ABSTRACT

Title of Thesis: CONTOUR BASED 3D FACE MODELING FROM MONOCULAR VIDEO

Himaanshu Gupta, Master of Science, 2003

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

Constructing 3D models from video is one of the most important problems in computer vision. We propose a novel 3D face modeling approach from monocular video captured by a conventional camera. An algorithm is proposed to estimate the head pose by comparing the edges of video frame, and the contours extracted from a generic face model. A generic 3D face model is assumed to be the initial estimate of the true 3D model. The generic face model is adapted to the actual 3D face model by global and local deformations. An affine model is used for global deformation. The 3D model is locally deformed by computing the optimal perturbations of a sparse set of control points using a stochastic search optimization method. The deformations are integrated over a set of poses in the video sequence, leading to an accurate 3D model.

CONTOUR BASED 3D FACE MODELING FROM MONOCULAR VIDEO

by

Himaanshu Gupta

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2003

Advisory Committee:

Professor Rama Chellappa, Chairman/Advisor
Professor Harhalabos Papadopoulos
Professor Min Wu

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Dr. Rama Chellappa. He has always been a constant source of inspiration. I am also grateful to Dr. Babis Papadopoulos and Dr. Min Wu for being on my defense committee. I would like to thank Dr. Amit Roy Chowdhury for the time spent discussing different technical issues that arose in the course of this work. I would also like to thank my officemates, Mr. Aravind Sunderesan and Mr. Amit Agrawal, for many fruitful discussions I have had with them over the last couple of years. I would like to take this opportunity to thank my roommates and friends who made my stay at Maryland very enjoyable. Last, but not the least, I would like to express my gratitude and indebtedness to my parents and sister for their love and support. Their inspiration and hope brought me here in the first place.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Constructing 3D models from video is one of the most important problems in Computer Vision. One particularly interesting application of 3D reconstruction from 2D images is in the area of modeling a human face from video. Successful solution of this problem has applications in multimedia, computer graphics, and face recognition. In multimedia, 3D face models can be used in video conferencing application for efficient transmission. In computer graphics application, 3D face models form the basic building block upon which facial movements and expressions can be added. The problem of 3D face reconstruction from video also has immense potential in the field of face recognition. Many researchers have worked on the face recognition problem, and come up with different algorithms [1]. While 2D view-based and other appearance-based methods perform reasonably well in constrained environments, the performance of these methods is far from satisfactory when face images are acquired in an uncontrolled environment, such as in surveillance video clips. The 2D images of 3D objects can change dramatically due to lighting and viewing conditions. The illumination

and pose variation are two of the most challenging problems encountered by most face recognition systems. Performance degradation due to the illumination problem in face recognition is clearly documented in the FERET test [2], the largest independent test of face recognition algorithms. The changes induced by illumination can be larger than the differences between individuals, causing systems based on comparing images to misclassify the identity of the input face image. Several algorithms have been proposed to handle this problem, including the illumination cone [3], and shape from shading [4], to name a few. Sensitivity to variations in pose is another challenging problem in face recognition. Various approaches have also been proposed to solve this difficult problem, including the linear class based method [5], graph matching based method [6], and the view-based eigenface approach [7]. Cases when illumination and pose variations are simultaneously present are even more difficult. By constructing a 3D model, the problems arising due to pose, illumination, and expression variations can be taken care of, and we can build a robust face recognition system that will work satisfactorily in uncontrolled environments.

Most current commercial systems address special cases of the 3D face reconstruction problem that are well-constrained by additional information. For example, the additional information could be in the form of depth estimates available from multiple cameras, projecting laser or other patterns on the face, decorating the face with special textures to make inter-frame correspondences simple, or using structured light to reveal the contours of the face. All of these methods require various combinations of high quality, high resolution sources, calibrated cameras, special lighting, and careful posing. All these constraints and the hardware

needed reduce the operational flexility of the system. We propose a method capable of working in an unconstrained environment, without any special illumination or sensors. Our system reconstructs the 3D face model from monocular video captured by a conventional camera.

1.2 Previous Work

The most common approach to solve the problem of 3D reconstruction from monocular video is Structure from Motion (SfM). Numerous SfM algorithms exist which can reconstruct a 3D scene from two or more images. Broida and Chellappa investigated the use of Extended Kalman Filter [8] for estimating motion and structure from a sequence of monocular images. Azarbayejani and Pentland extended this work to include the estimation of the focal length of the camera, along with motion and structure [9]. A method for recovering non-rigid 3D shapes as a linear combination of a set of basis shapes was proposed in [10]. Tomasi and Kanade developed an algorithm for shape and motion estimation under orthographic projection using the factorization theorem [11]. Poelman and Kanade [12] further extended this approach to apply the factorization theorem under paraperspective projection. In [13], the author proposed a method for self-calibration and metric 3D reconstruction in the presence of varying internal camera parameters. In [14] [15], the authors proposed a bundle-adjustment approach to solve the problem. The basic idea behind most SfM algorithms is to recover 3D points on a rigid object from 2D correspondences of points across images. Finding accurate inter-frame correspondences is a very challenging and difficult problem, and is a major reason for the poor performance of most SfM algorithms. Roy Chowdhury and Chellappa [16] proposed an algorithm for 3D

face reconstruction from monocular video sequence using uncertainty analysis. They incorporate the generic 3D face model after obtaining the 3D structure from a SfM algorithm. Romdhani, Blanz and Vetter [17] came up with an impressive appearance-based approach where they show that it is possible to recover the shape and texture parameters of a 3D morphable model from a single image. They use a sophisticated statistical head model which has been learned from a large database of human heads.

Shape from Contours [18] is another promising approach for 3D reconstruction. One of the strongest cues for the 3D information contained in a 2D image is the outline of an object in the image. The occluding contour (extreme boundary) in a 2D image directly reflects the 3D shape. Shape from Silhouette techniques have been used to reconstruct 3D shapes from multiple silhouette images of an object without previous knowledge of the object to be reconstructed [19]. The reconstructed 3D shape is called the visual hull, which is an approximation of the object consistent with the object's silhouettes. The accuracy of the visual hull depends on the number and location of the cameras used to generate the input silhouettes. However, it is impossible to recover the concavities in the shape of the object from the silhouettes or contours. But if we assume prior knowledge of the object being reconstructed, and use a generic model, contour information can be exploited as an important constraint for the exact shape of the object. Since human faces do not vary drastically from person to person, using the generic model as the prior estimate of the exact 3D face model is reasonable. Moghaddam et al [20] have developed a system to recover 3D shape of a human face from a sequence of silhouette images. They use a downhill simplex method to estimate the model parameters, which are the coefficients of the

eigenhead basis functions.

Estimating the pose of the face in the video is an integral part of the 3D face reconstruction algorithm. Researchers in computer vision have proposed several approaches for head pose estimation from a still image or an image sequence. In [21] the head orientation is modelled as a linear combination of disparities between facial regions and several face models. Gee and Cipolla [22] estimate head orientation based on the individual face geometry assuming a weak perspective imaging model. Their method depends on the distance between the eyes and mouth which often changes during facial expressions. Horprasert et al [23] recover the head orientation based on the structure configuration of five facial feature points and projective invariance relationship among these feature points. Their method has the constraint that both eyes and the nose should be visible, thus avoiding near-profile poses. The difficulty to locate and track these feature points accurately and consistently across multiple frames is the major concern for these pose estimation algorithms.

1.3 Thesis Contributions

In this thesis, we propose a novel 3D face modeling approach from a monocular video sequence, whereby the model is recursively updated to fit the outer contour of each image in the video sequence. An algorithm is proposed to estimate the head pose by comparing the edges of video frame, and the contours extracted from a generic face model. The pose estimate is continually refined in conjunction with the 3D face reconstruction algorithm to give an accurate pose estimate. A generic 3D face model is assumed to be the initial estimate of the true 3D model. The generic face model is adapted to the actual 3D face model by global and

local deformations. An affine model is used for global deformation. The 3D model is locally deformed by computing the optimal perturbations of a sparse set of control points using a stochastic search optimization method, and an edge disparity based cost function. The deformations are integrated over a set of poses in the video sequence, leading to an accurate 3D model.

1.4 Organization of the Thesis

The thesis is organized as follows. In Chapter 2, we describe our contour-based pose estimation algorithm. The algorithm for 3D reconstruction of the face is described in Chapter 3. Some of the 3D face modeling results obtained from real test video sequences are presented in Chapter 4. Conclusions and scope of future work are presented in Chapter 5.

Chapter 2

Pose Estimation

A contour-based pose estimation method is proposed to estimate the head pose for a video frame. To estimate the pose for the first frame without any prior knowledge about the 3D structure of the face, we use a generic 3D face model. Human shape variability is highly limited by both genetic and environmental constraints, and is characterized by a high degree of symmetry and approximate invariance of body lengths and ratios. Since the facial anthropometric measurements of the generic face are close to average, unless the person whose head pose is being estimated has a hugely deformed (far from average) face, the algorithm can estimate the pose robustly. The problem of pose estimation along the azimuth angle (Figure 2.3) is most commonly encountered in video sequences, and we limit ourselves to this case only in our work.

2.1 Generic Face Model

This section describes the generic 3D face model used in our algorithm for pose estimation. We use the generic face model proposed by Vetter et al [17]. The database consists of Laser Scans (Cyberware) of 200 heads of young adults (100

male and 100 female). Each laser scanned face is represented by approximately 70,000 vertices and the same number of color values. The geometry of a face can be represented by a shape vector $\mathbf{S} = (X_1, Y_1, Z_1, X_2, \ldots, Y_n, Z_n)^T \in \Re^{3n}$, that contains the X, Y, Z coordinates of its n vertices. The texture of the face can be represented by a texture vector $\mathbf{T} = (R_1, G_1, B_1, R_2, \ldots, G_n, B_n)^T \in \Re^{3n}$, that contains the R, G, B color values of the n corresponding vertices. The average shape ($\bar{\mathbf{S}}$) and texture ($\bar{\mathbf{T}}$) vectors are given by,

$$\bar{\mathbf{S}} = \frac{1}{m}\sum_{i=1}^{m}\bar{\mathbf{S}}_i \qquad \bar{\mathbf{T}} = \frac{1}{m}\sum_{i=1}^{m}\bar{\mathbf{T}}_i \tag{2.1}$$

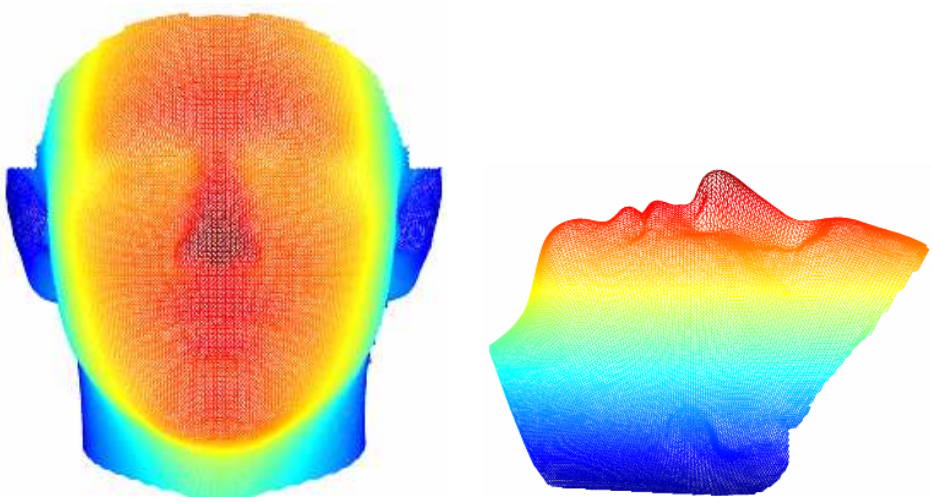
where m is the total number of individuals in the database.

The average 3D model is shown in Figure 2.1, and the average texture is shown in Figure 2.2(a). The average 3D model with the average texture pasted onto it is shown in Figure 2.2(b). We did not perform any of the data collection, pre-processing, etc. ourselves to extract the average shape and average texture. We got the average shape and average texture data from Vetter et al [17].

2.2 Contour-based Pose Estimation

2.2.1 Data Preprocessing

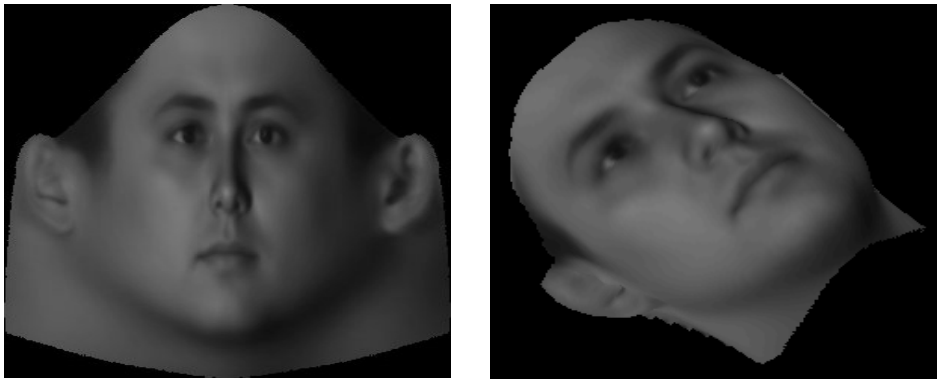
The frames extracted from the video are sub-sampled such that successive frames have a distinct pose variation. This sub-sampling step makes our 3D face reconstruction algorithm much faster without any appreciable degradation in reconstruction quality. We assume that there is no motion in the background. We use a simple image-difference method to detect the background pixels in the video. All edges in the background are removed to make sure they do not adversely affect the pose estimation algorithm. Since we assume that the head motion is



(a) (b)

Figure 2.1: Generic face mesh: (a): Frontal view (b): Profile view

only along the azimuth angle, the top and bottom pixels of the face region are marked out in the first frame, and are assumed to be constant across all video frames. All the edge maps extracted from the average 3D model are resized according to this scale. Also, only the edges in the region between the marked out top and bottom pixels of the face region are retained, thus removing the hair and shoulder edges. For our pose estimation algorithm, we need to know the coordinates of the nose tip in the image, since we pivot the edge maps from the 2D projections of 3D model about the nose tip in the edge map of the video frame. We use the Kanade-Lucas tracker [24] to automatically track the nose tip across multiple frames. Since, the area around the nose tip generally has smooth texture without any distinguishing features to identify it, the automatic tracking is not accurate in some frames. Currently we manually refine the coordinates of the nose tip in case of inaccurate tracking. The nose tip can also be accurately



(a) (b)

Figure 2.2: (a): Average texture map (b): Generic face mesh with average texture mapped onto it.

detected in an automatic manner by using a marker on the nose while collecting the video.

2.2.2 3D Model Edge Extraction

The texture mapped average face is rotated along the azimuth angle, and edges are extracted using the Canny edge detector [25]. The projection of the average 3D model also has edges which result from the boundaries of the 3D model. These edges are not the result of natural contours of the human face, and therefore need to be removed. To differentiate between edges from natural contours and edges resulting from 3D model boundaries, we have a modified average texture, the outer part of which has been manually marked out in white (Figure 2.4(a)). This modified average texture is mapped onto the 3D model from which the edges have been extracted (Figure 2.4(b)). To remove the unwanted boundary edges, we rotate this modified average texture mapped model by the same angle as

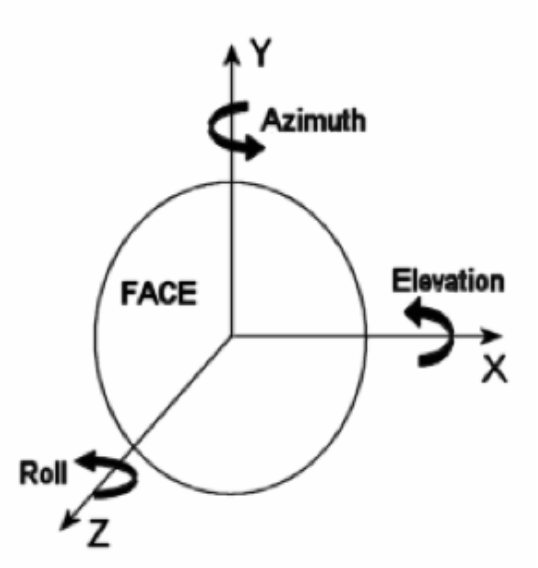


Figure 2.3: Pose variation of face along X, Y, and Z axes

the 3D face model from which edges are extracted, and remove the edges that lie on the white pixels. This procedure makes sure that all the unwanted edges induced due to the boundaries of the 3D model are removed. A few examples of the actual edges extracted from the 3D generic model, and the true face edges are shown in Figure 2.5. In this manner, edge maps are computed for 3D model rotation along the azimuth angle from -90° to $+90^\circ$ in increments of 5°. These edge maps need to be computed only once, and are stored offline in an image array to make the procedure fast.

2.2.3 Pose Estimation Algorithm

To estimate the head pose in a given video frame, we extract the edges of the image using the Canny edge detector. Figure 2.6(a) shows a video frame and

(a) (b)

Figure 2.4: (a): Average texture map with outer edges marked in white (b): Generic face mesh with modified average texture mapped onto it.

Figure 2.6(b) shows the extracted edges after preprocessing as described in Section 2.2.1. Each of the scaled 3D model edge maps (varying from $-90°$ to $+90°$ along the azimuth angle, in increments of $5°$) is compared to this edge map to determine which pose results in the best overlap of the edge maps. To compute the disparity between these edge maps, the Euclidean Distance Transform (DT) of the current video frame edge map is computed. For each pixel in the binary edge map, the distance transform assigns a number that is the distance between that pixel and the nearest nonzero pixel of the edge map (Figure 2.6(c)). Each of the 3D model edge maps are pivoted about the nose tip in the video frame, and the average distance transform value at the nonzero pixels of the binary 3D model edge map (EM) is computed (Figure 2.6(d)). The cost function, F, which measures the disparity between the 3D model edge map and the edges of the current video frame is of the form:

$$F = \frac{\sum_{(i,j)\in A_{EM}} DT(i,j)}{N} \tag{2.2}$$

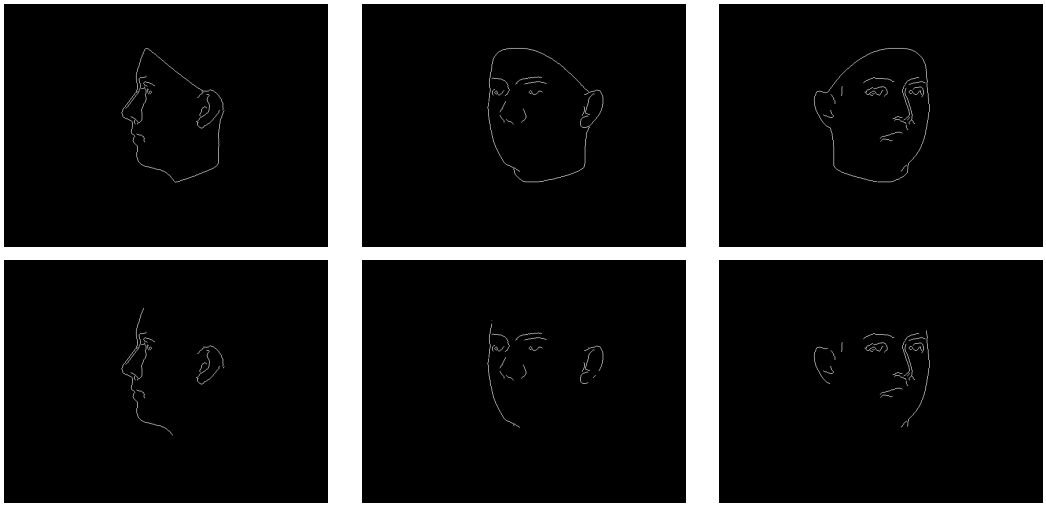


Figure 2.5: Images in the top row are the actual edges extracted from the model. Images in the bottom row show the true edges induced by natural face contours

where $A_{EM} \triangleq \{(i,j) : EM(i,j) = 1\}$ and N is the size of set A_{EM} (total number of nonzero pixels in the 3D model edge map EM).

The pose for which the corresponding 3D model edge map (EM) results in the lowest value of cost function, F, is the estimated head pose for the current video frame. Figure 2.6(e) shows the estimated head pose of the person in 2.6(a). The plot of the values of cost function for all candidate poses is shown in Figure 2.6(f).

2.3 Pose Estimation Results

The algorithm described above was used to estimate the head pose for a large number of individuals, and the results show that it can estimate the rough head pose robustly. Figure 2.7 and Figure 2.8 show the head pose estimation results for a few video frames of two subjects.

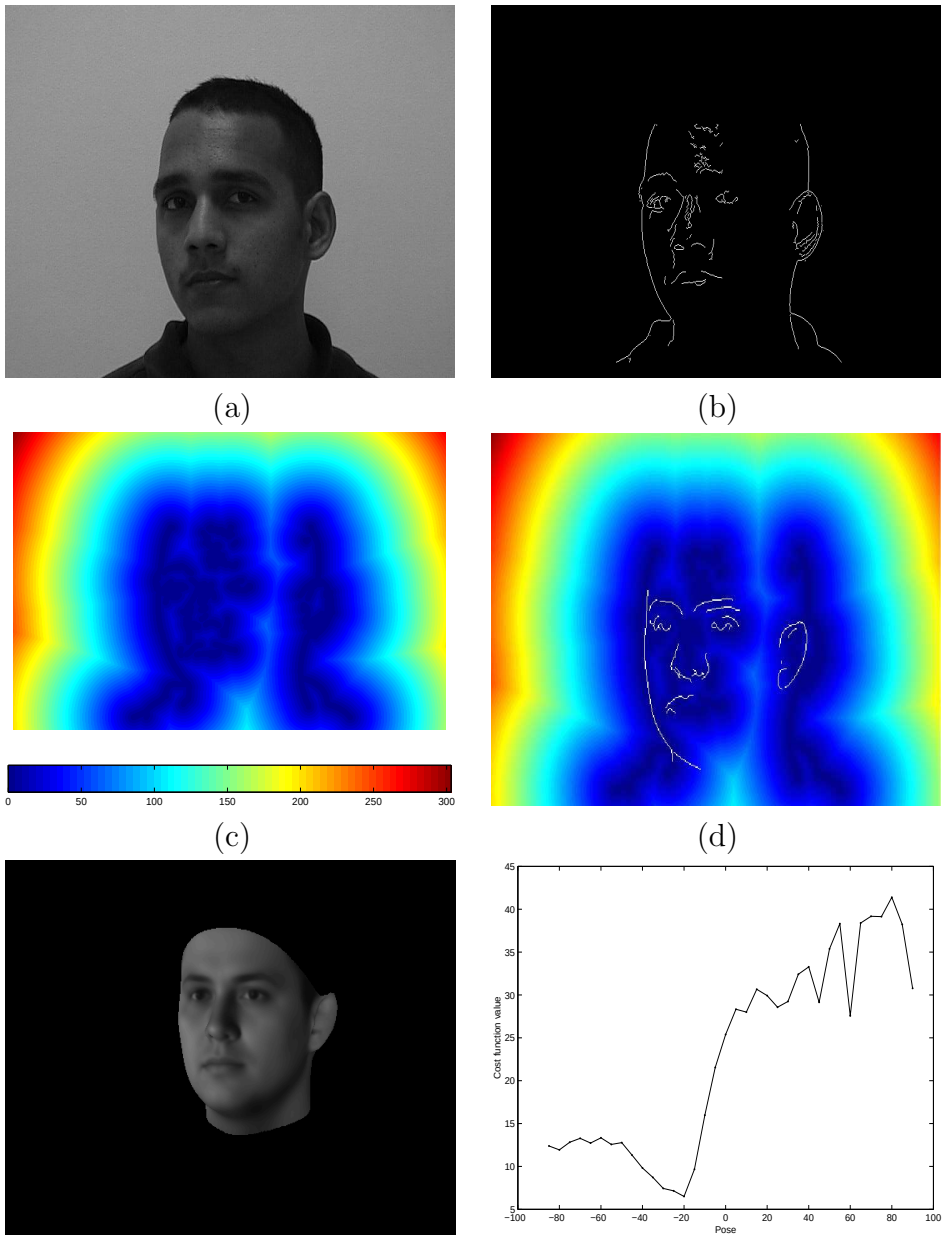


Figure 2.6: (a): Video Frame (b): Extracted edges after preprocessing (c): Distance Transform (DT) of edge map shown in (b). (d): The cost function value for a candidate pose is the average DT value at the white pixels, (e): Estimated head pose ($-20°$) for (a), (f): Plot of cost function values for all candidate poses

Figure 2.7: Left Column: Video Frames. Right Column: Generic model displayed at the estimated head pose

Figure 2.8: Left Column: Video Frames. Right Column: Generic model displayed at the estimated head pose

Chapter 3

3D Model Reconstruction

In this chapter, we propose a contour-based algorithm for 3D face reconstruction from a monocular video sequence. A generic 3D face model is assumed to be the initial estimate of the true 3D model. To adapt this generic face model to the actual 3D face model of the person, we deform the generic model globally and locally such that its 2D projection conforms to the face images over a set of poses in the video sequence.

3.1 Registration and Global Deformation

Once the pose is estimated using the method described in Chapter 2, the next step is to perform global deformation and register the 3D model to the 2D image. We use a scaled orthographic projection model, and an affine deformation model for the global deformation of generic face model. To determine a unique solution for the affine parameters, we need to know the position of atleast 3 non-collinear feature points. We use the coordinates of 4 feature points (left eye, right eye, nose tip, and mouth center), which seem to give a good solution. Since these feature point locations need to be known for just the first frame, currently we

mark these manually. The 3D coordinates of the corresponding feature points on the face mesh are known beforehand. Also, we know the rough head pose for the first video frame using the method described in Chapter 2.

Let $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ be the vector denoting the angular motion. The orientation of the X, Y, and Z axes is the same as shown in Figure 2.3. Since we assume that there is motion only along the azimuth angle, in our case ω_x, and ω_z is zero. ω_y is the pose along the azimuth angle determined by our contour-based pose estimation algorithm. The rotation matrix ($\mathbf{R}$) corresponding to this angular motion vector ($\boldsymbol{\omega}$) can be calculated by

$$\mathbf{R} = e^{[\boldsymbol{\omega}_\times]} \tag{3.1}$$

where,

$$[\boldsymbol{\omega}_\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

The face mesh is globally deformed using the following affine model. The affine model appropriately stretches/shrinks the 3D model along the the X, and Y axes and also takes into account the shearing in the X-Y plane. Considering the basic symmetry of human faces, the affine parameters contributing to shearing (a_{12}, a_{21}) are very small, and can often be neglected. Since we use an orthographic projection model, we can not have an independent affine deformation parameter for the Z-coordinate. For example, in a frontal pose, the Z-coordinate will have no influence over the projection of the 3D feature points, and hence the estimated affine deformation parameter could be totally arbitrary.

$$\begin{bmatrix} X_{\text{gbdef}} \\ Y_{\text{gbdef}} \\ Z_{\text{gbdef}} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & \frac{a_{11}+a_{22}}{2} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix} \tag{3.2}$$

where the subscript "gbdef" denotes global deformation. The aim is to minimize the reprojection error of the rotated deformed 3D feature points, and their corresponding 2D locations in the current frame. The rotated deformed points can be computed by,

$$\begin{bmatrix} X^{\text{rot}}_{\text{gbdef}} \\ Y^{\text{rot}}_{\text{gbdef}} \\ Z^{\text{rot}}_{\text{gbdef}} \end{bmatrix} = \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} X_{\text{gbdef}} \\ Y_{\text{gbdef}} \\ Z_{\text{gbdef}} \end{bmatrix} \tag{3.3}$$

In our scaled orthographic projection model, we choose the scale parameter to be equal to 1. The 2D projection (x, y) of the 3D feature points (X, Y, Z) can be computed by,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{bmatrix} \begin{bmatrix} a_{11}X + a_{12}Y + b_1 \\ a_{21}X + a_{22}Y + b_2 \\ (\frac{a_{11}+a_{22}}{2})Z \end{bmatrix} \tag{3.4}$$

To determine the affine deformation parameters $\mathbf{P} = [a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2]^T$, we construct a linear system of equations and solve for these parameters using a Least-Squares (LS) method. Equation (3.4) can be rewritten as

$$\underbrace{\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} (r_{11}X_1 + \frac{r_{13}Z_1}{2}) & r_{11}Y_1 & r_{12}X_1 & (r_{12}Y_1 + \frac{r_{13}Z_1}{2}) & r_{11} & r_{12} \\ (r_{21}X_1 + \frac{r_{23}Z_1}{2}) & r_{21}Y_1 & r_{22}X_1 & (r_{22}Y_1 + \frac{r_{23}Z_1}{2}) & r_{21} & r_{22} \\ (r_{11}X_2 + \frac{r_{13}Z_2}{2}) & r_{11}Y_2 & r_{12}X_2 & (r_{12}Y_2 + \frac{r_{13}Z_2}{2}) & r_{11} & r_{12} \\ (r_{21}X_2 + \frac{r_{23}Z_2}{2}) & r_{21}Y_2 & r_{22}X_2 & (r_{22}Y_2 + \frac{r_{23}Z_2}{2}) & r_{21} & r_{22} \\ (r_{11}X_3 + \frac{r_{13}Z_3}{2}) & r_{11}Y_3 & r_{12}X_3 & (r_{12}Y_3 + \frac{r_{13}Z_3}{2}) & r_{11} & r_{12} \\ (r_{21}X_3 + \frac{r_{23}Z_3}{2}) & r_{21}Y_3 & r_{22}X_3 & (r_{22}Y_3 + \frac{r_{23}Z_3}{2}) & r_{21} & r_{22} \\ (r_{11}X_4 + \frac{r_{13}Z_4}{2}) & r_{11}Y_4 & r_{12}X_4 & (r_{12}Y_4 + \frac{r_{13}Z_4}{2}) & r_{11} & r_{12} \\ (r_{21}X_5 + \frac{r_{23}Z_4}{2}) & r_{21}Y_4 & r_{22}X_4 & (r_{22}Y_4 + \frac{r_{23}Z_4}{2}) & r_{21} & r_{22} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ b_1 \\ b_2 \end{bmatrix}}_{\mathbf{P}} \tag{3.5}$$

$$\mathbf{P} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A} \tag{3.6}$$

where, X_i, Y_i, Z_i, for i=1,...4 are the 3D feature point coordinates (left eye, right eye, nose tip, mouth center), and x_i, y_i are the corresponding 2D feature point coordinates in the image.

Once, these affine deformation parameters are obtained by the LS method, the generic mesh is globally deformed according to these parameters. Figure 3.1 shows the actual feature point locations, and the 2D projections of corresponding 3D feature points in the globally deformed mesh.

3.2 Local Deformation

The global deformation of the 3D face mesh using the affine model ensures that the face mesh matches the approximate shape of the face. To adapt the 3D model to a particular individual more accurately from the video sequence, we

Figure 3.1: Green crosses are the actual feature locations, and the red crosses are the 2D projections of corresponding 3D feature points in the globally deformed mesh.

introduce local deformations in this globally deformed mesh. The algorithm for local deformation of the face model is described in this section.

3.2.1 Control Points

The globally deformed dense face mesh is sampled at a small number of points to obtain a sparse mesh (Figure 3.2). Each of the vertices of this sparse mesh is a control point in the optimization procedure. The idea behind sub-sampling the dense mesh is to reduce the number of free parameters in the optimization algorithm. A large number of free parameters greatly slows down the algorithm without any appreciable enhancement in the quality of 3D reconstruction.

Each of the control points is imparted a random perturbation in the X, Y, and Z direction. The perturbations for all vertices of the dense face mesh are computed from the perturbations of these control points using a triangle-based

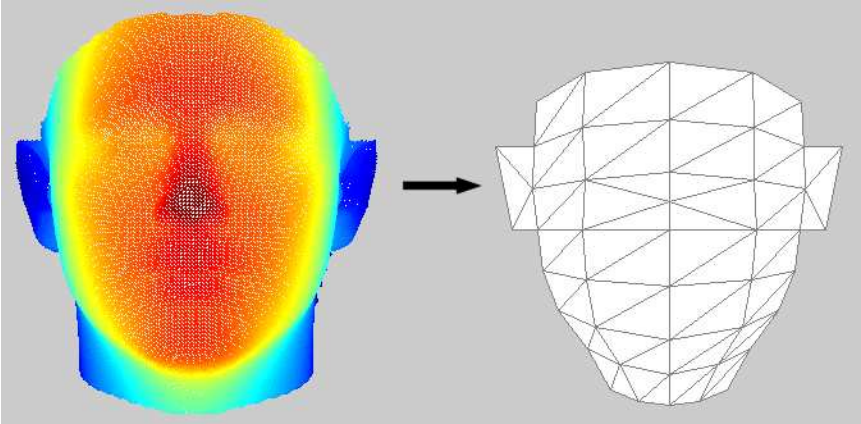


Figure 3.2: Mesh of control points used for local deformation

linear interpolation. These random local perturbations are applied to each of the vertices of the face mesh, and the contours obtained from this locally-deformed face mesh are compared with the edges from the current video frame. The optimum local perturbations of the control points that result in the minimum disparity between the 3D model contours and the edges of the video frame are determined using a stochastic search method described in the next section.

3.2.2 Direct Random Search

Direct Random Search methods [26] are based on exploring the domain D in a random manner to find a point that minimizes the cost function L. They are "direct" in the sense that the algorithms use minimal information about the cost function $L = L(\boldsymbol{\theta})$. The minimal information is essentially only input-output data of the form input $= \boldsymbol{\theta}$, output $= L(\boldsymbol{\theta})$. This contrasts with other algorithms that impose requirements such as: L is differentiable and the gradient

is possibly computable, or the initial condition $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$ is close to the solution $\boldsymbol{\theta}^*$. Direct random search methods have the following advantages relative to most other search methods:

1. **Ease of programming.** Direct Random Search methods are very easy to code and thereby significantly reduce the human cost of an optimization process.

2. **Use of function measurements.** The reliance on function measurements alone can significantly reduce the incentive that might otherwise be present to pick a loss function largely for analytical convenience.

3. **Generality.** The algorithms can apply to virtually any function. The user simply needs to specify the nature of the sampling distribution to allow an adequate search in D. Thus if $\boldsymbol{\theta}$ is continuous-valued, the sampling distribution should be continuous, likewise, a discrete-valued $\boldsymbol{\theta}$ calls for a discrete sampling distribution with non-zero probability of hitting the candidate points.

4. **Ease of Handling constraints.** Direct random search has another desirable feature that it is relatively easy to perform constrained optimization by restricting the values of random samples of $\boldsymbol{\theta}$ that are accepted. This can be applied directly when the constraint on $\boldsymbol{\theta}$ can be explicitly specified as the domain D, and the sampling distribution is restricted to be over D.

5. **Theoretical foundation.** Direct Random Search has supporting theory to guarantee convergence, unlike many other optimization algorithms.

In the direct random search method, we repeatedly sample over D such that the current sampling for $\boldsymbol{\theta}$ does not take into account the previous samples. The

domain D is a hypercube (a p-fold Cartesian product of intervals on the real line, where p is the problem dimension), and we use uniform samples.

Algorithm: Global Random Search

Step 0 (initialization): Generate an initial value of $\boldsymbol{\theta}$, say $\hat{\boldsymbol{\theta}}_0 \in D$, according to uniform probability distribution on the domain D. Calculate $L(\hat{\boldsymbol{\theta}}_0)$. Set k=0.

Step 1: Generate a new independent value of $\boldsymbol{\theta} \in D$, say $\boldsymbol{\theta}_{new}(k+1)$, according to the uniform probability distribution. If $L(\boldsymbol{\theta}_{new}(k+1)) < L(\hat{\boldsymbol{\theta}}_k)$, set $\hat{\boldsymbol{\theta}}_{k+1} = \boldsymbol{\theta}_{new}(k+1)$. Else take $\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k$.

Step 2: Stop if the maximum number of L evaluations has been reached; else return to Step 1 with $k = k+1$.

Convergence proof [26] below shows that the two-step (after initialization) approach described above will converge almost surely ($a.s$) to the solution $\boldsymbol{\theta}^*$ under reasonable conditions. Informally, the proof of convergence says that if values of L at or near $L(\boldsymbol{\theta}^*)$ can be reached with non-zero probability based on the chosen sampling probability (uniform in our case) for generating $\boldsymbol{\theta}_{new}(k)$ at each k, then $\hat{\boldsymbol{\theta}}_k$ will converge to $\boldsymbol{\theta}^*$ as $k \to \infty$.

Proof of Convergence

Suppose that $\boldsymbol{\theta}^*$ is the unique minimizer of L on the domain D and that $L(\boldsymbol{\theta}^*) > -\infty$. Suppose further that for any $\eta > 0$ and $\forall\ k$, there exists a

$\delta(\eta) > 0$ such that

$$P(\boldsymbol{\theta}_{new}(k) : L(\boldsymbol{\theta}_{new}(k)) < L(\boldsymbol{\theta}^*) + \eta) \geq \delta(\eta). \tag{3.7}$$

Then, for Global Random Search algorithm , $\hat{\boldsymbol{\theta}}_k \to \boldsymbol{\theta}^*$ $a.s$ as $k \to \infty$.

Proof. Since $L(\boldsymbol{\theta}^*) > -\infty$ and $L(\hat{\boldsymbol{\theta}}_k)$ is monotonically decreasing, $\lim_{k\to\infty} L(\hat{\boldsymbol{\theta}}_k)$ exists for each underlying sample point, say $\boldsymbol{\omega}$. Let $S_\eta = \{\theta : L(\boldsymbol{\theta}) < L(\boldsymbol{\theta}^*) + \eta\}$. By condition (3.7), $P(\boldsymbol{\theta}_{new}(k) \in S_\eta) > \delta(\eta)$ for any $\eta > 0$. Hence, by independence of the sampling distribution, $P(\boldsymbol{\theta}_{new}(k) \notin S_\eta \ \forall \ k) \leq (1 - \delta(\eta))^k \to 0$ as $k \to \infty$. So, $\lim_{k\to\infty} L(\hat{\boldsymbol{\theta}}_k) < L(\boldsymbol{\theta}^*) + \eta$ in probability. Since $\eta > 0$ is arbitrarily small, $L(\hat{\boldsymbol{\theta}}_k) \to L(\boldsymbol{\theta}^*)$ in probability as $k \to \infty$. Then by a standard result in probability theory ([27], pg. 314), this implies that $L(\hat{\boldsymbol{\theta}}_{k_j}) \to L(\boldsymbol{\theta}^*)$ $a.s$ for some subsequence $\{k_j\}$. Given the convergence of $L(\hat{\boldsymbol{\theta}}_k)$ for each sample point $\boldsymbol{\omega}$, and the fact that the subsequence has to converge to the same point as the full sequence, we know that the $a.s$ limit of $L(\hat{\boldsymbol{\theta}}_k)$ must also be $L(\boldsymbol{\theta}^*)$. Hence, $\hat{\boldsymbol{\theta}}_k \to \boldsymbol{\theta}^*$ $a.s$ by the uniqueness of $\boldsymbol{\theta}^*$.

Realistically these convergence results may have limited utility in practice since the algorithm may take a prohibitively large number of function evaluations to reach a value very close to $\boldsymbol{\theta}^*$, especially if the problem dimension is large. Nevertheless, the formal convergence proof provides a guarantee for convergence to the global minimum for very large number of iterations. As we increase the number of function evaluations, the probability of the algorithm getting stuck in a local minima decreases.

3.2.3 Application of Random Search in our algorithm

In our application of the Global Random Search algorithm to achieve the optimum local deformation of the face mesh, the cost function L is the disparity between the contours obtained from the 3D face mesh after applying the local deformation ($\boldsymbol{\theta}$) and the edges from the current video frame. The stochastic optimization algorithm determines the value of $\boldsymbol{\theta}$ that minimizes the cost function L. The parameter vector $\boldsymbol{\theta}$ whose optimum value is being sought is of the form:

$$\boldsymbol{\theta} = [\Delta X_c \quad \Delta Y_c \quad \Delta Z_c]_{P\times 3} \tag{3.8}$$

where ΔX_c, ΔY_c, and ΔZ_c are the perturbations in the X, Y, and Z directions of the P control points. The perturbations for all vertices of the dense face mesh, $[\Delta X_{\text{model}}\ \Delta Y_{\text{model}}\ \Delta Z_{\text{model}}]$, are computed from the perturbations of these control points using triangle-based linear interpolation. The coordinates of the vertices of the locally deformed face model are given by,

$$\begin{bmatrix} X_{\text{lcdef}} \\ Y_{\text{lcdef}} \\ Z_{\text{lcdef}} \end{bmatrix} = \begin{bmatrix} X_{\text{gbdef}} \\ Y_{\text{gbdef}} \\ Z_{\text{gbdef}} \end{bmatrix} + \begin{bmatrix} \Delta X_{\text{model}} \\ \Delta Y_{\text{model}} \\ \Delta Z_{\text{model}} \end{bmatrix} \tag{3.9}$$

where the subscript "lcdef" denotes local deformation. Let $EM_{\boldsymbol{\theta}}$ be the binary edge map of the 2D projection of the 3D mesh after applying local deformation $\boldsymbol{\theta}$. The unwanted edges due to the boundaries of the 3D model are removed by the method described earlier in Section 2.2.2. Let DT be the distance transform of the edge map of the current video frame. The cost function, L, is of the form:

$$L(\boldsymbol{\theta}) = \frac{\sum_{(i,j)\in A_{EM_{\boldsymbol{\theta}}}} DT(i,j)}{N} \tag{3.10}$$

where $A_{EM_{\boldsymbol{\theta}}} \triangleq \{(i,j) : EM_{\boldsymbol{\theta}}(i,j) = 1\}$ and N is the size of set $A_{EM_{\boldsymbol{\theta}}}$. The structure of this cost function is similar to the one used in Equation 2.2. In Chapter 2, we compare the edge maps to determine the head pose, but here we compare them to determine the optimal local deformation for the globally deformed generic face model.

3.2.4 Multi-Resolution Search

The random perturbations are applied to the face mesh at two different resolutions. Initially, M iterations of large random perturbation are performed to scan a large search area at a coarse resolution. Within each iteration k, each of the control points is imparted a large random perturbation $\boldsymbol{\theta}^c(k)$, where superscript "c" denotes coarse resolution. Perturbations for all vertices of the face mesh are determined by triangle-based interpolation, and the cost function L is evaluated. If $L(\boldsymbol{\theta}^c(k)) < L(\hat{\boldsymbol{\theta}}_{k-1})$, $\hat{\boldsymbol{\theta}}^c_k = \boldsymbol{\theta}^c(k)$, else $\hat{\boldsymbol{\theta}}^c_k = \hat{\boldsymbol{\theta}}^c_{k-1}$. The final estimate at the end of M coarse-level iterations is denoted by $\hat{\boldsymbol{\theta}}^c_M$. Once we have a coarsely deformed mesh which is close to the global minimum, we use this coarsely deformed mesh as the starting estimate for the fine resolution iterations, i.e. $\hat{\boldsymbol{\theta}}^c_M = \hat{\boldsymbol{\theta}}^f_0$. N small random perturbations $\boldsymbol{\theta}^f(k)$, where superscript "f" denotes fine resolution, are applied to the coarsely deformed mesh to get even closer to the actual solution. The final estimate at the end of N fine-level iterations is denoted by $\hat{\boldsymbol{\theta}}^f_N$. This coarse to fine resolution search helps the algorithm converge to the solution much faster, compared to a search strategy where we sample the search space at a fixed fine resolution. As we increase the number of iterations performed for the coarse resolution and fine resolution search, the performance of the algorithm gets better, but, at the cost of computational time. We observed that the

algorithm converges to a reasonable solution in 300 iterations of large random perturbations (coarse), and 100 iterations of small random perturbations (fine). As a final step to fine tune the control point perturbations estimate, we exhaustively search for the optimal perturbation $\hat{\boldsymbol{\theta}}_N^{sf}(i)$ for each control point i, individually, in a small neighborhood of the solution from fine resolution iterations $\hat{\boldsymbol{\theta}}_N^{f}(i)$, while fixing the perturbations of all other control points to the value determined by $\hat{\boldsymbol{\theta}}_N^{f}$. The perturbation that results in the minimum value of the cost function L, is chosen to be the optimal perturbation $\hat{\boldsymbol{\theta}}_N^{sf}(i)$ for that particular control point. The superscript "*sf*" denotes super-fine estimate. Ideally, we would like to perform this exhaustive search in a combinatorial manner (instead of individual) over the entire search space, but the computational complexity for this strategy would be extremely prohibitive.

3.2.5 Constraints

We impose a few constraints on the values of perturbations that are imparted to each of the control points. The basic idea behind imposing these constraints is to limit the search space to the domain of all realistic looking human faces. Without any constraints, the algorithm will also search in the domain of unrealistic faces, thus unnecessarily slowing down the algorithm. The following are the constraints imposed on the perturbations of the control points:

- Symmetry along the vertical axis passing through nose tip, based on the fact that most human faces are symmetric about this axis.
- The maximum possible perturbation for the control points is defined.
- The perturbation of a few control points is dependent on the perturbations

of their neighboring control points.

- Only control points whose movement might alter the contours of the 3D model (and hence change the cost function) are perturbed. For example, in a frontal pose, the control points responsible for the movement of the nose in the Z-direction (depth) will not be perturbed in that direction.

Most of these constraints have been determined heuristically, keeping in mind the basic geometry of a human face. These constraints greatly help us limit our search to the domain of realistic-looking human faces, thus making our algorithm faster.

3.3 Pose Refinement and Model Adaptation across time

Once we have an adapted 3D model from a particular frame, we refine the rough pose estimate (obtained using a generic model) of the next frame to obtain a more accurate head pose estimate. The method we use for this pose refinement is similar to the method described in Chapter 2, except that we extract the contours to be compared to the edges of the current frame from the adapted 3D model upto that stage, instead of the generic face mesh. For the first video frame, the adapted 3D model used for pose refinement is just the globally deformed generic mesh. But, for later frames, we use the globally and locally adapted 3D model for pose refinement. This pose refinement step is critical to the shape estimation procedure because the contour-based algorithm that we describe is very sensitive to the head pose estimate. If our pose estimate is not accurate, the 3D model will adapt itself in an inappropriate manner so that its contours conform with the edges extracted from the video frame. To refine the current

estimate of the pose, we obtain the contours from the adapted 3D face model rotated about the azimuth angles in the neighborhood of the rough pose estimate, and choose the refined pose estimate as the angle which results in contours closest to the edges extracted from the current frame. The similarity criterion used to compare the edge maps is again the Distance Transform based measure described in Chapter 2, Equation 2.2.

Once we have the refined pose estimate and the adapted (global and local) 3D face model from the first video frame, we apply the algorithm for pose refinement and local deformation described earlier in this chapter to all the subsequent frames to improve the quality of the reconstructed 3D face model. The adapted 3D model from the previous frame is used as the starting estimate for the next frame. As we get more and more frames of the video, if the person is changing pose with time, we have more information available to model the face. For example, in a full frontal face, its is difficult to determine the height of the nose, but in a profile pose, we can easily extract that information and model the nose accordingly. Therefore, as we get more frame with varying poses, we get more and more cues to model certain aspects of the face more accurately.

Figure 3.3 shows the rough pose estimate (with generic model), and the refined pose estimate (with adapted 3D model) for a few video frames of a subject. The 3D models have been texture mapped with the texture extracted using the algorithm described in the next section. We can see from the figure that in cases when the rough pose estimate is not good (3.3(b,e)), we are still able to estimate the accurate head pose at the refinement step. Figure 3.4 shows a sequence of contours extracted from the adapted 3D model (green) and contours

extracted from the generic model (blue), overlapped over the edges from the video frames of a subject (red). The figure illustrates the extent of conformity of the contours of the adapted 3D model, and the edges from the video frame.

3.4 Texture Extraction

Once the pose refinement and the 3D model adaptation has been done for all the frames, we need to extract the texture of the person's face and map it onto the adapted 3D model. We create a single texture-map image in contrast to the view-independent texture extraction approach used by Moghaddam et al [20]. We found from our experiments that the weighted texture sampling across multiple frames approach smears the texture slightly, making it appear like a smoothed texture. This smearing occurs because of the fact that the registration (using estimated pose) across multiple frames is not exact.

In our approach, we extract the texture from a single frame. Now the question arises which frame to choose for texture extraction? Based on studies that face recognition systems perform best on 3/4 profile view [28], we believe that this view is most representative of the person. Hence, we choose the frame closest to 3/4 profile view ($\pm 45°$) for texture extraction. In a 3/4 profile view, part of the face would be occluded for one half of the face. To get the texture for this half of face (part of which is occluded), we assume symmetry of face texture about the vertical axis passing through the nose tip. Figure 3.5 displays the multi-frame weighted texture, and the single frame symmetrical texture from the 3/4 profile view that we use for our algorithm.

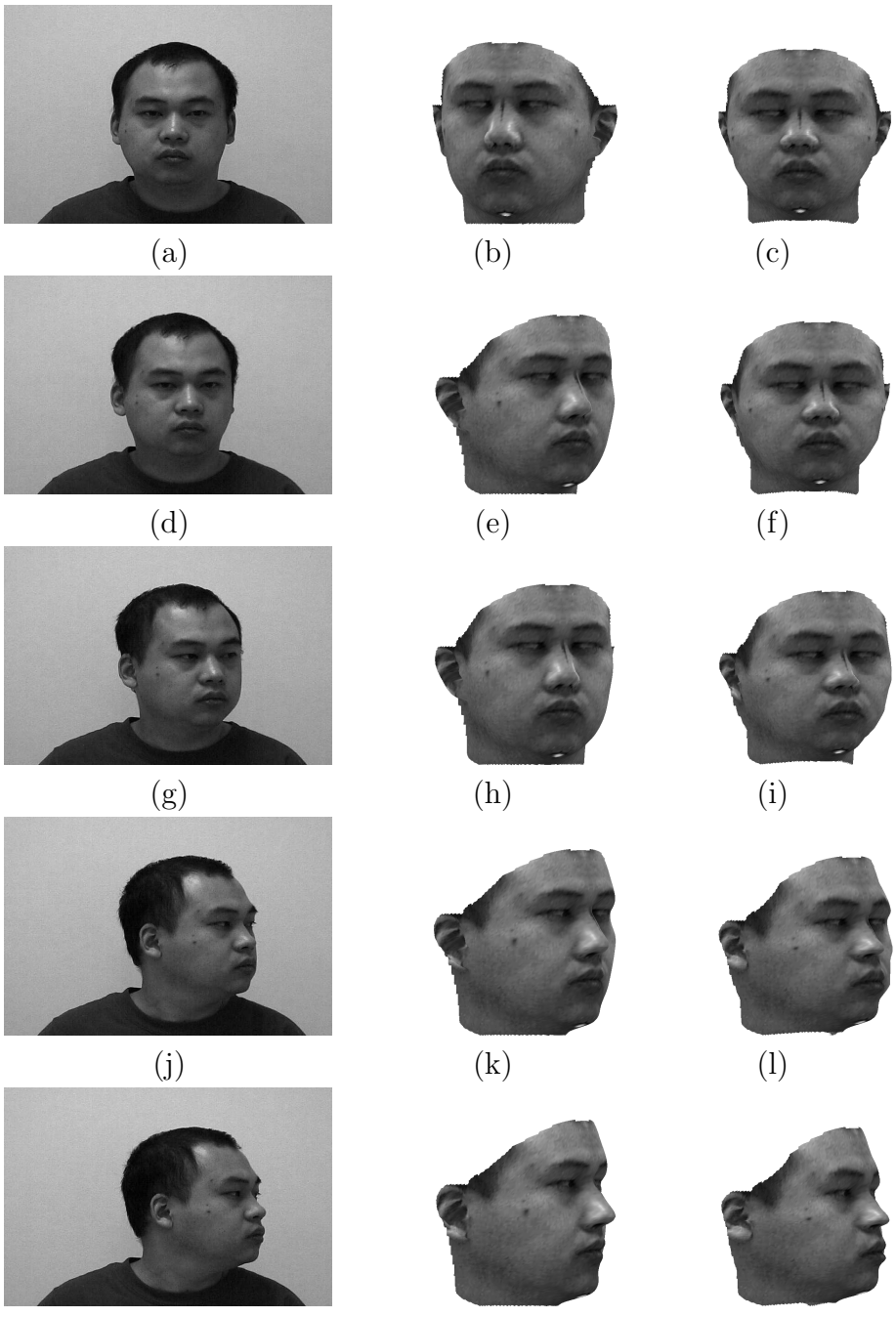


Figure 3.3: Left Column: Video Frames. Middle Column: Generic 3D model (texture mapped) at the rough pose estimate. Right Column: Adapted 3D model (texture mapped) at the refined pose estimate.

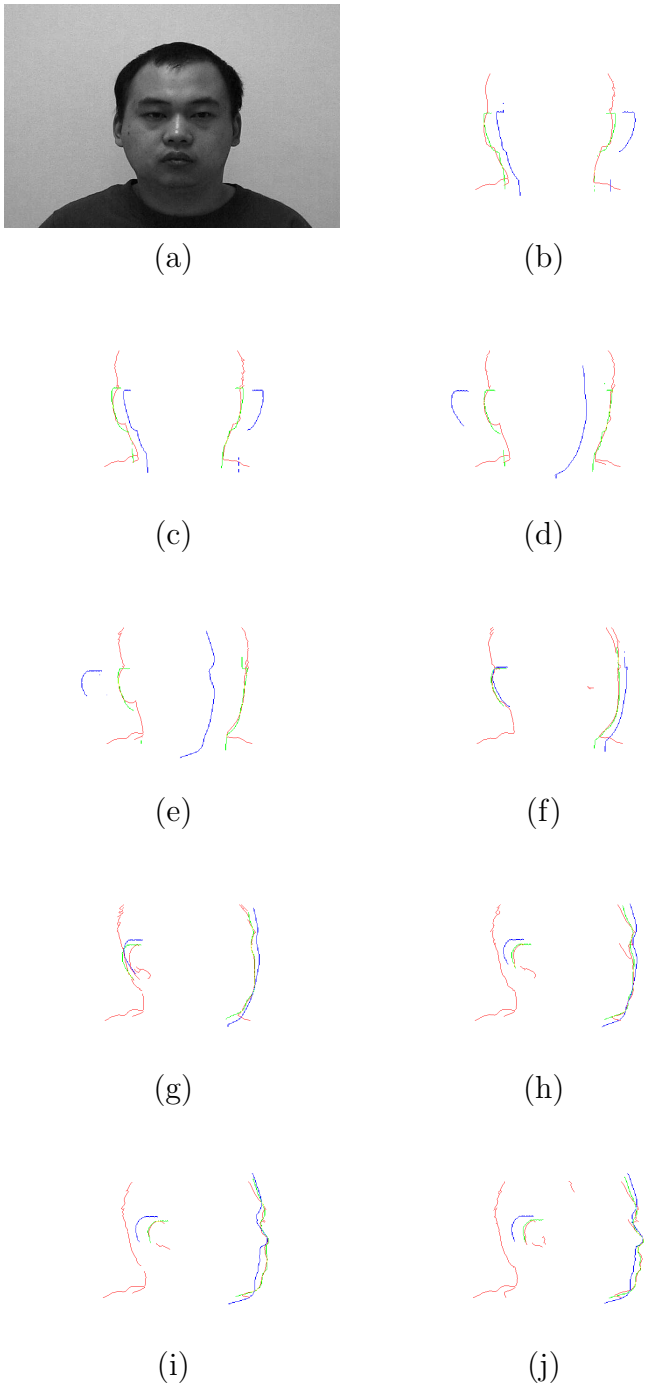
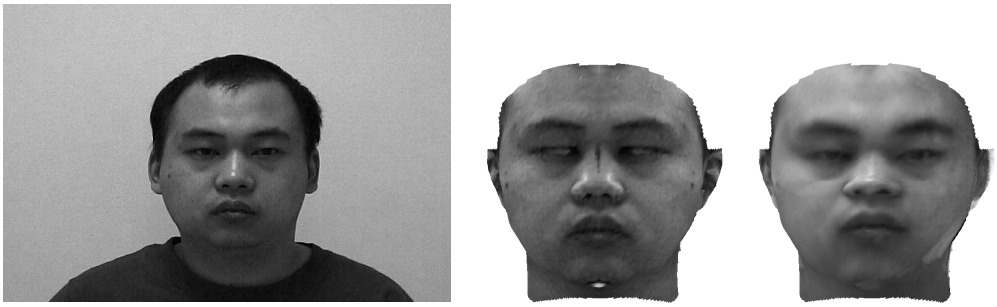


Figure 3.4: (a): First Video Frame. (b)-(j): Sequence of contours extracted from the adapted 3D model (green) and contours extracted from the generic model (blue), overlapped over the edges from the video frames (red).



(a) (b) (c)

Figure 3.5: (a): First Video Frame (b): Single-frame symmetrical texture (c): Multi-frame weighted texture

3.5 Summary of the 3D Face Reconstruction Algorithm

The algorithm for 3D face reconstruction described in this thesis can be summarized as follows:

1. Estimate the rough head pose for all the video frames using the method described in Chapter 2.
2. Apply an affine model for global deformation of the generic 3D face model.
3. Refine the pose estimate based on the current adapted 3D model .
4. Deform the 3D model locally in a stochastic search optimization framework.
5. Go to Step 3 if there are more frames.

Figure 3.6 shows an illustration of this algorithm.

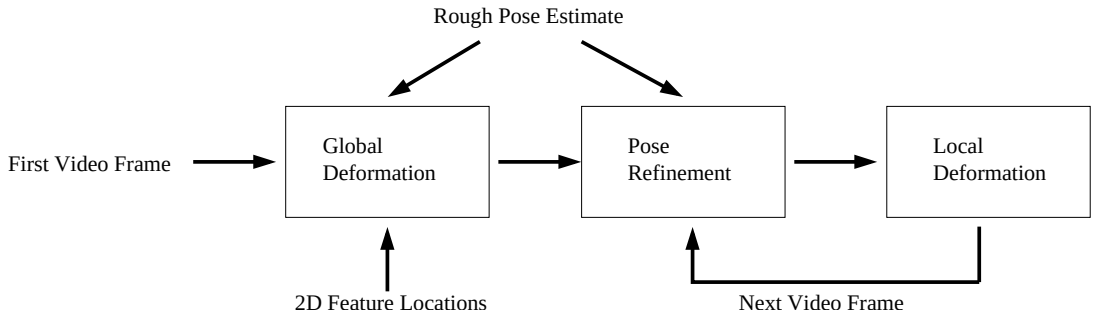


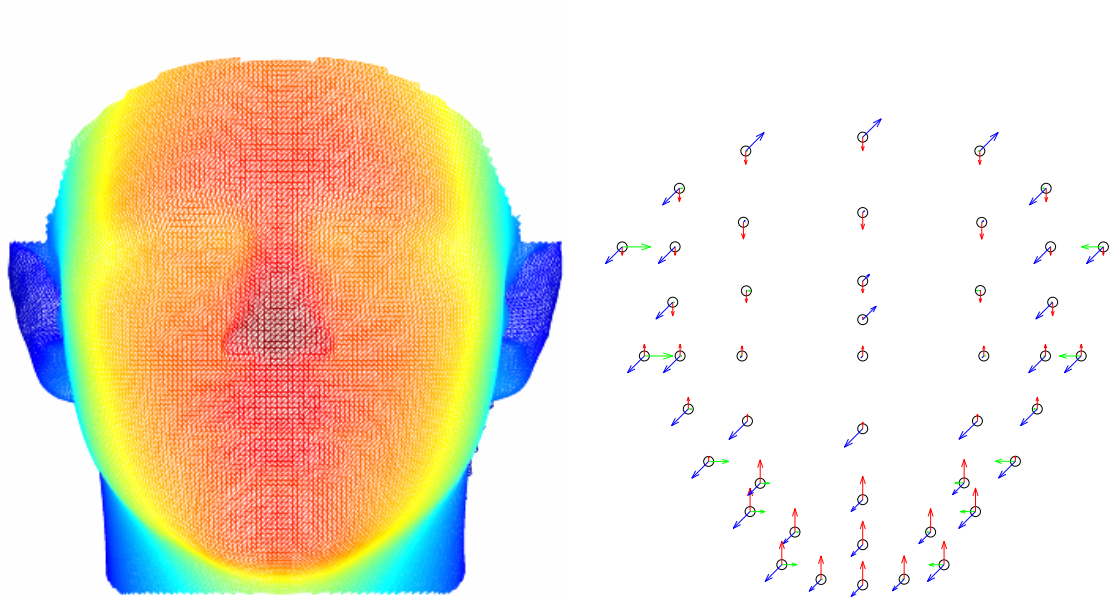
Figure 3.6: 3D face reconstruction algorithm

Chapter 4

Results

In this chapter we present the experimental results of our proposed contour-based 3D face reconstruction algorithm. The results of experiments performed on the videos of three subjects are presented. Figures 4.1, 4.5, 4.9 show the video frames used to reconstruct the 3D face model of the three subjects. Figures 4.2(a), 4.6(a), 4.10(a) show the globally deformed generic face models for the respective individuals. Figures 4.2(b), 4.6(b), 4.10(b) show the optimal local perturbations of the sparse control points obtained from the stochastic search optimization method. The globally deformed face model is locally deformed according to these local perturbations to obtain the final adapted 3D face model. Green arrows are the perturbation along X-axis (width), red arrows are the perturbation along Y-axis (height), and the blue arrows are the perturbation along Z-axis (depth). The direction of the positive X, Y, Z axes is the same as shown in Figure 2.3. Figures 4.3, 4.7, 4.11 show the minimum perturbation error (value of cost function L) at each stage of the multi-resolution search across all frames of the video sequence. Figures 4.4, 4.8, 4.12 show the reconstructed 3D face model of the subjects from different viewpoints.

Figure 4.1: Subject A: Video frames used for 3D model reconstruction



(a) (b)

Figure 4.2: (a): Generic mesh after global affine deformation (b): Optimal perturbations applied to each control point to obtain the final adapted model. Green: Perturbation along X-axis (width), Red: Perturbation along Y-axis (height), Blue: Perturbation along Z-axis (depth). The results are for Subject A.

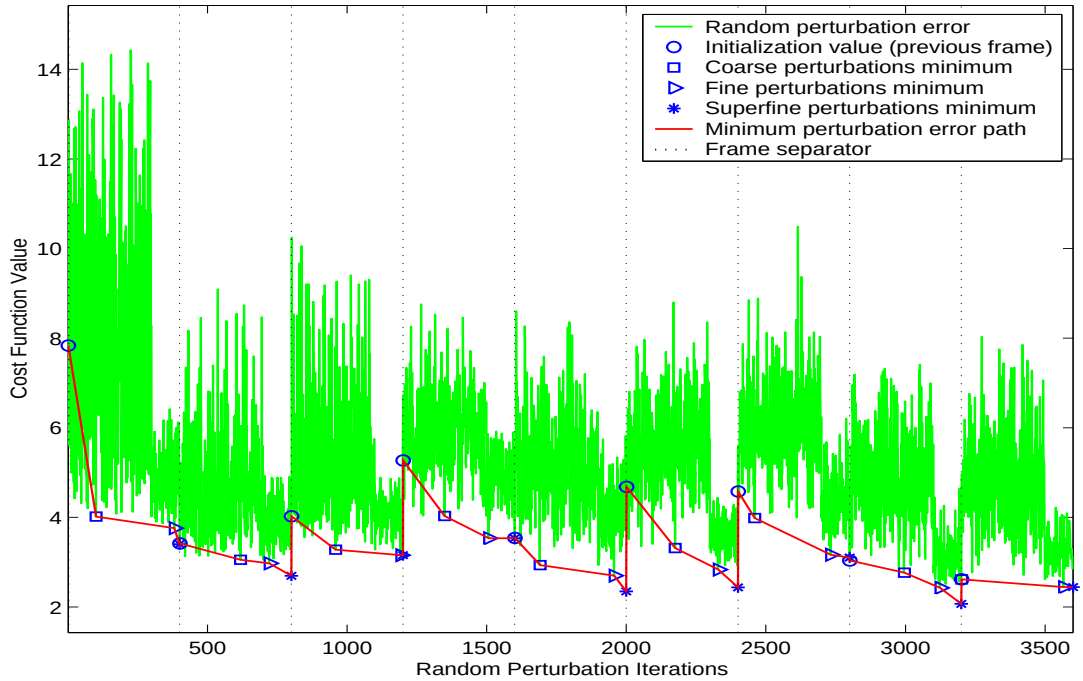


Figure 4.3: Perturbation errors at each stage of the multi-resolution search, for all frames. The results are for Subject A.

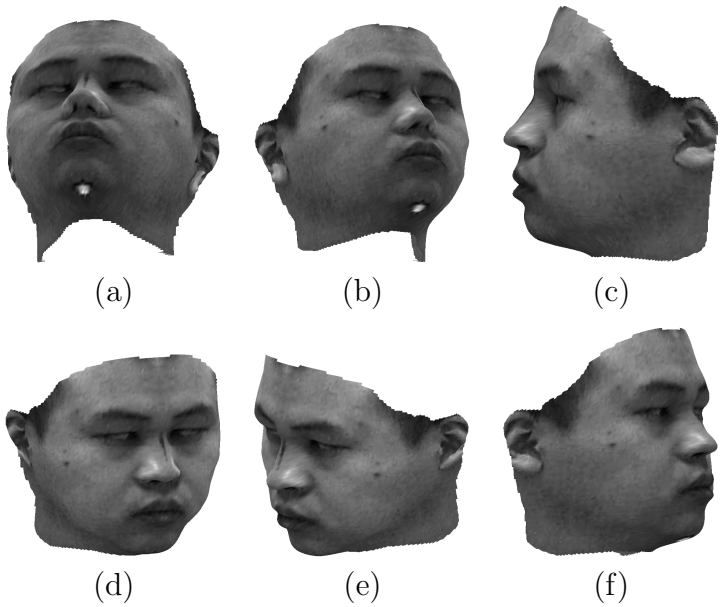
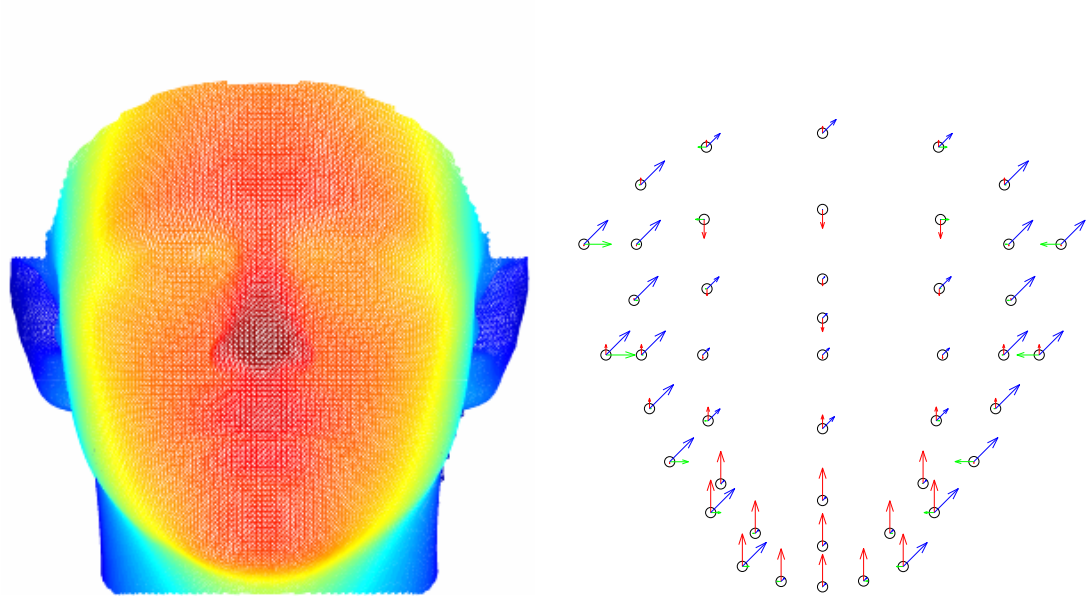


Figure 4.4: Reconstructed 3D face model of Subject A seen from different viewpoints

Figure 4.5: Subject B: Video frames used for 3D model reconstruction



(a) (b)

Figure 4.6: (a): Generic mesh after global affine deformation (b): Optimal perturbations applied to each control point to obtain the final adapted model. Green: Perturbation along X-axis (width), Red: Perturbation along Y-axis (height), Blue: Perturbation along Z-axis (depth). The results are for Subject B.

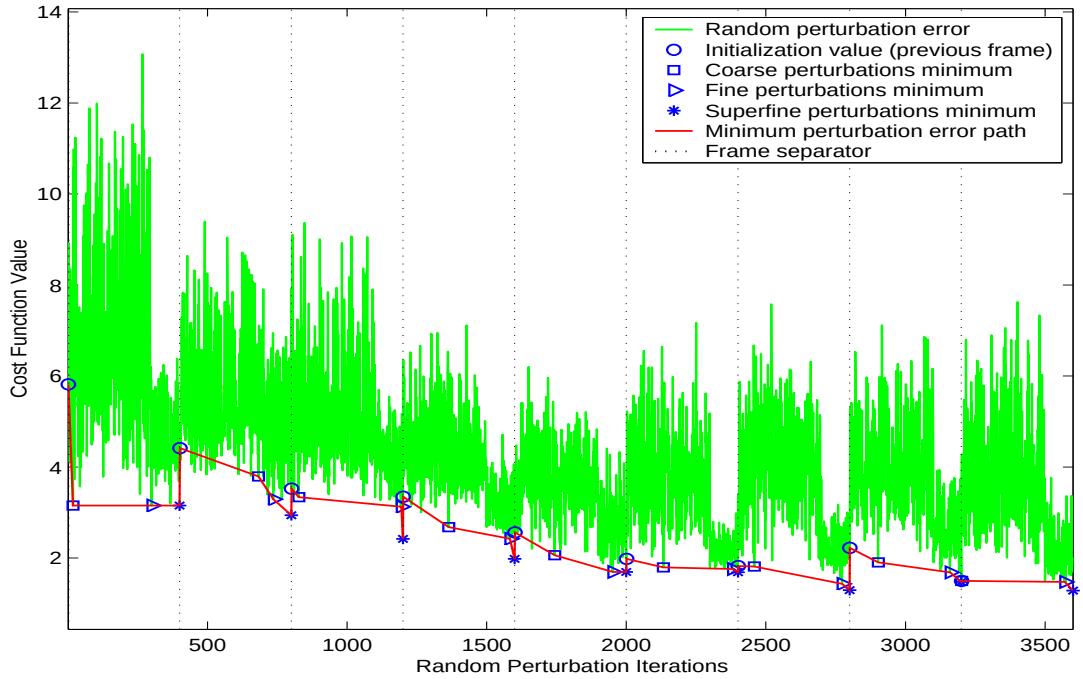


Figure 4.7: Perturbation errors at each stage of the multi-resolution search, for all frames. The results are for Subject B.

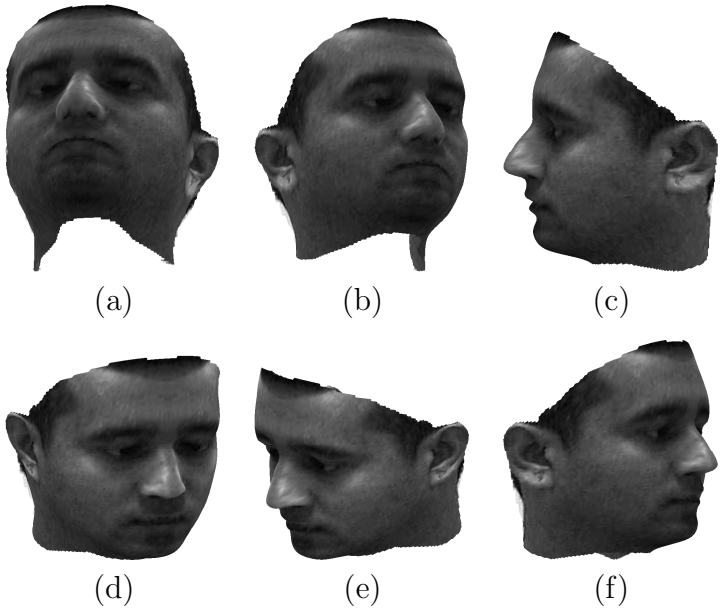
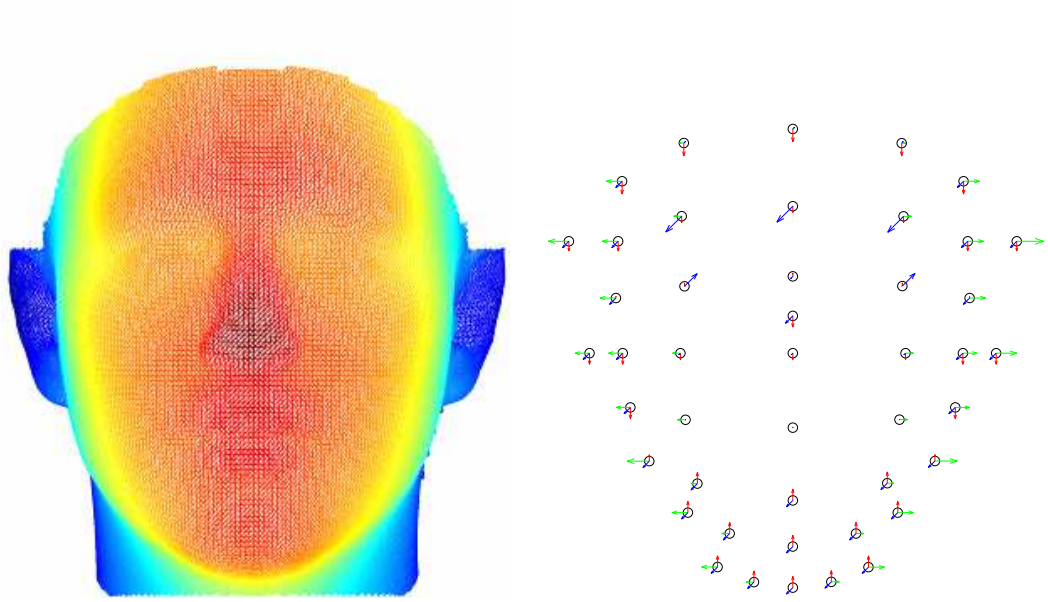


Figure 4.8: Reconstructed 3D face model of Subject B seen from different viewpoints

Figure 4.9: Subject C: Video frames used for 3D model reconstruction



(a) (b)

Figure 4.10: (a): Generic mesh after global affine deformation (b): Optimal perturbations applied to each control point to obtain the final adapted model. Green: Perturbation along X-axis (width), Red: Perturbation along Y-axis (height), Blue: Perturbation along Z-axis (depth). The results are for Subject C.

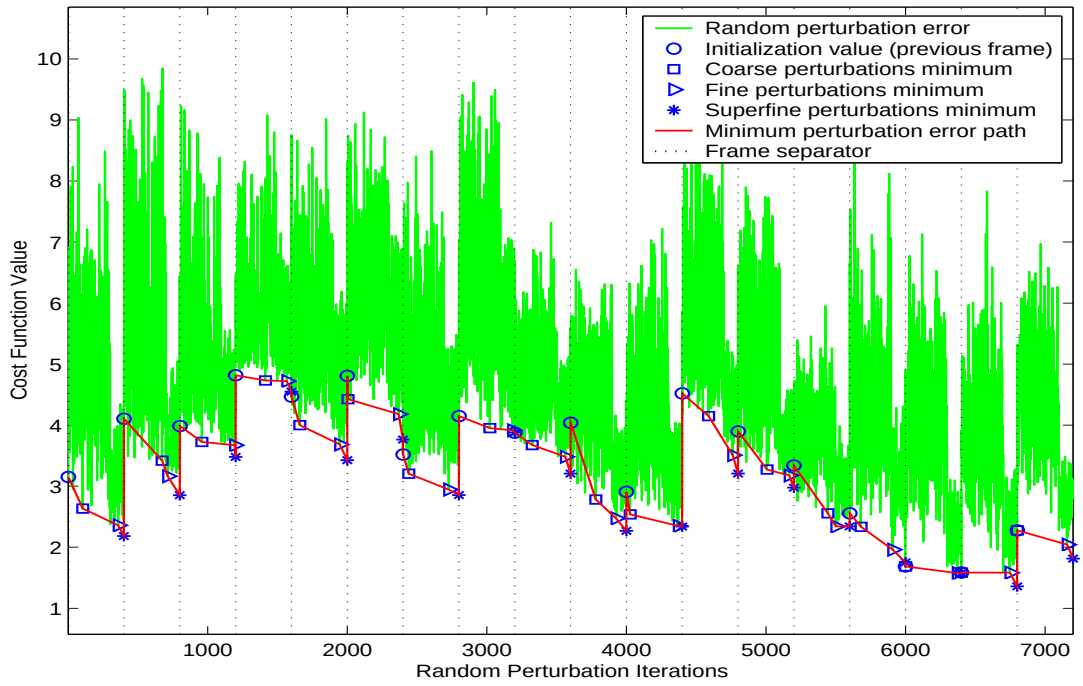


Figure 4.11: Perturbation errors at each stage of the multi-resolution search, for all frames. The results are for Subject C.

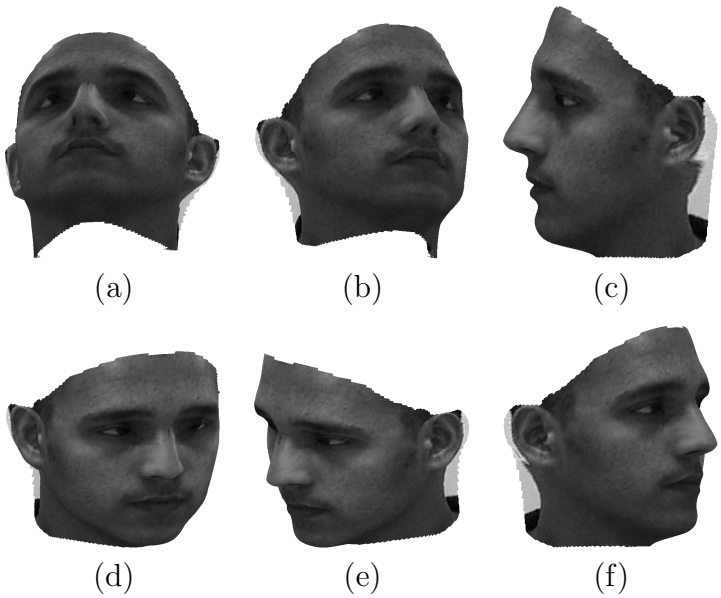


Figure 4.12: Reconstructed 3D face model of Subject C seen from different viewpoints

Chapter 5

Conclusions and Future Work

In this thesis, we have addressed the problem of head pose estimation and 3D face modeling from a monocular video sequence. We presented an algorithm to estimate the rough head pose by comparing the edges from the video frame with the contours extracted from a generic 3D face model rotated by different azimuth angles, using a Distance Transform based similarity criterion. Next, an algorithm for 3D reconstruction of face from a monocular video sequence is presented. We assume the generic 3D model to be the initial estimate of the true 3D face model. To adapt this generic face model to the actual 3D face model of the person, the deformation of the generic 3D mesh is performed in 2 stages: Global and Local. An affine model is used for global deformation of the mesh. To impart local deformations to the globally deformed mesh, we have proposed a solution where we search for the optimal perturbations of control points at multiple resolutions in a stochastic optimization framework. A method to refine the rough pose estimate (from generic model) for each frame is described. The deformations are integrated over a set of poses in the video sequence, leading to an accurate 3D face model.

Since our algorithm for pose estimation and 3D face reconstruction relies

solely on contours, we do not require knowledge of rendering parameters (e.g light direction, intensity, etc.) which might otherwise be needed. Using contours separates the geometric subtleties of the human head from the variations in shading and texture. Hence, we believe our system is much more robust to illumination changes compared to most structure from motion (SfM) algorithms that rely on finding accurate point correspondences across frames.

3D face model reconstruction has immense potential for face recognition from still images and video sequences. We believe that 3D face models will be the backbone of robust face recognition systems. The algorithm described in this thesis for 3D face reconstruction can be applied to build pose-invariant face recognition system. If the probe is a face image from a viewpoint never seen before, the 3D face model can be used to create novel synthetic views to populate the gallery, thus improving the recognition rate. Similarly, incorporating a 3D model into a face recognition system can greatly help in making the system robust to illumination variations. In case of extreme illumination conditions when part of the outer contours of face are not clearly visible, we might have to do illumination normalization as a preprocessing step for our algorithm. Illumination normalization is a difficult problem in itself, and this could be another interesting extension to our work.

BIBLIOGRAPHY

[1] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Journal of Computing Surveys*, December 2003.

[2] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, October 2000.

[3] P. Belhumeur and D. Kriegman, "What is the set of images of an object under all possible lighting conditions," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA*, June 1996, pp. 270–277.

[4] W. Zhao and R. Chellappa, "Shape from shading based view synthesis for robust face recognition," in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France*, March 2000, pp. 285–292.

[5] T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 733–742, July 1997.

[6] T. Maurer and C.v.d Malsburg, "Single-view based recognition of faces rotated in depth," in *First International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland*, June 1995, pp. 248–253.

[7] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recogition, Seattle, WA*, June 1994, pp. 84–91.

[8] T.J. Broida and R. Chellappa, "Estimating the kinematics and structure of a rigid object from a sequence of monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 497–513, June 1991.

[9] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 562–575, June 1995.

[10] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recogition, Hilton Head, SC*, June 2000, pp. II:690–696.

[11] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *International Journal of Computer Vision*, vol. 9, pp. 137–154, November 1992.

[12] Conrad J. Poelman and Takeo Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206–218, March 1997.

[13] M. Pollefeys, "Self-calibration and metric 3D reconstruction from uncalibrated image sequences," 2000, PhD Thesis, ESAT-PSI, K.U.Leuven.

[14] P. Fua, "Regularized bundle-adjustment to model heads from image sequences without calibration data," *International Journal of Computer Vision*, vol. 38, no. 2, pp. 153–171, July 2000.

[15] Y. Shan, Z. Liu, and Z. Zhang, "Model-based bundle adjustment with application to face modeling," in *International Conference on Computer Vision, Vancouver, BC*, July 2001, pp. 644–651.

[16] A.K. Roy Chowdhury and R. Chellappa, "Face reconstruction from video using uncertainty analysis and a generic model," *Computer Vision and Image Understanding*, vol. 91, no. 1-2, pp. 188–213, 2003.

[17] S. Romdhani, V. Blanz, and T. Vetter, "Face identification by fitting a 3D morphable model using linear shape and texture error functions," in *European Conference on Computer Vision, Copenhagen, Denmark*, May 2002, pp. 3–19.

[18] M. Brady and A.L. Yuille, "An extremum principle for shape from contour," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 3, pp. 288–301, May 1984.

[19] W. Matusik, C. Buehler, R. Raskar, S.J Gortler, and L. McMillan, "Image-based visual hulls," in *Computer Graphics Proceedings, Siggraph 2000, New Orleans, LA*, July 2000, pp. 369–374.

[20] B. Moghaddam, J. Lee, H. Pfister, and R. Machiraju, "Model-based 3-D face capture with shape-from-silhouettes," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures, Nice, France*, October 2003, pp. 20–27.

[21] A. Tsukamoto, C. Lee, and S. Tsuji, "Detection and pose estimation of human face with synthesized image models," in *International Conference on Pattern Recognition, Jerusalem, Israel*, October 1994, pp. 754–757.

[22] A. Gee and R. Cipolla, "Estimating gaze from a single view of a face," in *International Conference on Computer Vision, Cambridge, MA*, June 1995, pp. 758–760.

[23] T. Horprasert, Y. Yacoob, and L. Davis, "Computing 3-D head orientation from a monocular image sequence," in *Second International Conference on Automatic Face and Gesture Recognition, Killington, VT*, October 1996, pp. 242–247.

[24] C. Tomasi and J. Shi, "Good features to track," in *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recogition, Jerusalem, Israel*, October 1994, pp. 593–600.

[25] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, November 1986.

[26] J.C. Spall, *Introduction to Stochastic Search and Optimization*, Wiley, 2000.

[27] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, Oxford University Press, 2001.

[28] V. Bruce, T. Valentine, and A. Baddeley, "The basis of the 3/4 view advantage in face recognition," *Applied Cognitive Psychology*, vol. 1, pp. 109–120, 1987.