	90	96	94	92	88
$f09$	100	100	98	100	99	100	98	100	96	100	98	99	96
$f10$	100	100	98	100	100	100	100	96	94	100	98	92	92
$f11$	100	100	100	100	100	100	100	98	96	100	100	98	96
$f12$	96	94	92	94	94	95	95	90	88	92	92	90	86
$f13$	98	96	92	96	94	94	94	92	88	94	92	90	88
$f14$	100	100	98	100	100	100	100	98	94	99	96	96	92
$f15$	100	100	100	100	100	100	100	100	97	100	98	98	96
$f16$	98	97	95	98	96	98	96	96	92	97	95	95	90
$f17$	95	94	92	95	95	95	95	92	88	94	90	90	86
$f18$	92	90	88	92	90	90	88	86	82	90	86	86	80
$f19$	96	95	90	94	92	92	92	90	86	94	90	84	82
$f20$	96	95	92	96	94	95	94	92	88	94	90	90	84
$f21$	97	97	97	97	96	97	95	94	92	95	95	94	90
$f22$	97	97	95	96	95	95	95	94	90	95	94	92	90
mean	97	96	94	96	95	96	95	93	90	95	93	92	89
std	3.2	3.8	4.6	3.7	4.2	4.2	4.6	4.3	5.1	4.2	4.7	4.6	5.2

Table 4.3: The correct recognition rates at six rotated poses under multiple light sources for 68 subjects in the CMU-PIE database, where L is the lighting condition and N is the number of component images.

$L \setminus N$	2	3	4	5	6	7	8
c05	96%	96%	96%	97%	98%	97%	100%
c07	92%	94%	97%	97%	97%	100%	98%
c09	94%	96%	94%	98%	92%	96%	97%
c11	98%	97%	95%	92%	94%	97%	96%
c29	97%	99%	100%	96%	97%	95%	98%
c37	96%	94%	96%	95%	96%	98%	95%

the test image to the space spanned by the basis images of each model. The impact of some empirical factors (i.e., correspondence and interpolation) due to the warping is embedded in a sparse transformation matrix, and we prove that the recognition performance is not affected after warping the test image to the frontal view. Experimental results on both synthetic and real images show that high recognition rates can be achieved when the test image is at a different pose and under arbitrary illumination condition. Furthermore, the recognition results can be better verified by easily generated face image of the chosen subject at the same pose as the test image, using the linear transformation between the spherical harmonic basis images across poses.

In scenarios where only one training image is available, finding the cross-correspondence between the training images and the test image is inevitable. Unfortunately, automatic computation of these correspondences is not a trivial task and manual operation is required in existing methods. We are looking into possible solutions to this challenging problem.

Chapter 5

Homography-based View Synthesis and Robust Tracking for Surveillance Video

Target tracking and object verification from airborne video is of great importance for both military and civilian applications. Due to the distance between the camera and the observed object, all the visible points on the object are coplanar leading to structure degeneracy. Thus the image data do not contain enough information to recover the epipolar geometry. Consequently, it is not possible to compute the 4×4 projective transformation between two sets of 3D points if the only correspondences available are coplanar. This will cause difficulties in being able to design accurate and stable target tracking and object verification algorithms. However, in this case, it is reasonable to assume the observed object moves on a dominant plane (the ground plane) which induces a homography relation between two views. In this chapter, we present a robust two camera tracking method and an end-to-end verification system for moving objects, both utilizing the homography relation induced by the dominant plane.

5.1 Introduction to Homography

Suppose P is a scene point lying on a plane π. Let p and p' be the projections of P in view 1 and view 2 respectively. Then there exists a 3×3 matrix H_π such that $p' \cong H_\pi p$ where H_π is the homography matrix of the plane π [40]. For simplicity we will omit the subscript of H_π if there is no confusion in the following sections.

5.1.1 Homography Estimation

Given a set of corresponding points $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$, where $\mathbf{x_i}$ come from view 1 and $\mathbf{x_i'}$ come from view 2, and writing $\mathbf{x_i'} = (x_i', y_i', \omega_i')^T$ with homogeneous coordinate, we can estimate the homography H between the two views using $\mathbf{x_i'} \times H\mathbf{x_i} = 0$ [40]. For each pair of corresponding points, three linear equations are written as

$$\begin{bmatrix} \mathbf{0}^T & -\omega_i'\mathbf{x_i}^T & -y_i'\mathbf{x_i}^T \\ \omega_i'\mathbf{x_i}^T & \mathbf{0}^T & -x_i'\mathbf{x_i}^T \\ y_i'\mathbf{x_i}^T & x_i'\mathbf{x_i}^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{pmatrix} = \mathbf{0} \tag{5.1}$$

where $\mathbf{h_i}, i = 1, 2, 3$ is a 3×1 vector made up of the entries in the i^{th} row of H.

By stacking the coordinates of all the corresponding points into a coefficient matrix A as shown in (5.1), H is the solution to the linear equation $A\mathbf{h} = 0$ where $\mathbf{h} = (\mathbf{h_1}^T, \mathbf{h_2}^T, \mathbf{h_3}^T)^T$. For a more accurate result, robust estimation methods like RANSAC [31] or LMedS [30] estimation can be used. Before feeding into the linear equation, the coordinates of all the points are normalized such that the centroid of the points is the coordinate origin $(0, 0)^T$, and their average distance from the origin is $\sqrt{2}$.

5.2 Homography Based Robust Two View Tracking

Multi-view tracking has the obvious advantage over single-view tracking because of its wide coverage range. When a scene is viewed from different viewpoints, there are often regions which are occluded in some views but visible in other views. A visual tracking system must be able to track objects which are partially or even fully occluded. In this section we present a wide baseline, two-view visual tracking method which handles occlusions using the homography relation between the two views. An adaptive appearance model is incorporated in Sequential Monte Carlo (SMC) framework to accomplish the single view tracking. Occlusion is detected using robust statistics. When occlusion is detected in one view, the homography between the two views is estimated from previous tracking results. Correct transformation of the target in the occluded view can be inferred with the homography and the tracking result of the un-occluded view [111].

Some work has been done in handling occlusion for both single view tracking [86, 102] and multi view tracking [13, 20, 27]. In [86], an appearance model is used to accomplish tracking. When occlusion is detected, the ”disputed” pixels are classified using a maximum likelihood classifier to infer the depth order of the objects, and update the appearance model accordingly. In [102], a dynamic Bayesian network which accommodates an extra hidden process for occlusion is used to cope with occlusion. Both [86] and [102] assume that the target is occluded by a known object, which gives a clue to infer the depth ordering or compute the observation likelihood. [13] presents a multi view tracking method using a set of calibrated cameras. A Kalman filter is used to track each object in 3D world coordinates and 2D image coordinates. In [20], the correlation of visual information between different cameras is learnt using Support Vector Regression and Hierarchical PCA to estimate the subject appearance across cameras. When occlusion is detected for one camera, correspondences across cameras are built using the appearance models acquired during training, and different cues are fused based on the Bayes's theorem to make a final tracking report. [27] uses a Bayesian network to fuse the independent observations from multiple cameras and produce the most likely 3D state estimates.

The method we propose in this section uses the homography relation between two views to infer the transformation for the occluded view. Even when the target is partially or fully occluded by an unknown object, the tracker still can follow the target as long as it is visible from another view. No complicated inference scheme is used to fuse the multiple camera observations, nor 3D information needs to be explicitly recovered. The homogrphy can be robustly estimated from previous tracking results, and the motion inference for the target in the occluded view is also estimated robustly by utilizing all the points inside the tracking region. The computation is simple and fast. The result is satisfactory as shown in the experimental results.

5.2.1 Single View Appearance Tracking

We first present an appearance model-based tracking system for a single view. The system processes the video frames captured under one single view and produces the tracking parameters for later use. The task of an appearance tracker is to infer the deformation (or tracking) parameter best describing the differences between the observed appearances and the appearance model. To accommodate the dynamics embedded in the video sequence, we employ a state space time series model.

Suppose $\{Y_1, ..., Y_t, ...\}$ are the observed video frames containing the appearances of the object to be tracked. We use an affine transformation $\mathcal{T}$ parameterized by θ_t and denote the appearance model by A_t. Our time series model is fully defined by (a) *a state transition equation* and (b) *an observation equation.*

$$(a)\ \theta_t = \theta_{t-1} + U_t,\ (b)\ Z_t \doteq \mathcal{T}\{Y_t; \theta_t\} = A_t + V_t, \tag{5.2}$$

where U_t is the system noise and V_t is the observation noise. Our goal is to compute the posterior probability $p(\theta_t|Y_{1:t})$, which is used to estimate the 'best' parameter $\hat{\theta}_t$. Because this model is nonlinear (e.g. the affine transformation part), we use SMC techniques [56, 64] to approximate $p(\theta_t|Y_{1:t})$ using a set of particles. We now specify the actual model choices.

The appearance model A_t is crucial in a tracker. If a fixed template, say $A_t \equiv A_0$, is used, it is difficult to handle appearance changes in the video. On the other hand, one could use a rapidly changing model, say $A_t = \hat{Z}_t \doteq \mathcal{T}\{Y_t; \hat{\theta}_t\}$, i.e., the 'best' patch of interest in the previous frame, but this is susceptible to drift. Thus, it is necessary to have a model which is a compromise between these two cases. Mixture models are used in [53, 124]. In this chapter, we simply adapt the appearance model to the changing appearances at a moderate pace.

We assume that (i) the appearance model A_t is associated with a mean image μ_t (the actual A_t in (5.2)) and a variance image σ_t^2 (included in V_t in (5.2)), and (ii) A_t summarizes the past observations under an exponential envelop with a forgetting factor α. When the appearance in the current frame has been tracked, i.e. $\hat{Z}_t$ is ready, we compute an updated appearance model A_{t+1} and use it to track in the next frame. Using the maximum likelihood (ML) principle, one can show that μ_{t+1}

and σ^2_{t+1} can be updated in the following manner:

$$\mu_{t+1} = \alpha\mu_t + (1-\alpha)\hat{Z}_t;\ \sigma^2_{t+1} = \alpha\sigma^2_t + (1-\alpha)(\hat{Z}_t - \mu_t)^2. \tag{5.3}$$

Notice that in the above equations, all μ's and σ^2's are vectorized and the operation is element-wise. Also, V_t is distributed as a multivariate normal density $\mathcal{N}(0, D(\sigma^2_t))$, where $D(\sigma^2_t)$ denotes a diagonal matrix with diagonal elements σ^2_t.

The system noise U_t constrains the particle coverage. It is ideal to draw particles such that they are close to the object. In addition, the particle coverage should also accommodate the extent of clutter in the observation. To this end, we use $U_t \sim \mathcal{N}(\nu_t, r_t I)$, where ν_t is the 'instantaneous' velocity in the tracking parameter, r_t is the noise variance measuring the extent of clutter, and I is an identity matrix.

However, we have no knowledge of ν_t and r_t. We use a linear prediction scheme to estimate them. This prediction scheme is in spirit similar to finding an affine flows for the current 'best' patch in the next frame. Refer to [124] for details. As a consequence, the prediction scheme produces an estimate of ν_t and a prediction error ϵ_t. We take r_t as a monotone function of ϵ_t. Also, we vary the number of particles according to r_t.

When occlusion happens in one view, we need a mechanism to detect it. We assume that occlusions produce large image differences which can be treated as 'outliers'. Outlier pixels cannot be explained by the underlying process. If a pixel x satisfies $|\hat{Z}_t(x) - \mu_t(x)|/\sigma_t(x) > c$ (we take $c = 0.75$), we declare the pixel to be an outlier. This actually corresponds to using a robust statistics [47]. If the number of the outlier pixels in $\hat{Z}_t$, say d_{out}, exceeds a certain threshold, i.e., $d_{out} > \lambda d_{total}$ (we take $\lambda = 0.13$), we declare an occlusion. Once occlusion is declared, we stop updating the appearance model and estimating the motion velocity and start using the information derived from other views to maintain tracking. To cancel the occlusion alert, we compare the image warped from the other views with our observation till the error is consistently small. Tracking is then resumed.

5.2.2 Occlusion Handling With Homography

We consider a wide baseline two view tracking system. In order to estimate the homography, we have to build the correspondence between the two views, which is always challenging especially for wide baseline views. Although H can be estimated from at least 4 pairs of corresponding points (the more we can find, the more robust H will be) in the initial frame, it is more robust to utilize the corresponding points in all frames. Assuming that the object moves on the same dominant plane for all the frames, it is clear that the corresponding points in all frames will contribute in estimating H. Suppose n pairs of corresponding points $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}, i = 1, 2, \ldots, n$ on the object were picked in the initial frame, then their corresponding relation is kept for all the frames (through the inter-frame affine transformation $\mathcal{T}$'s known from the tracking result) and can be used to estimate H. One assumption used here is that for the corresponding points in the previous frames, after taking the affine transformations in both views for the current frame (i.e., we have $\mathbf{y_i} \leftrightarrow \mathbf{y'_i}$ where $\mathbf{y_i} = \mathcal{T}_1\mathbf{x_i}, \mathbf{y'_i} = \mathcal{T}_2\mathbf{x'_i}$), they are still linked to each other with the same homography H as in previous frames. This assumption usually will not hold since an affine transformation concatenated with a homography gives another homography instead of another affine transformation. Considering this, we do not directly assume $\mathbf{y_i} \leftrightarrow \mathbf{y'_i}$ as the true corresponding points. Instead, after getting $\mathbf{y_i}$'s in view 1, we do a random local search around $\mathbf{y'_i}$'s in view 2 to find the correct corresponding points for $\mathbf{y_i}$'s. Nevertheless, since the tracker works well enough in our experiment, which means the difference between the two

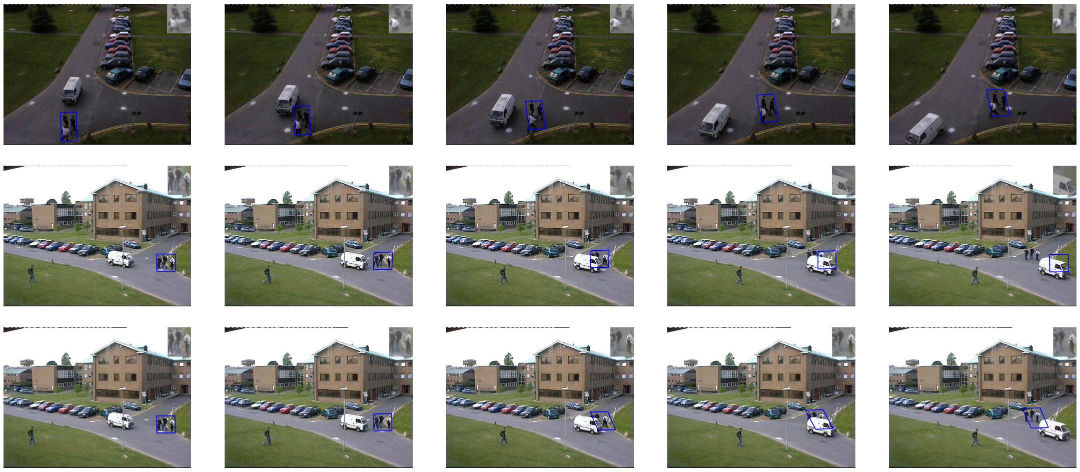


Figure 5.1: Two view tracking result with the target partially occluded by an unknown object, with the appearance model A_t shown at the upper right corner. Top row: tracking result for the unoccluded view. Middle row: tracking result for the partially occluded view without occlusion handling. Bottom row: tracking result for the partially occluded view with occlusion handling

frames can be satisfactorily described with an affine transformation, we can always find the correct correspondences in a very close neighborhood around $\mathbf{y_i'}$'s.

5.2.3 Transformation inference for the occluded view

Suppose at frame j occlusion is detected for view 2, but not for view 1. Denote $\mathcal{T}_1^j$ and $\mathcal{T}_2^j$ as the affine transformations from frame $j-1$ to frame j for view 1 and view 2, respectively. We need to derive $\mathcal{T}_2^j$ from H and $\mathcal{T}_1^j$. Let $\mathbf{x}^{j-1}$ and $\mathbf{x}'^{j-1}$ be a pair of corresponding points at frame $j-1$ for view 1 and view 2 respectively. Then we have

$$\mathbf{x}^j = \mathcal{T}_1^j \mathbf{x}^{j-1};\ \mathbf{x}'^j = \mathcal{T}_2^j \mathbf{x}'^{j-1}, \tag{5.4}$$

and

$$\mathbf{x}'^{j-1} = H\mathbf{x}^{j-1};\ \mathbf{x}'^j = H\mathbf{x}^j. \tag{5.5}$$

Knowing H and $\mathcal{T}_1^j$, it is easy to derive from (5.4) and (5.5) that

$$\mathcal{T}_2^j = H\mathcal{T}_1^j H^{-1}. \tag{5.6}$$

Although (5.6) gives a theoretically correct solution for $\mathcal{T}_2^j$, it gives a homography while the sought solution is an affine transformation in accordance with the tracker. Practically $\mathcal{T}_2^j$ can be obtained from $\mathbf{x}'^{j-1}$'s and the inferred $\mathbf{x}'^j$'s. Writing $\mathbf{x}'^k = (x'^k, y'^k, 1)^T, k = j-1, j$, and $\mathcal{T}_2^j = \begin{pmatrix} \alpha_1 & \alpha_2 & t_x \\ \alpha_3 & \alpha_4 & t_y \\ 0 & 0 & 1 \end{pmatrix}$, we have

$$\begin{pmatrix} x'^j \\ y'^j \end{pmatrix} = \begin{pmatrix} x'^{j-1} & y'^{j-1} & 0 & 0 & 1 & 0 \\ 0 & 0 & x'^{j-1} & y'^{j-1} & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ t_x \\ t_y \end{pmatrix}. \tag{5.7}$$

A minimum of 3 pairs of corresponding points is needed to solve for $\mathcal{T}_2^j$ from (5.7). To get a more robust solution, we want to use all the points inside the tracking region to form an over constrained linear equation and seek the least square estimate. To this end, we have to infer the coordinates for all the points inside the tracking region at frame j. Given 3 non-collinear points $\mathbf{p_i}, i = 1, 2, 3$ on the image of an object, the relation between $\mathbf{p_i}'$s and any other image point $\mathbf{q}$ on the object stays invariant under affine transformation $\mathcal{T}$, i.e., if $\mathbf{q} - \mathbf{p_1} = \beta_1(\mathbf{q}-\mathbf{p_2})+\beta_2(\mathbf{q}-\mathbf{p_3})$, then we have $\mathcal{T}(\mathbf{q}-\mathbf{p_1}) = \beta_1\mathcal{T}(\mathbf{q}-\mathbf{p_2})+\beta_2\mathcal{T}(\mathbf{q}-\mathbf{p_3})$. Recall that up until frame j we have stored $n(j-1)$ pairs of corresponding points in order to estimate H. With H and $\mathbf{x_i}^j, i = 1, 2, \ldots, n$, we can compute $\mathbf{x}'^j_\mathbf{i}, i = 1, 2, \ldots, n$

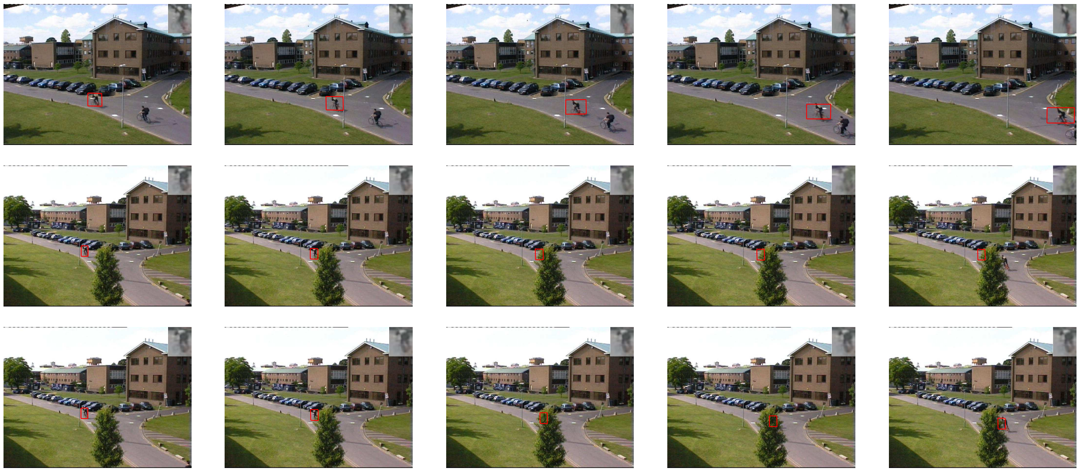


Figure 5.2: Two view tracking result with the target fully occluded by an unknown object, with the appearance model A_t shown at the upper right corner. Top row: tracking result for the un-occluded view. Middle row: tracking result for the fully occluded view without occlusion handling step included. Bottom row: tracking result for the fully occluded with occlusion handling step included.

with (5.5). Then the coordinates for all the other points inside the tracking region can be obtained accordingly. Here the number of initially picked correspondence pairs n can be as few as 3 if they are non-collinear, so the difficulty of finding the required number of corresponding points in the initial frame is greatly reduced.

5.2.4 Experimental Results

Experiments were conducted on the PETS2001 test sequence [2]. Figure 5.1 shows the sequence where three walking humans are visible in all the frames for view 1, and are partially occluded by an incoming vehicle in some frames and reappear afterwards for view 2. The appearance model A_t is shown at the upper right corner of each frame. The top row of Figure 5.1 shows the tracking result for view 1 (the un-occluded view). The middle row shows the tracking result for view 2 (the partially occluded view) without using homography to handle occlusion. We see that the appearance model keeps updating even when there is occlusion, and the tracker stays with the vehicle instead of the walking humans. The bottom row of Figure 5.1 shows the tracking result for view 2 using homography and occlusion handling step included. If there is no occlusion detected, the two views are tracked independently. When occlusion is detected in view 2, the appearance model is not updated, and the affine transformation is inferred from the tracking result for view 1 and the computed H. It is clear that the tracker in view 2 still tracks the walking people even when they are partially occluded by the vehicle and regains control as soon as they fully reappear.

Figure 5.2 shows similar experiment results, except that the to-be-tracked walking person is fully occluded by the tree in view 2. The tracking results for view 1 (un-occluded view), view 2 (occluded view) without using occlusion handling step, and view 2 using homography to handle occlusion are shown in the top, middle and bottom rows of Figure 5.2, respectively. We can see from the bottom row that the tracker can track the person even though he/she is fully occluded by the tree, while the tracker stays where the tree is when the occlusion is not handled (as shown in the middle row).

5.3 Moving Object Verification from Airborne Video

Object verification differs from the Automatic Target Recognition (ATR) problem in that it does not seek to identify the observed object, only to confirm that it is the same object that has been observed recently, and thus does not require prior training data. Verification remains challenging due to the potentially large changes of the object's pose, illumination, or occlusion between initial and subsequent observations. Appearance characteristics of the object must be captured during the initial observation and stored for later verification. A typical application for a verification system is within vehicle trackers; verification is required after the vehicle is obscured for a while or leaves the field of view.

The problem of object verification of vehicles, which is the objects of interest in this section, has received intense attention in recent years. The scale-invariant feature transform (SIFT) method [65] represents an image by a collection of features that are invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and affine or 3D projection. It is very effective in high-resolution videos where the features can be extracted reliably, but does not work well in airborne videos with lower resolution. In the detection and classification method by Gupta et al. [38], vehicles are modeled as rectangular patches with certain dynamic behavior. This method is based on the establishing correspondences between regions and vehicles, and only deals with stationary cameras. In [63], a classification metric, together with a temporal consistency constraint, is applied to classify all moving blobs into human, vehicle or background clutter. Sivic et al. [92] proposed a method for automatically associating image patches from frames of a movie shot into object-level groups. Multiple parts of an object can be matched from many different frames using the patch-based multi-view feature grouping. Inspired by this approach, Guo et al. [37] customized the alignment and flexible matching components to suit the resolution constraints as well as the goal of exact matching. A good review can be found in [45] on the recent developments and the processing framework of visual surveillance in dynamic scenes,

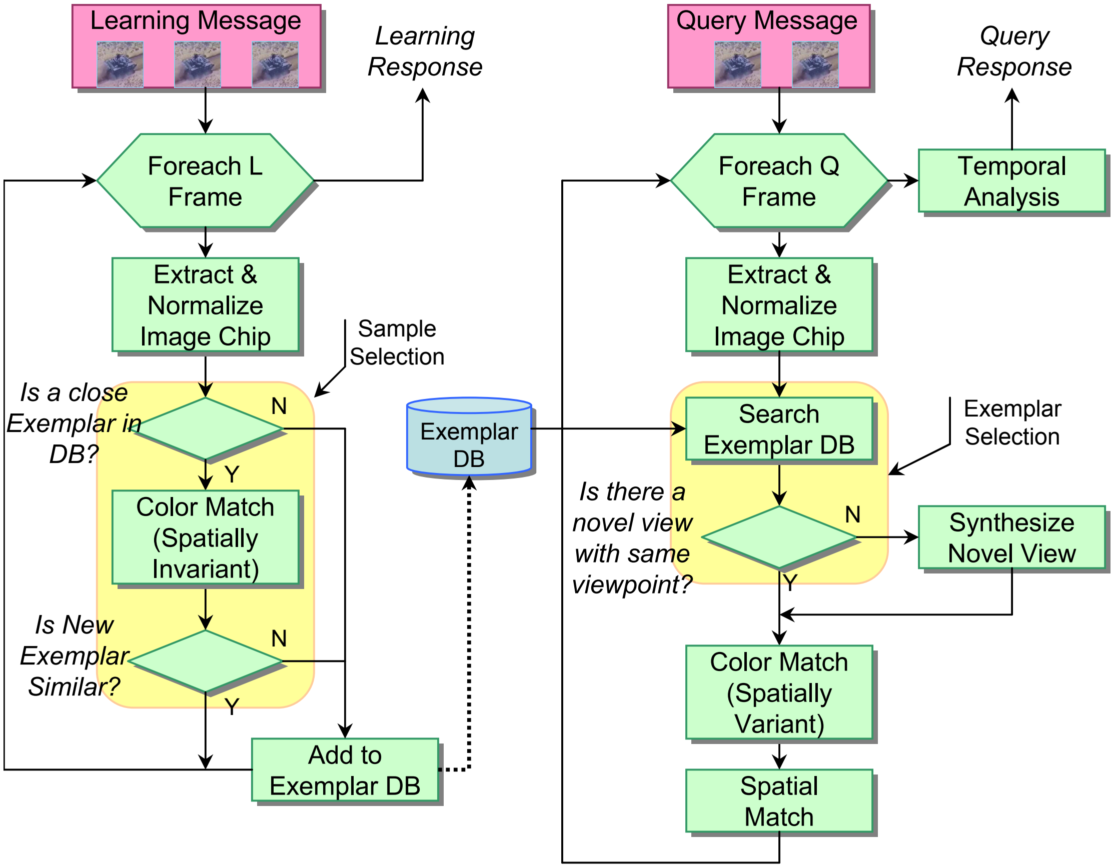


Figure 5.3: An overview of the proposed object verification system. The learning and query processes are independent but share the examplar database.

including modeling of environments, detection of motion, classification of moving objects, human identification, and fusion of data from multiple cameras.

In this section we present an end-to-end verification system for moving vehicles in airborne video. The system has separate learning and verification (or query) functions that share a common database. The flow chart of the proposed system is shown in Fig. 5.3. Key contributions of the proposed system include: 1) a homography-based view synthesis method to handle the varying appearance of an object due to changing viewpoints between the learning exemplar and the query image; and 2) the use of both spatial and temporal models to match how an object looks and how it behaves, respectively. Spatial models describe the color or grayscale variations, texture, and geometric features of the object. Temporal models describe an object's expected behavior.

Since it is impossible to know in advance which objects will be verified, models must be generated on the fly from real-time video. Objects are typically selected and segmented by a tracker or other modules which request the verification. To learn, the system is provided a short video, referred to as a learning message (usually around 1 second long), containing the object of interest, the associated metadata, and an arbitrary identification (ID) number for later reference. This is all the information the system uses to build a model of the object. Samples from the learning message are selected and stored in an exemplar database for use during verification. The collection of samples (exemplars) for a given object are referred to as the object model. A sample selection module is exploited to reduce the number of exemplars by saving only those that differ appreciably from the existing ones. Because of the short learning message, the database will typically be sparse which is another major difference from traditional ATR systems.

For verification, the system is provided another short video, called a query message (usually 0.5 second long), containing the object to verify, the associated metadata, and a set of model ID numbers to verify it against. Due to the sparse nature of the database, it is unlikely that an exemplar will exist with the desired viewpoint and resolution for matching. Thus, a homography-based view synthesis method is used to generate a novel view of the object with the necessary viewpoint and resolution.

Novel view synthesis can be accomplished by recovering the 3D information of the object from the available images using SfM techniques and projecting to the desired view. When the object to be synthesized is at a great distance from the camera, its depth-relief is negligible, and it is a reasonable approximation to assume the object moves on a dominant plane (the ground plane). According to [96], if all the visible scene points are coplanar (i.e., structure degeneracy), the image data does not contain enough information to recover the epipolar geometry. Consequently, it is not possible to compute a 4×4 projective transformation between two sets of 3D points if the only correspondences available are coplanar. Therefore, SfM methods, which essentially need the projective transformation between the 3D scene and the images, may not be accurate or stable. For the same reason, the "Plane + Parallax" approach [49] do not apply here since the object of interest is approximately a flat scene and the 3D structure can not be reliably estimated. Alternatively, image-based rendering techniques can be used, which rely on view interpolation or pixel reprojection and do not explicitly build a 3D model. In order to accomplish the view synthesis task for moving objects in airborne video, we resort to homography induced by the ground plane. The region of interest (ROI, in our case, the moving object) is tracked with the appearance based visual tracking method described in Section 5.2.1. The on-object point correspondence is built from the tracking parameters and used to estimate the homography induced by the ground plane for each pair of frames. With known camera focal length, the surface normal to the ground plane and the camera motion between the frame pair are factored out from the homography using Singular Value Decomposition (SVD), as shown in [97]. With the tracking result across multiple frames and the estimated rotation between each frame pair available, a rank one constraint is applied to decompose a matrix that contains the homographies from multiple frame pairs [81] to ensure robust surface normal estimation. Given a desired viewing direction, the novel image of the object is generated by warping the reference frame using the new homography between the desired viewpoint and the reference frame.

Efficient integration of spatial and temporal model matching assures the robustness of the verification step. A color matcher is also employed which incorporates color co-occurrence histograms described in [19] to compare colors and color adjacency. A rotationally invariant color matcher can be achieved by measuring the color adjacency without respect to any direction. The spatial matcher extracts features from the query image chip, and compares them (type and location) against features extracted from the novel view. Feature locations are compared using the Distance Transform [51]. The color matcher and the spatial matcher scores are combined using a weighted average rule.

Temporal analysis enforces consistency over time by applying a temporal model that requires the object's orientation to vary smoothly with time. Temporal models are a distribution on the probability that an object will rotate a given amount between frames. The temporal model is based on physics, and on expected object size and operating conditions. Spatial matches to an incorrect object tend to match at random orientations, which are interpreted within the temporal analysis as erratic behavior; correct matches appear to have a smooth behavior consistent with the temporal model. Because temporal analysis can distinguish between these cases, the system quickly converges to a decision even when the underlying spatial matches are weak.

5.3.1 The System

In this section, we describe in detail the components of the proposed moving vehicle verification system for airborne video in detail [36, 113, 114].

Image Normalization

Consistent and comparable image statistics are desired to optimize color and spatial features for matching. Considerable differences in image statistics occur between the learning and the query messages due to changing backgrounds, and to the time gap between them. Statistics can also change within a learning or query message due to glints and shadows that cause the sensor's automatic gain control (AGC) to adjust. Consequently we apply a normalization process to all image data in the learning and the query messages to accommodate for these variations.

An adaptive histogram stretching method is used to redistribute the brightness of the images, enhancing and normalizing their contrast characteristics. For color images, histogram stretching is applied only to the Y component (luminance) of YUV color images (images are converted to YUV from other formats). For infrared (IR) images, histogram stretching is applied to the single grayscale component. The adaptive nature of the algorithm spreads the majority of values to 80% of the luminance range while compressing the highest and lowest values through the use of pivotal points in the histogram stretch mapping function. The lowest and highest luminance values primarily represent dark shadows and glints; compressing them minimizes their impact. This adjustment leads to more consistently defined edges for spatial matching while maintaining the color information. Fig. 5.4 shows several examples before and after normalization.

Sample Selection

We wish to restrict the number of exemplars saved in an object model for two reasons. First, the storage and processing requirements grow as a function of the number of exemplars. Second, exemplars with very similar appearance do not

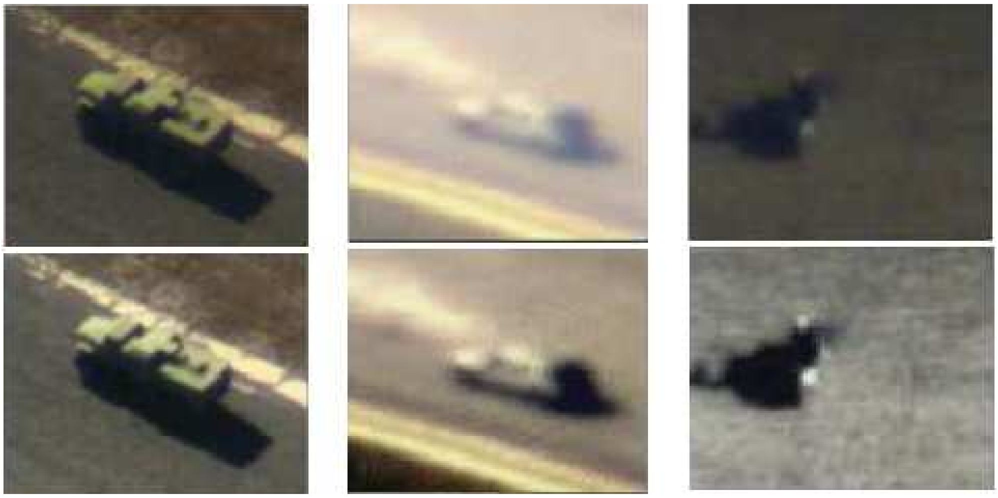


Figure 5.4: Normalization provides consistent image statistics for downstream processing. Top row: the images before normalization. Bottom row: results after normalization .

necessarily increase the model fidelity. Since the appearance of an object does not change considerably between consecutive frames, this provides a convenient way to restrict the number of exemplars retained. We choose to retain exemplars only if the collection geometry or appearance changes considerably. Changes in collection geometry are derived from the metadata provided with each learning message. It contains the location and the extent of the object in the frame, an aspect angle on the ground (usually the direction of velocity as determined by a tracker), the observation angle from the sensor, and other camera parameters. We arbitrarily segment the exemplar database into 5-degree bins for both aspect and elevation angles. As each frame within a learning message is processed, an exemplar is always saved if its collection geometry falls within an empty bin. If the bin is not empty, then the new exemplar is saved only if its appearance is different from those already in that bin. Differences in appearance may result from illumination, focus, or background changes; atmospheric effects; or the presence of nearby objects within the defined extent. Similarity in appearance is measured using a rotationally variant version of the color matcher described later. A simple threshold on the match score determines the similarity.

Each frame within a learning message is considered only with respect to exemplars already in the database. Thus, the system can process additional learning messages as an object is tracked or otherwise observed over time. This allows the fidelity of an object model to increase as additional exemplars are collected.

Color Matching

Being an important component of the proposed target verification system, the color matching algorithm is based on a technique proposed by Chang and Krumm [19] that utilizes a color co-occurrence histogram to recognize objects in images. It has been adapted to work with either color images or IR images.

The foundation of the Color Matching algorithm lies in the generation of high-quality color co-occurrence matrices (CCMs). Creation of CCMs is accomplished by examining how colors in the image are distributed in relation to each other. The CCM is $n \times n \times nDist$ where n is the number of colors and $nDist$ is the number of distances. Each cell contains the number of times that a particular color (specified by the row index) is at a given distance away (specified by the nDist index) from another color (specified by the column index). If all eight directions from a pixel are counted, then the CCM will enable a rotationally invariant match. If only a single direction is counted, say to the right, then some geometric information is also captured and rotation dependent matches are made. To reduce the obvious computational complexity involved in examining all colors at all distances the image is first quantized down to a manageable number of colors. Quantization is accomplished by thresholding and clustering RGB color bands into discrete groups. For IR images, grayscale values are dynamically quantized into eight discrete levels. Next, an empirically defined distance measure is set that manages the extent of the search area from the reference pixel to sample pixels. This distance measure is dynamically set according to the resolution of the image.

The intersection between a pair of CCMs is calculated to determine the color match score. This score provides a qualitative metric to gage the difference between two images. Each CCM is normalized so that its elements sum to one. The intersections of the diagonal and non-diagonal matrix elements are determined separately and then combined in a weighted sum. The diagonal elements for a given distance represent areas of uniform color in the image. Given a CCM for the query (qCCM) and for the exemplar (lCCM), the intersection of the diagonal elements is given by: $diag = \sum_{k}^{nDist} \sum_{i}^{n} min(qCCM_{ii}^{k}, lCCM_{ii}^{k})$.

The non-diagonal matrix elements represent areas where different colors are adjacent to each other. This typically occurs at the boundaries of different areas of the image, thus providing a measure of image texture. The intersection of the non-diagonal elements is given by: $ndiag = \sum_{k}^{nDist} \sum_{i}^{n} \sum_{j}^{n} min(qCCM_{ij}^{k}, lCCM_{ij}^{k}), i \neq j$.

To calculate a final color score, the diagonal and nondiagonal intersection values are combined in a weighted average. Proper selection of the weight for the diagonal intersection (dWt) and the weight for the non-diagonal intersection ($ndWt$) allows emphasis of one feature over the other. The final color score is calculated as:

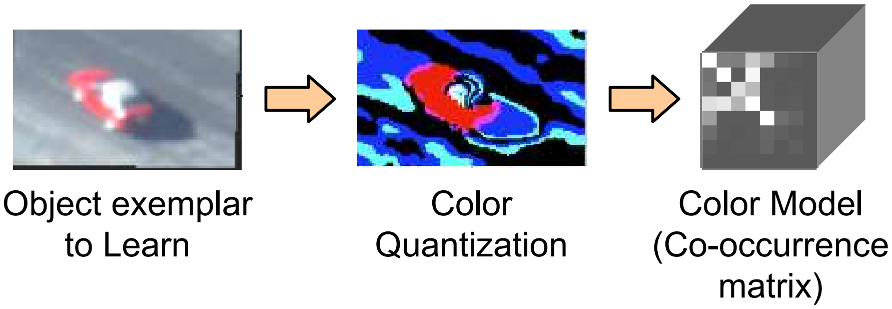


Figure 5.5: The color model is a set of square co-occurrence matrices whose dimensions are the number of quantized colors.

$d = 1.0 - \frac{(dWt \cdot diag) + (ndWt \cdot ndiag)}{nDist \cdot (dWt + ndWt)}$. Since the CCMs were normalized, the color score ranges from zero to one. The weighted average is subtracted from one to force a lower-is-better score. An example development of a CCM is shown in Fig. 5.5.

Exemplar Selection

The spatial and color matchers perform best when the viewpoints of the selected exemplar and the object to be verified (the query object) are the same. For spatial matching, view synthesis is used to create a novel view with the required viewpoint from exemplars in the database. However, a homography-based view synthesis (in Section 5.3.2) can only modify the viewpoint of those parts of the object already appearing in the exemplar. That is, parts of the object not observed in the exemplar cannot be projected to the novel view. Thus, the exemplar with the closest collection geometry to that of the query object is selected from the database. We also assume symmetry between the left and right sides of the object. This selection metric ensures the greatest overlap of the observed parts between the exemplar and the query object.

Notice that more effective selection method could be employed since multiple exemplars are stored within a geometry bin if they do not appear similar. Currently, collection geometry is the only criteria used. A different method might select the newest exemplar within a bin, the one with the closest sun angle, or some other measure. This is an issue for future investigation.

Spatial Feature Extraction

The spatial matcher compares the spatial features of two objects. These features typically describe the extent, shape, distinguishing characteristics, or textures on the object. We chose to use the horizontal and vertical ridges because they are relatively stable with respect to small rotations, project well into novel views, and capture both the object's outline and surface characteristics (lines, points, segments). Thus, discontinuities in color or texture are captured, such as those between the windshield and the hood, or the line between doors. Other features, such as the circular Laplacian of Gaussians (LoG) were found to give little or no improvement for the added computational complexity and processing time.

Ridges are defined as either a dark-to-light or light-to-dark transition. Horizontal ridges are extracted using a sliding 1×3 window. If the difference between the center pixel and either side pixel is greater than a threshold, then a ridge is detected. The value of the threshold is selected to accommodate image noise and to set the minimum strength of the ridge. We also use a threshold of zero to capture clean edges. The location of the feature is retained using a binary representation to indicate if the feature is present at that pixel location. Vertical ridges are extracted in a similar manner using a 3×1 window and retained in a separate binary plane. Locations of each feature type are independently retained so that

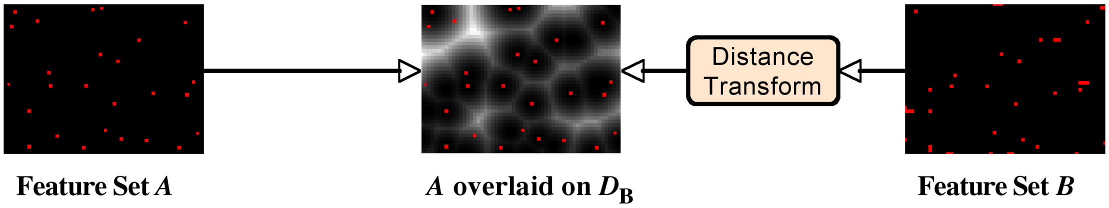


Figure 5.6: The Distance Transform provides a similarity measure between two point patterns.

only similar features can be matched.

Spatial Matcher

Spatial matching operates by determining the similarity of the extracted feature locations between the query object and a novel view. We use a spatial matcher based on the Distance Transform (DT) [82]. It provides a convenient method for measuring a difference between two feature sets.

An example of using the DT for matching a single feature type is shown in Fig. 5.6. Let feature set $A(\mathbf{p})$ have the value 1 at each pixel location $\mathbf{p}_i = (x_i, y_i)$ where a feature was extracted from the novel view, and zero otherwise. Similarly, let feature set $B(\mathbf{p})$ be a binary representation of the feature locations from the query object. Then the DT of feature set B, represented by $D_{B(\mathbf{p})}$, specifies the distance from each pixel to the nearest feature location in B. Larger distances appear as a whiter gray in the example shown in Fig. 5.6.

A quantitative measure for the strength of the match is obtained by pixel-wise multiplication of feature set A with D_B and summing over the entire range. Intuitively, this can be thought of as overlaying feature set A on D_B and summing the DT values at each feature location in A. Smaller values indicate a better match with zero indicating a perfect match. Because the different collection geometry and

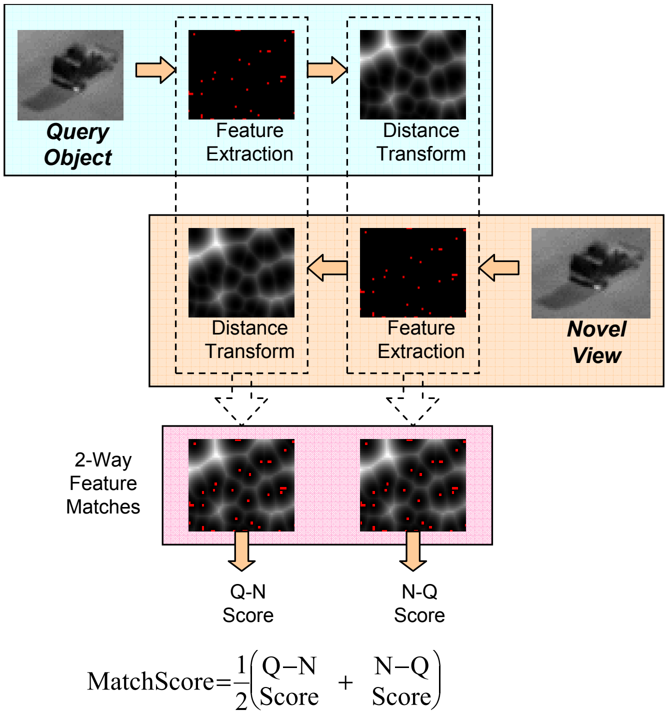


Figure 5.7: Matching the query to the novel view and vice versa increases the match robustness when either the query or the examplar is partially obscured.

segmentation of each view will slightly vary the location of extracted features, we slide A over D_B until a best fit (minimum match score) is obtained. Algebraically, for N pixels in the region to be matched the score S_{AB} is given by

$$S_{AB} = \sum_{i}^{N} D_B(\mathbf{p}_i)A(\mathbf{p}_i - \mathbf{u}) \tag{5.8}$$

where $\mathbf{u} = (\Delta x, \Delta(y))$ is an arbitrary translational offset used to minimize S_{AB}. Typical values for S_{AB} range from 0.5 to 2.0.

Missing feature points in feature set A, possibly due to occlusion, will decrease the score and indicate a better match. Likewise, additional points, generally due to noise, will increase the score. But matching can be performed in either the forward direction (comparing feature set A against D_B) or the reverse direction (comparing

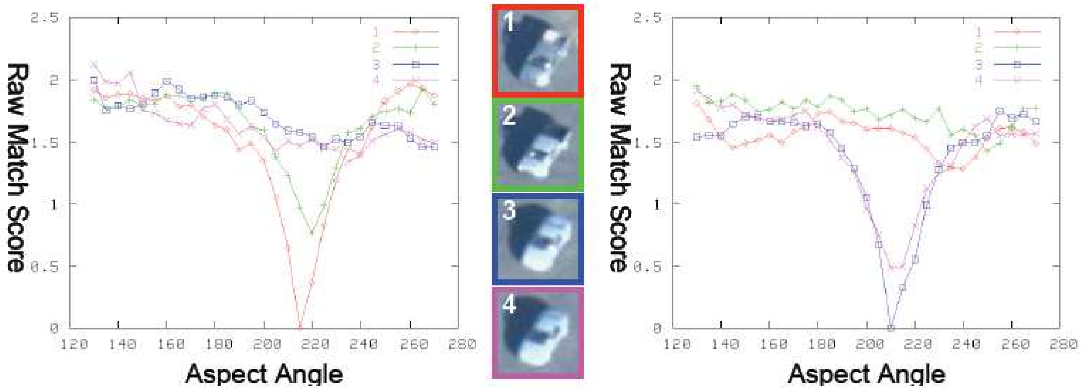


Figure 5.8: The DT matcher easily distinguishes between two similar trucks (left graph) and between two similar sedans (right graph). Performance degradation is graceful as the orientation changes.

feature set B against D_A). To increase robustness to noise and occlusion, both the forward and reverse matches are performed and the final match score is the average of the two. Fig. 5.7 illustrates the complete two-way spatial matching procedure.

Two examples in Fig. 5.8 show the typical behavior of the spatial matcher. Samples of four vehicles (two trucks, a 2-door sedan, and a 4-door sedan) were obtained at approximately 5-degree intervals. In each example, a single aspect angle of one of the vehicles was selected as the query object and matched against all four vehicles at each aspect angle. The graphs plot the match scores as a function of the aspect angle where the match occurred. For the example on the left, the selected query object was vehicle 1 (a truck) at 215 degrees. The red line shows the match scores between this query object and itself; it is zero to indicate a perfect match when the object is matched against itself at the same aspect angle. Within this neighborhood, the query object always matches to itself better than to other objects. Notice that the match to the other truck is better than to either sedan; this is a desirable feature that similar vehicles match better than dissimilar ones. Outside the neighborhood, all matches are poor. For the example on the right, the selected query object was vehicle 3 (the 2-door sedan) at 210 degrees. Again, the graph indicates a perfect score when the query object is matched against itself at the same aspect angle. In this example, the two sedans are almost identical and appear nearly the same in the picture, yet the spatial matcher detects a measurable difference between them. Notice in both examples how the match scores degrade gracefully as the difference in aspect angle increases or the similarity of vehicles decreases.

View Synthesis

To successfully match the spatial features of two objects, both objects must be viewed with the same perspective and have the same resolution. That is, they must have the same collection geometry. Because this system is trained on the fly using short learning messages, the exemplar database typically contains only a sparse set of samples that are closely bunched around one collection geometry. Thus it is unlikely that an existing exemplar will have the same collection geometry as the query object.

The objective of view synthesis is to create a novel view from object exemplars that will simulate the collection geometry of the query object. Features extracted from the novel view should then be at the correct locations for a robust spatial match. In our application, as the distance from the camera to the object is much greater than the height of the object, we use a homography to create a novel view from the selected exemplar. The homography implements a full projective transform to simultaneously handle changes in aspect angle, depression angle, and resolution. Details of view synthesis are presented in Section 5.3.2.

Temporal Analysis

Temporal Analysis combines the spatial matching with an object's temporal behavior. The behavior, by which we mean the object's rate of rotation over time, is tracked using a Dynamic Bayesian Net (DBN); a separate DBN is used for each reference model that the query object is compared against. For each video frame, or one observation in time, the DBN has 72 states representing the orientations of every 5 degrees in aspect angle. The value in each state is the probability that the query object is at that aspect angle, given the previous observations up to the current frame. The probability is updated for each frame based on the current match score, state probabilities from the previous frame, and a set of transition probabilities called the temporal model.

The temporal model is a probabilistic description of how fast an object can reasonably be expected to rotate based on physics and the frame rate. It represents the probability that the query object will rotate from orientation j at frame $f-1$ to orientation i at frame f. We use a discrete probability distribution defined as

$$A_{ji} = \begin{cases} 0.6 & \text{if } j = i \\ 0.2 & \text{if } |j-i| = 1 \\ 0.0 & \text{otherwise} \end{cases} \tag{5.9}$$

The temporal model can easily accommodate skipped frames by convolving A_{ji} with itself once for each skipped frame.

For use in the DBN, the individual raw match scores must first be converted to likelihoods. As described earlier, a lower match score indicates a better match, and a score of zero indicates a perfect match. Therefore we choose an exponential decay as an appropriate model for the probability density of match scores. If $S_{i,f}$ is the raw match score at orientation i during frame f, then we model the likelihood as $p(S_{i,f}|\theta_i) = e^{-S_{i,f}}$. Let $S_F = (S_{1,1}, S_{2,1}, \ldots, S_{72,1}, S_{1,F}, S_{2,F}, \ldots, S_{72,F})$ be the set of scores for all 72 orientations up to frame F, and $p_{i,f}(S_F|\theta_i)$ be the state probability in the DBN of the observed scores up through frame $F = f$ given that the query object is currently at orientation i. With each new frame, the state probabilities are recursively updated using those from the previous frame, the current match scores $S_{i,f}$, and the temporal model. The forward algorithm [74] provides a convenient approach to the calculation $p_{i,f}(S_F|\theta_i) = \sum_j p_{j,f-1}(S_{F-1}|\theta_j)A_{j,i}e^{-S_{i,f}}$.

The forward algorithm needs to be initialized for the first frame. Since we have no information about the orientation of the query object prior to the first time it was observed (i.e., the first frame), we assume that the object has equal probability of being in any of the 72 orientations. Thus we define $p_{i,0}(S_0|\theta i) = 1/72$ where $S_0 = \o$. Other definitions for $p_{i,0}$ also can be made, for example, we can assume that the object's orientation immediately prior to being observed is the same as that provided by the tracker for frame 1.

Once the state probabilities are calculated for a given frame, the likelihood that the query object is the same type as the reference model it is being compared to, is simply the sum of the probabilities for all orientations. This likelihood will decrease with each added frame as the probabilities are multiplied. To overcome this, we use the geometric average of the likelihood over the number of frames. That is $p_f S_F = (\sum_i p_{i,f}(S_F|\theta_i))^{1/F}$.

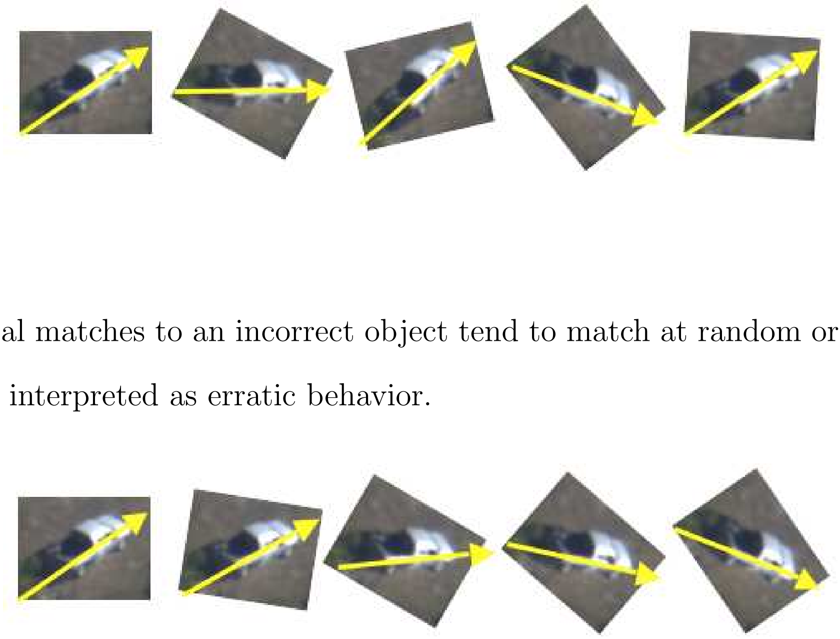
The DBN improves the overall performance by requiring temporal consistency of an object's orientation. In practice, an incorrect reference model either matches well at erratic orientations (implying erratic behavior), or matches poorly at all orientations. Thus its likelihood will decrease rapidly over time in the DBN. A correct reference model will have a strong match at the correct orientation and somewhat weaker matches at the orientations on either side (i.e., at ±5 degrees); its likelihood will decrease at a much slower rate. Fig. 5.9 shows the comparison of the temporal models for the incorrect and correct matches, indicating that the temporal analysis helps the system to assure the robustness of the verification.

Ideally a novel view is created at each of the 72 aspect angles and matched against the query object. In a practical application, the aspect angle provided by a tracker is derived from the forward velocity of the object; thus it will be close to the true orientation. To reduce the computational load, matches are only performed at ±20 degrees about the orientation of the query object, reducing the number of matches from 72 to 9. We assume that match scores for the remaining states will be poor and arbitrarily give them a correspondingly large match score of 5.0.

5.3.2 A Homography-based View Synthesis Method

The objective of view synthesis is to generate the novel view for each image in the exemplar database that has the same pose as the query object in each frame. The transformation which links the on-object points across frames can be approximated by a homography.

Establishing point correspondence is always challenging, especially for unstabilized video without rich texture. We use the appearance based visual tracking method in Section 5.2.1 to build the point correspondence. The inter-frame transformation we obtain from the tracking algorithm captures the combined motion of the object and the camera. Without loss of generality, we pick the first frame



(a): Spatial matches to an incorrect object tend to match at random orientations, which are interpreted as erratic behavior.

(b): Correct matches have smooth behavior consistent with the temporal model.

Figure 5.9: Temporal models describe how an object behaves over time, and help the system to assure the robustness of the verification.

as the reference frame wherein the object of interest is specified by the user and tracked through the entire video sequence. Because of the rigidity of the object, all pixels on the object undergo the same transformation so the point correspondence across frames can be found using some sampling techniques.

Two tracking results are shown in Figure 5.10 and Figure 5.11.

Plane Function Estimation

For a specified viewing direction $[R_{new}|\mathbf{t_{new}}]$ relative to the reference frame, the homography H_{new} between the reference frame and the desired viewpoint induced

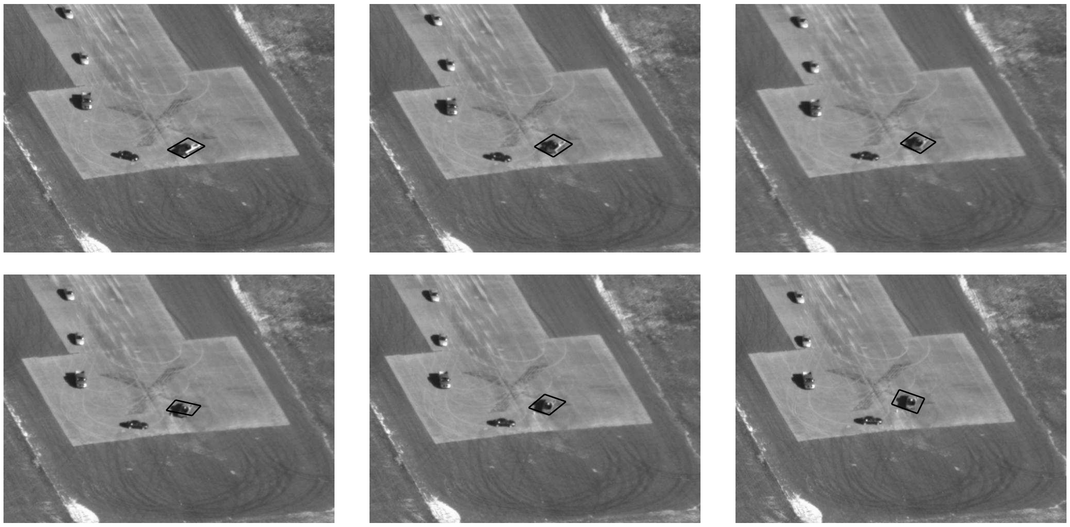


Figure 5.10: The appearance based visual tracking result, with the ROI marked as a black box in each frame.

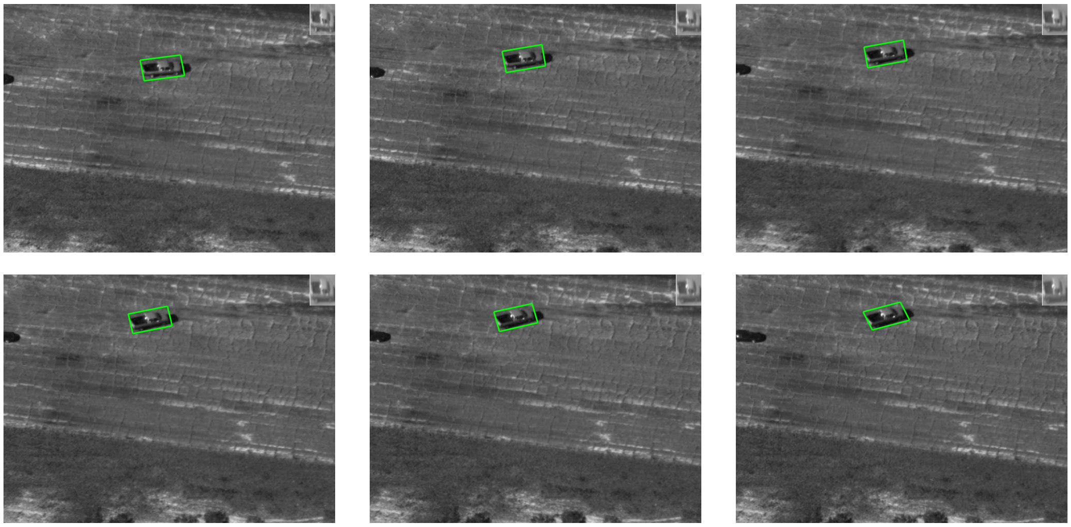


Figure 5.11: Another visual tracking result, with the ROI marked as a black box in each frame and the top right corner showing the appearance model updated at each frame.

by the ground plane is given by

$$H_{new} = K_{new}(R_{new} - \mathbf{t_{new}}\mathbf{n}^T){K_1}^{-1} \tag{5.10}$$

where K_{new} is the camera calibration matrix for the desired view, K_1 is the camera calibration matrix for the reference frame, and $\mathbf{n}^T$ is the surface normal to the ground plane in the coordinate system of the reference frame. Therefore, the on-object points $\mathbf{p}_i$'s in the reference frame and the corresponding points $\mathbf{p}_i'$'s in the desired view are related by $\mathbf{p}_i' = H_{new}\mathbf{p}_i$, which can be used to generate the desired image by warping the points from the reference frame.

In order to get H_{new}, we need to know K_1, K_{new}, and $\mathbf{n}^T$. By assuming that the principal point of the camera is at the center of the image and there is no skewing effect, the camera calibration matrix solely relies on the focal length f. As suggested in [40], f can be estimated using the inter-frame homographies H_k's and two imaged circular points $\mathbf{c}_j, j = 1, 2$ in the reference frame if the calibration matrix is assumed to be constant ($K_k = K_1$) throughout the video sequence. In this chapter, we simply obtain the focal length f from the metadata comes with the surveillance video.

It is not possible to get the ground plane information $\mathbf{n}^T$ from only one view. Triggs [97] gives an SVD based factorization method to decompose a calibrated homography $\hat{H} = {K_2}^{-1}HK_1$ into the plane normal $\mathbf{n}^T$ and the relative orientation between the two cameras $R(I_{3\times3}|-\mathbf{t})$. In the coordinate system of the first camera ($P_1 = (I_{3\times3}|0)$), let the 3D plane be $\mathbf{n} \times \mathbf{x} = z = 1/\zeta$, where $z = 1/\zeta > 0$ is the inverse distance to the plane. Let the matrix of the second camera be $P_2 = R(I_{3\times3}| - \mathbf{t})$ where $\mathbf{t}$ is the inter camera translation and R the inter camera rotation. Then the homography from image 1 to image 2 is $\hat{H} = R\hat{H}_1$ where $\hat{H}_1 = I_{3\times3} - \zeta\mathbf{t}\mathbf{n}^T$. For a 3D point $\mathbf{x}$ on the plane $\hat{H}\mathbf{x} = R(\mathbf{x} - \zeta\mathbf{t}\mathbf{n}^T) = R(\mathbf{x} - \mathbf{t}) \approx P_2\mathbf{x}$, since $\zeta \mathbf{n}^T \mathbf{x} = 1$ there. Treating $\mathbf{x}$ as a point in image 1 changes only the overall scale factor. Only the product $\zeta \mathbf{t}\mathbf{n}^T$ is recovered, so we normalize to $\|\mathbf{t}\| = \|\mathbf{n}\| = 1$ and use visibility tests to work out the allowable signs. The detailed decomposition of $\hat{H}$ can be found in appendix 1 of [97]. For a distant plane $\zeta \rightarrow 0$ as in an airborne video, the estimated $\mathbf{n}^T$ and $\mathbf{t}$ might be unreliable but R is still accurate. The inaccuracy of $\mathbf{n}^T$ and $\mathbf{t}$ is compensated using multiple image pairs, and the accurate R is used to compute the infinite homography H^∞.

Fusion Scheme

Every other frame, together with the reference frame, gives an estimate to the plane function $\mathbf{n}^T$ in the coordinate system of the reference frame. However, for a distant plane, the estimated $\mathbf{n}^T$ for each pair of frames is not reliable as pointed in [97]. Also, the information from the whole video sequence has not been fully utilized. Therefore, a suitable fusion scheme that can fuse available two-view estimates is needed to achieve a robust estimate of $\mathbf{n}^T$.

In [81], a rank one constraint is applied to factorize a matrix, which stacks the planar homographies between the reference frame and all the other frames, into plane functions and camera motions. This is a good fit to our problem as a fusion scheme. The only information that is needed other than the inter-frame homography H_k is the infinite homography H_k^∞ for each pair of frames. As mentioned before, for a distant plane the estimate to the camera rotation R_k is still accurate. Thus the infinite homography H_k^∞ is computed as $H_k^\infty = K_k R_k {K_1}^{-1}$ [40]. A block matrix W is constructed by stacking all the transformed inter-frame homographies $\hat{H}_k$ as in (5.11). By applying the constraint that W has rank at most 1, W can be factored into the camera center vector $[\bar{t}_k]$ and the ground plane surface normal

$\mathbf{n}^T$ using SVD:

$$W = \begin{pmatrix} \hat{H}_2 \\ \hat{H}_3 \\ \vdots \\ \hat{H}_n \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{t}}_\mathbf{2} \\ \bar{\mathbf{t}}_\mathbf{3} \\ \vdots \\ \bar{\mathbf{t}}_\mathbf{n} \end{pmatrix} \mathbf{n}^T, \tag{5.11}$$

where $\hat{H}_k = \lambda^{-1} H_k^\infty H_k - I_{3\times 3}$. The scale λ for $\hat{H}_k$ is computed from the double eigenvalues of the planar homology $H_k^{\infty -1} H_k$.

Having the robust estimate of $\mathbf{n}^T$, we can use (5.10) to compute H_{new} and then the points on the object in the reference frame are warped to the desired viewpoint. A cubic interpolation is used to get the final synthesis result. With the metadata available, we can simply assume that the camera calibration matrix $K_{new} = K_k$ because mostly the desired image is in a comparable range of the available images, which relaxes the requirement that the focal length f be constant throughout the whole video sequence.

The advantages of the proposed method include (i) avoiding the degeneracy in estimating the perspective projection relation across views. (ii) the desired viewpoint $[R_{new}|\mathbf{t_{new}}]$ is easy to be incorporated in the framework as shown in (5.10). (iii) the rank one constraint fusion scheme can help to improve the ground plane function estimation and view synthesis by using the information from the whole video sequence. (iv) no dense point correspondences are needed for view synthesis. (v) the computation is simple and fast. Refer to [112] for detail.

Two examples of the view synthesis results are shown in Fig. 5.12. In both figures, the center image is the reference frame with the object inside the bounding box, and the surroundings are the synthesized images w.r.t. different viewing directions. As we can observe from the figures, the synthesized images are very

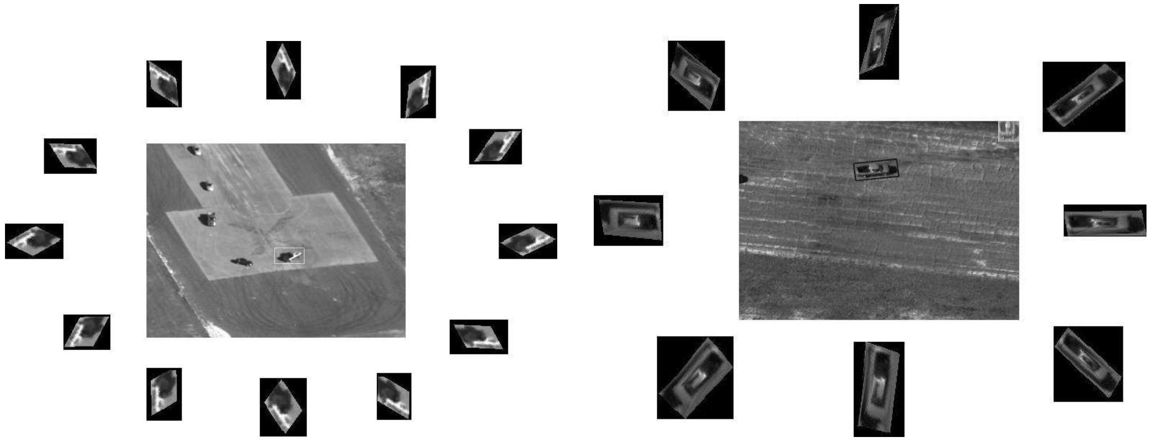
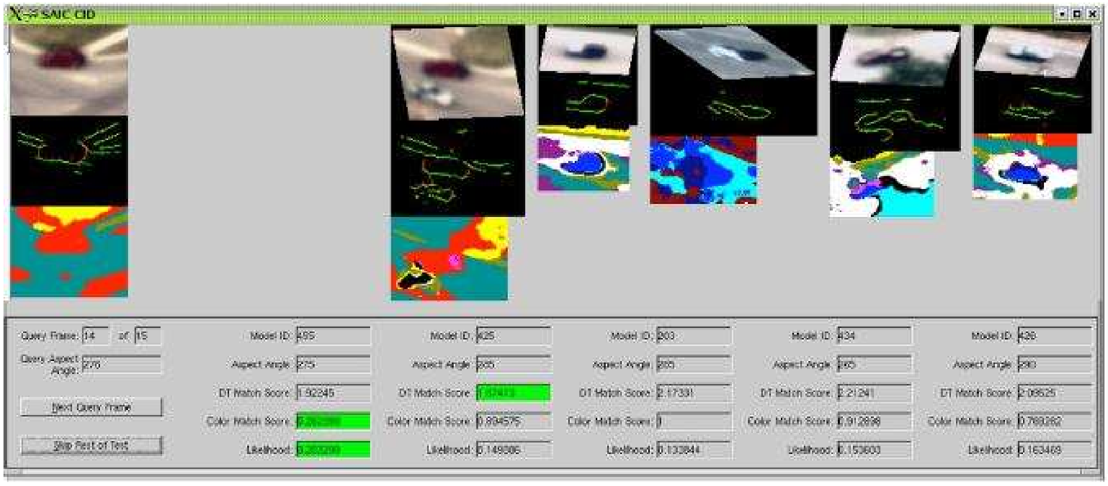


Figure 5.12: View synthesis results. The center image is the reference frame with the object inside the bounding box, and the surroundings are the synthesized images corresponding to different viewing directions.

good in following the changing viewpoints although it is not easy to see the fine details because of the large distance between the camera and the scene.

5.3.3 Experimental Results

The verification system was designed to operate non-interactively, but a graphical display was created for testing and evaluation. Fig. 5.13 shows an example of the graphic output for a typical trial. On the far left is the current query frame; the query target is displayed at the top with the spatial and color models directly below it. To the right of the query are the five learned models it is being compared against. The novel views created by view synthesis are displayed at the top with the spatial and color models directly below. At the bottom of the screen are the spatial and color match scores for the current frame, and the cumulative likelihood for the trial. The winning color match, spatial match and likelihood are highlighted in green.



Query Frame: 14 of 15

Query Aspect Angle: 276

Model ID:	495	425	203	434	429
Aspect Angle:	275	265	265	265	280
DT Match Score:	1.92245	[illegible]	2.17331	2.21241	2.08525
Color Match Score:	[illegible]	0.894575	0	0.912898	0.769262
Likelihood:	[illegible]	0.149806	0.133844	0.153603	0.163469

Figure 5.13: Graphical display for monitoring the operation of the CID module. The Query frame on the left is currently being compared against the five targets on the right.

Several thousands trials were conducted to test the verification performance of the proposed system. Each trial compared one query vehicle against five different vehicle IDs. The system was given five learning messages, one for each of the five known vehicle IDs, followed by a query message. In all cases the query vehicle ID matched one of the five learned vehicle IDs. For each of the five matches, the system returned a likelihood that the query vehicle was the same as the vehicle it was matched against. If the highest likelihood was associated with the correct vehicle ID, then the trial succeeded, otherwise it failed.

Trial parameters were chosen to mimic a real operational scenario. The five vehicle IDs for each trial were randomly chosen from a pool of over 100, and the query vehicle ID was randomly chosen from these five. A number of video sequences, averaging approximately 1 second (30 frames) in length, were selected in advance for each of the 100 vehicles in the pool. One of these sequences was

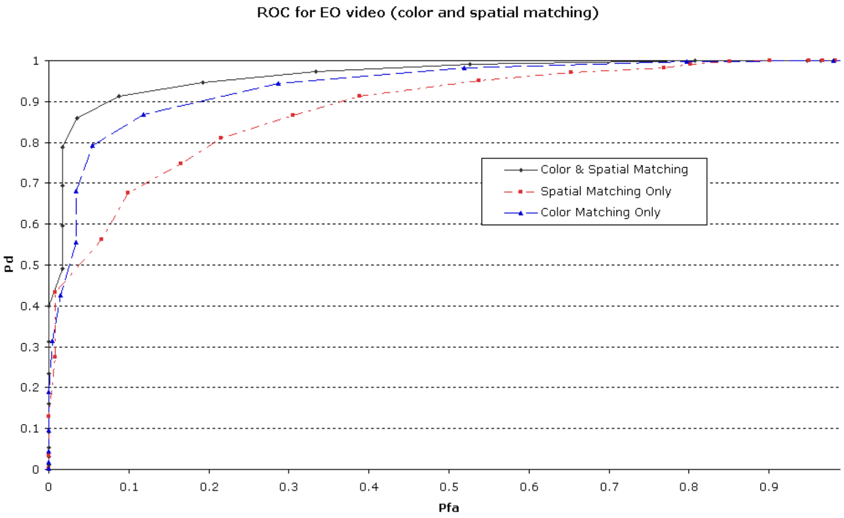


Figure 5.14: The comparison of ROC curves for EO trials with color matching only, spatial matching only and both.

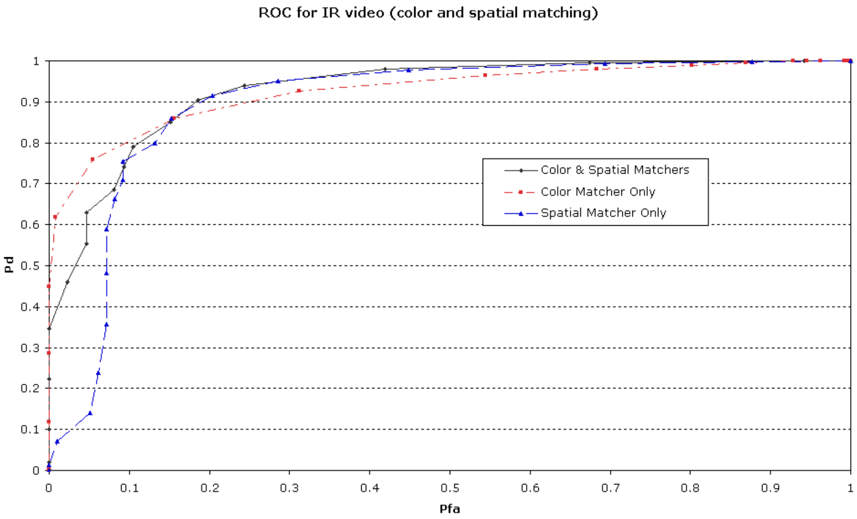


Figure 5.15: The comparison of ROC curves for IR trials with color matching only, spatial matching only and both.

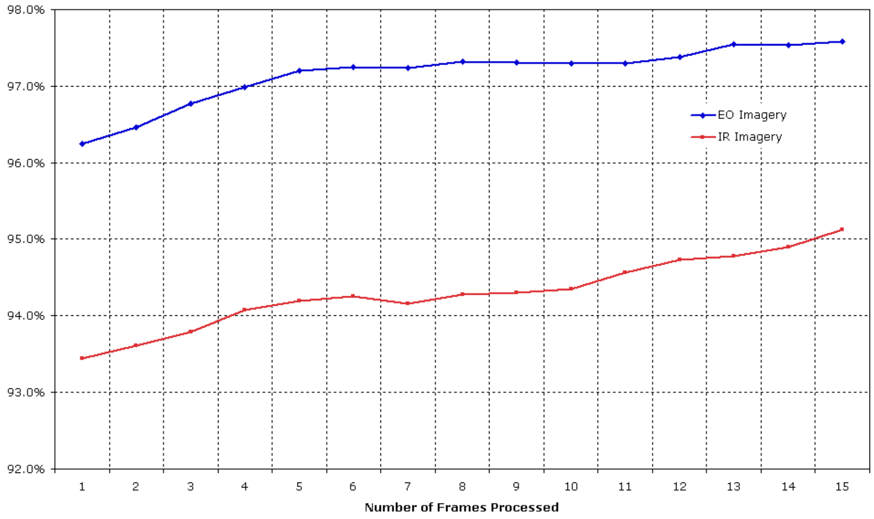


Figure 5.16: The verfication performance scored after the specified number of frames in the query message for both EO and IR imagery, which demonstrates the improvement by using the temporal processing.

randomly selected as the learning message for each of the five known targets. The query sequence was selected from this set to have parameters consistent with a 10-second gap after the learning sequence, and modified to average about half a second (14 frames) in length.

On visible band (EO) imagery, the system correctly matched the query vehicle for 97.5% of 2289 trials. On IR imagery, the system correctly matched 95.0% of 1723 trials. To determine the individual contribution of each matcher, the same set of trials were run under three conditions: color matcher only, spatial matcher only, and both. Receiver Operating Characteristic (ROC) curves for each condition are shown in Fig. 5.14 and Fig. 5.15 for EO and IR imagery respectively. A clear distinction between the matchers is visible in the ROC curves for the EO imagery. For IR imagery, the distinction is not as clear. Initially the color matcher performs much better, most likely due to the typically lower resolution and lack of sharp edges in IR imagery that are needed for the spatial matcher. Additionally, the thermal characteristics, corresponding to color, do not change significantly over the short time between learning and query, thus the color matcher can perform well. However, glints and strong shadows do change over that short time, creating phantom features that confuse the spatial matcher. Accommodating these is an area for further research.

The advantage of temporal processing can be seen in Fig. 5.16. The same EO trials described above were processed, but the results were scored after the specified number of frames in the query message. Performance improves steadily for the first five frames and then levels off. Additional frames provide no significant improvement for EO imagery, but continue to gradually improve the performance for IR imagery.

Processing time for the system was measured on a 2.8GHz Pentium P4 running Red Hat Linux. Average execution time is 3.35 seconds per trial for EO imagery and 1.25 seconds per trial for IR. The longer time for the EO trials is a result of warping three bands (red, green, and blue) to create a novel view instead of the single band for IR imagery. Most of the time per trial was consumed during the query process. Improvements in processing time are possible, making it very likely that real-time processing can be achieved.

5.4 Summary and Future Work

We have described a two view tracking approach which uses the homography relation between two views to handle occlusions. An adaptive appearance model is used in a particle filter to accomplish single view tracking. We showed how to robustly estimate the homography with the previous tracking results and how to infer the correct transformation for the occluded view with the estimated homography and the tracking result for the un-occluded view. Experimental results show that the proposed multiple view tracking method can follow the target when it is partially or fully occluded by an unknown object.

In addition, an end-to-end verification system for moving objects in airborne video has been presented in this section. The object information is collected on the fly from a short real-time learning sequence to avoid the requirement for prior training data. The components of the system have been described in detail, including image normalization, exemplar selection, feature extraction, spatial matching, a homography-based view synthesis method and temporal analysis. The key contributions of novel view synthesis method and the integration of spatial and temporal models have been demonstrated by the experimental results. Very good verification performance is achieved in thousands of trials for both EO and IR sequences using the proposed system.

One problem encountered at the verification stage was that the vehicle was not tightly segmented from the background. This caused the system to match background in addition to the vehicle itself. Moderate to severe illumination changes also caused the performance to decrease. Our future work will integrate automatic object detection and shadow removal modules into the system to make it more robust. We will also conduct some experiments on simultaneous verification and tracking from video to video for airborne sequences.

Chapter 6

Conclusions and Future Research

In this dissertation, we have presented a number of view synthesis algorithms from image and video to improve the object recognition performance for various applications.

We first presented a complete framework combining the active image based visual hull algorithm and a contour based body part segmentation technique for a better synthesis and understanding of the human pose from a limited number of available silhouette images. No 3D body model needs to be explicitly reconstructed. Pose normalized silhouette images are generated using an active virtual camera and an image based visual hull technique, with the silhouette turning function distance being used as the pose similarity measurement. In order to overcome the inability for visual hull technique to reconstruct concave regions, we utilized a contour-based human body part localization algorithm to segment the input silhouette images into convex body parts, and then assembled the separately processed body parts for a better visual hull reconstruction. Furthermore, these two components improve each other for better performance through the correspondence across viewpoints built via the inner distance shape context measurement.

We then examined the most challenging scenarios in face recognition. That is, to identify a subject from a test image that is acquired under different pose and illumination condition from the only one training sample of this subject in the database. Two cases on the lighting condition are considered. When the test face image is taken under a single light source, we presented a pose-normalized face synthesis approach on a pixel-by-pixel basis from a single view by exploiting the bilateral symmetry of the human face. For a more general illumination condition, we extended the spherical harmonics representation, which has proved to be effective in modeling illumination variations for a fixed pose, to encode pose information by utilizing the fact that 2D harmonic basis images at different poses are related by close-form linear transformations. Very efficient face recognition and synthesis algorithms were proposed based on the orthonormality of the linear transformations.

Furthermore, we investigated a robust two view tracking problem for airborne video. The homography relation induced by the dominant plane (ground plane) is used to handle the structure degeneracy caused by the distance from the camera to the object. We showed that when occlusion happens in one view, the inter-frame transformation in the occluded view can be reliably inferred from the homography and the tracking result in the un-occluded view. We also proposed an end-to-end moving object verification system for airborne video, wherein a homography based view synthesis algorithm was used to simultaneously handle the object's changes in aspect angle, depression angle, and resolution. Efficient integration of spatial and temporal model matching assures the robustness of verification step.

6.1 Suggestion for Future Research

Image/video based view synthesis has been an active research topic in computer vision for decades. Despite the recent progress, there are still many more interesting research directions that need to be further investigated.

First, in almost all image based view synthesis techniques except volume carving methods, dense correspondence is needed to warp the image from the available view to the desired view. Finding correspondence has long been a challenging fundamental problem for computer vision community. Various stereo algorithms have been proposed. Most of them used the intensity-invariant assumption and few considered the occlusion due to the viewpoint change. Recently, promising results have been shown by using the 4 planes, 4 transitions stereo matching algorithm described in [25]. The disparity map can be reliably built for a pair of images of the same object taken under the same lighting condition, even with some occlusions. We plan to further study this algorithm to make it work for images taken under different lighting conditions by utilizing the bilateral symmetry between a given image and its mirror image. In addition, we are in the process of extending this stereo algorithm to build the disparity map for a pair of images at arbitrary pose, where the epipolar lines are not necessarily the horizontal scanlines.

Secondly, temporal information has not been fully exploited in video based view synthesis methods. Instead, these methods have been restricted to treating each time instant sequentially and independently. For a moving object, since the motion between the nearby frames is usually small, it is possible to generate the virtual view using the motion information with less effort/time than to generate it independently at each time instant. Therefore, it is beneficial to use motion information to reduce the computation burden for video base view synthesis algorithms. We will take efforts towards efficient view synthesis techniques which fully utilize the spatial/temporal information.

Finally, there are always empirical factors to be considered for each image/video based view synthesis technique, depending on the specific object recognition application it serves. For example, the shadows and severe illumination changes may cause the virtual images generated from homography based view synthesis method do not look like the images taken from the real scene, thus seriously degrading the object verification performance; or the accuracy of the 2D-3D registration procedure may determine the quality of model-based image rendering results. View synthesis techniques offer great help to boost the performance of object recognition applications, and extra care must be paid for the related practical issues when we deal with each specific view synthesis technique.

BIBLIOGRAPHY

[1] 3dfs-100 3 dimensional face space library (2002 3rd version). *University of Freiburg, Germany.*

[2] http://www.cvg.cs.rdg.ac.uk/pets2001/pets2001-dataset.html.

[3] N. Ahuja and J. Veenstra. Generating octrees from object silhouettes in orthographic views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):137–149, Feb. 1989.

[4] E. Arkin, L. Chew, D. Huttenlocher, K. Kedem, and J. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–216, Mar. 1991.

[5] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Transactions on Visualization and Computer Graphics*, 4(4):293–306, 1998.

[6] P. Barral, G. Dorme, and D. Plemenos. Visual understanding of a scene by automatic movement of a camera. *International Conference GraphiCon'99*, Moscow, Russia, Aug. 26 - Sep. 3 1999.

[7] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, Feb. 2003.

[8] P. Beardsley, P. Torr, and A. Zisserman. 3d model acquisition from extended image sequence. *Proc. European Conference on Computer Vision*, Cambridge, UK, Volume 2:683–695, Apr. 1996.

[9] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, Jul. 1997.

[10] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions. *Proc. Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, pages 270–277, Jun. 1996.

[11] S. Belongie J. Malik and J. Puzicha. Shape matching and object recognition using shape context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4) 509–522, Apr. 2002.

[12] B. Beyme. Face recognition under varying pose. *Technical Report 1461, MIT AI Lab*, 1993.

[13] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. *Proc. the Workshop on Motion and Video Computing*, Orlando, FL, pages 169–174, Dec. 2002.

[14] V. Blanz and T. Vetter. Morphable model for the synthesis of 3d faces. *Proc. SIGGRAPH*, Los Angeles, CA, pages 187–194, Aug. 1999.

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

[16] E. Borovikov and L. Davis. 3d shape estimation based on density driven model fitting. *Proc. 1st International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, Padova, Italy, pages 116–125, Jun. 2002.

[17] M. Brand. Shadow puppetry. *Proc. International Conference on Computer Vision*, Corfu, Greece, pages 1237–1244, Sep. 1999.

[18] C. Castillo and D. Jacobs. Using stereo matching for 2-d face recognition across pose. *Proc. Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, Jun. 2007.

[19] P. Chang and J. Krumm. Object recognition with color co-occurrence histograms. *Proc. Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, pages 498–504, Jun. 1999.

[20] T. Chang, S. Gong, and E. Ong. Tracking multiple people under occlusion using multiple cameras. *Proc. British Machine Vision Conference*, Cardiff, UK, pages 566–575, Sep. 2002.

[21] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *Proc. Conference Computer Vision and Pattern Recognition*, Madison, WI, pages 77–84, Jun. 2003.

[22] G. Cheung, S. Baker, and T. Kanade. Visuall hull alignment and refinement across time: A 3d reconsturcion algorithm combing shape-from-silhouette with stereo. *Proc. Conference on Computer Vision and Pattern Recognition*, Madison, WI, pages 375–382, Jun. 2003.

[23] A. Chowdhury and R. Chellappa. Robust estimation of depth and motion using stochastic approximation. *Proc. International Conference on Image Processing*, Thessaloniki, Greece, pages 642–645, Oct. 2001.

[24] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, Jun. 2001.

[25] A. Criminisi, J. Shotton, A. Blake, C. Rother, and P. Torr. Efficient dense stereo with occlusion for new view synthesis by four-state dynamic programming. *International Journal of Computer Vision*, 71(1):89–110, Jan. 2007.

[26] L. Davis, E. Borovikov, R. Cutler, D. Harwood, and T. Hoprasert. Multi-perspective analysis of human action. *Proc. International Workshop on Cooperative Distributed Vision*, Kyoto, Japan, Nov. 1999.

[27] S. Dockstader and A. Tekalp. Multiple camera fusion for multi-object tracking. *Proc. IEEE Workshop on Multi-Object Tracking*, Vancouver, Canada, pages 95–102, Jul. 2001.

[28] R. Dovgard and R. Basri. Statistical symmetric shape from shading for 3d structure recovery of faces. *Proc. European Conference on Computer Vision*, Prague, Czech Republic, pages 99–113, May 2004.

[29] C. Dyer. *Foundations of Image Understanding.* The Kluwer International Series in Engineering and Computer Science, ISBN 0-7923-7457-6, 2001.

[30] J. Erickson, S. Harpeled, and D. Mount. On the least median square problem. *Discrete and Computational Geometry*, pages 593–607, Dec. 2006.

[31] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communication of the ACM*, 24(6):381–395, 1981.

[32] D. Forsyth, S. Ioffe, and J. Haddon. Bayesian structure from motion. *Proc. International Conference on Computer Vision*, pages 660–665, Sep. 1999.

[33] W. Freeman and J. Tenenbaum. Learning bilinear models for two-factor problems in vision. *Proc. Conference Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pages 554–560, Jun. 1997.

[34] A. Geoghiades, P. Belhumeur, and D. Kriegman. Illumination-based image synthesis: Creating novel images of human faces under differing pose and lighting. *Proc. Workshop on Multi-View Modeling and Analysis of Visual Scenes*, Fort Collins, CO, pages 47–54, Jun. 1999.

[35] R. Green. Spherical harmonic lighting: The gritty details. *Game Developers' Conference*, San Jose, CA, Mar. 2003.

[36] D. Guarino, B. Walls, and E. Miles, Confirmatory Identification of Targets in Video, *VIVID Automated Video Processing for Unmanned Aircraft, Edited by T. Strat and L. Hollan*, DARPA, 2005.

[37] Y. Guo, S. Hsu, Y. Shan, H. Sawhney, and R. Kumar. Vehicle fingerprinting for reacquisition and tracking in videos. *Conference Computer Vision and Pattern Recognition*, San Diego, CA, pages 761–768, Jun. 2005.

[38] S. Gupte, O. Masoud, R. Martin, , and N. Papanikolopoulos. Detection and classification of vehicles. *IEEE Transations on Intelligent Transportation Systems*, 3(1):37–47, Mar. 2002.

[39] I. Haritaoglu, D. Harwood, and L. S. Davis. Ghost: A human body part labeling system using silhouettes. *Proc. International Conference on Pattern Recognition*, Brisbane,Australia, pages 77–82, Aug. 1998.

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

[41] D. Hoffman and W. Richards. Salience of visual parts. *Cognition*, 63(1): 29–78, Jan. 1997.

[42] T. Horprasert, Y. Yacoob, and L. Davis. Computing 3-d head orientation from a monocular image sequence. *Proc. International Conference on Automatic Face and Gesture Recognition*, Killington, VT, pages 77–82, 1996.

[43] N. Howe. Silhouette lookup for automatic pose tracking. *IEEE Workshop on Articulated and Nonrigid Motion*, Washington DC, pages 15–22, Jun. 2004.

[44] N. Howe, M. Leventon, and W. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. *Advances in Neural Information Processing Systems*, Volume 12, pages 820–826, 1999.

[45] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):334–352, Mar. 2004.

[46] T. Huang and A. Netravali. Motion and structure from feature correspondences: A review. *Proc. IEEE*, 82(2):252–268, Feb. 1994.

[47] P. Huber. *Robust Statistics.* Wiley, 1981.

[48] T. Inui, Y. Tanabe, and Y. Onodera. *Group Theory and its Applications in Physics.*

[49] M. Irani, P. Anandan, and M. Cohen. Direct recovery of planar-parallax from multiple frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1528–1534, Nov. 2002.

[50] D. Jacobs, P. Belhumeur, and R. Basri. Comparing images under variable illumination. *Proc. Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pages 610–617, Jun. 1998.

[51] A. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, 1989.

[52] T. Jebara, A. Azarbayejani, and A. Pentland. 3-d structure from 2-d motion. *IEEE Signal Processing Magazine*, 16(3): 66–84, May 1999.

[53] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance model for visual tracking. *Proc. Conference on Computer Vision and Pattern Recognition*, Kauai, HI, pages 415–422, Dec. 2001.

[54] B. Johansson. View synthesis and 3d reconstruction of piecewise planar scenes using intersection lines between the planes. *Proc. International Conference on Computer Vision*, Corfu, Greece, pages 54–59, Sep. 1999.

[55] S. Kang. A survey of image-based rendering techniques. *Proc. SPIE 3641*, pages 2–16, 1999.

[56] G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1): 1–25, Jan. 1996.

[57] R. Koch, M. Pollefeys, and V. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. *Proc. European Conference on Computer Vision*, Freiburg, Germany, pages 55–71, Jun. 1998.

[58] D. Kriegman, P. Belhumeur, and A. Georghiades. Shape and enlightenment: Reconstruction and recognition under variable illumination. *International Symposium on Robotics Research*, Snowbird, UT, pages 79–88, Oct. 1999.

[59] K. Kutulakos and C. Dyer. Recovering shape by purposive viewpoint adjustment. *International Journal of Compuater Vision*, 12(2-3): 113–136, Apr. 1994.

[60] K. Kutulakos and S. Seitz. A theory of shape by space carving. *International Journal of Compuater Vision*, 38(3): 199–218, Jun. 2000.

[61] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2): 150–162, Feb. 1994.

[62] H. Ling and D. Jocobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2): 286–299, Feb. 2007.

[63] A. Lipton, H. Fujiyoshi, and R. Patil. Moving target classfication and tracking from real-time video. *IEEE Workshop Applications of Computer Vision*, Princeton, NJ, pages 8–14, Nov. 1998.

[64] J. Liu and R. Chen. Sequential monte carlo for dynamic systems. *Journal of the American Statistical Association*, 93(443): 1031–1041, 1998.

[65] D. Lowe. Object recognition from local scale-invariant features. *Proc. International Conference on Computer Vision*, Corfu, Greece, pages 1150–1157, Sep. 1999.

[66] E. Marchand and N. Courty. Image-based virtual camera motion strategies. *Proc. Graphics Interface Conference*, Montreal, Quebec, pages 69–76, May 2000.

[67] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. *Proc. Eurographics Workshop on Rendering*, Manchester,UK, pages 115–125, Jun. 2001.

[68] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. *Proc. SIGGRAPH*, New Orleans, LA, pages 369–374, Jul. 2000.

[69] L. McMillan. An image-based approach to three-dimensional computer graphics. *Ph. D Dissertation, University of North Carolina*, 1997.

[70] G. Mori and J. Malik. Estimating human body configurations using shape context matching. *Proc. European Conference on Computer Vision*, Copenhagen, Denmark, pages 666–680, May 2002.

[71] W. Niem and M. Steinmetz. Camera viewpoint control for the automatic reconstruction of 3d objects. *Proc. International Conference on Image Processing*, Lausanne, Switzerland, pages 655–658, Sep. 1996.

[72] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *Proc. Conference on Computer Vision and Pattern Recognition*, Seattle, WA, pages 84–91, Jun. 1994.

[73] M. Potmesil. Generating octree models of 3d objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40(1): 1–20, Jan. 1987.

[74] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of IEEE*, 77(2): 257–286, Feb. 1989.

[75] R. Ramamoorthi. Analytic pca construction for theoretical analysis of lighting variability in images of a lambertian object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10): 1322–1333, Oct. 2002.

[76] R. Ramamoorthi and P. Hanrahan. A signal processing framework for reflection. *ACM Transactions on Graphics*, 24(4): 1004–1042, Oct. 2004.

[77] T. Raviv and A. Shashua. The quotient image: Class based re-rendering and recognition with varying illuminations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2): 129–139, Feb. 2001.

[78] D. Roberts and A. Marshall. Viewpoint selection for complete surface coverage of three dimentional objects. *Proc. British Machine Vision Conference*, Southampton, UK, pages 740–750, Sep. 1998.

[79] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. *Proc. European Conference on Computer Vision*, Copenhagen, Denmark, pages 700–714, May 2002.

[80] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of body pose from a single image. *IEEE Workshop on Human Motion*, Austin, TX, pages 19–24, Dec. 2000.

[81] C. Rother, S. Carlsson, and D. Tell. Projective factorization of planes and cameras in multiple views. *Proc. International Conference on Pattern Recognition*, Quebec City, Canada, pages 737–740, Aug. 2002.

[82] T. Ryan. Mstar indexing final report. *Prepared for DARPA and AFRL, Contract F33615-95-C-1642, Wright-Patterson Air Force Base*, 2000.

[83] B. Scassellati, S. Alexopoulos, and M. Flickner. Retrieving images by 2d shape: A comparison of computation methods with human perceptual judements. *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 2–14, 1994.

[84] S. Seitz and C. Dyer. Physically-valid view synthesis by image interpolation. *Proc. Workshop on Representations of Visual Scenes*, Cambridge, MA, pages 18–25, Jun. 1995.

[85] S. Seitz and C. Dyer. View morphing. *Proc. SIGGRAPH*, New Orleans, LA, pages 21–30, Aug. 1996.

[86] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. *2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, Kauai, HI, Dec. 2001.

[87] G. Shakhnarovich, L. Lee, and T. Darrell. Integrated face and gait recognition from multiple views. *Proc. Conference on Computer Vision and Pattern Recognition*, Kauai, HI, pages 439–446, Dec. 2001.

[88] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. *Proc. International Conference on Computer Vision*, Nice, France, Oct. 2003.

[89] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3d human figures using 2d image motion. *Proc. European Conference on Computer Vision*, Dublin, Ireland, pages 702–718, Jun. 2000.

[90] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database of human faces. *Proc. International Conference on Automatic Face and Gesture Recognition*, Washington DC, pages 46–51, May 2002.

[91] T. Sim and T. Kanade. Illuminating the face. *Tech. Report CMU-RI-TR-01-31, Robotics Institute, CMU*, 2001.

[92] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. *Proc. European Conference on Computer Vision*, Prague, Czech Republic, pages 85–98, May, 2004.

[93] S. Soatto and R. Brockett. Optimal structure from motion: Local ambiguities and global estimation. *Proc. Conference on Computer Vison and Pattern Recognition*, pages 282–288, Jun. 1998.

[94] R. Szeliski. Rapid octree construction from image sequences. *Compuater Vision, Graphics and Image Processing: Image Understanding Archive*, 58(1): 23–32, Jul. 1993.

[95] C. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 80(3): 349–363, Dec. 2000.

[96] P. Torr, A. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision*, 32(1): 27–44, Aug. 1999.

[97] W. Triggs. Autocalibration from planar scenes. *Proc. European Conference on Computer Vision*, Freiburg, Germany, pages 89–105, Jun. 1998.

[98] P. Vazquez, M. Feixas, M. Sbert, and W. Hendrich. Viewpoint selection using viewpoint entropy. *Proc. Vision, Modeling, and Visualization*, pages 273–280, 2001.

[99] T. Vettor, M. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. *Technical Report, Artificial Intelligence Laboratory and Center for Biological and Computer Learning, MIT*, 1997.

[100] T. Vettor and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Patter Analysis and Machine Intelligence*, 19(7): 733–742, Jul. 1997.

[101] Y. Wexler. Tensor methods for vision and graphics with applications to dynamic morphing. *Ph. D Dissertation, University of Maryland, College Park*, 2000.

[102] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. *Proc. Conference on Computer Vision and Pattern Recognition*, Madison, WI, pages 785–795, Jun. 2003.

[103] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. *Proc. Conference on Computer Vision and Pattern Recognition*, Washington DC, pages 535–542, Jun. 2004.

[104] Z. Yue and R. Chellappa. Pose-Invariant view synthesis using image-based visual hull. *Proc. 7th Joint Conf. on Info. Sciences*, pages 781–784, Cary, NC, Sep. 2003.

[105] Z. Yue and R. Chellappa. Pose-normalized view synthesis from silhouettes. *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Philadelphia, PA, Mar. 2005.

[106] Z. Yue, L. Zhao and R. Chellappa. View synthesis of articulating humans using visual hull. *Proc. Intl. Conf. on Multimedia and Expo* , Volume 1, pages 489-492, Baltimore, MD, Jul. 2003.

[107] Z. Yue and R. Chellappa. View synthesis for articulated humans from silhouettes. *IEEE Transactions on Multimedia*, under review.

[108] Z. Yue and R. Chellappa. Pose-normalized view synthesis of a symmetric object using a single image. *Proc. 6th Asian Conference on Computer Vision*, Jeju City, South Korea, pages 915–920, Jan. 2004.

[109] Z. Yue, W Zhao and R. Chellappa. Pose-encoded spherical harmonics for robust face recognition using a single image. *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, Beijing, China, Oct. 2005.

[110] Z. Yue, W Zhao and R. Chellappa. Pose-encoded spherical harmonics for robust face recognition using a single image. *EURASIP Journal on Advances in Signal Processing*, to appear, Jan. 2008.

[111] Z. Yue, S Zhou and R. Chellappa. Robust two-camera visual tracking with homography. *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May, 2004.

[112] Z. Yue and R. Chellappa. Synthesis of novel views of moving objects in airborne video. *Proc. British Machine Vision Conference*, Oxford, UK, pages 290–299, Sep. 2005.

[113] Z. Yue, D. Guarino and R. Chellappa. Moving Objects verification in airborne video. *IEEE International Conference on Computer Vision System*, New York, NY, Jan. 2006.

[114] Z. Yue, D Guarino and R. Chellappa. Moving objects verification in airborne video. *IEEE Transactions on Circuits and Systems for Video Technology*, under review.

[115] L. Zhang and D. Samaras. Face recognition under variable lighting using harmonic image examplars. *Proc. Conference on Computer Vision and Pattern Recognition*, Madison, WI, pages 19–25, Jun. 2003.

[116] L. Zhang, S. Wang, and D. Samaras. Face recognition from a single image under arbitrary unknown lighting using spherical harmonics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3): 351–363, Mar. 2006.

[117] Z. Zhang. A flexible new technique for camera calibration. *Technical Report, Microsoft Research*, 1998.

[118] L. Zhao. Dressed human modeling, detection, and parts localization. *Ph. D Dissertation, Carnegie Mellon University*, 2001.

[119] W. Zhao and R. Chellappa. Sfs based view synthesis for robust face recognition. *Proc. International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, pages 285–292, 2000.

[120] W. Zhao and R. Chellappa. Symmetric shape-from-shading using self-ratio image. *International Journal of Computer Vision*, 45(1): 55–75, Oct. 2001.

[121] W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4): 399–458, 2003.

[122] Q. Zheng and R. Chellappa. Estimation of illumination direction, albedo, and shape from shading. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 13(7): 680–702, Jul. 1991.

[123] S. Zhou, R. Chellappa, and D. Jacobs. Characterization of human faces under illumination variations using rank, integrability, and symmetry constraints. *Proc. European Conference on Computer Vision*, Prague, Czech Republic, pages 588–601, May 2004.

[124] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-based modeling in particle filters. *IEEE Transactions on Image Processing*, Vol. 11, pages 1491–1506, Nov. 2004.