Creation of 3D Face Models from 2D Images

A Project Report

Submitted by

J.ELANGKUMARAN
EE10B009

for the award of the degree

of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

CHENNAI – 600036

MAY 2014

CERTIFICATE

This is to certify that the project entitled "**Creation of 3D Face Models from 2D Images**" submitted by **Mr J.Elangkumaran (EE10B009)** is a bonafide record of work carried out by him in partial fulfilment for the award of degree of **Bachelor of Technology** in **Electrical Engineering**. The contents of this report, in full or in parts, have not been submitted and will not be submitted to any other Institute or University for the award of any degree or diploma.

Dr. B. Ravindran

Professor

Department of Computer Science and Engineering

Indian Institute of Technology Madras

Chennai – 600036

Date:

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and indebtedness to my project guide, Dr. B.Ravindran, Professor in the Department of Computer Science and Engineering, for his most valuable guidance, discussions, suggestions and encouragement, from the conception to the completion of this project. His moral support, unreserved co-operation and generosity, which enabled me to complete the work successfully, will be everlasting in my memory.

I would also like to thank my professors and friends from my undergraduate studies while at Indian Institute of Technology, Madras. The students pursuing their Masters at the laboratories in the Department of Electrical Engineering and Department of Computer Science and Engineering were also extremely instrumental in the project's completion.

A special thanks goes out to the users of Stackoverflow, without whom my project would not have materialised.

ABSTRACT

Creating 3D images still remains a difficult task. It requires the use of expensive scanners and sensors, or spending a lot of time with 3D modelling software. In this project, an attempt was made to construct 3D images of faces from their 2D images.

The well-known method was to use the Shape from Motion algorithm described in [2], along with the combined 2D+3D AAM approach to construct a 3D model of the face. An approach that used estimated depth information, as described in [3] was adopted as it needed a lesser number of images. Due to a lack of an AAM implementation that could accommodate the varying angles and provide adequate facial feature information, a different approach was adopted.

Two images were used, a frontal face image and a profile view of the same subject. Depth information of the feature points extracted from the former was estimated from the latter, as described in [3]. The sparse mesh was interpolated to get a 3D face shape.

Key words: Image Processing, Active Appearance Models, Active Shape Models, 3D images, Shape from Motion, Profile face

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AAM	Active Appearance Model
ASM	Active Shape Model

1. Introduction

1.1 Project Motivation

The first digital photograph was taken by Russell Kirsch in 1957 by scanning a photograph of his son's face into a 176x176 pixel image. Technology has progressed a lot since, and 2D digital photographs can be found at every nook and corner of the World Wide Web. Instagram, the photo based social-networking site averages 60 million photographs every day.

On the other hand, 3D images do not enjoy the same popularity, mostly because they are difficult to create. There are multiple applications for 3D face images in the field of cosmetics, for example an app for trying out new hairstyles or spectacle frames. It can also be used to quickly create models which are used for 3D animated sequences, or map the user's face to a player in a computer game.

1.2 Report Structure

Chapter 2 introduces Active Shape Models and Active Appearance Models. They used for facial feature extraction from generic face images, and thus form the foundation of the project.

Chapter 3 gives the details of the estimated depth algorithm that can be used to construct a 3D AAM. It describes how it works on an annotated dataset, and the difficulties in working with generic face images.

Chapter 4 discusses the construction of 3D face images using a frontal and a profile view of the same subject's face. The algorithm based on [3] is described and results are shown with various faces.

Chapter 5 gives a summary, and **Chapter 6** discusses possible future work

1.3 Terminology

1. The terms frontal image, frontal face or frontal face image refer to an image of a face facing the camera, for example the image in Figure 1.1(a)
2. The terms frontal shape or frontal face shape refer to the shape i.e. set of locations of facial features detected in a frontal face image, shown by the numbers in Figure 1.1(a)
3. The terms profile image, profile face or profile face image refer to an image of a face facing left or right, such that only one ear is visible. An example is shown in Figure 1.1(b)
4. The terms profile shape or frontal face shape refer to the shape detected in a profile face image, such as the set of numbers shown in Figure 1.1(b)

(a) (b)

***Figure 1.1**:Frontal face image and shape (a), profile face image and shape (b)*

2. Active Shape Models and Active Appearance Models

Active Appearance Models (AAMs) belong to a set of linear shape and appearance models. Other elements in said class include Direct Appearance Models, Active Blobs and Morphable Models. Although they can be used for other object classes, their most popular application has been face modelling. AAM is generally used to refer to the model (it's components as described below), or the process by which it is 'fit' to a new face.

2.1 Parameters of AAM

2.1.1 Shape (*s*)

The shape of an AAM is a mesh consisting of different vertices that correspond to various features of the face. It is the set s of n vertices $(x_i, y_i)^T$ that constitute the mesh

$$s = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T \tag{1}$$

The shape varies linearly i.e. it can be expressed as a combination of a base shape s_0 and a linear combination of shapes s_i (usually orthonormal) with shape parameters p_i.

$$s = s_0 + \sum_{1}^{n} p_i s_i \tag{2}$$

AAMs are made from a set of annotated training images. Procrustes analysis is done to normalise the shapes, and Principal Component Analysis is done to extract the shape data.

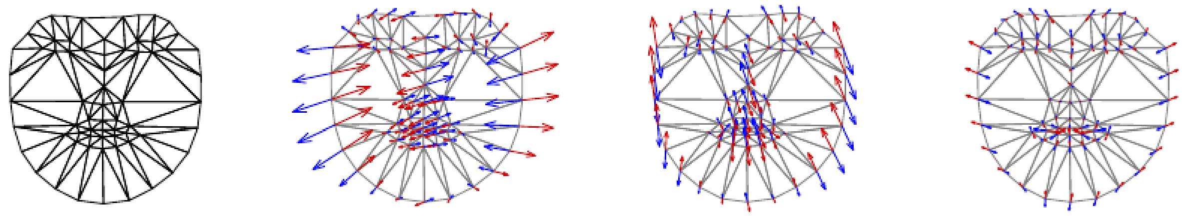

***Figure 2.1**: Shape model of an AAM. The base shape s_0 is shown on the left, while three of the shape vectors*

2.1.2 Appearance (A)

The appearance of an AAM is the image contained within the base mesh s_0. If we let s_0 to also denote the set of all vertices $v = (x_i, y_i)^T$ lying within it, then the appearance A(v) is the image defined over it. Similar to shape, it also be expressed as s base appearance and a linear combination of m appearance images $A_i(v)$ with λ_i as the appearance parameters.

$$A(v) = A_0(v) + \sum_{1}^{m} \lambda_i A_i\,(v) \qquad \forall\, x \in s_0 \tag{3}$$

***Figure 2.2** Appearance of an AAM. $\boldsymbol{A_0}(\boldsymbol{x})$ is the base appearance, and is shown on the left while $\boldsymbol{A_1}(\boldsymbol{x}), \boldsymbol{A_2}(\boldsymbol{x}), \boldsymbol{A_3}(\boldsymbol{x})$ are appearance images, all defined over the base shape $\boldsymbol{s_0}$*

Similar to shape, the base appearance and the appearance images are normally computed by applying PCA to a set of shape normalised training images. Each training image is shape normalised by warping the (hand labelled) training mesh onto the base mesh. Usually the mesh is triangulated and a piecewise affine warp is defined between corresponding triangles in the training and base meshes

2.2 Fitting an AAM

The process of starting from an initial shape and finally getting the AAM to ‘fit’ on top of the subject’s face i.e. get the shape mesh vertices to correspond with the facial features is called fitting. The initial shape is generally generated after a face detector has run.

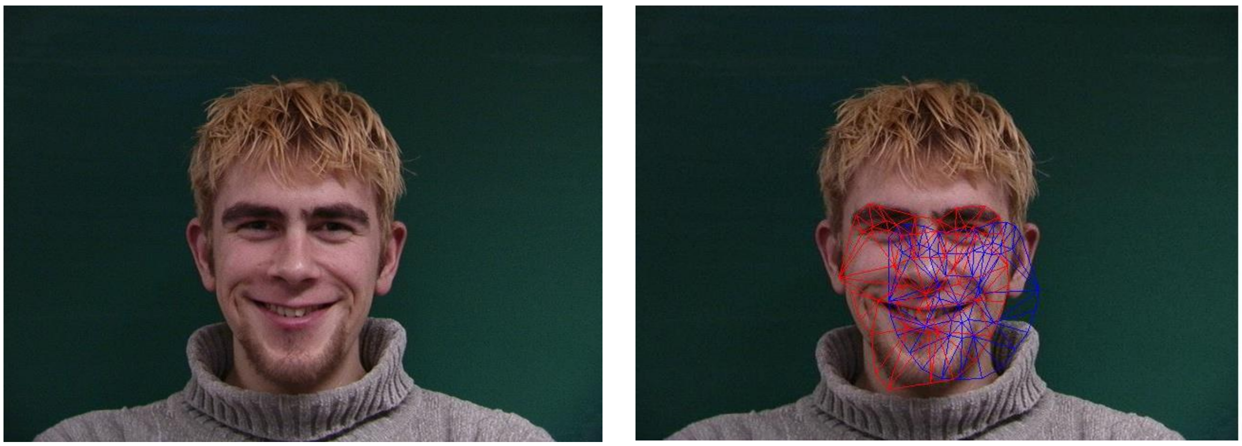

Figure 2.3: *Fitting of an AAM. The image on the left is the face from the IMM dataset. The image on the right shows a semi-randomised initial shape in blue, and the final ‘fitted’ AAM in red*

There are multiple algorithms for fitting AAMs, and each has its merits. The algorithm used in this project for the estimated depth algorithm is an Inverse Computational one described in [4], which does not assume a constant linear relationship between the error image and incremental parameter value changes.

2.3 Active Shape Model

An ASM is similar to an AAM, except for the fact that it uses only shape information and not texture information i.e. the pixel intensities across the image that is stored as Appearance information in the latter.

3. Creation of 3D shape by estimation of depth

There have been many attempts at modelling 3D face shape. Most of them, like the view based approach used in [8] involve complex computations. Simon Baker, author of the widely known reference [4] proposed an SVD method for the same in [1] using the shape recovery algorithm [2], which given the proper dataset, gives results in very little time.

3.1 Facial Feature Extraction

A lot of tools were tried for training and fitting AAMs. Since AAM fitting algorithms do not do face detection, they need a close initial shape to start from. For generic images that are not annotated, this is usually done by taking a face detector like the one based on Ramanan and Zhu's work [6]. Milbrow and Nicolis' STASM package [7] includes its own detector, which works only for frontal facing images, failing detection even when the face is turned 15 degrees sideways. Luca Vezzaro's Inverse Compositional AAMs package [10] could be trained with annotated public datasets of frontal face views, and also included a dataset of a single subject in other poses for reconstruction as shown in [1]. Unfortunately, the latter also worked only for images very similar to those in the dataset i.e. similar face size, features, lighting, contrast etc. There was no publicly available dataset which enabled fitting of AAMs on faces over the range of poses described in Section 3.2, and 3D reconstruction was only possible on images in the dataset or query set of [10].

3.2 Sparse 3D Mesh Formation

A 2D AAM is trained over a dataset of 91 images, which includes face images of a single subject in various poses, enabling fitting of 2D AAM to new face images of the subject. To recover 3D information, i.e. estimated depth of the features, there is a separate dataset of 390 images, but using just 35 images covering a wide range of angles (left-down, left-up, frontal, up etc.) from the 2D training set makes minimal difference to the depth created in the sparse 3D mesh, thereby eliminating the need for the secondary set. A sample mesh is shown in Figure 3.1

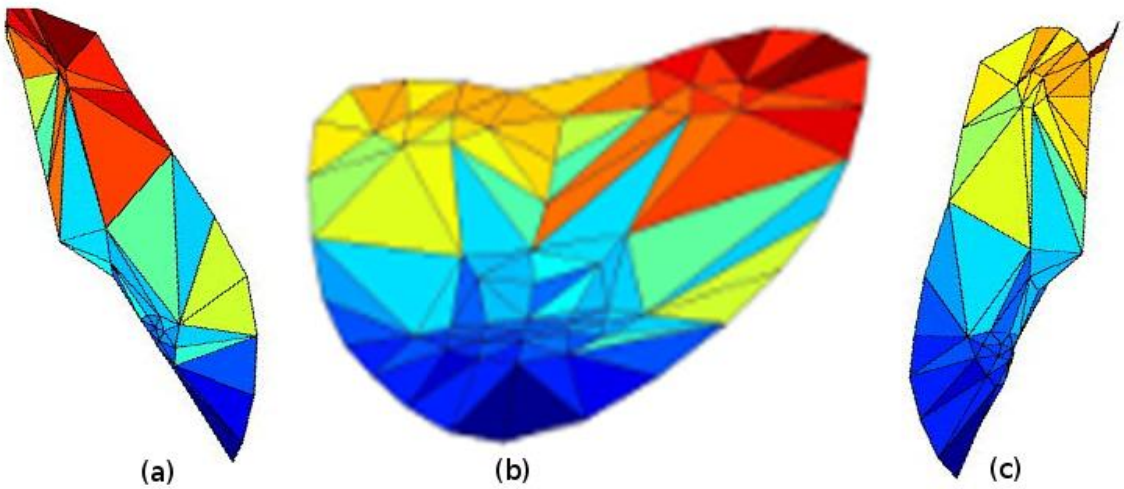

Figure 3.1: *Sparse 3D mesh shown in 3 angles, left-facing(a), front- facing(b) and right-facing(c)*

3.3 Interpolation and 3D Image Display

Cubic interpolation is done on the sparse 3D mesh to make it 'fuller'. The image is also interpolated and the 3D image is displayed as a surface with an indexed colourmap. These steps are explained in detail in Sections 4.4 and 4.5

3.4 Results

Result for one of the images is shown in Figure 3.2. It can be seen that the estimated depth of the features is very less, and the image seems fairly flat.

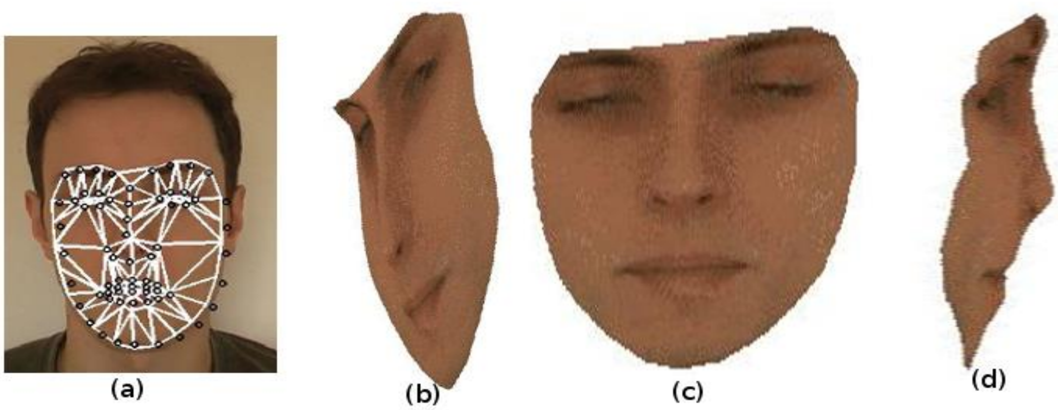

Figure 3.2: *Fitted 2D AAM(a) and generated 3D image from different angles (b),(c),(d).*

4. 3D Image from frontal and profile face images

A straightforward method for estimating the 3D shape of a face is described by Jingu Heo and Marios Savvides in [5]. It involves locating facial features on the frontal image, and estimating their depth from the profile image.

The steps in this process are

1. Extraction of facial features from the frontal and profile images
2. Scaling and rotation of the features to correspond to each other
3. Creation of a sparse 3D mesh by using a lookup table
4. Interpolation of the sparse mesh to create a ‘fuller’ mesh
5. Interpolation of the 2D image and generation of the 3D image

4.1 Facial Feature Extraction

Two shape models were used to extract the facial features of the frontal image. The first was the Zhu and Ramanan face detector based on [6]. This came trained with the MultiPIE dataset and labelled faces with 68 points. The second one was STASM, based on [7], which was trained on the MUCT dataset, which consisted of 3755 faces. The latter extracted 77 feature points from frontal images, and gave a more descriptive shape for the eyes, tip of the nose and the lips. The results are better if the subject looks straight at the camera, as that results in the most generic 3D model of a face.

Zhu and Ramanan face detector was used again for the profile face. When a purely profile (i.e. only one eye and ear visible) face was detected, 39 facial features were detected.

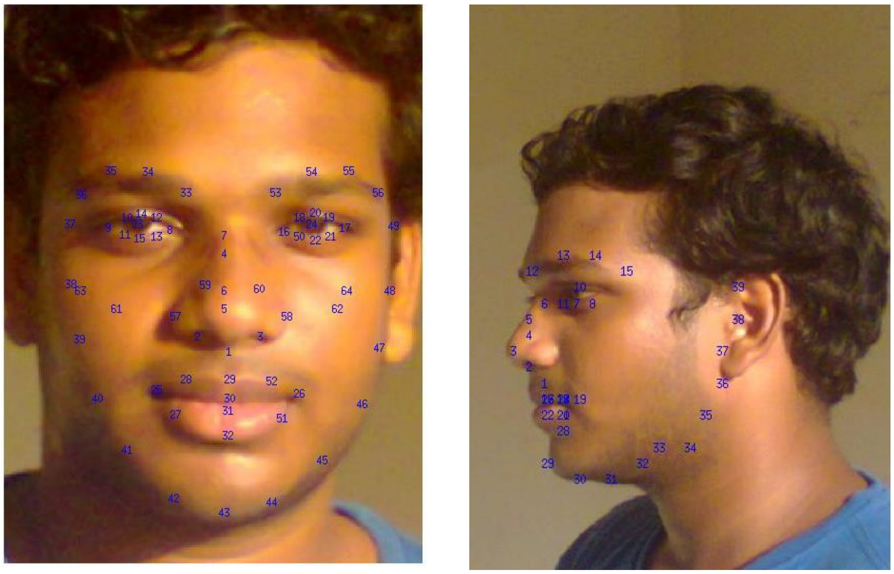

Figure 4.1: *The image on the left shows the final set of detected frontal features, while the one on the right shows the extracted profile face features*

4.2 Scaling and rotation of shapes

The shapes and corresponding images are rotated so that they are straight. This is done by using the following process

1. Rotating the frontal shape and image counter clockwise around the point 13 (center of right pupil) by the angle $\cot^{-1}(\frac{24.y-13.y}{24.x-13.x})$
2. Rotating the profile shape and image counter clockwise around the point 1 (nose tip) by the angle $\cot^{-1}(\frac{5.y-1.y}{5.x-1.x})$
3. Computing the Cartesian distances between points 1 and 4 of the frontal image, and points 1 and 5 of the profile image. i.e.

$$d_f = \sqrt{(4.x - 1.x)^2 + (4.y - 1.y)^2} \tag{4}$$

$$d_p = \sqrt{(5.x - 1.x)^2 + (5.y - 1.y)^2} \tag{5}$$

4. Scaling the profile shape using the multiplier $\frac{d_f}{d_p}$. This makes the profile shape's size correspond to the frontal shape's size

4.3 Creation of a sparse 3D mesh

A lookup table is created between corresponding points in the frontal shape and the profile shape based on depth. For example, points 53 and 43 in the frontal shape correspond best to points 12 and 29 respectively in the profile shape.

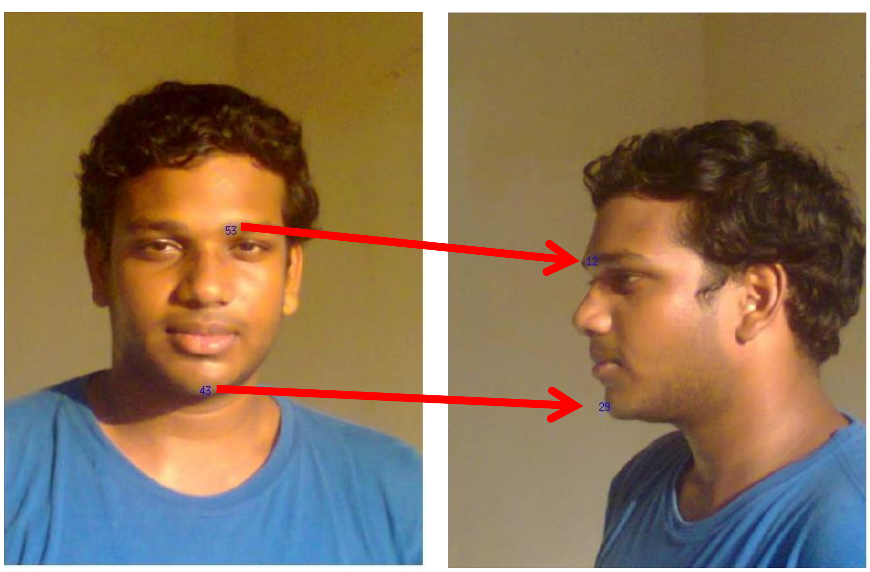

Figure 4.2*: Image showing correspondence of frontal and profile feature points*

The table is established for all the frontal points, with approximations being made when exact depth cannot be determined. Once this is done, a mx3 matrix shape3d is made to store the sparse 3D shape, with the first column storing the x-coordinate of the feature point, second storing the y-coordinate, and the third storing z-coordinate information. A sample mesh is shown in Figure 4.3

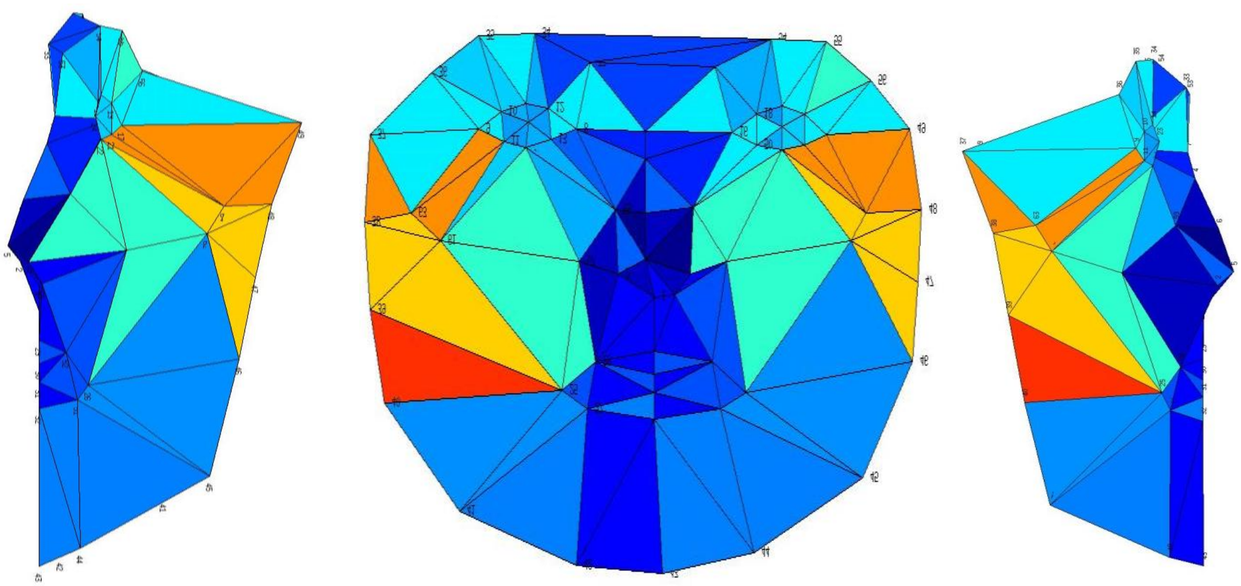

Figure 4.3 *Sparse 3D mesh generated, shown in left facing, front facing and right facing*

.

4.4 Interpolating the sparse 3D mesh

The sparse 3D mesh cannot be used to get a proper 3D image. It is interpolated over a mesh twice the size of the frontal face image to get a 'fuller' mesh. Cubic interpolation yielded the best results for facial structure. An example is shown in Figure 4.4

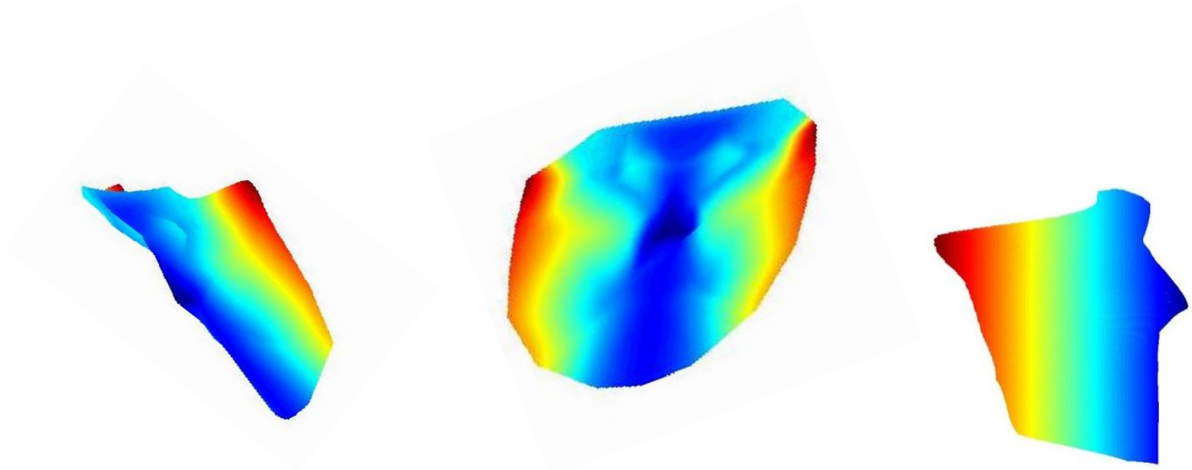

Figure 4.4 *Interpolated 3D mesh with a depth-based colourmap shown in left-facing, frontal and right-facing poses*

4.5 Interpolating the 2D frontal image and displaying the 3D image

RGB information was extracted from the frontal face image and stored in separate matrices. Since the values vary numerically and are independent of each other, we are free to interpolate the image to twice its size, i.e. if the old image matrices contained mxn elements, the new matrices will have $2mx2n$ elements. Treating this as a new image, we now have colour information for each location on the interpolated 3D mesh.

Now, we have the position of each pixel of the image, and the corresponding colour information stored in different matrices. To display this in MATLAB, the RGB image was converted to an indexed image. This was then used as a colourmap for a 3D surface object created using the interpolated mesh's information. A case is demonstrated in Figure 4.5

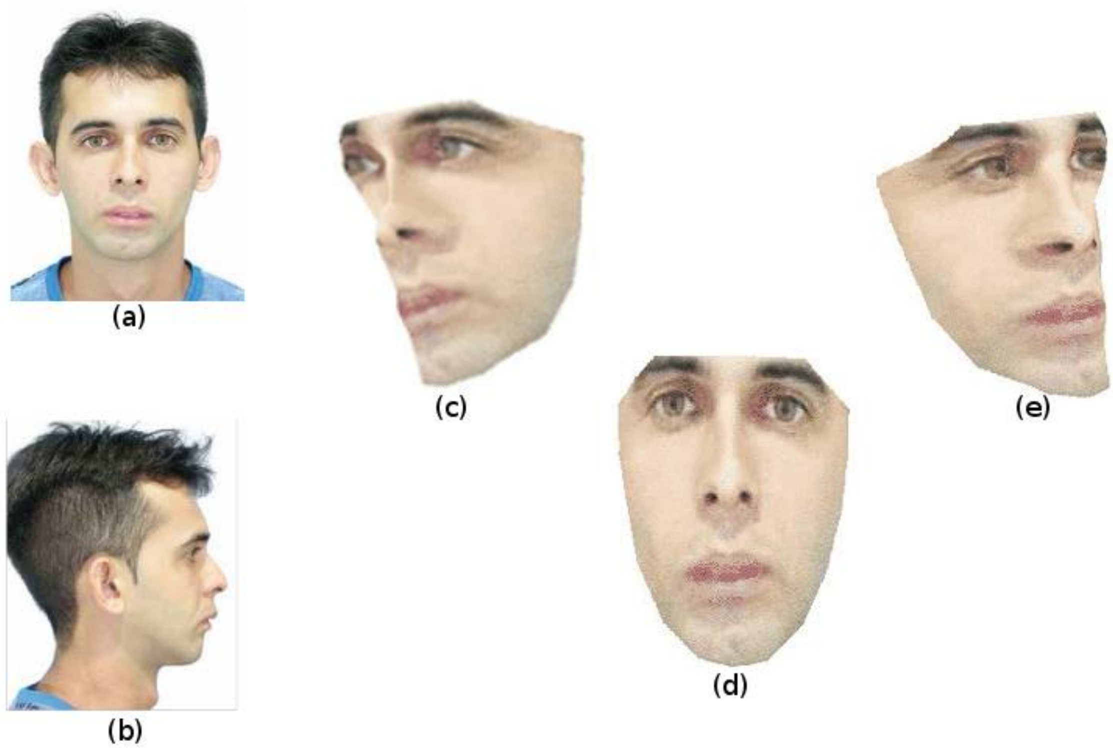

Figure 4.5 *3D face image generated from a frontal face(a) and profile face(b), shown in three angles, left-facing(c), frontal(d) and right-facing(e). All the faces are not displayed on the same scale.*

4.6 Results

The process was carried out on 20 subject's faces. It was found that the time taken by the Ramanan and Zhu face detector [6] increased with size of the image. It was also the same case for the algorithm described above, as degree of interpolation increases with image size. In order to get a size-independent estimate of time taken, all the images were resized to a 480x640 resolution. The average, minimum and maximum times for the process are shown in Table 4.1. Note that the total time is calculated for each case by addition and the extreme and average values need not match with the sum of other tasks' times.

Task	Time taken (s)		
	Min.	Avg.	Max.
Frontal face feature extraction – Zhu and Ramanan (Matlab)	7.21	12.13	16.24
Frontal face feature extraction – STASM (OpenCV in C++)	0.20	0.35	0.41
Profile face feature extraction – Zhu and Ramanan (Matlab)	8.26	12.41	15.63
Formation of 3D image from above data (Matlab)	2.12	4.63	6.41
Net process	22.33	31.46	39.21

Table 4.1 *Time taken for different tasks on 480x640 images*

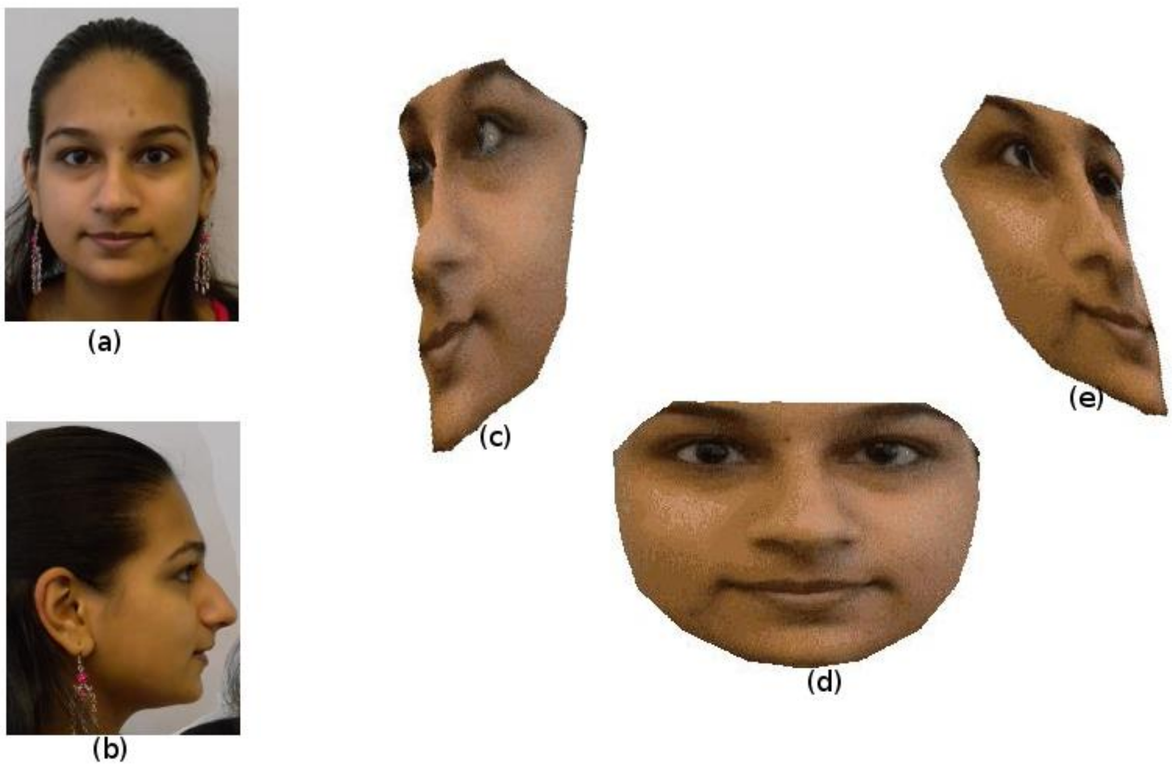

Figure 4.6 *The frontal face(a) and profile face(b) are used to create a 3D image, shown in three angles, left-facing(c), frontal(d) and right-facing(e). All the faces are not shown on the same scale.*

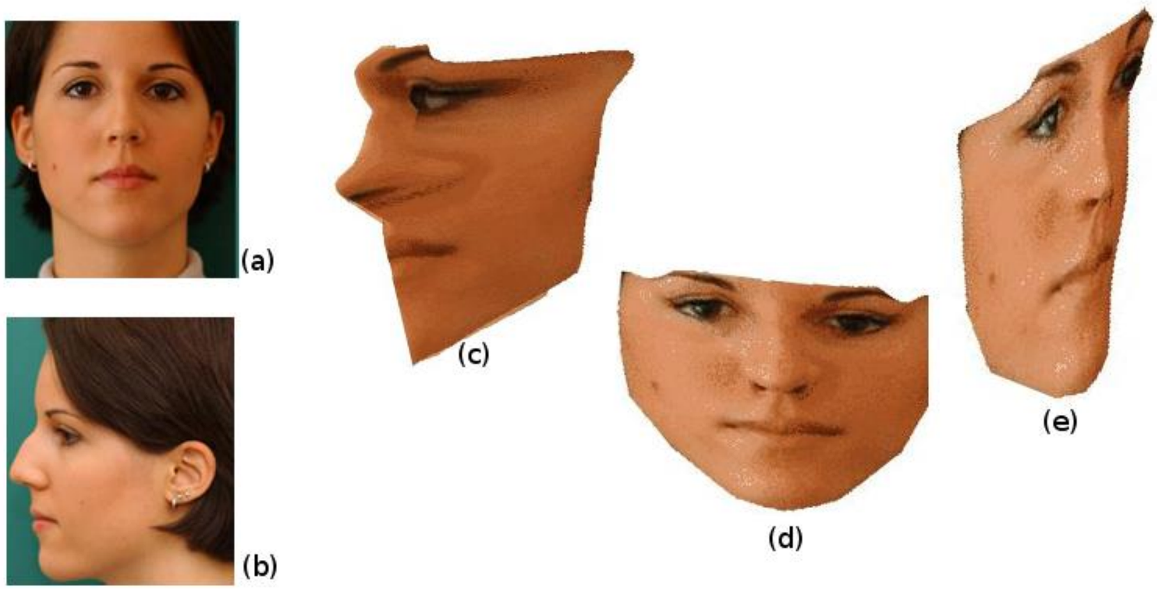

Figure 4.7 *The frontal(a) and profile(b) faces are used to create a 3D image, shown in three angles, left-facing(c), frontal(d) and right-facing(e). All the faces are not displayed on the same scale.*

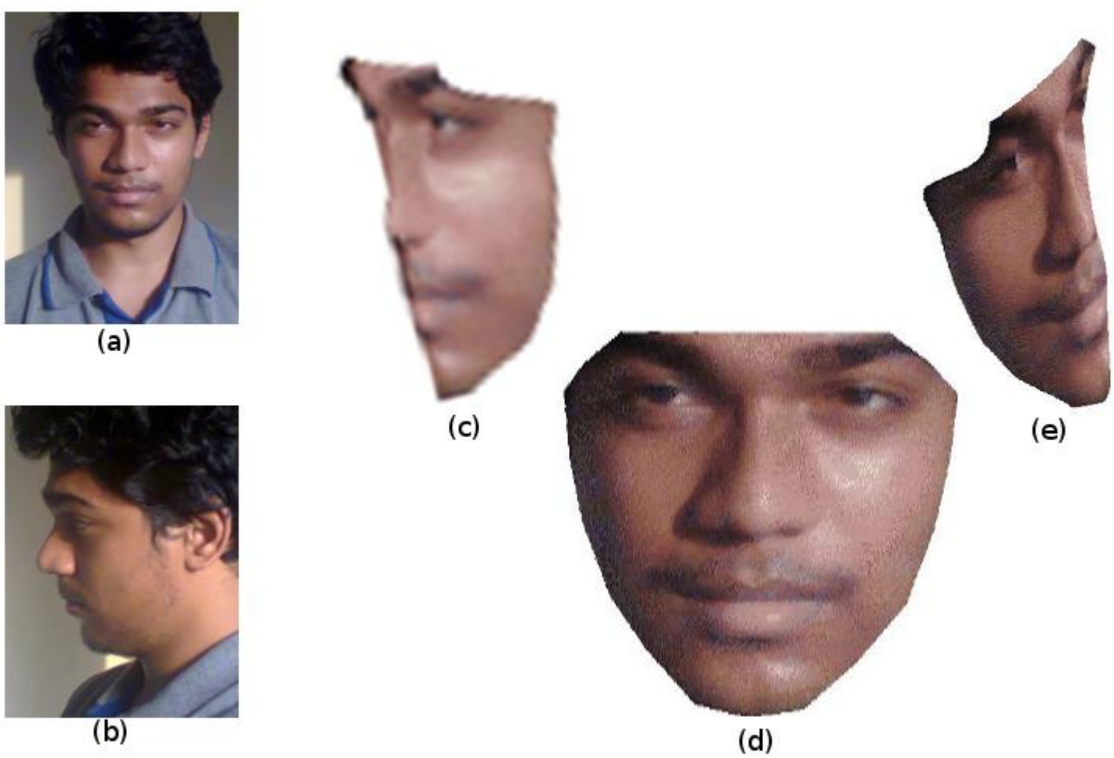

Figure 4.8 *Poses of 3D face image (c), (d) and (e) made from images (a) and (b). All the faces are not shown on the same scale.*

Some results are shown in Figures 4.6, 4.7 and 4.8. It is to be noted that the colours of the frontal image are mapped on the 3D image, and differences in lighting will be reflected in all poses of the 3D image. Normalisation cannot be generalised, as asymmetrical facial features such as the mole on the left side of the subject in Figure 4.7 will be lost or minimised. The best models are made by taking the image in even light, as shown in Figures 4.5 and 4.6.

The face is ‘editable’ as points on the sparse 3D mesh can be adjusted in z-direction to alter their depth. Interpolation upon the new mesh will yield the desired 3D image.

5 Conclusion

Attempts were made to convert 2D face pictures to 3D images. Two methods were attempted, estimating 3D depth from motion and calculating depth from a profile image. The former did not work for images significantly different from the provided dataset, and provided fairly flat images. The latter provided significantly better images, and the whole process took under a minute even when most of the execution was in Matlab.

6 Future work

3D images have been generated for most of the face. Other aspects of the face like the ears and hair are yet to be modelled, and would give a complete 3D representation of a person's head.

Selective interpolation can be done between specific points on the mesh and the image, such that the facial features are editable across x, y and z directions. This will allow manipulation of features like nose tip width, lip width etc. rather than simply changing protrusion.

REFERENCES

[1] **Jing Xiao, Simon Baker, Iain Matthews** and **Takeo Kanade**, Real-Time Combined 2D+3D Active Appearance Models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June, 2004, pp. 535 - 542.*

[2] **Jing Xiao, Jinxiang Chai** and **Takeo Kanade**, A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. *The 8th European Conference on Computer Vision (ECCV 2004), May, 2004.*

[3] **Myung-HoJu** and **Hang-Bong Kang**, A Fast 3D-AAM Method using the Estimated Depth Information. *Progress in Informatics and Computing (PIC), 2010*

[4] **Iain Matthews** and **Simon Baker**, Active Appearance Models revisited. *International Journal of Computer Vision, Vol. 60, No. 2, November, 2004, pp. 135 - 164.*

[5] **Jingu Heo** and **Marios Savvides**, Rapid 3D Face Modelling using a Frontal Face and a Profile Face for Accurate 2D Pose Synthesis. *Ninth IEEE International Conference on Automatic Face and Gesture Recognition (FG 2011)*

[6] **X. Zhu** and **D. Ramanan**, Face detection, pose estimation and landmark localization in the wild. *Computer Vision and Pattern Recognition (CVPR) Providence, Rhode Island, June 2012.*

[7] **S. Milborrow** and **F. Nicolls**, Active Shape Models with SIFT Descriptors and MARS. *The IEEE International Conference on Computer Vision (ICCV) Workshops, 2013, pp. 378-385*

[8] **J. Sung** and **D. Kim**, Pose-Robust Facial Expression Recognition Using View-Based 2D+3D AAM. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*

[9] **M. Brown** and **D. G. Lowe**, Automatic Panoramic Image Stitching using Invariant Features. International Journal of Computer Vision, vol. 74, issue 1, pp. 59-73, 2007.

[10] **Luca Vezzaro,** "ICAAM - Inverse compositional Active Appearance Models" based on *Active Appearance Models - Aspetti Metodologici e Casi di Studio, Universit degli Studi di Verona*