USING MORPHABLE FACE MODEL TO IMPROVE STEREO RECONSTRUCTION AND VISUALISING THE MODEL ON A SMARTPHONE

A DISSERTATION

submitted in partial fulfillment of the
requirements for the award of the degree
of

MASTER OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

(With specialization in Instrumentation & Signal Processing)

By

HARDIK JAIN

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247667 (INDIA)
MAY 2016

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

CANDIDATE'S DECLARATION

I hereby declare that this thesis which is being presented as the final evaluation of the dissertation **Using Morphable Face Model to Improve Stereo Reconstruction and Visualising the Model on a Smartphone** in partial fulfillment of the requirement of award of Degree ”Master of Technology” in Electrical Engineering with specialization in Instrumentation and Signal Processing, submitted to the Department of Electrical Engineering, Indian Institute of Technology Roorkee, India is an authentic record of the work carried out during a period from May 2015 to May 2016 under the joint supervision of **Prof. Dr. R. S. Anand** (IIT Roorkee, India) and **Prof. Dr.-Ing. Olaf Hellwich** (TU Berlin, Germany).
The matter presented in this report has not been submitted by me for the award of any other degree of this institute or any other institute.

Date:
Place: (Hardik Jain)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to best of my knowledge.

Prof. Dr. R. S. Anand
Professor
Dept. of Electrical Engineering
Indian Institute of Technology
Roorkee, India

Prof. Dr.-Ing. Olaf Hellwich
Professor
Computer Vision and Remote Sensing
Technical University Berlin
Germany

"It doesn't matter where you come from
what matters is who you choose to be."

Papa Smurf

Acknowledgements

This dissertation was an integral part of me for the last few months. I want to take this opportunity to thank all of them without whom this thesis could not be finished, though this short acknowledgement is not enough to match their favors.

I am greatly indebted to my supervisors Prof. Dr. R. S. Anand and Prof. Dr.-Ing. Olaf Hellwich (Computer Vision and Remote Sensing, TU-Berlin, Germany) for giving this wonderful opportunity to work under their joint supervision. The encouragement that I got from Prof. Anand boosted my confidence and enthusiasm. Prof. Hellwich was a great source of inspiration for me. The conversations with him always used to inject a lot of positive vibes within me.

My sincere gratitude goes to all the faculty members of Instrumentation and Signal Processing Group, Department of Electrical Engineering, IIT Roorkee for their encouragements and help.

I would like to thank DAAD (Deutsch Akademischen Austausch Dienst) for funding my research in Germany. This project was only possible because of DAAD. I am extremely happy that I was able to work in Computer Vision and Remote Sensing group of TU-Berlin in such a friendly yet sincere environment; I thank all my colleagues and friends back there. I would also like to thank Carl Martin Grewe at Zuse Institute Berlin for the High Quality Face Scan.

I thank TU-Berlin International Relations Office (Akademisches Auslandsamt), who made my stay in Berlin a lot more smoother and enjoyable than what it could turn out to be without them.

I feel fortunate to get my family and friends by my side, who were always a great source of motivation for me. My parents were always great pillars of strength who have always encouraged me to reach for my dreams, and I want to dedicate this work to them.

Begging pardon of all those well-wishers whose name could not be taken.

HARDIK JAIN

ABSTRACT

Human faces are similar in global properties, including location of main features, size, aspect ratio, but can vary considerably in details across individuals gender, race or facial expression. Due to the loss of one dimension in the image acquisition process, the retrieval of the true 3D geometry is difficult and a so called ill-posed problem, 3D reconstruction from images. Stereo Reconstruction from image pair is a standard method for 3D acquisition of human faces. Depending on available imagery and accuracy requirements the resulting 3D stereo reconstructions may have deficits. The stereo surface reconstruction has lots of holes, and limited texture information. In this work we remedy such deficits combining the 3D stereo reconstruction with a generic Morphable Face Model. The 3D morphable face model has smooth shape information which can be modified specific to a face image. For the improved reconstruction, prior shape information can be obtained by already developed methods, which uses landmarks to fit a morphable model to a single image. A Major part of the thesis is devoted to improvement in stereo face reconstruction pipeline by allowing to prefer information from the single image reconstruction whenever the stereo reconstruction shows untypical deviations from the expected 3D features of a human face. From a pair of stereo images, one is used for single image reconstruction and the combination gives the stereo model. The two reconstruction are then combined to result in the deformed face model. The fusion of models is conducted in a global and a local transformation stages. To include high frequency color information in the model, texture is extracted from the face image. A comparison of the output with high quality face scan is also presented. The fusion outcome results in more accurate face reconstruction than the either of the two. Finally, the resultant deformed face model is visually presented on a smartphone using cardboard, which addresses the modern trend of low cost devices in virtual 3D visualization.

Contents

List of Figures

Abbreviations

AAM	**A**ctive **A**ppearance **M**odel
3DMM	**3 D**imensional **M**orphable **M**odel
BFM	**B**asel **F**ace **M**odel
CAVE	**C**ave **A**utomatic **V**irtual **E**nvironment
HMD	**H**ead **M**ounted **D**isplay
VR	**V**irtual **R**eality
SDK	**S**oftware **D**evelopment **K**it
SFM	**S**urrey **F**ace **M**odel
PCA	**P**rincipal **C**omponent **A**nalysis
IMDR	**I**terative **M**ulti-resolution **D**ense 3D **R**egistration
RBF	**R**adial **B**asis **F**unction
AR	**A**ugmented **R**eality
APK	Android Application Package
Open GL ES	**O**pen **G**raphics **L**ibrary **E**mbedded **S**ystem
API	**A**pplication **P**ackage **I**nterface
MMOD	**M**ax **M**argin **O**bject **D**etection
HOG	**H**istogram of **O**riented **G**radient
CCD	**C**harge **C**oupled **D**evice
MPO	**M**ulti **P**icture **O**bject
ICP	**I**terative **C**losest **P**oint

Chapter 1

Introduction

Researchers have been investigating methods to acquire 3D information from objects and scenes for many years. Earlier the main application of 3D information was visual inspection. Nowadays the emphasis is shifting, there is more demand of 3D content in computer graphics, virtual reality and communication. A lot of significance to the visual quality is given.

An image is a description of a scene or an object which has been captured. Due to the nature of image formation process, depth information is lost. The three-dimensional point corresponding to a specific two-dimensional image point is constraint to be on the associated line of sight. From a single image in two-dimensional plane it is not possible to determine which point of this line corresponds to the image point. One image is however not enough to reconstruct the 3D scene.

Due to the loss of one dimension in the projection process, the estimation of the true 3D geometry is difficult and a so called ill-posed problem, because usually infinitely many different 3D surfaces may produce the same set of images. So the computer aided object modeling is an area which still requires a great deal of expertise and various manual instructions to get valuable information.

In Computer Vision quite often 3D data is generated, often after a 3D reconstruction process, the data has to be visually presented – for instance in order to check its validity. Techniques of visual presentation of 3D data have advanced a lot recently. Even smartphones being omnipresent in everyday life are used for 3D representation most spectacularly as a head-mounted display.

1.1 Literature Survey

In Stereo Vision, two cameras which are displaced horizontally from one another are used to capture two views of a scene. In a manner similar to human binocular vision. Upon comparing these two images, the lost depth information can be recovered. For the purpose of 3D reconstruction from stereo pair, given two projections of the same point in the world onto the two images, its 3D position can be found as the intersection of the two projection rays. This process can be repeated for several points yielding the shape and configuration of the objects in the scene [1]. Stereo Vision can be used with 3D Pattern Matching and Object Tracking and is therefore used in applications such as bin picking, surveillance, robotics, and inspection of object surfaces, height, shape, etc.

Stereo reconstruction from image pairs is a standard method for 3D acquisition of human faces. Stereo vision has a couple of synchronized camera with known parameters and fixed mutual positions. Depending on available imagery and accuracy requirements the resulting 3D reconstructions may have deficits. The stereo image pair needs to be acquired with a relatively short baseline length in order to avoid occlusions [2] e.g. When capturing the human face if the two stereo cameras are far away the person's nose would occlude some portion of face in both the images, resulting in steep intersection angles of the two view rays limiting the accuracy of depth measurement. Furthermore, in some areas of the face the view rays are relatively tangential to the surface making surface reconstruction more difficult. Another problem, in particular for imagery lacking spatial resolution, is the lack of texture in some parts of faces. The unavoidable consequence of these problems is the limited accuracy of surface reconstruction.[1]

The deficits of 3D stereo reconstruction can be alleviated by the use of prior knowledge. In principle as well as from a historic viewpoint the use of prior knowledge in shape reconstruction has a long history in craftsmanship and engineering. One can even claim that the more limited measurement devices were in previous periods of technological

[1] Depending on defined demands this is certainly true for any measurement process.

development, the more prominent the use of prior knowledge was. Only with the advent of mass measurement devices such as cameras and laser scanners the use of prior knowledge disregarded to larger extent.[2]

In order to be generally usable the type of prior knowledge to be used should be rather generic and not particularly object specific. In our example, prior knowledge about face geometry will be provided by the same 3D model to be used for any person's face to be reconstructed. In the processing steps of the procedure the 3D model will first be adapted to better represent one of the two images of a stereo pair.

The 3D model used for reconstruction is referred to as 3D Morphable Model of face. Based on input image the model parameters are modified to fit the face image [3], the technique is referred to as face modeling. When discussing about the prior knowledge, the major limitation of automated technique for face modeling from single image is either the problem of locating features in faces or the dilemma of separating realistic from non real faces, which could never exists [4].

For the first problem, the feature location (50 to 100 points) when done manually usually requires hours and accuracy of an expert. Human perception about faces is necessary to compensate for variations between different faces and to ensure a valid feature point assignment. Till date, automated feature point matching algorithms are available for salient features like the corners of lips or the tip of nose.

In the second problem of face modeling, the human knowledge is more significant for separation of natural (real) faces from unnatural looking faces and avoid non realistic face results. Some application of face modeling even involves the design of completely new natural looking face, which can occur in real world but has no actual counterpart. Others require the handling of existing face in accordance with changes in age, sex, body weight and race or to simply enhance the characteristics of the face. These tasks might require plenty of time with combined efforts of an artist. The problem of non face like structures can be solved by creating a model from real faces, which has scan of real world human faces of different age, sex and race.

One of the tasks required in face analysis is to perform face reconstruction from input images. Given a single face input image, a generalized face model is commonly used to recover shape and texture via a *fitting process*. However, fitting the model to the

[2] For instance in medieval time periods the 3D geometry of a cubical wooden box would have been acquired by the three distance measurements of height, length and depth plus the prior knowledge that the object of interest has a cubical shape, and not by thousands of 3D point measurements.

facial image remains a challenging problem. Many models have been proposed for this purpose. The face models are a powerful tool of computer vision and are classified in two groups: 2D face models and 3D face models.

The Active Appearance Model (AAM) by Cootes et al. [5] belong to the 2D group. It models the shape and texture variations statistically. The authors in [6] when observed the relationship between the resolution of the input image and the model, concluded that the best fitting performance is obtained when the input image resolution is slightly lower than the 2D model resolution.

Although AAMs exhibit promising face analysis performance, there is still a problem that the reconstruction with AAMs fails if the in-depth rotation of the face becomes large. Almost all the 2D-based models suffer from the same problem.

The 3D model has distinctive advantage over the 2D, in Active Appearance Models the correlations between texture and shape are learned to generate a combined appearance model. Where as in a 3D model, the shape of a face is clearly separated from the pose. Its projection to the 2D is modeled by affine or perspective camera model. Also, the use of a 3D face model allows us to model the light explicitly since the surface normals, depth and self-occlusion information are available. The illumination model separates light from the face appearance and is not incorporated with the texture parameters, as it is for the case in 2D AAMs. The main advantage of 3D based models is that 3D shape does not change under different viewpoint and so are more robust than their 2D counter part.

The generative model in this study focuses on 3D Morphable Model (3DMM) which were first proposed by Blanz and Vetter [4]. In 2009, Basel Face Model (BFM) [7] was introduced which spurred the research with 3D Morphable Models. It surpassed the existing models due to the accuracy of 3D scanners used and the quality of registration algorithms. The multi segment BFM, along with some fitting results and metadata can be obtained by signing a license agreement. This led to its implementation in face recognition in video [8], and Linear Error function based Face identification by fitting a 3DMM [9]. While the BFM provides the model, they only provide fitting result for limited database and do not provide algorithms to implement the model to new images [10]. This restricted its application to a closed domain.

The 3DMM attempts to recover the 3D face shape which has been lost through projection from 3D into 2D image. Given a single facial image, a 3DMM can retrieve both intra-personal (pose, illumination) and inter-personal (3D shape, texture) via a fitting algorithm. Furthermore, a 3DMM can be used in a productive way to create specific faces or to generate annotated training data sets for other algorithms that covers a variety of pose angles. Thus the 3DMM is crucial for variety of applications in Computer Vision and Graphics. The application of 3DMMs include face recognition, 3D face reconstruction, face tracking.

The so-called fitting algorithms that solve these cost functions are often very complex, slow, and are easily trapped in local minima. These methods can be roughly classified into two categories: first ones with linear cost functions and those with non-linear. The algorithms falling in first category generally use only prominent facial landmarks like eye or mouth corners to fit the shape and camera parameters, and use image pixel value to fit color and light model [9]. The algorithms of later category consist of more complicated cost functions applying a non-linear solver to iteratively solve for all the parameters [8, 11, 12].

The use of 3D Visualization is standard today, but Visualization devices are still controlled by standard PC and screens. Different low-cost-systems for 3D Visualization are present as inexpensive alternatives to complex virtual reality systems such as a CAVE or a 3D power wall. Low-cost-systems for the visualization (display and control) are defined by the cost of the hardware not exceeding smartphones for ₹15000 (€200).

Low-cost-system components such as mobile phone can be used for the stereoscopic display of the objects. A smartphone can be used to visually inspect results of image-based 3D reconstruction. An application (app) for an Android smartphone allowing to view 3D point clouds could be best suited. By using a smart phone app, the device becomes a Head Mounted Display (HMD) to create an even more immersive exploration of the data. The inertial sensors of the phone can be used for the tracking of the head, Since the interactive visualization should allow free movement or navigation of the user in the 3D model. Necessary input commands must be captured and processed using appropriate hardware, which must be sent as input data to the Visualization software. The navigation parameters are continuous changes of the camera position and orientation, which are defined interactively by the user.

1.2 Motivation and Objectives of the Dissertation

3D reconstruction is an ill-posed problem, various modern algorithm have been developed to achieve this. Majority of these methods either involves multiple images as input or a single image with some object information. we would be dealing in the combination of two methods for human face and the accomplished object is the geometrically rich deformed face model.

3D reconstruction of object based on measurement data from e.g.imaging sensors would quite generally profit from the use of prior information about the object's shape. As the stereo reconstruction poses few deficits in the reconstruction, the prior knowledge in the form of single image reconstruction could be used.

The use of morphable model as the basis for 3D reconstruction was introduced by Blanz and vetter and the majority of literature regarding the morphable face models is based on their work. Recently, 3DMMs provided by University of Surrey have been used with regression-based methods [3] which is a leap forward in the direction of face modeling. No significant improvement in the face modeling has been brought in it since then.

Automation in the process of single image face reconstruction has not been still achieved because of manual landmark annotations or premarked landmark coordinates being used. This gap has also been bridged in this work, by using the regression tree method [13] by V. Kazemi et al. This method identifies landmark positions of faces like tip of eyes, eyebrows, lips and Nose; which are used for fitting the morphable model to the single face image.

The Morphable model has an advantage of giving 3D structure from a single image, but it doesn't conforms to the actual congruous of the face. The reason being it is based on few landmark positions only. A geometrically more definite description of the face is stereo reconstruction. Deformation, which is one of the crucial step of this pipeline and obviously some of the most difficult procedure has to be carried out on the face model, so as to result in a deformed face model. Thus concluding model holds the best of face model and stereo model.

Cardboard has brought a revolution in the Virtual Reality and its depended applications in healthcare, entertainment and scientific visualization. An economical head mounted cardboard has changed the way 3D objects can be anticipated on a screen. Various well known companies have developed there own Virtual Reality (VR) visualisers like

Samsung Gear VR, Carl Zeiss VR One, hTC Vive. This development in the hardware has also made the software development or more specifically VR android app development much easier by providing open source Software Development Kit (SDK)[3] . This SDK when customized, served our purpose of 3D face surface visualization with texture information as well.

Conclusively, the objectives of this dissertation work include:

- Automatic 68 Landmark Point Detection on a face Image.
- Shape Fitting of the 3DMM from Image.
- Face Texture synthesis.
- Stereo Reconstruction from two face images.
- Fusion of face model and stereo model.
- Visualization of deformed face model on Smart phone using cardboard.

1.3 Organization of Report

This report is organized on a purpose basis. The current chapter introduces the theme of the work along with the literature survey of the topic. Chapter 2 describes the technical background required for accomplishing various tasks, along with the overview of fusion and deformation are mentioned in it. The proposed reconstruction pipeline is described in chapter 3, which includes automatic landmarking, procedure of single image reconstruction and stereo reconstruction. It also includes the deformation of face model and android application development. Results of the deformation along with its comparison to the High Quality Scan Model is mentioned in Chapter 4. Finally the last chapter presents the conclusions and scope for future work. The publication derived from the work and bibliography are mentioned at the end.

[3] `https://developers.google.com/cardboard/overview`

Chapter 2

Background

For a better understanding of the basic concepts used in this work, some prerequisites are discussed in this chapter. The first one is Facial Point Annotation which is used for landmarking on face image. Next section introduces the 3D morphable model with its composition information. Texture representation uses isomap algorithm, details of which are discussed in third section. Another reconstruction technique i.e.stereo reconstruction is included next. Later section includes the Deformation of Face Model which is an integral part of this thesis. The last section discusses the Visualization of the resultant model on smartphone.

2.1 Facial Point Annotation

Detecting facial features is the problem of detecting semantic facial points in an image such as tip of lips, eyes, mouth and boundary of face. Apart from its application in Face reconstruction from a 3DMM, facial feature detection concerns various other research applications which include **Face Recognition** of pre-learnt faces in unseen images, **Face Hallucination** (synthesizing a high-resolution facial image from a lower resolution), **Face Animation** (real-time animation of a graphical avatars expression based on real facial expression).

Facial point annotation can be achieved by variety of algorithms. One of these, Cascade based regression methods by V. Kazemi et al. uses an Ensemble of Regression Trees [13], which can be used to estimate face's landmark positions directly from sparse subset of pixel intensities. These methods iteratively improve the estimate by working in a stage-by-stage cascade. Our Facial annotation method is based on its implementation in dlib c++ Library[1].

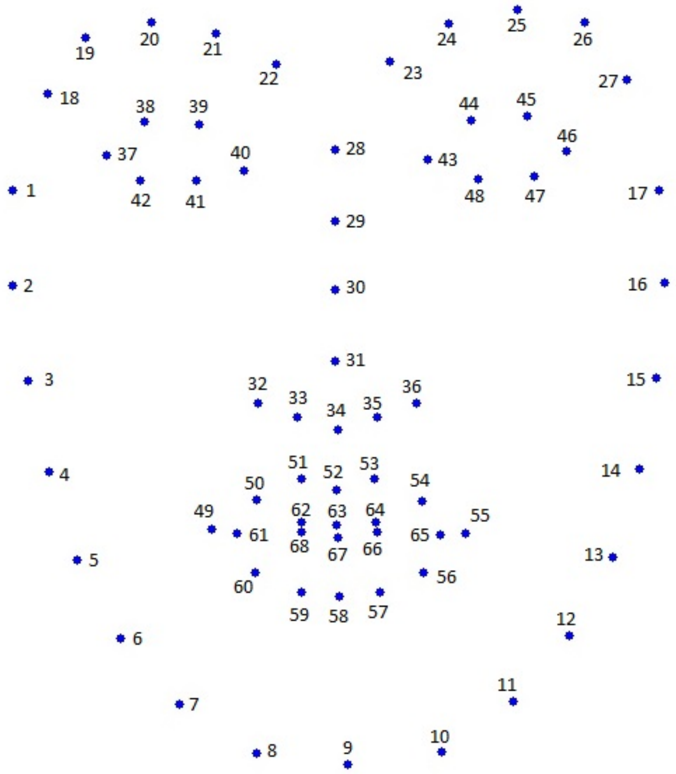


FIGURE 2.1: 68 points mark-up used for face annotations

The face alignment problem is solved with a cascade of regression function. Each regression function in the cascade efficiently evaluates the feature from an initial estimate and the intensities of set of pixels indexed relative to this initial estimate.

Two key elements that are incorporated into the learnt regression function, which are also present in several successful algorithms. The first revolves around the indexing of pixel intensities relative to the current estimate of the feature. The extracted features in the vector representation of a face can greatly differ due to both illumination condition and shape deformation. This makes the estimation process difficult.

The second considers the difficulty of solving the inference or prediction problem, as the problem is non-convex with many local minimas. The method of Ensemble of regression trees by V. Kazemi addresses both the issues, by incorporating them in their learnt regression functions.

[1] `http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html`

The facial feature points are typically represented in two dimensional system as a feature vector $\mathbf{F} = (x_1\ y_1, \ldots, x_p\ y_p)^T \in \mathrm{R}^{2p}$. Where each pair $(x_i\ y_i)$ is the coordinate of i-th facial landmark in a face image I and p is the number of landmarks ranging from 10 to 68 (or more), depending upon the application. The objective of facial point annotation is to produce such a face feature vector $\mathbf{F}$. Figure 2.1 shows 68 points mark-up which is used for annotation of faces.

2.1.1 The Cascade of Regressors

Let $\hat{\mathbf{F}}^{(t)}$ denote the current estimate of facial vector $\mathbf{F}$. Each regressor $r_t(\cdot,\cdot)$, in the cascade predicts an update vector from the image and $\hat{\mathbf{F}}^{(t)}$, which is added to current shape estimate $\hat{\mathbf{F}}^{(t)}$ according to the equation 2.1 to improve the estimate:

$$\hat{\mathbf{F}}^{(t+1)} = \hat{\mathbf{F}}^{(t)} + r_t(I, \hat{\mathbf{F}}^{(t)}) \tag{2.1}$$

The analytical point in this method of cascade is that, the regressor r_t makes its prediction based on features, such as the pixel intensities values, computed from I and indexed relative to the current shape estimate $\hat{\mathbf{F}}^{(t)}$. Geometric invariance is introduced into the process because of the above.

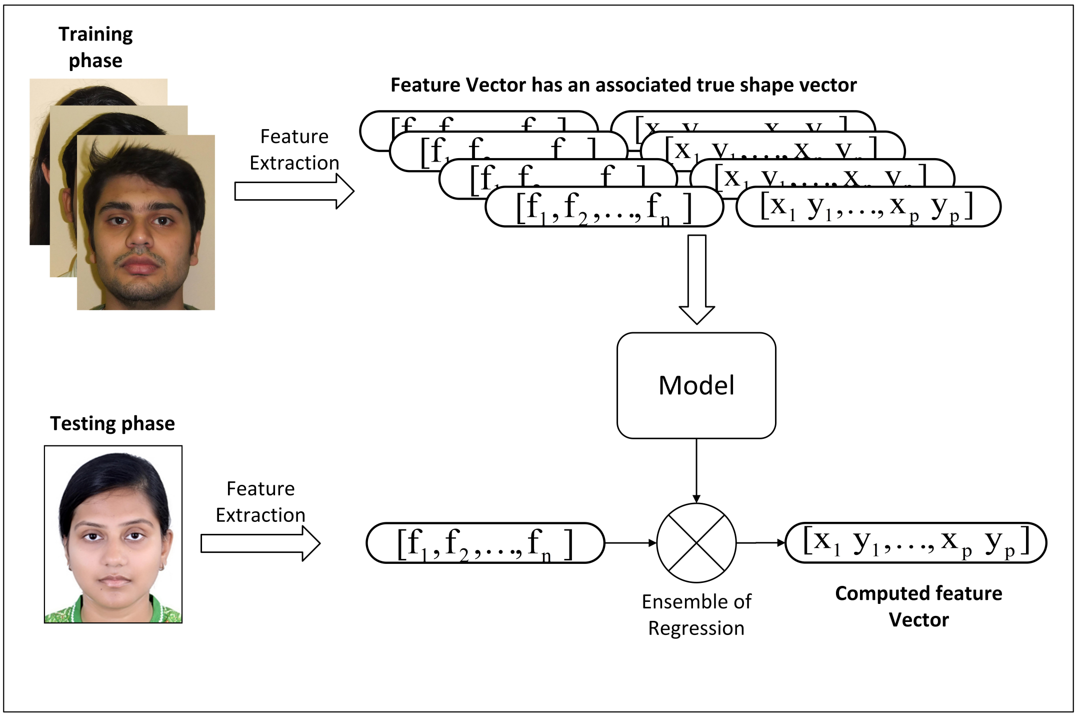


FIGURE 2.2: Training and Testing phase in context with facial annotation

2.1.2 Learning each Regressor in Cascade

For training each r_t, the gradient tree boosting algorithm with a sum of square error loss is implemented [13]. Assume the training data $(I_1, \mathbf{F}_1), \ldots, (I_n, \mathbf{F}_n)$, where each I_q is a face image and $\mathbf{F}_q$ is its conforming feature vector. The training data is created in triplets to learn the first regression function r_0 in the cascade. The triplet is made up of face image, an initial shape estimate and the target update step, *i.e.* $(I_{\pi q}, \hat{\mathbf{F}}_q^{(0)}, \Delta\mathbf{F}_q^{(0)})$ where:

$$\pi_q \in \{1, \ldots, n\} \tag{2.2}$$

$$\hat{\mathbf{F}}_q^{(0)} \in \{\mathbf{F}_1, \ldots, \mathbf{F}_q\}/\mathbf{F}_{\pi q} \tag{2.3}$$

$$\Delta\mathbf{F}_q^{(0)} = \mathbf{F}_{\pi q} - \hat{\mathbf{F}}_q^{(0)} \tag{2.4}$$

for $q = 1, \ldots, Q$. the total of $Q(= nR)$ triplets are taken, where R is the number of initialization used per image I_q.

From this data the regression function r_o is learnt, using gradient tree boosting with a sum of square error loss. The training triplet is then updated to provide the new training data $(I_{\pi q}, \hat{\mathbf{F}}_q^{(1)}, \Delta\mathbf{F}_q^{(1)})$, for the next regressor r_1 in the cascade by setting $(t = 0)$ in Equation 2.1

$$\hat{\mathbf{F}}_q^{(1)} = \hat{\mathbf{F}}_q^{(0)} + r_t(I, \hat{\mathbf{F}}_q^{(0)}) \tag{2.5}$$

$$\Delta\mathbf{F}_q^{(t+1)} = \mathbf{F}_{\pi q} - \hat{\mathbf{F}}_q^{(t+1)} \tag{2.6}$$

This process is iterated till a cascade of T regressors $r_0, r_1, \ldots, r_{T-1}$ are learnt which can be combined to give satisfactory level of accuracy. Figure 2.2 shows the training and test phases of landmarking algorithm.

2.2 The Surrey Face Model

The Surrey Face Model (SFM) consists of shape and color (or so-called albedo) Principal Component Analysis (PCA) models. PCA is a way of identifying patterns in data, and expressing the data in a way to highlight their similarities and differences. The identified pattern can be used to compress the data by reducing the number of dimensions, without much loss of information.

SFM is available in three different resolution levels. In addition to the high resolution models accessible via the University of Surrey[2], an open source low-resolution shape-only model is freely available on Github[3], which has been used in this work. The following sections would describe the model in detail.

2.2.1 Construction of a Face Model

The first step in the generation of model involves, collection of suitable database of 3D face scans that include shape and texture. For the robustness of the model it is essential to be a representative of the high inter-person variability. The recorded subjects in SFM have a diversity in skin tone and face shape to well represent multicultural make up of many modern societies. The ideal 3D face scans only capture the intrinsic facial characteristics, removing hair occlusions, makeup, or facial expressions and other extraneous factors; since these are not intrinsic to the shape or texture of the face. Figure 2.3 shows the racial distribution of all 169 faces used in model construction. Unlike BFM, the Surrey Face model has included significant number of non-Caucasian people to generalize the model [10]. The pie-chart in Figure 2.4 shows the age groups of the face scans, its evident that the SFM contains more people from 20+ age category.

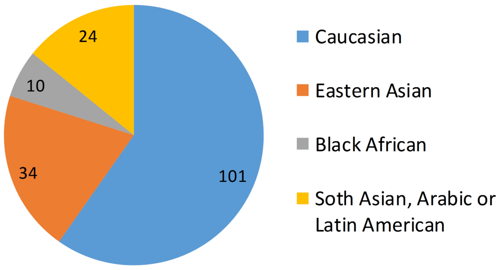


FIGURE 2.3: Racial distribution of 169 scans used to train the Surrey Face Model.

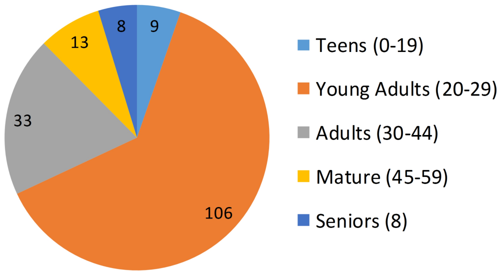


FIGURE 2.4: Age groups distribution of the 169 scans.

3dMDface[4] camera system was used to capture these scans. The system consists of two structured light projectors, 4 infrared camera and 2 RGB cameras. The infrared camera captures the light pattern and are used to reconstruct the 3D shape. The high resolution face texture is recorded by RGB cameras. One half of the cameras record 180° view of the face from left side and other half from the right side. Uniform lighting condition

[2] `http://www.cvssp.org/faceweb/`
[3] `https://github.com/patrikhuber/eos/blob/master/share/sfm_shape_3448.bin/`
[4] `http://www/3dmd.com`

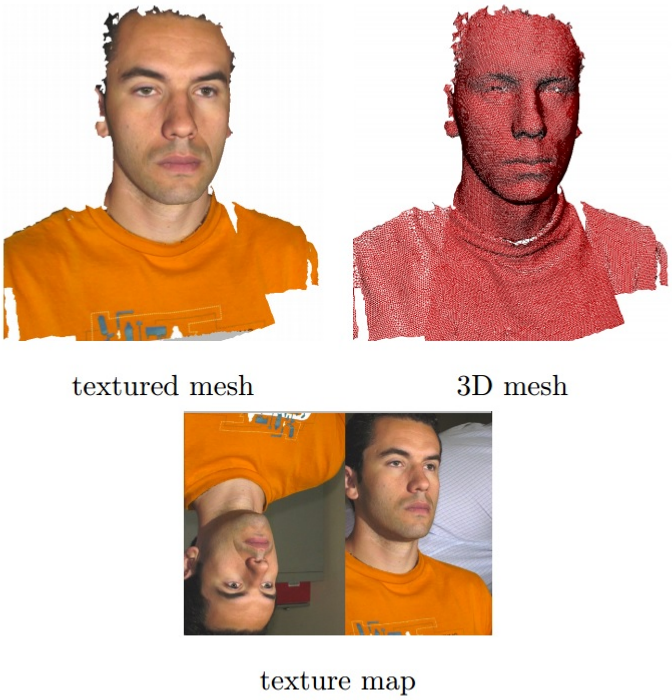
are maintained to avoid shadow and specularities and ensures the model texture is representative of face color (albedo) only and reduces the significance of the components which are not inherent characteristics of a human face.

The most challenging task in building the model is establishing dense correspondence across the database. Each raw face scan comprises a 3D mesh and a 2D RGB texture map (as shown in figure 2.5). The (x, y, z) coordinates of the vertices of the 3D mesh of the j^{th} face scan can be concatenated into a shape vector as:

$$\mathbf{Shape}_j = [x_1\ y_1\ z_1,\ \ldots,\ x_n\ y_n\ z_n]^T \tag{2.7}$$

And similarly for the RGB values of texture map in the texture vector as:

$$\mathbf{Texture}_j = [R_1\ G_1\ B_1,\ \ldots,\ R_n\ G_n\ B_n]^T \tag{2.8}$$



textured mesh

3D mesh

texture map

Figure 2.5: 3D data acquired with a 3dMD$^{\text{TM}}$ sensor. Textured mesh, 3D mesh, and texture map are shown [14].

However for each new scan the values of n (the number of shape and texture components in a scan) will be different. In addition to it, the new face scan will not be aligned since they are captured with different pose. This problem is solved by using Iterative Multi-resolution Dense 3D Registration (IMDR) algorithm [15], which brings these scans in correspondence. To establish dense correspondence among all scans, a deformable reference 3D face model is used to perform a combination of local matching, global mapping and energy-minimization.

2.2.2 3D Morphable Model

In Face reconstruction, 3D Morphable Models can be used to infer the 3D information from a 2D image. The 3D Morphable Model is a three dimensional mesh of faces which have been registered to a reference mesh through dense correspondence. It is required for all the faces to be have same number of vertices in corresponding face locations. Under these constraints a shape vector is represented by $s_j = [x_1\ y_1\ z_1,\ \ldots,\ x_v\ y_v\ z_v]^T \in \mathbb{R}^{3v}$, containing the x, y and z components of the shape, and a texture vector $t_j = [R_1\ G_1\ B_1,\ \ldots,\ R_v\ G_v\ B_v]^T \in \mathbb{R}^{3v}$, containing the per vertex RGB color information, where v is the number of dense registered mesh vertices.

Principle component analysis is performed on these set of shape $\tilde{\mathbf{S}} = [s_1, \ldots, s_M] \in \mathrm{R}^{3v \times M}$ and texture $\tilde{\mathbf{T}} = [t_1, \ldots, t_M] \in \mathrm{R}^{3v \times M}$ vectors, where M is the number of face meshes. PCA performs a basis transformation to an orthogonal coordinate system formed by the eigenvectors $\mathbf{S}_i$ and $\mathbf{T}_i$ of the shape and texture covariance matrices, respectively in ascending order of their eigen values.

For M face meshes, PCA provides a mean face $\bar{s}$, a set of $M - 1$ principal components, the i^{th} of which is denoted by $\mathbf{S}_i$ (also called as the eigen vector) with corresponding variance $\sigma_{s,i}^2$ (also called as eigen value). Unique face meshes can be generated by varying the shape parameter vector $\mathbf{c}_s = [\alpha_1, \ldots, \alpha_{M-1}]^T$ in the equation 2.9

$$\mathbf{S}_{mod} = \bar{s} + \sum_{i=1}^{M-1} \alpha_i \sigma_{s,i}^2 \mathbf{S}_i \tag{2.9}$$

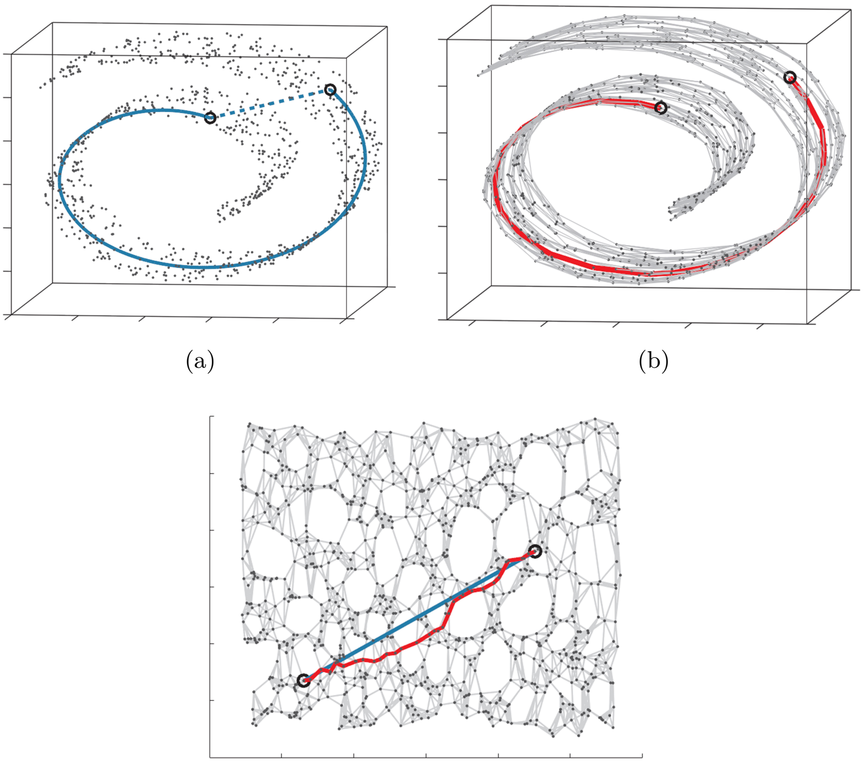
Similarly the equation for texture representation can be given by $\mathbf{T}_{mod} = \bar{t} + \sum_{i=1}^{M-1} \beta_i \sigma_{t,i}^2 \mathbf{T}_i$ with $\mathbf{c}_t = [\beta_1, \ldots, \beta_{M-1}]^T$ as the texture parameter vector.

2.3 Texture Representation

The PCA color model obtained from 3DMM is a useful illustration for the appearance of a face. But in some case it is either desirable to use pixel color information from image or a combination of the two. The texture (pixel color information) from the input image which when remapped onto the mesh sustain all details of a face's appearance, while some high frequency information can be absent when the face is represented by PCA color model only. Therefore it is desirable to use a 2D representation of the whole face mesh that can be used to store the remapped texture. This generic representation is created by *Isomap algorithm*.

2.3.1 Isomap Algorithm

The Isomap algorithm initially proposed by Tenenbaum et al. [16] is a non-linear dimensionality reduction technique which focuses on retaining the *geodesic distance*, between data points, measured on the data's manifold. The geodesic distance between two points is defined as the shortest path connecting the two points without cutting through the surface. The algorithm is used to remove the 3^{rd} dimension of the 3D mesh to convert it to a flat 2D surface, while preserving the area of mesh's triangle. Figure 2.6 depicts the working of the algorithm diagrammatically.



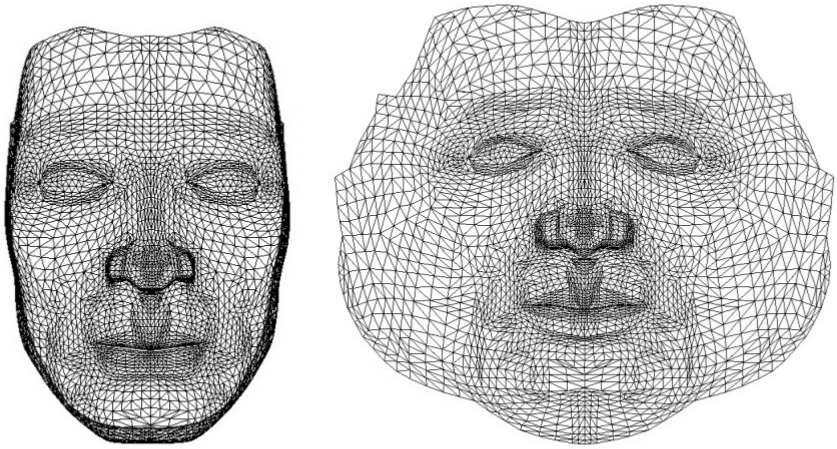
(a)

(b)

(c)

Figure 2.6: The Isomap Algorithm Illustration

With reference to Figure 2.6, A data set of 3D points on a spiral manifold illustrates how the algorithm exploits geodesic distance for non-linear dimensionality reduction. For two arbitrary points (circled on non-linear manifold (a)), their Euclidean distances (blue dashed line) in the high dimensional space may not depict their intrinsic similarity, as measured by geodesic distance (solid blue line) along the low-dimensionality manifold (a). The geodesic distance can be obtained (red segments in (b)) by taking successive Euclidean distance between neighboring points. The Isomap in (c) recovers a 2D-embedding, that preserves geodesic distance (red line) between points, where blue straight line approximates the true geodesic distances between points in the original manifold.



(a) Generic Model (b) isomap dimensionality reduction

Figure 2.7: Dimensionality reduction with the isomap algorithm, the 3D model before and after the application of isomap algorithm [14].

2.4 Stereo Reconstruction

The stereo vision setup for 3D reconstruction is the closest to the human perception of 3D reality. The images produced from two calibrated cameras provide disparity information (or distance) between corresponding points in the two images. The resulting “disparity map” is used to determine the relative depths of objects in the scene [17].

The typical setup contains multiple cameras oriented in the same viewing direction with the same axis system and projection [17]. Let us assume that the setup shown in Figure 2.8. The distance between the projection centers is called the base (b) and should be greater than zero. The center of the left camera is located at point C_1, the right one lies at a distance b on X axis at C_2. The camera constant describing the idealized camera parameters is f and f' respectively for left and right camera. The 3D point $\mathbf{P}$ in space has coordinates $(\mathbf{x_p}, \mathbf{y_p}, \mathbf{z_p})$, and U_L and U_R are the projection of the point $\mathbf{P}$ on left and right image plane.

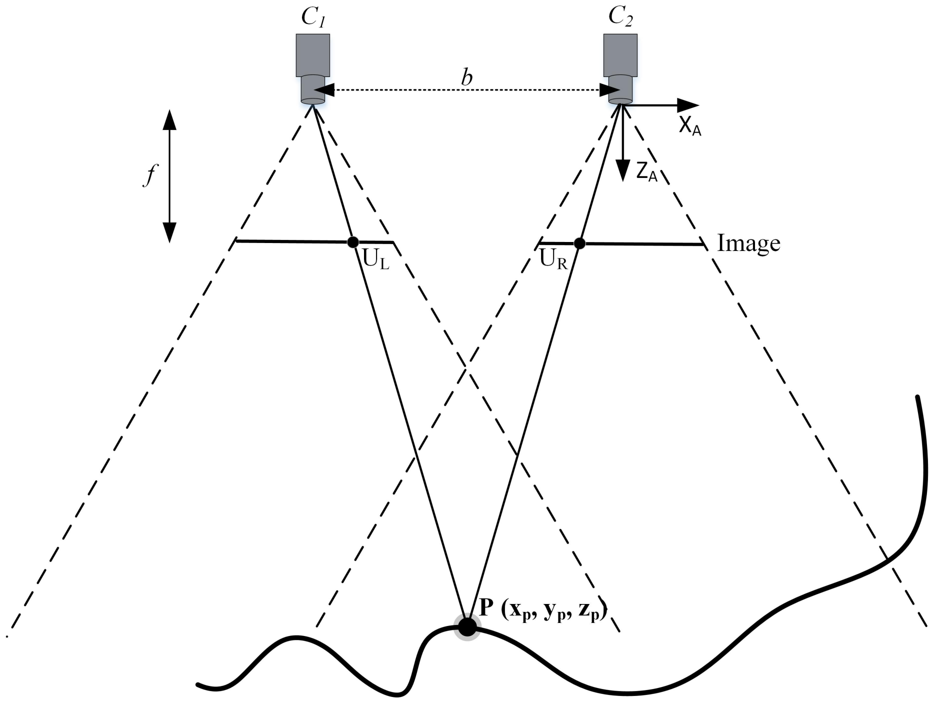


FIGURE 2.8: Basic stereo imaging scheme: The two cameras have overlapping field of view, the points lying in this region are observed in both the images.

2.4.1 Epipolar Geometry

Epipolar geometry describes the relations between two images. It is an outcome of the idea that projection of a 3D point gives a 2D point on an image. The 2D point on image and the corresponding projection center gives a 3D line. The image of this 3D line as seen from the other camera gives a 2D line on the image. It is called *epipolar line.* The matching projection point of one image lies on the epipolar line of another image.

Figure 2.9 explains the epipolar geometry for stereo vision. The Projection of point **P** on right image plane is at $\mathrm{U_R}$. The image of line joining P and $\mathrm{U_R}$ on left image plane is $\mathrm{l_L}$. The point $\mathrm{U_R}$ has a corresponding point on this line $\mathrm{l_L}$. Similarly for $\mathrm{U_L}$ the epipolar line is $\mathrm{l_R}$.

If the cameras are in ideal position, the epipolar lines are parallel. These epipolar lines are used for key point matching. Some problems, however persist: similar textures around the epipolar line, big distance discontinuity, occluded objects [17]. Overall, stereo imaging can be seen as robust, reliable and relatively precise technique. If using a single stereo camera the inter calibration of camera is not required.

The epipolar relation is used to derive Fundamental Matrix which is the key to the projection of image points back to the 3D points.

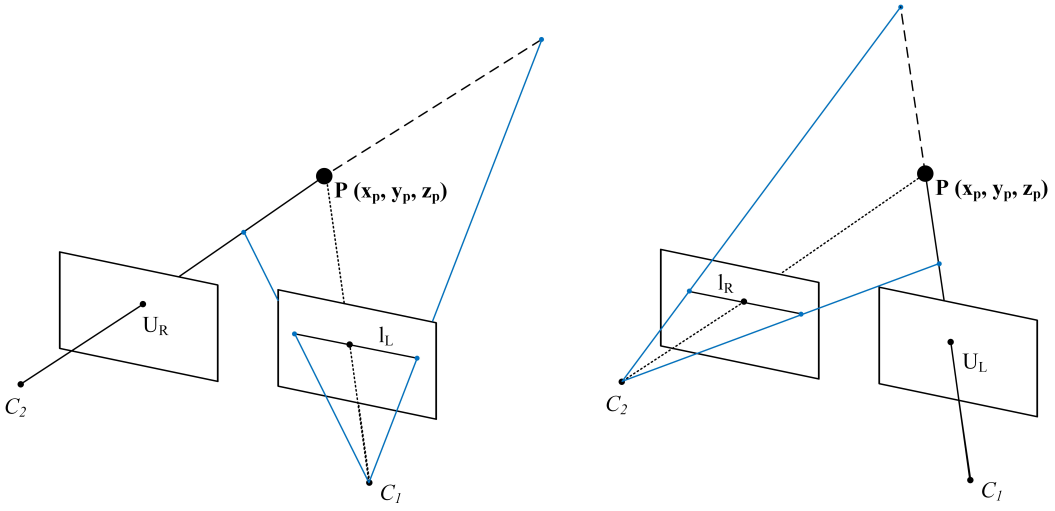


FIGURE 2.9: Epipolar Geometry

2.5 Deformation of Face Model

Deformation which is sometimes a crucial step of any reconstruction scheme, could be required when a particular surface or reconstruction is unable to denote the object. In such cases, additional information is needed to correct the geometry of the model. When observing the deformation from a different view point, it is a element of surface registration. And has problem similar to that of registration, it is not rigid which makes the registration problem significantly more challenging. When comparing the target mesh with the source mesh, the source would have to undergo both rigid and non-rigid deformation.

There are various sophisticated deformation algorithms which propagate the attributes of target mesh to the source mesh. But performing the deformation in natural way is a challenging task and requires a weighted or combinational scheme to overcome the problem. A deformation model which is general and favors high quality natural shape deformation is required. The system should be robust to noisy targets composed of multiple unwanted connecting components.

For matching a source mesh to a target, generally surface registration by the rigid and non-rigid transformations are to be carried out on the source. In our case the source mesh is the face model and the target is the stereo reconstruction. The choice of stereo reconstruction as the target mesh and face model as the source mesh has been explained in the later section. As the target is rather rough and not the exact goal, we aim to fuse the information of source and target. This fusion can be achieved by the non-rigid component of surface registration, i.e. by the deformation. Our procedure defuses the mesh continuity of source with the geometry of target, thus the result holds the best of the two.

The approach differs from standard surface registration in two ways. In the first place, natural deformation which is performed iteratively for surface registration is carried out only once. Secondly, the requirement of optimization based on energy function is ruled out, which significantly reduces the computational complexity.

Our deformation model is motivated by the Global Correspondence Optimization approach of H. Li et al. [18], which treats deformation as a two step approach. The choice of routines for the deformation, differs us from global correspondence optimization. Our work augments the parts of embedded deformation framework with a global transformation. But unlike the approach, we do not use any deformation graph and the whole mesh is taken as a single entity.

One of the also possible ways could be treating deformation as the three dimensional data interpolation problem. This brings us to the use of Radial Basis Function (RBF) because of its applications in data interpolation [19], reconstruction and representation of 3D objects [20].

A two step deformation arrangement, which addresses the local as well as global deformation scheme is implemented, such that a vertex $\mathbf{v}_j$ on the source is transformed to $\tilde{\mathbf{v}}_j$ according to:

$$\tilde{\mathbf{v}}_j = \Phi_{\text{local}} \circ \Phi_{\text{global}}(\mathbf{v}_j) \tag{2.10}$$

The source mesh vertices are deformed by first applying the global deformation and then the local deformation routine. Our approach does not require source and target to share equal number of vertices, or to have identical connectivity. This deformation technique is our central research contribution, details of which are discussed below.

2.5.1 Global Deformation

By Global Deformation, we mean the global change in the source, by using few anchor points on the source mesh. These anchor points are selected by subsampling the source. As the Radial Basis Function (RBF) have been proved successful in shape deformation [21], it is our choice for non-rigid global deformation. RBF are means to approximate multi variable functions by linear combinations of terms based on a single univariate function. RBF interpolation sums a set of replicates of single basis function, where each replicate is centered at a data point (called knot or center) and scaled by interpolation condition [22].

A RBF is a real-valued function whose value depends only on the distance from the origin, so that $\phi(x) = \phi(\|x\|)$. The norm $(||\cdot||) : \mathbb{R}^n \to \mathbb{R})$ is usually Euclidean distance, although other distance functions are also possible. Alternatively, the RBF value could depend on the distance from some other point X_c, called a center, such that:

$$\phi(x, X_c) = \phi(\|x - X_c\|), \qquad x \in \mathbb{R}^n \tag{2.11}$$

Radial basis function can be used as a weighted combination of a kernel function, which is used to build up mesh deformation of the form:

$$g(x) = \sum_{c=1}^{N} \lambda_c \, \phi(\|x - X_c\|) \tag{2.12}$$

where the approximating function $g(x)$ is represented as a sum of N radial basis functions, each associated with a center X_c weighted by an appropriate coefficient λ_c. These coefficients are calculated by considering the position of target vertices, which brings in the influence of target in the deformation.

It should be clear from the equation 2.12, why this technique is called ”radial”, the influence of a single data point is constant on a sphere centered at that point. The weights λ_c can be estimated using the matrix methods of linear least squares, because the approximating function is linear in the weights.

A variety of functions can be used as radial basis kernel, some of them include: Gaussian, Thin Plate Spline, Multiquadric and Inverse Mutiquadric ([23], Appendix D). The choice of kernel depends upon quality features required in the output. In reality there is not yet a general characterization of what functions are suitable as RBF kernels [23].

We choose the Gaussian kernel of the form $\phi(r) = e^{-\frac{r^2}{2\sigma^2}}$, where σ^2 is the variance of the normal distribution. Now we can compose the Gaussian RBF with Euclidean distance function:

$$\phi_{i,c}(x) = e^{-\frac{||x_i - X_c||^2}{2\sigma^2}} \tag{2.13}$$

where x_i is the set of vertices on source mesh. Changing the value of σ changes the shape of the deformation function. A Gaussian RBF monotonically decreases with distance from center X_c. A larger variance causes the function to become *flatter* [24].

2.5.2 Local Deformation

After the global deformation step the deviations between face model and stereo reconstruction are locally still large. The aim of the subsequent non-rigid local deformation is to smoothen the overshoots caused by noise or errors of the stereo reconstruction. However, to maintain individual shape features that are only contained in the stereo reconstruction. In other words, the local deformation step allows to transfer shape information from the stereo reconstruction to the final face model.

The deformation of a vertex on the source is influenced by its neighboring vertices, referring to as local deformation. The local deformation approach is derived from a work by Sumner et al. [25] using a modified weighting scheme.

The non rigid transformation of each source vertex w.r.t. its nearest vertices on the target is obtained by Procrustes Analysis [26]. But instead of applying non rigid transformation of a source vertex to itself, the transformations of k-nearest neighbors is applied by a weighted scheme. This technique creates a smooth localized deformation.

Let $\mathbf{B}_i$ and $\mathbf{t}_i$ denote the components of affine transformation for all k-nearest neighbors of node v. These transformations are used to transform the mesh at each node v according to the equation:

$$\Phi_{\text{local}}(v) = w_0(v)[\mathbf{B}_0 v + \mathbf{t}_0] + \sum_{i=1}^{k} w_i(v)[\mathbf{B}_i v + \mathbf{t}_i], \forall v \in \mathbf{V} \tag{2.14}$$

where $\mathbf{V}$ denotes the set of all the source mesh vertices, k is the number of nearest neighbors of node v. The $w_0(v)$ limits the influence of the node on its own transformation, $\mathbf{B}_0$ and $\mathbf{t}_0$ denote the components of affine transformation of the node. The weights $w_i(v)$ are derived from the argument that, it must be a function of Euclidean distance of the nearest neighbor of node v.

$$w_i(v) = f(||v_i - v||) \tag{2.15}$$

where v_i is the i^{th} nearest neighbor of node v. Also a vertex far from the node should have less influence on the transformation, and hence a smaller value of weight:

$$w_i(v) \propto \left(1 - \frac{||v_i - v||}{d}\right)$$
$$w_i(v) = \frac{1}{k}\left(\frac{d - ||v_i - v||}{d}\right) \tag{2.16}$$

where d is the sum of distance of k-nearest neighbors of node v. The factor $1/k$ is used to scale down the sum of weights to 1. From the above follows that the value of $w_0(v) = 1/k$.

Every vertex on source would have more than one weight, the number of weights of a vertex is equal to the number of nodes to which it is nearest neighbor. Given the source and target mesh, for local deformation, weights of source vertices are calculated. These weights when used in equation 2.14 gives the local deformation.

2.6 Point Cloud Visualization

The cardboard project of Google[5] aims to develop inexpensive Virtual Reality tools to allow everyone enjoy it in a simple, fun and natural way. This Software Development Kits (SDKs) focuses on applications of VR in entertainment but by proper environment object modification, our reconstruction could also be visualized. The simplifications include:

- User head tracking
- Side-by-side stereo rendering
- Detecting cardboard-only user inputs such as the trigger
- Automatic stereo configuration for a specific cardboard model
- Distortion correction for cardboard lenses
- An alignment marker to help center the screen under the lenses
- A settings button that links to cardboard app for managing headset parameters

Even the hardware and software are kept open to encourage the community participation and compatibility with VR content available elsewhere.

[5] https://developers.google.com/cardboard/overview

2.6.1 Smart Phone based 3D HMD

The smart phone serves as a display and provides the necessary inertial sensors for head tracking during its rotation. Due to the limited performance of the smart phone the files to be used need slight modification. The motion of onscreen object is a reflection of head's rotation and are directly derived from the inertial sensors of the smart phone. Thus, rotation of the head results in rotation of the virtual camera. Due to a fast implementation the latency of the visualization is low.

2.6.2 Cardboard Viewer

A simple device that unlocks the power of your smart phone as a Virtual Reality (VR) or Augmented Reality (AR) platform. Its a virtual reality head-mounted smartphone mount made of cardboard. Named for its fold-out cardboard viewer, the platform is intended as a low-cost solution to encourage interest and development in VR and AR applications. It can be easily made by inexpensive components using the specifications published by google or can be purchased from a third party manufacturer. The cardboard was introduced in 2014, and through January 2016, over 5 million cardboard viewers have been alone shipped by google.

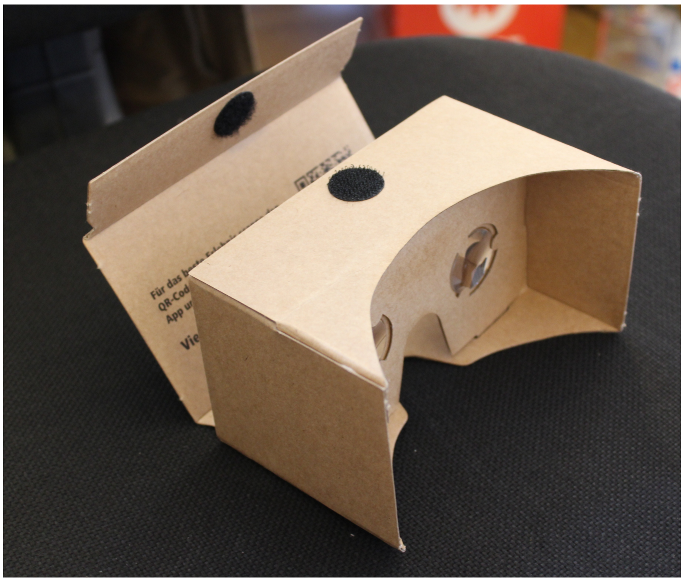


FIGURE 2.10: A Cardboard Viewer

Cardboard viewer can work with smart phone to display 3D scenes with binocular rendering, track and react to head movements, and interact with apps through a trigger input. The head mounting positions the two lenses in between the eyes and the half image (as shown in figure 2.10). The eyes are focused on the display using the lens stereoscope that is connected to the mounting. Both lenses can also be positioned autonomously to adapt for personal inter eye distance and for ametropia. It has a provision to position a smartphone in landscape mode. Some of the cardboards have a magnetic switch, which can be used as a trigger. The mounting is fixed on the head with a headband (or could be hold by hand) to create a head mounted display. Figure 2.11 illustrates the ease of use of cardboard with hand.

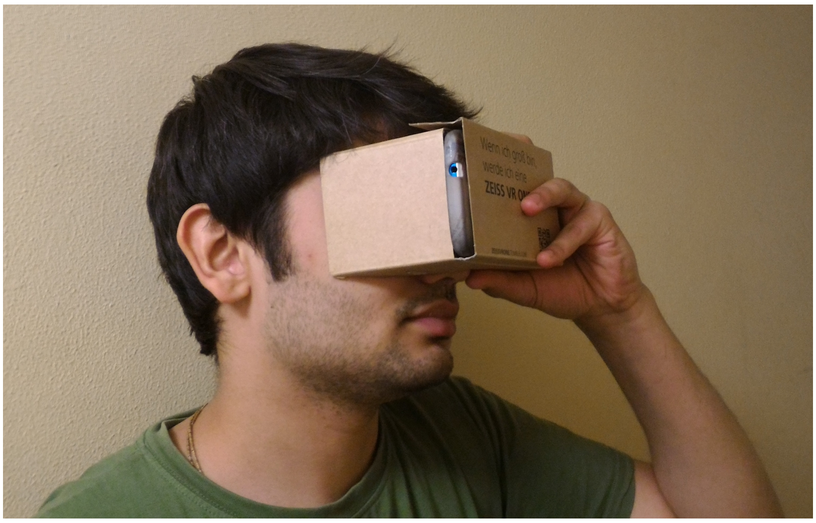


FIGURE 2.11: Using cardboard with smartphone as a Head Mounted Display

2.6.3 Android Framework

Android is one of the open source platforms. It was created by Google and owned by open handset Alliance. It is designed with goal *accelerate innovation in mobile.* As such android has taken over the field of mobile software innovation. It gives possibility of friendlier environment for developers and users. Android is a complete software package for a mobile device, it uses Android application package (APK) for distribution and installation of mobile apps.

Since the beginning, android team offers the developing kit (tools and framework) for creating mobile application quick and easy as possible. The Integrated Android development environment tool by Google is Android Studio[6] and is freely available under Apache License 2.0. Another prominent cross platform game engine is Unity3D[7] by Unity Technologies and can be used to develop android applications. Although the two softwares are provided by different vendors, the project files can be interchangeably used.

[6] http://developer.android.com/sdk/index.html

[7] https://unity3d.com/

Earlier it was not possible to view and interact with point clouds on a mobile device, since the Visualization of point clouds was only based on DirectX while android based mobile devices run OpenGL ES. Google has bridged this gap by providing the point cloud visualization support in OpenGL ES 3.0, which is present in the android devices with API 21 and Higher. A 4.7 inches (or higher till 5.1 inches) Android smartphone with Operating System *Android Lollipop* along with OpenGL ES 3.0 support could be an ideal combination for visualization. Quickly many VR games were also available which aims to give the real feel to the player. Over 1000 compatible applications had been published on VR and AR. Majority of these applications on Google Play store are for entertainment purpose or virtual tours of cities, space, roller coaster. But the data volume and the detail of 3D models are still restricted due to the limited computational performance of a smart phone.

Chapter 3

Proposed Scheme

Our work uses existing single face reconstruction method by P. Huber et al. [10] on 3448 vertices shape-only Surrey 3D Morphable Face Model. It is carried out by sequentially following Pose Estimation, Shape Fitting and Texture Extraction to generate a face model. An improvement in the stereo reconstruction from the obtained face model by fusing the information of the two is proposed. For the natural deformation, a combination of global and local optimization is implemented. The outcome of the deformation is referred to as *Deformed Face Model* in this work. The block diagram of the complete procedure is shown in figure 3.1

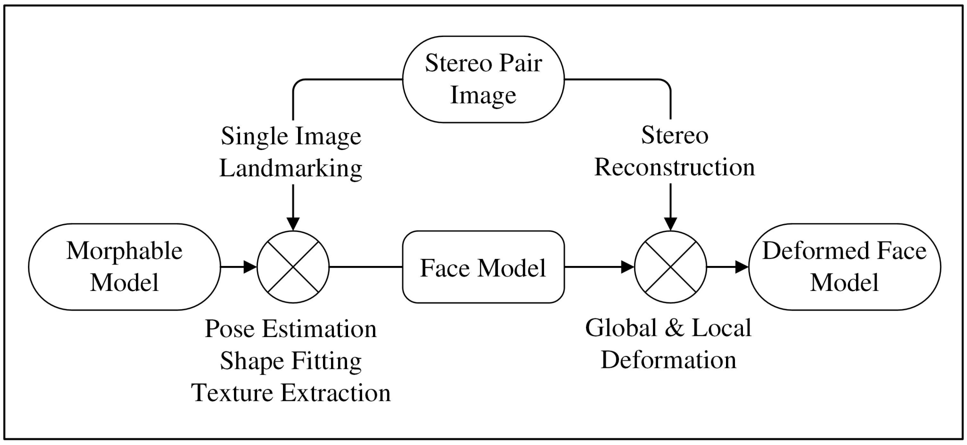


FIGURE 3.1: Block Diagram of proposed scheme

A face is captured by stereo camera, which gives two images. One of which is used for single image reconstruction and the pair is used to generate the stereo reconstruction of the face. The process begins by landmarking the face image.

The automated process of 3D reconstruction from an image involves some correspondence between the 2D image and 3D model. These correspondences may be of different forms depending upon the accuracy required. In our case, some feature points on Morphable model are marked which uniquely identifies a human face. These points includes corners of eyes, mouth, tip of nose and eye brows and their connecting points. These points on image are identified by landmarking annotation algorithm.

3.1 Landmarking Annotation

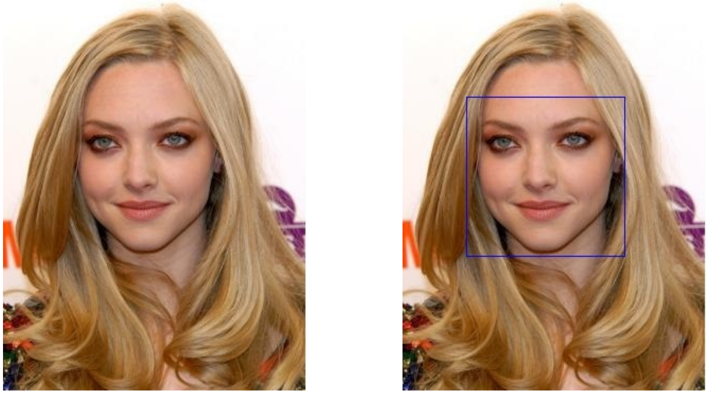
The Regression based methods do not build any parametric models of shape or appearance, but study the correlations between the image features to infer a facial vector. The synthesis pipeline starts with a face detector, then the image together with the bounding box in fed to face alignment component which return a facial vector. The face alignment component is approached by supervised machine learning, whereby a model is trained from a large amount of human-labeled images and can then be used for facial feature vector (**F**) estimation on new face images.

3.1.1 Training data

To facilitate training, 3283 faces data collected by *300 Faces In-The-Wild Challenge (300-W)* [27] has been used. The face databases covers large variety including: different subjects, poses, illumination, occlusions. A well established landmark configuration of 68 points mark up (as shown in figure 2.1) were also available with the images. To enhance accuracy, the annotations have been manually corrected by an expert. Additionally, the IBUG data set consists of 135 images with highly expressive faces, difficult poses and occlusions. The training is performed with $T = 15$ regressors (r_t), in the cascade. $R = 20$ different initializations for each training example have been used.

3.1.2 Implementing Regression Algorithm

Typically, the problem assumes an image with an annotated bounding box which surrounds a face. In this case we have used an implementation of Max-Margin Object Detection (MMOD) by D. king [28] found in the dlib c++ library. Features are extracted from the sliding boxes on the image and these are compared with the trained values, resulting in the bounding box as shown in Figure 3.2



(a) (b)

Figure 3.2: Illustration of original image and Face bounding box obtained by Histogram of Oriented Gradient (HOG) detector

In order to start with regression, the initial estimate $\hat{\mathbf{F}}^{(0)}$ is chosen as the mean shape of the training data centered at the bounding box output of the generic face. This also assures the output of ensembles to lie in the linear subspace of training data. When the mean facial vector is overlapped on the face, the shape-indexed local features are extracted from the corresponding positions. To ensure invariance to the lightning conditions, the features used in the regressor are differences in pixel intensities extracted from image.

For a face image with an arbitrary shape, the differential points are indexed relative to the mean shape. To achieve this the coordinates of test positions u and v can be warped to the mean shape, before extracting the feature (Pixel Intensity). Instead of transforming the whole image, warping location of points once at each level of cascade is more efficient.

The first regressor is applied to the set of differential pixel intensities and then the coordinates are transformed to the global system by inverse transformation. These regressor r_t are used in equation 2.1 to update the vector estimate. This process is repeated for $T(= 15)$ regressions.

The 2D-3D correspondence between the annotated point on image and feature points on model are used, to get the camera orientation which is a crucial step in any reconstruction procedure. From this camera orientation matrix, more details such as orientation of face in image and texture of the model can be identified.

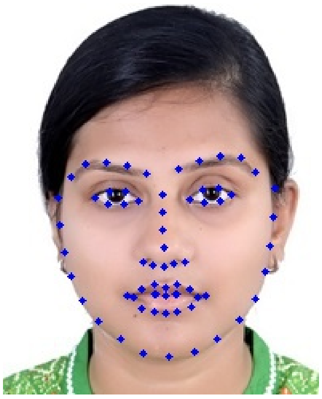


Figure 3.3: The 68 point landmarks on a face Image

3.2 Pose Estimation

Given a set of 3D points on Morphable Model and corresponding 68 2D Landmark Point (obtained in section 3.1) on a face image. The goal is to estimate the position of the camera (or the pose of the face) in the world coordinate. An affine camera model is assumed and *Gold Standard Algorithm* of Hartley & Zisserman [1] is used to compute camera matrix P, a 3 × 4 matrix such that $\mathbf{x}_i = \mathrm{P}\mathbf{X}_i$ for a subset of 68 points.

First, the labeled 2D Landmark Points in the face image $x_i \in \mathbb{R}^2$ and the corresponding 3D model points $X_i \in \mathbb{R}^3$ are represented in homogeneous coordinate by $x_i \in \mathbb{P}^2$ and $X_i \in \mathbb{P}^3$ respectively. Then the points are normalized by similarity transform that translate the centroids of the point set to the respective origin and scale them so that the Root-Mean-Square distance from their origin is $\sqrt{2}$ for 2D landmark points and $\sqrt{3}$ for model points. This makes the further algorithm invariant of similarity transform and also nullify the effect of arbitrary position of origin in the two spaces.

$$\tilde{\mathrm{x}}_{\mathrm{i}} = \mathrm{Tx}_{\mathrm{i}}, \text{ with } \mathrm{T} \in \mathbb{R}^{3\times3} \tag{3.1}$$

$$\tilde{\mathrm{X}}_{\mathrm{i}} = \mathrm{UX}_{\mathrm{i}}, \text{ with } \mathrm{U} \in \mathbb{R}^{4\times4} \tag{3.2}$$

The similarity transform T and U can be obtained by equations 3.3 and 3.4. Where the letter s and t are used to denote scaling and translation respectively in the subscript direction.

$$\mathrm{T} = \begin{bmatrix} s_x & 0 & -t_x \\ 0 & s_y & -t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

$$\mathrm{U} = \begin{bmatrix} s'_x & 0 & 0 & -t'_x \\ 0 & s'_y & 0 & -t'_y \\ 0 & 0 & s'_z & -t'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

Now for the 2D point transformed point $\tilde{\mathbf{x}}_i = (\tilde{u}_i, \tilde{v}_i, \tilde{w}_i)$, each correspondence between $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{X}}_i$ gives the relation:

$$\begin{bmatrix} \mathbf{0}^T & -\tilde{w}_i\tilde{\mathbf{X}}_i^T & \tilde{v}_i\tilde{\mathbf{X}}_i^T \\ \tilde{w}_i\tilde{\mathbf{X}}_i^T & \mathbf{0}^T & -\tilde{u}_i\tilde{\mathbf{X}}_i^T \\ -\tilde{v}_i\tilde{\mathbf{X}}_i^T & \tilde{u}_i\mathbf{X}^T & \tilde{\mathbf{0}}_i^T \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{P}}^1 \\ \tilde{\mathbf{P}}^2 \\ \tilde{\mathbf{P}}^3 \end{pmatrix} = \mathbf{0} \tag{3.5}$$

where each $\tilde{\mathbf{P}}^{iT}$ is a 4-vector, which is the i-th row of camera matrix $\tilde{\mathbf{P}}$. Although in the matrix relation 3.5 there are three equations, but only two of them are linearly independent (since the third row is a linear combination of first two). Thus each 2D-3D point correspondence gives two equations in the entries of $\tilde{\mathbf{P}}$. Rewriting the equation 3.5 with $\tilde{\mathbf{x}}_i = (\tilde{x}_i, \tilde{y}_i, 1)$ implies $\tilde{x}_i = \frac{\tilde{u}_i}{\tilde{w}_i}, \tilde{y}_i = \frac{\tilde{v}_i}{\tilde{w}_i}$:

$$\begin{bmatrix} \mathbf{0}^T & -\tilde{\mathbf{X}}_i^T & \tilde{y}_i\tilde{\mathbf{X}}_i^T \\ \tilde{\mathbf{X}}_i^T & \mathbf{0}^T & -\tilde{x}_i\tilde{\mathbf{X}}_i^T \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{P}}^1 \\ \tilde{\mathbf{P}}^2 \\ \tilde{\mathbf{P}}^3 \end{pmatrix} = \mathbf{0} \tag{3.6}$$

solving equation 3.6 for affine camera matrix *i.e.* with $\tilde{\mathbf{P}}^3 = [0\ 0\ 0\ 1]^T$, we get

$$\begin{bmatrix} \mathbf{0}^T\tilde{\mathbf{P}}^1 - \tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^2 + \tilde{y}_i\tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^3 \\ \tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^1 + \mathbf{0}^T\tilde{\mathbf{P}}^2 - \tilde{x}_i\tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^3 \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{0}^T\tilde{\mathbf{P}}^1 + \tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^2 - \tilde{y}_i \\ \tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^1 + \mathbf{0}^T\tilde{\mathbf{P}}^2 - \tilde{x}_i \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} \mathbf{0}^T\tilde{\mathbf{P}}^1 + \tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^2 \\ \tilde{\mathbf{X}}_i^T\tilde{\mathbf{P}}^1 + \mathbf{0}^T\tilde{\mathbf{P}}^2 \end{bmatrix} = \begin{bmatrix} \tilde{y}_i \\ \tilde{x}_i \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{0}^T & \tilde{\mathbf{X}}_i^T \\ \tilde{\mathbf{X}}_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{P}}^1 \\ \tilde{\mathbf{P}}^2 \end{pmatrix} = \begin{bmatrix} \tilde{y}_i \\ \tilde{x}_i \end{bmatrix} \tag{3.7}$$

Each correspondence of $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{X}}_i$ contributes to equation 3.7 which are stacked into a $2n \times 8$ matrix equation $\mathbf{A}_8\mathbf{p}_8 = \mathbf{b}$, where $\mathbf{p}_8$ is the 8-vector containing the first two rows of $\tilde{\mathbf{P}}$. The affine camera matrix has 8 degrees of freedom (8 unknowns) for which we require four equations (n = 4), each solving for two unknowns.

In order to reduce the observation error, the actual number of correspondences is more than four, which results in an non square matrix $\mathbf{A}_8$ of size $m \times n, m > n$, so the solution is obtained by taking pseudo-inverse of $\mathbf{A}_8$ denoted by $\mathbf{A}_8^+$.

$$\mathbf{p}_8 = \mathbf{A}_8^+\mathbf{b} \tag{3.8}$$

Horizontal concatenation of three vectors $\tilde{\mathbf{P}}^{1T}$, $\tilde{\mathbf{P}}^{2T}$ and $\tilde{\mathbf{P}}^{3T}(= [0\ 0\ 0\ 1])$ gives the normalized affine camera matrix $\tilde{\mathbf{P}}$. The actual camera matrix by is obtained by performing de-normalization step by transformation matrices:

$$\mathbf{P} = \mathrm{T}^{-1}\tilde{\mathbf{P}}\mathrm{U} \tag{3.9}$$

Once the pose has been estimated, now the task is to reach to the actual model shape which resembles to the image. This is accomplished by estimation of shape parameter vector, which depends upon the landmark positions. These shape parameter vector when used in the 3DMM gives the *Face Model* corresponding to the face. Subsequently, the camera orientation matrix along with the Isomap algorithm is used for texture map extraction from the single image.

3.3 Shape Fitting

The shape parameters are recovered using a probabilistic approach, which maximizes the posterior probability. The aim is to find the most likely shape vector $\mathbf{c}_s$ given an observation of N 2D feature landmark points in homogeneous coordinates: $\mathbf{y} = [x_1\ y_1\ 1, ..., x_N\ y_N\ 1]^T$ and taking into consideration the model prior. From Baye's rule we can state the posterior probability of variance normalized shape parameter $(\mathbf{c}_s)$ w.r.t. $\mathbf{y}$ as

$$P(\mathbf{c}_s|\mathbf{y}) = v \cdot P(\mathbf{y}|\mathbf{c}_s) \cdot P(\mathbf{c}_s) \tag{3.10}$$

where $v = (\int P(\mathbf{y}|\mathbf{c}'_s) \cdot P(\mathbf{c}'_s) d\mathbf{c}'_s)^{-1}$ is a constant factor. The coefficients of shape parameters are normally distributed with zero mean and unit variance, *i.e.* $\mathbf{c}'_s \sim \mathcal{N}(0,\mathrm{I_N})$, so the probability of observing a given $\mathbf{c}_s$ is

$$P(\mathbf{c}_s) = v_c \cdot e^{\frac{-1}{2}\ ||c_s||^2} \tag{3.11}$$

where $v_c = (2\pi)^{-m'/2}$. The probability of observing $\mathbf{y}$ for a given $\mathbf{c}_s$ is given by:

$$P(\mathbf{y}|\mathbf{c}_s) = \prod_{i=1}^{3N} v_N \cdot e^{-\frac{1}{2\sigma^2_{2D,i}}[y_{model2D,i}-y_i]^2} \tag{3.12}$$

Here, $y_{model2D,i}$ are the homogeneous coordinates of the 3D feature points (marked on model) projected to 2D, defined as follows:

A matrix $\hat{\mathrm{S}} \in \mathbb{R}^{\mathrm{3N\times m-1}}$ is constructed by sub-selecting the rows of eigenvalues $\mathbf{S}$ associated with N landmark points. In order to give matrix a homogeneous shape, the matrix is further modified by inserting a row of zeros after third row of $\mathbf{S}$, resulting in matrix $\hat{\mathrm{S}}_{\mathrm{h}} \in \mathbb{R}^{\mathrm{4N\times m-1}}$. Now a block diagonal matrix $\mathbf{C} \in \mathrm{R}^{\mathrm{3N\times 4N}}$ is formed, in which the the camera matrix is placed on the diagonals:

$$C = \begin{bmatrix} P & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & P \end{bmatrix} \tag{3.13}$$

Finally the 2D point obtained by projecting the 3D model point given by

$$y_{model2D,i} = \mathbf{C}_i \cdot (\hat{\mathbf{S}}_h \mathrm{diag}(\sigma_s^2)\mathbf{c}_s + \bar{\mathbf{v}}) \tag{3.14}$$

where $\mathbf{C}_i$ is the i^{th} row of $\mathbf{C}$.

Substituting equation 3.11 and 3.12 in the Baye's rule equation 3.10, the conditional probability is given by

$$\begin{aligned} P(\mathbf{c}_s|\mathbf{y}) &= v \cdot v_N^l \cdot v_c \cdot e^{-\sum_{i=1}^{3N} \frac{[y_{model2D,i}-y_i]^2}{2\sigma^2_{2D,i}}} \cdot e^{-\frac{1}{2}||\mathbf{c}_s||^2} \\ &= v \cdot v_N^l \cdot v_c \cdot e^{-\frac{1}{2}\left[\sum_{i=1}^{3N} \frac{[y_{model2D,i}-y_i]^2}{\sigma^2_{2D,i}} + ||\mathbf{c}_s||^2\right]} \end{aligned}$$

which can be maximized by minimizing the exponent:

$$E = -2 \cdot logP(\mathbf{c}_s|\mathbf{y}) = \sum_{i=1}^{3N} \frac{[y_{model2D,i} - y_i]^2}{\sigma^2_{2D,i}} + ||\mathbf{c}_s||^2 + constant \tag{3.15}$$

Substituting the statistical model equation 3.14 with $\mathbf{a} = \text{diag}(\sigma_s^2)\mathbf{c}_s$ into the equation 3.15:

$$
\begin{aligned}
&E = \sum_{i=1}^{3N} \frac{[\mathbf{C}_i \cdot \hat{\mathbf{S}}_h\mathbf{a} + \mathbf{C}_i \cdot \bar{\mathbf{v}} - y_i]^2}{\sigma_{2D,i}^2} + ||\mathbf{c}_s||^2 \\
&E = \sum_{i=1}^{3N} \frac{[\mathbf{C}_i \cdot \hat{\mathbf{S}}_h\mathbf{a} + \mathbf{C}_i \cdot \bar{\mathbf{v}}]^2 - 2[\mathbf{C}_i \cdot \hat{\mathbf{S}}_h\mathbf{a} + \mathbf{C}_i \cdot \bar{\mathbf{v}}]y_i + y_i^2}{\sigma_{2D,i}^2} + ||\mathbf{c}_s||^2 \\
&E = \sum_{i=1}^{3N} \frac{(\mathbf{C}_i \cdot \hat{\mathbf{S}}_h\mathbf{a})^2 + (\mathbf{C}_i \cdot \bar{\mathbf{v}})^2 + 2\mathbf{C}_i \cdot \hat{\mathbf{S}}_h\mathbf{a}\mathbf{C}_i \cdot \bar{\mathbf{v}} - 2\mathbf{C}_i \cdot \hat{\mathbf{S}}_h\mathbf{a}y_i - 2\mathbf{C}_i \cdot \bar{\mathbf{v}}y_i + y_i^2}{\sigma_{2D,i}^2} + ||\mathbf{c}_s||^2
\end{aligned}
$$

For clarity, using $\mathbf{R}_i = \mathbf{C}_i \cdot \hat{\mathbf{S}}_h$, $k_i = \mathbf{C}_i \cdot \bar{\mathbf{v}}$ and $\mathbf{c}_s = \frac{a}{diag(\sigma_{s,i}^2)}$ in the above equation

$$
E = \sum_{i=1}^{3N} \frac{(\mathbf{R}_i\mathbf{a})^2 + k_i^2 + 2\mathbf{R}_i\mathbf{a}k_i - 2\mathbf{R}_i\mathbf{a}y_i - 2k_iy_i + y_i^2}{\sigma_{2D,i}^2} + \|\frac{a}{diag(\sigma_{s,i}^2)}\|^2 \tag{3.16}
$$

The error function E has to be minimized so differentiating with respect to $\mathbf{a}$ and set the derivative to zero.

$$
\begin{aligned}
\triangledown E = \sum_{i=1}^{3N} \frac{2\mathbf{R}_i^T\mathbf{R}_i\mathbf{a} + 2\mathbf{R}_i^T k_i - 2\mathbf{R}_i^T y_i}{\sigma_{2D,i}^2} &+ \frac{2a}{[diag(\sigma_{s,i}^2)]^2} = 0 \\
\sum_{i=1}^{3N} \frac{\mathbf{R}_i^T\mathbf{R}_i\mathbf{a} + k_i\mathbf{R}_i^T - y_i\mathbf{R}_i^T}{\sigma_{2D,i}^2} &+ \frac{\mathbf{c}_s}{diag(\sigma_{s,i}^2)} = 0
\end{aligned} \tag{3.17}
$$

Since the system of equation has to be solved for $\mathbf{c}_s$ instead of $\mathbf{a}$, so multiplying the equation by $diag(\sigma_{s,i}^2)$, and using $\mathbf{Q}_i = \mathbf{R}_i diag(\sigma_{s,i}^2)$, we obtain:

$$
\begin{aligned}
\sum_{i=1}^{3N} \frac{diag(\sigma_{s,i}^2)\mathbf{R}_i^T\mathbf{R}_i\mathbf{c}_s diag(\sigma_{s,i}^2) + k_i diag(\sigma_{s,i}^2)\mathbf{R}_i^T - y_i diag(\sigma_{s,i}^2)\mathbf{R}_i^T}{\sigma_{2D,i}^2} + \mathbf{c}_s = 0 \\
\sum_{i=1}^{3N} \frac{\mathbf{Q}_i^T\mathbf{Q}_i\mathbf{c}_s + k_i\mathbf{Q}_i^T - y_i\mathbf{Q}_i^T}{\sigma_{2D,i}^2} + \mathbf{c}_s = 0
\end{aligned}
$$

$$
\sum_{i=1}^{3N} \frac{\mathbf{Q}_i^T\mathbf{Q}_i\mathbf{c}_s}{\sigma_{2D,i}^2} + \mathbf{c}_s = \sum_{i=1}^{3N} \frac{y_i\mathbf{Q}_i^T}{\sigma_{2D,i}^2} - \sum_{i=1}^{3N} \frac{k_i\mathbf{Q}_i^T}{\sigma_{2D,i}^2} \tag{3.18}
$$

For simplicity we set:

$$
\mathbf{T}_1 = \sum_{i=1}^{3N} \frac{\mathbf{Q}_i^T\mathbf{Q}_i}{\sigma_{2D,i}^2} \qquad \text{and} \qquad \mathbf{T}_2 = \sum_{i=1}^{3N} \frac{y_i\mathbf{Q}_i^T}{\sigma_{2D,i}^2} - \sum_{i=1}^{3N} \frac{k_i\mathbf{Q}_i^T}{\sigma_{2D,i}^2}
$$

and obtain the following equation: $\mathbf{T}_1\mathbf{c}_s + \mathbf{c}_s = \mathbf{T}_2$. This can be solved by applying a *cholesky decomposition* to $\mathbf{T}_1$:

$$\text{using } \mathbf{T}_1 = \mathbf{M}^T\mathbf{M} \implies \mathbf{M}^T\mathbf{M}\mathbf{c}_s + \mathbf{c}_s = \mathbf{T}_2 \tag{3.19}$$

Decomposing $\mathbf{M}$ with *Singular Value Decomposition*, $\mathbf{M} = \mathbf{U}\mathbf{W}\mathbf{V}^T$, we get:

$$\begin{aligned} (\mathbf{U}\mathbf{W}\mathbf{V}^T)^T\mathbf{U}\mathbf{W}\mathbf{V}^T\mathbf{c}_s + \mathbf{c}_s &= \mathbf{T}_2 \\ \mathbf{V}\mathbf{W}\mathbf{U}^T\mathbf{U}\mathbf{W}\mathbf{V}^T\mathbf{c}_s + \mathbf{c}_s &= \mathbf{T}_2 \end{aligned} \tag{3.20}$$

since $\mathbf{U}$ is orthogonal in all columns i with $w_i \neq 0$, and multiplying by $\mathbf{V}^T$

$$\mathbf{V}^T\mathbf{V}\mathbf{W}^2\mathbf{V}^T\mathbf{c}_s + \mathbf{V}^T\mathbf{c}_s = \mathbf{V}^T\mathbf{T}_2 \tag{3.21}$$

$$\mathbf{diag}(w_i + 1)\mathbf{V}^T\mathbf{c}_s = \mathbf{V}^T\mathbf{T}_2 \tag{3.22}$$

$$\mathbf{c}_s = [\mathbf{diag}(w_i + 1)\mathbf{V}^T]^{-1}\mathbf{V}^T\mathbf{T}_2 \tag{3.23}$$

Hence, using only a series of matrix multiplications, we are able to recover the maximum likelihood of shape estimate vector $\mathbf{c}_s$ given the location of 2D projected feature points and projection matrix.

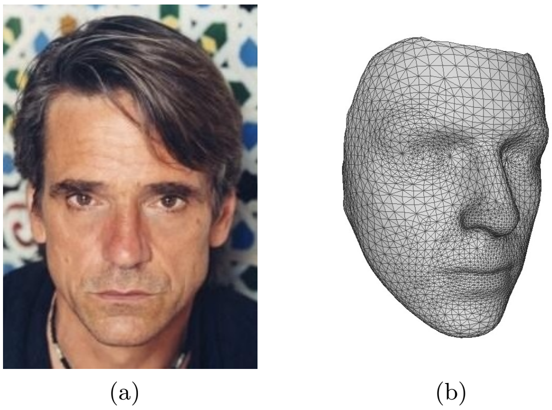

(a) (b)

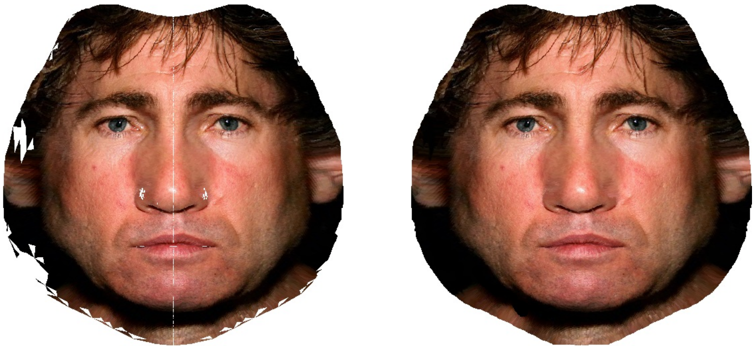
FIGURE 3.4: Face image and its obtained shape Model

3.4 Texture Extraction from Image

A Texture Extraction methodology has to be implemented to have the option of using the original image texture instead of texture reconstruction from the color morphable model. The input to the process is the face image and camera matrix, hence it has to be carried out after pose estimation. When carried out after shape fitting, the fitted shape can be used for precise texture element extraction. After the matching process is completed, the mesh triangles of optimized morphable model are projected to the image plane using isomap algorithm and camera matrix.

When the model is projected on image, the triangles with normal $> 90°$ are not visible on image. If a triangle is visible in the image, the RGB values of the pixels of the triangle are taken from the input image and copied to the location that corresponds to that same triangle in the newly created texture map. In the iso texture map a fourth quantity called alpha value is added which describes the visibility of a triangle, where the value α is given by $\alpha = 255 \times \text{normal angle}$. Using the normal values of optimized model and the z-buffer algorithm, these triangles are tested for visibility.

The texture map algorithm by P. Huber et al. [10] is efficient in evaluating the texture components from the image but it produces holes on some positions, as observed in Figure 3.5 (a). This holes in the texture map can be cleared by using a 3×3 averaging filter, without considering the holes. The filtering is only applied in the mask area specified by isomap algorithm as shown in figure 2.7 (b). The continuous texture can be observed in Figure 3.5 (b).

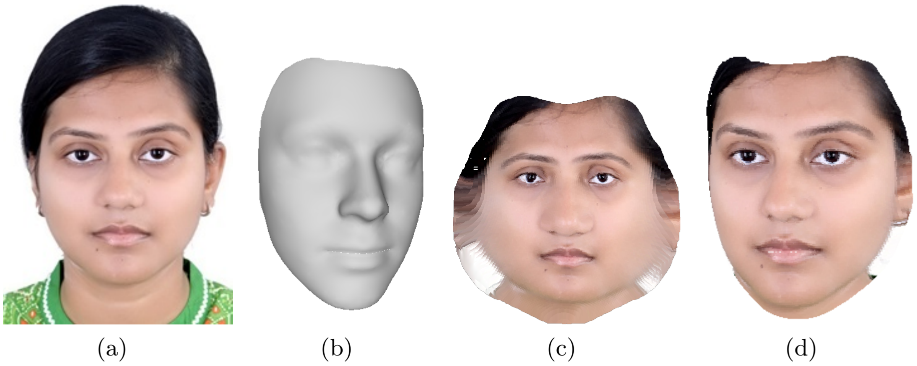

(a)

(b)

Figure 3.5: The texture maps before and after the application of averaging filter.

A complete combination of shape model and texture representation on a single image is shown in Figure 3.6, the combination of shape and texture as shown in 3.6 (d) is the resultant face model.

The Figure 3.7 shows two face models side by side in a comparative way. The face model in the last row look quite identical to each other but not to the respective images. We can observe the aspect ratio of the model, which is quite different in images. The observation is analytically supported by the face model comparison in Fig. 3.8. The



(a) (b) (c) (d)

Figure 3.6: A Face Image and the corresponding shape model, texture and face model

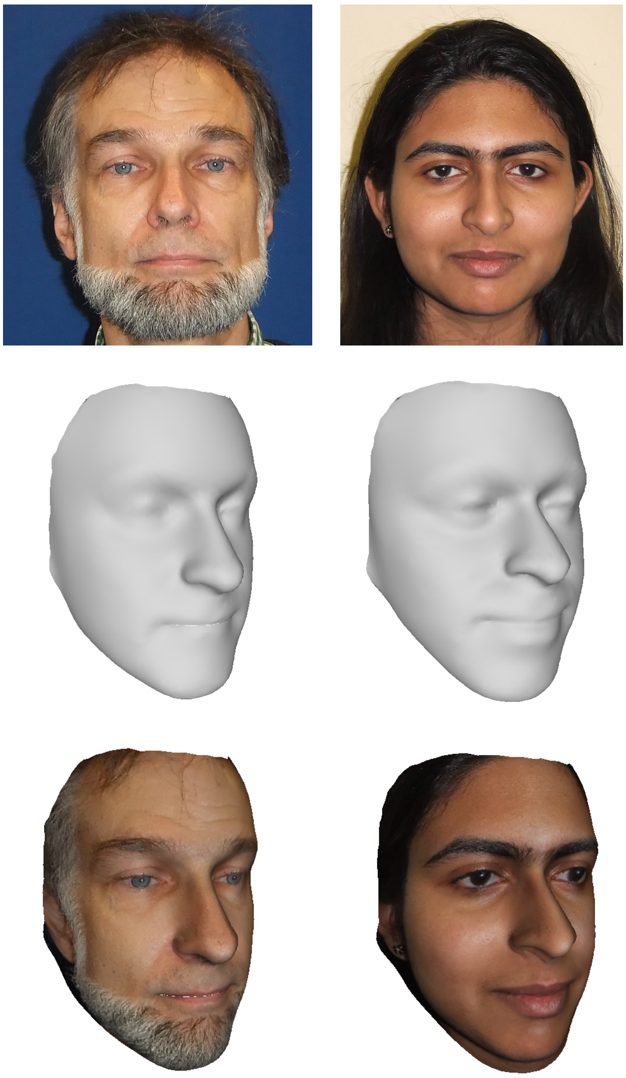


Figure 3.7: Two face models obtained from the images of first row, the center row illustrates the shape model.

result of cloud comparison states that the flatness of the face is similar in face model but is not in reality. The face cloud comparison, along with visual observation shows the need of some deformation in the face model to reach to more accurate shape.

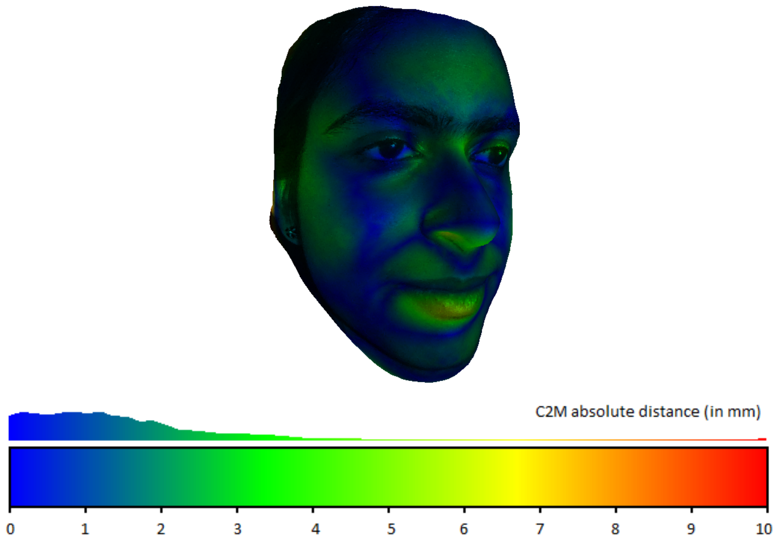


FIGURE 3.8: The two face models obtained from Fig. 3.7 are evaluated for distance between mesh nodes using Cloudcompare[1]. Even for the two distinct faces, the face model is almost similar.

3.5 Stereo Pair Reconstruction

Human being ability to perceive depth comes from viewing an object from two different line of sight assisted by the two eyes. Stereo reconstruction is also based on similar principle with digital images. By comparing information from two vantage points captured from two Charge Coupled Device (CCD) camera, the depth information can be obtained. In stereo vision, two camera which are displaced horizontally from one another are used to obtain two different views of a 3D scene, in a manner similar to human binocular vision. Upon comparing these two images, the relative depth information can be obtained in the form of disparity map.

For human eyes to compare two images, they must be superimposed in a stereoscopic device, such that image from left camera being shown on observer's left eye and similarly for right image. This can be digitally mimicked by capturing the images from a digital stereoscopic camera.

3.5.1 Fujifilm FinePix REAL 3D W3

For stereo reconstruction, the image pair is captured by a commercially available camera Fujifilm FinePix REAL 3D W3. It is worlds's first 3D digital imaging consumer grade camera designed to capture stereoscopic images that can be used to recreate the perception of 3D depth. The camera has a pair of lenses separated by a baseline distance that approximates the distance between an average pair of human eyes[2].

Fujifilm launched the W3 version (used in this work) in August 2010. It has a resolution of 720p. The images captured are saved as a pair of still images in Multi Picture Object (MPO) files. This MPO file can contain more than just two pictures as well. The Real Photo Processor 3D, developed by fujifilm over years, synchronizes image data passed to it by the two lens and two CCD sensor, which instantaneously blends the focus and brightness into a single symmetrical image. The Figure 3.9 shows the stereoscopic camera used in this work, for capturing face images.

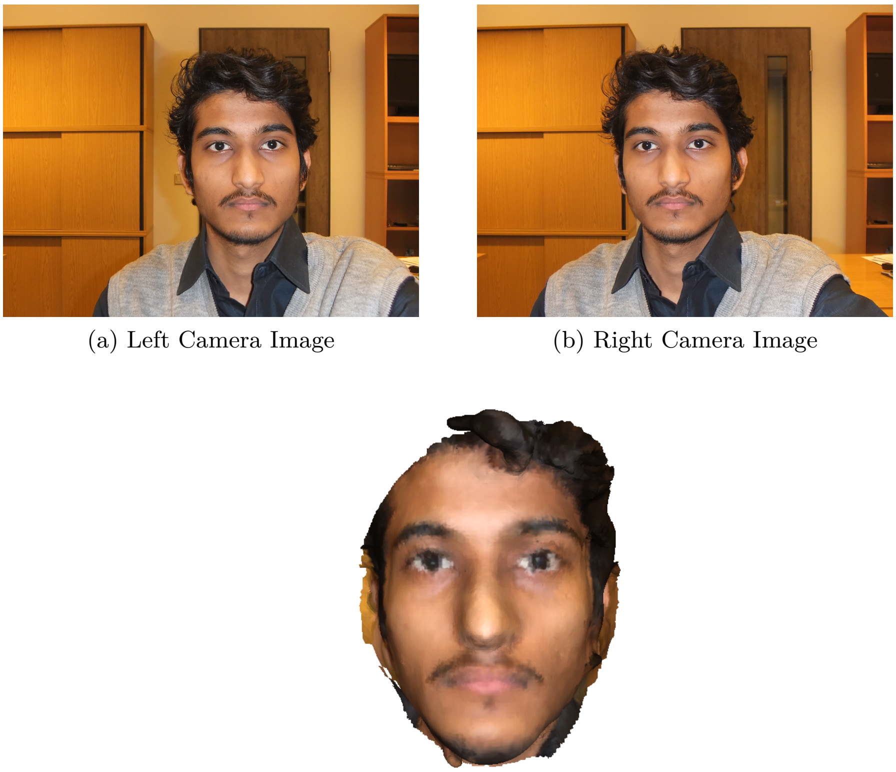
Figure 3.9: FujiFilm FinePix REAL 3D W3 Camera

For the deformation of face model a geometrically rich information is required. The comparison of stereo reconstruction with the high quality scan in Fig. 3.11 shows that even though the stereo reconstruction has deficits in texture and lacks smoothness, it holds sufficient shape information.

3.6 Deformation of Face Model

The face model comparison in Figure 3.8 shows that face model is not geometrically sufficient to discriminate between faces. Deformation on the face model has to be carried out. Stereo pair reconstruction, which is geometrically more appropriate description (justified in Figure 3.11) of the face is used for the deformation. A global and then a local optimization which are based on radial basis function and embedded deformation algorithm respectively are carried out.

[2] `http://www.fujifilm.com/products/3d/camera/finepix_real3dw3/`



(a) Left Camera Image

(b) Right Camera Image

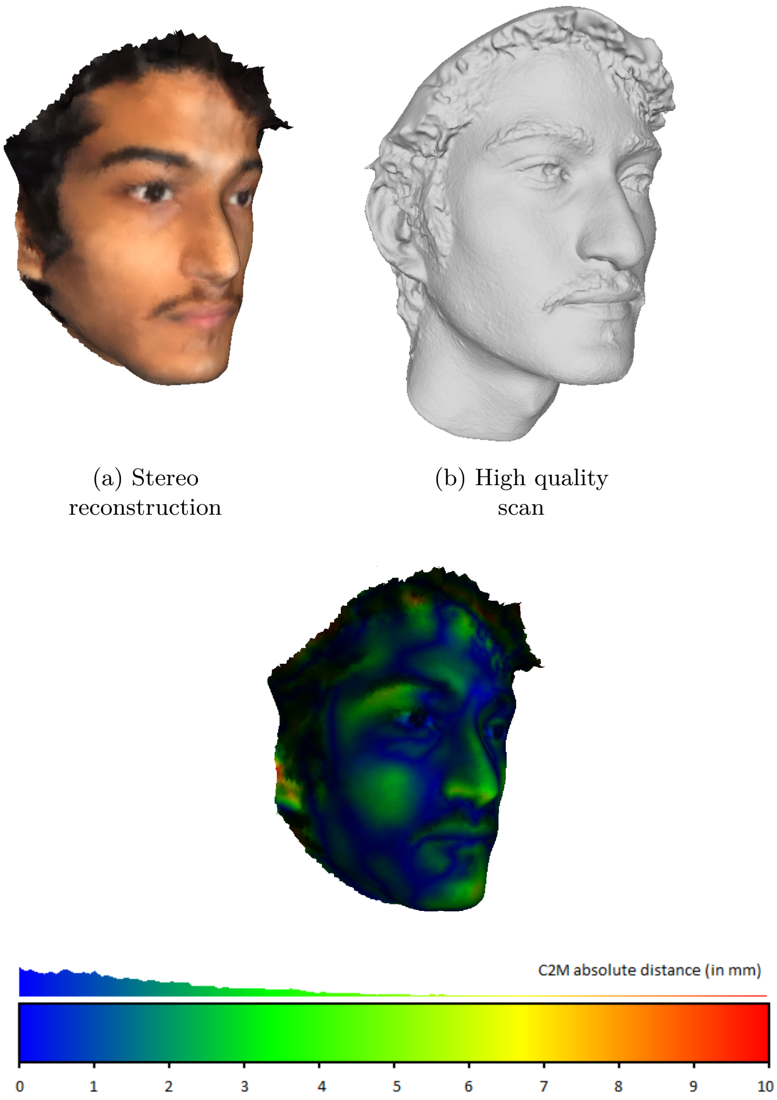
(c) Stereo Reconstruction

FIGURE 3.10: Images obtained from stereo camera and the surface reconstruction obtained from the images.

For the purpose of deformation, the face model is aligned with the its stereo reconstruction by rigid Iterative Closest Point (ICP) algorithm [29]. In the two step non-rigid deformation, the first stage of global deformation is carried out by computing the weights λ_c. The weights can be estimated using inverse matrix multiplication, because the global deformation function in Equation 2.12 is linear in the weights. So the equation for computing weights can be roughly written as $\lambda = \phi^{-1}g$.

Generally, all source vertices have a nearest correspondence on the target. But if the target mesh is incomplete or partial, some of these correspondences may be wrong. For instance, assume a source vertex having a nearest neighbor inside a hole region of the target would now have a correspondence at the boundary vertex of the hole. Let there be a set of all these source vertices which correspond to border vertices on the target. A large portion of this set originates from wrong correspondences, which is why this set is excluded from the source nodes further considered. The remaining (valid) nodes are called *sourcepartial* in the following.

Target correspondences of the *sourcepartial* vertices contribute to the matrix g. The



(a) Stereo reconstruction

(b) High quality scan

(c) Stereo reconstruction and high quality scan cloud comparison color map

FIGURE 3.11: A comparison of stereo reconstruction with the high-quality scan demonstrated that the stereo reconstruction is an appropriate choice of information for deformation estimation.

sourcepartial is sampled to obtain uniformly distributed centers which are used for the computation of ϕ.

$$\begin{bmatrix} \lambda_{r,1} \\ \vdots \\ \lambda_{r,N} \end{bmatrix} = \begin{bmatrix} \phi_{1,1} & \dots & \phi_{1,N} \\ \vdots & \ddots & \vdots \\ \phi_{V,1} & \dots & \phi_{V,N} \end{bmatrix}^{-1} \begin{bmatrix} g_{r,1} \\ \vdots \\ g_{r,V} \end{bmatrix} \tag{3.24}$$

where $\phi_{i,c}$ is solved from Equation 2.13 for $i \in \{1, 2, \dots\}$ and N centers. The Equation 3.24 has to be solved thrice for $r = (x, y, z)$ resulting in $3N$ weights.

These weights are used in Equation 2.12 for all the source vertices to obtain global optimization. The value of N has significant impact on the global deformation. Practically it was observed that for $N = 25$ good face-unspecific results were obtained. When implemented in MATLAB, the global deformation computation took few hundreds milliseconds on an Intel quad core computer.

In local deformation, $\mathbf{B}_i$ is a 3×3 matrix and $\mathbf{t}_i$ a 3 element vector that collectively represent an affine transformation. The affine non-rigid transformation for each source vertex with respect to its nearest correspondence on target is computed using Procrustes analysis. $k = 12$ nearest neighbors are chosen, whose weights are calculated from equation 2.16. The non-rigid transformation for each source vertices is estimated as formulated in Equation 2.14. Local deformation has quantitatively lesser impact on the deformation but is important to smoothen the deformed face model. It also consumes a significant amount of time compared to global deformation.

For the Visualization of this corrected model, an app is developed with the deformed face model as the input. The face model is fed in wavefront .obj file format with 2D texture map.

3.7 Visualization APK

Developing a visualization application usually takes lot of time and needs professional knowledge of software. As the android development was not the focus of this work, some prebuilt SDKs were used to accomplish the visualization on the smartphone device.

For our framework the Unity3D is much easier to use as it contains graphic assets, specific handling of user input, animations. There is a game editor, which enables to change the game property directly and observe the outcomes immediately.

The Unity game scene of the apk had following objects:

- **Plane**: A virtual plane on which the deformed face model is placed.
- **Point Light** : The light source used for illumination of game object (model). It is placed above the camera so that the section of model visible to camera is always illuminated.
- **Camera Right** : The camera which shows the model to the right eye.
- **Camera Left** : The camera which shows the model to the left eye.

- **Game Object** : The deformed face model in wavefront object file format. A script is attached to this object which has transformation details to visualize the complete model on a single screen.

The wavefront .obj file was added as a game object along with the 2D texture as a material file. The size of the file has to be adjusted so that the whole face can be observed on a part of split screen. In the scene, the light is placed above the camera, so that parts of object exposed to the viewer are properly illuminated. The same can be seen in the screenshot of Unity3D game scene in figure 3.12

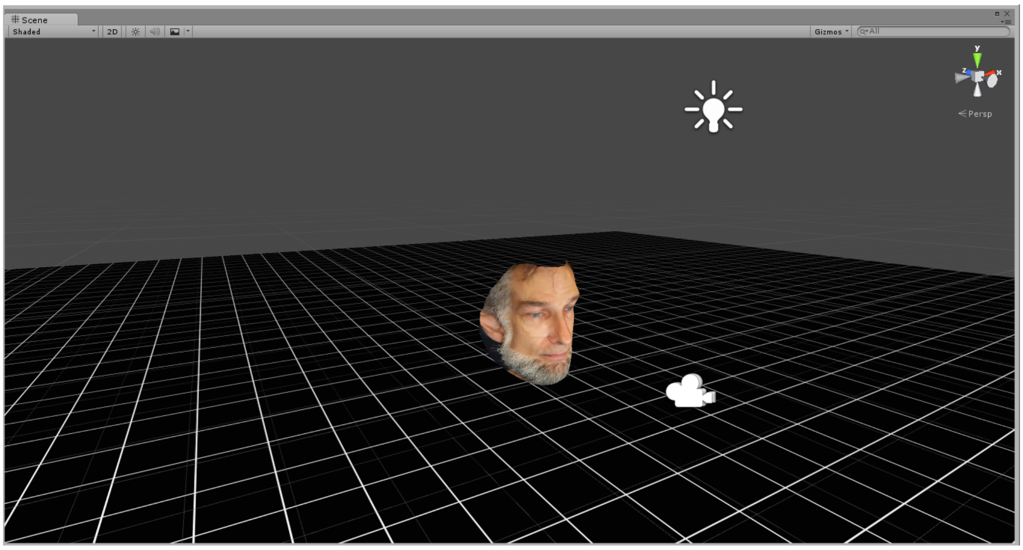


Figure 3.12: The game scene of Unity3D.

Chapter 4

Results and Discussions

4.1 Single Image Reconstruction

The single image reconstruction output is shown in Figure 3.6. The face model is obtained by landmarking face image, and sequentially following pose estimation, shape fitting to get the shape information. The texture is extracted from face image by isomap algorithm. Observing the results of Figure 3.7 separately i.e. the left column and right column reconstruction individually, the shape models seems to hold the sufficient 3D information of the face. The accomplished face model with overlapped texture pretend to be exact conformation of the face.

But the moment both the face models are compared, the visual accuracy of the model becomes questionable. The faces differ significantly in aspect ratio, size of nose and roundness of face and forehead, the same is not carried to the model. The nose in the shape model is also same but its not the ground truth. By human perception of visualisation, the two faces images cannot have similar 3D model. The visual observation is supported by face mesh comparison in fig 3.8, where a large amount of 3D points lie close to each other. These investigations bring us to the need of modification in the shape of the face model so as to adapt to the actual 3D structure.

4.2 Qualitative Analysis

Figure 4.1 shows the results after adapting stereo reconstruction to the morphable model. The correct difference in the aspect ratio of two face models depicts the accuracy visually. Deformed Face model is visibly more identical to the image as compared to the face model.

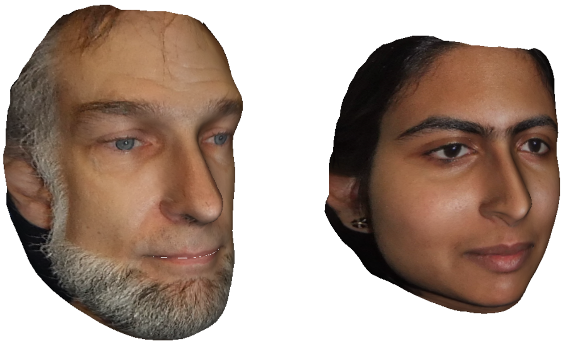
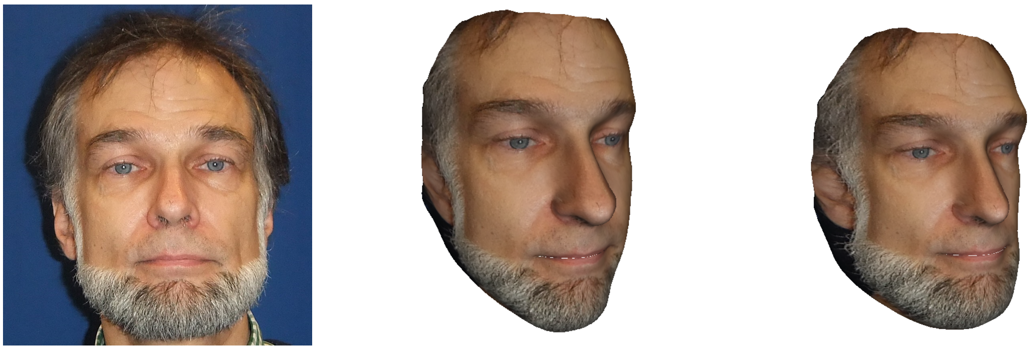


FIGURE 4.1: Deformed face model for faces in Fig. 3.7

In order to have a fair comparison, the results of face reconstruction from single image and from our method is mentioned in Figure 4.2. Our method depicts the face roundness near the beard in a proper form. The nose which was earlier bigger in face model is appropriate in the deformed face model. The cheeks in the deformed face model are more flat, which obeys the shape information from the stereo reconstruction. Qualitatively, the deformed face model which is obtained by adapting stereo reconstruction to the face model is a more accurate 3D face description. The results are also supported by quantitative analysis, i.e. comparing the various stages of reconstruction with the High quality face scan.



(a) Face Image

(b) Face model obtained by P. Huber et al. [10]

(c) Deformed model by our approach

FIGURE 4.2: Comparison of Face Model with Deformed Face Model

4.3 Comparison with High Quality Scans

Figures 4.3, 4.4 and 4.5 show a whole bunch of comparison of various different face reconstructions. The top row shows the references, the face image and 3D high quality face scan. The second row depicts the face model, stereo model and deformed model from left to right respectively. All the models are aligned in same 3D orientation for visual comparison. After comparison of all models with the high quality scan, the third row shows the models in different orientation and overlapped color map. Histograms obtained by computing distances are shown in last row.

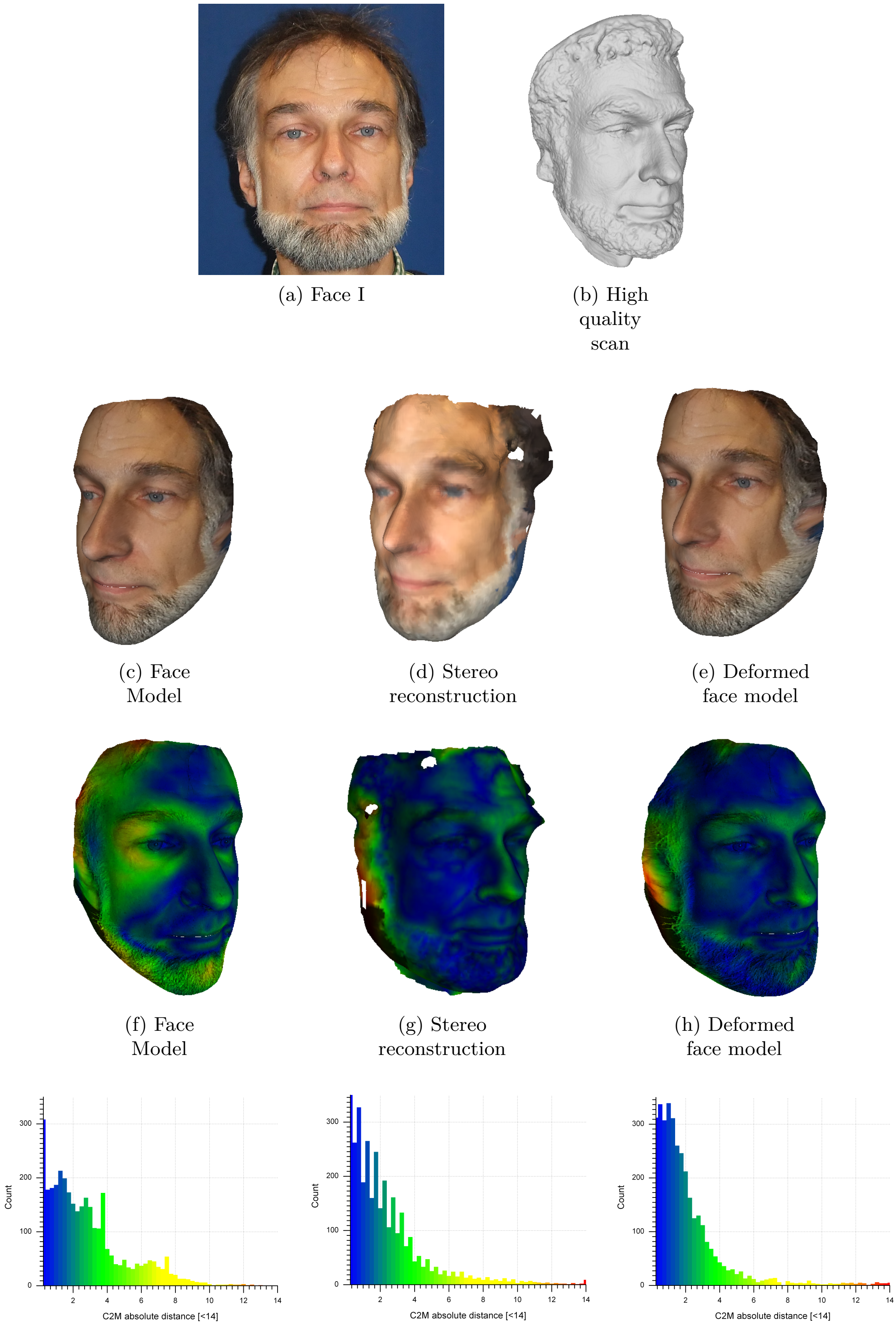
The histogram of distances of cloud points in the face model is more spread up. The stereo model has better shape information compared to the face model, as the histogram is less spread. The deformed face model, has majority of pixel distances concentrated to the origin. The deformed face model even has few points at a large distance which lie on the edges of model and have lesser significance.

A quantised description of improvement in the reconstruction is mentioned in table 4.1. It shows the Root-mean-square error of face model, stereo reconstruction, global deformation of face model and the combined deformation (all are aligned and compared to the high quality scan).

	Single image Reconstruction	Stereo pair Reconstruction	Global Deformation	Global + Local Deformation
Face I	3.5532	3.3073	3.0315	2.8738
Face II	4.2734	2.9985	2.2674	2.0133
Face III	2.9982	2.5212	2.3728	2.174

Table 4.1: RMS cloud to cloud distance of Models and High Quality Scan, in all the stages.

In Table 4.1 we observe that RMS error for all the three faces decreases from left to right meaning that the output of our method has smallest root mean square error than both, the shape from single image and the stereo reconstruction methods. The line chart of table 4.1 in figure 4.6 shows a decreasing value in all the cases.



(a) Face I

(b) High quality scan

(c) Face Model

(d) Stereo reconstruction

(e) Deformed face model

(f) Face Model

(g) Stereo reconstruction

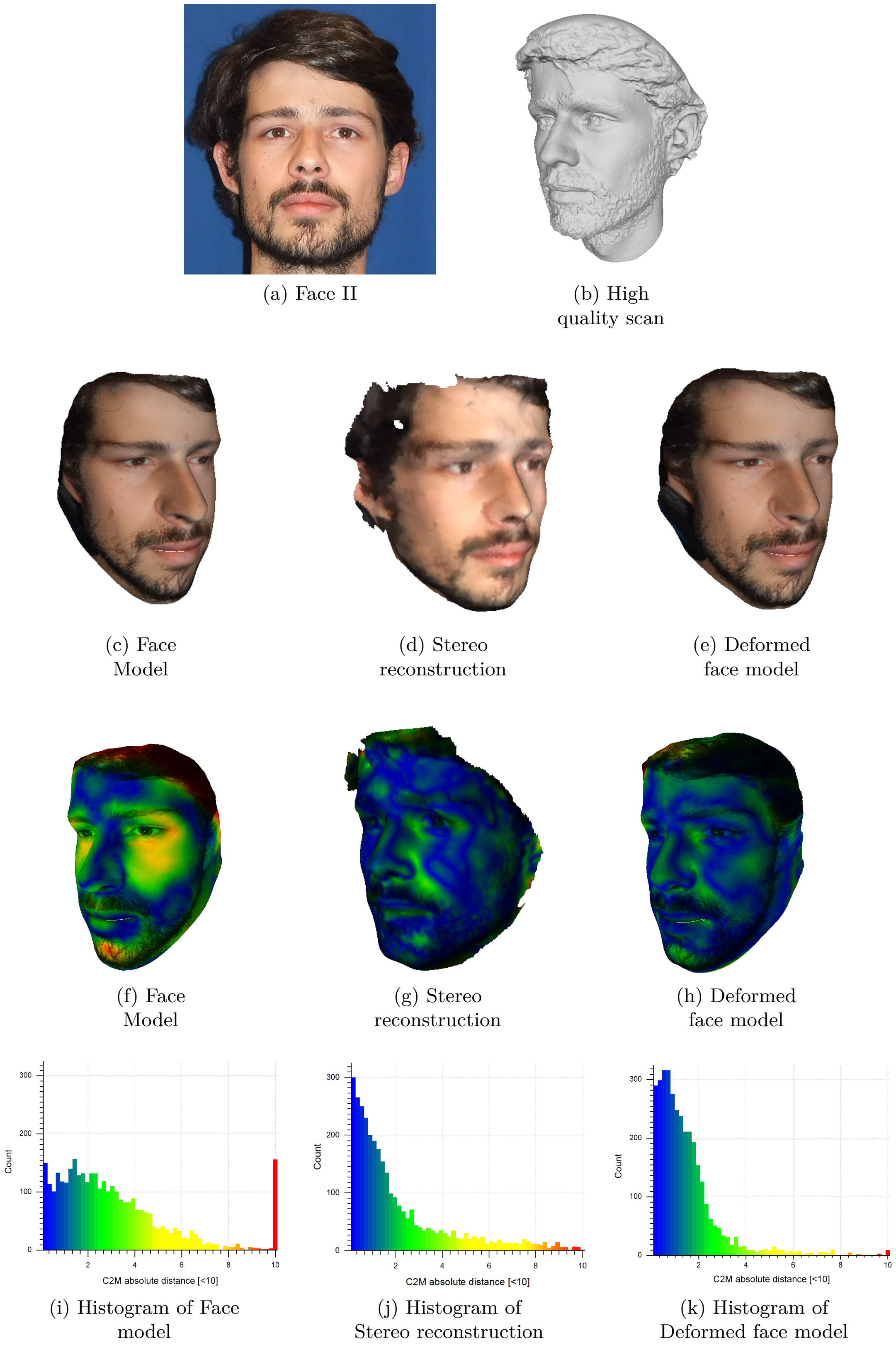
(h) Deformed face model

(i) Histogram of Face model

(j) Histogram of Stereo reconstruction

(k) Histogram of Deformed face model

FIGURE 4.3: The face and its face model, stereo reconstruction, deformed face model with superimposed color map showing the cloud to mesh distance from high-quality scan in the respective histograms.



(a) Face II

(b) High quality scan

(c) Face Model

(d) Stereo reconstruction

(e) Deformed face model

(f) Face Model

(g) Stereo reconstruction

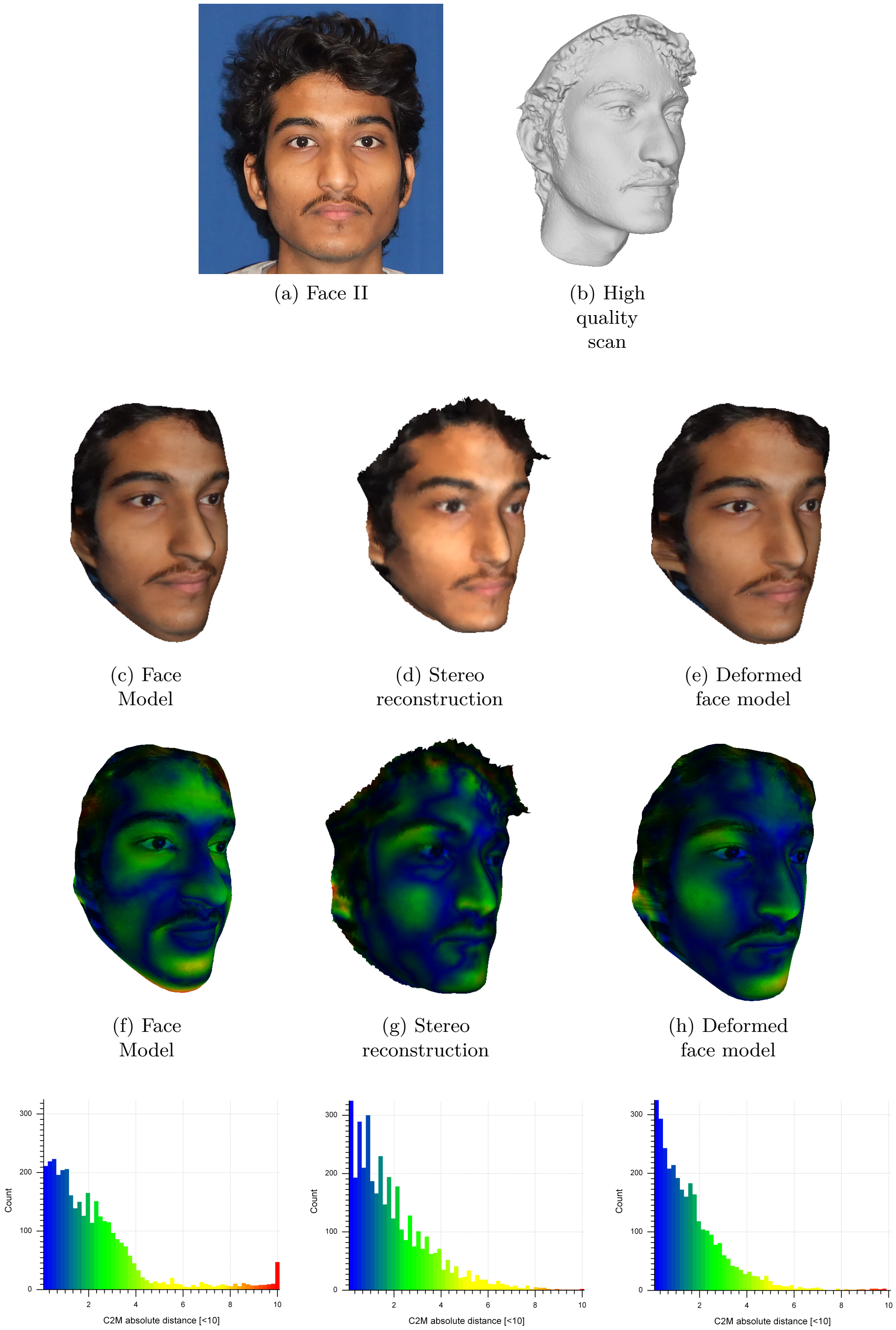
(h) Deformed face model

(i) Histogram of Face model

(j) Histogram of Stereo reconstruction

(k) Histogram of Deformed face model

FIGURE 4.4: Same analysis as for Face II



(a) Face II

(b) High quality scan

(c) Face Model

(d) Stereo reconstruction

(e) Deformed face model

(f) Face Model

(g) Stereo reconstruction

(h) Deformed face model

(i) Histogram of Face model

(j) Histogram of Stereo reconstruction

(k) Histogram of Deformed face model

Figure 4.5: Same analysis as for Face III

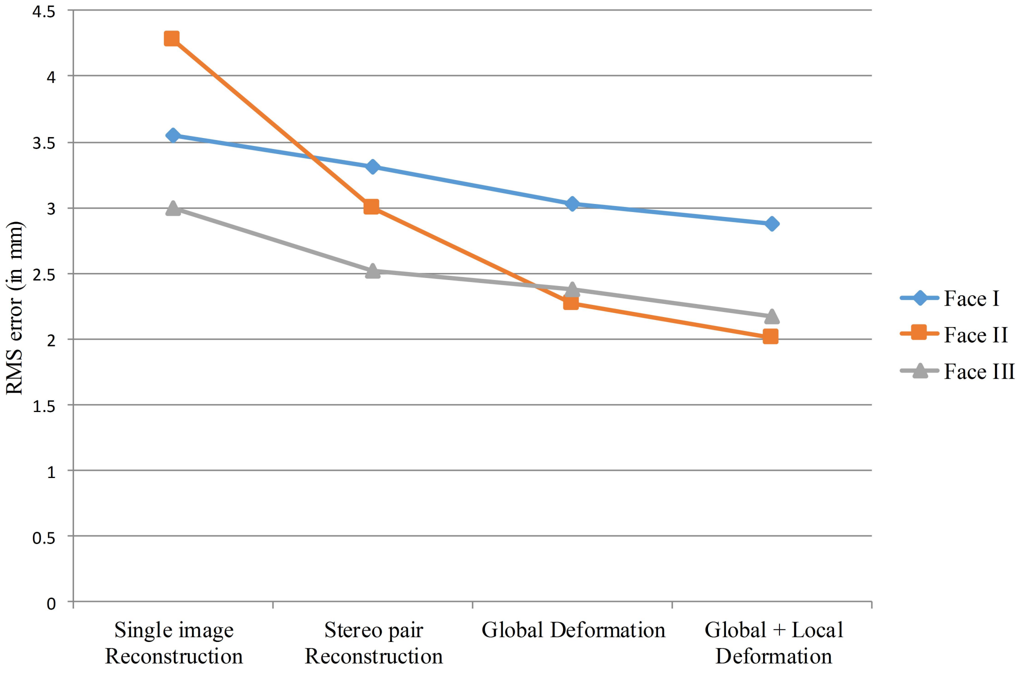


FIGURE 4.6: RMS error for different stages for faces(I-III)

More results are shown in the Fig. 4.8. The middle Column of which is face model and are all similar in appearance. The deformed model, which has been adapted to the stereo reconstruction, is descriptively more correct representation than the face model. Table 4.2 show the average distance of models of Fig. 4.8 from stereo reconstruction. A line chart in figure 4.7 is shown for data of table 4.2

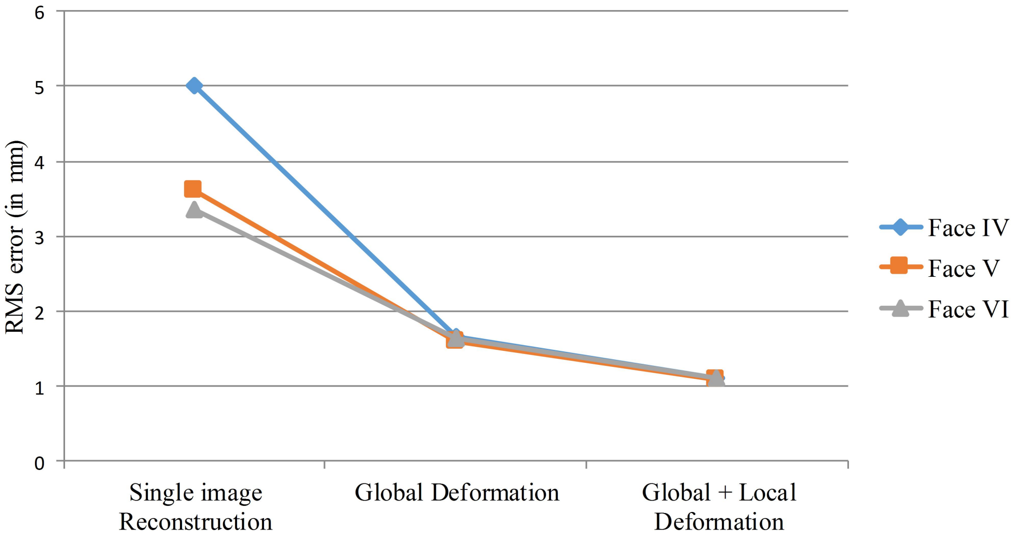


FIGURE 4.7: RMS error for different stages for faces(IV-VI)

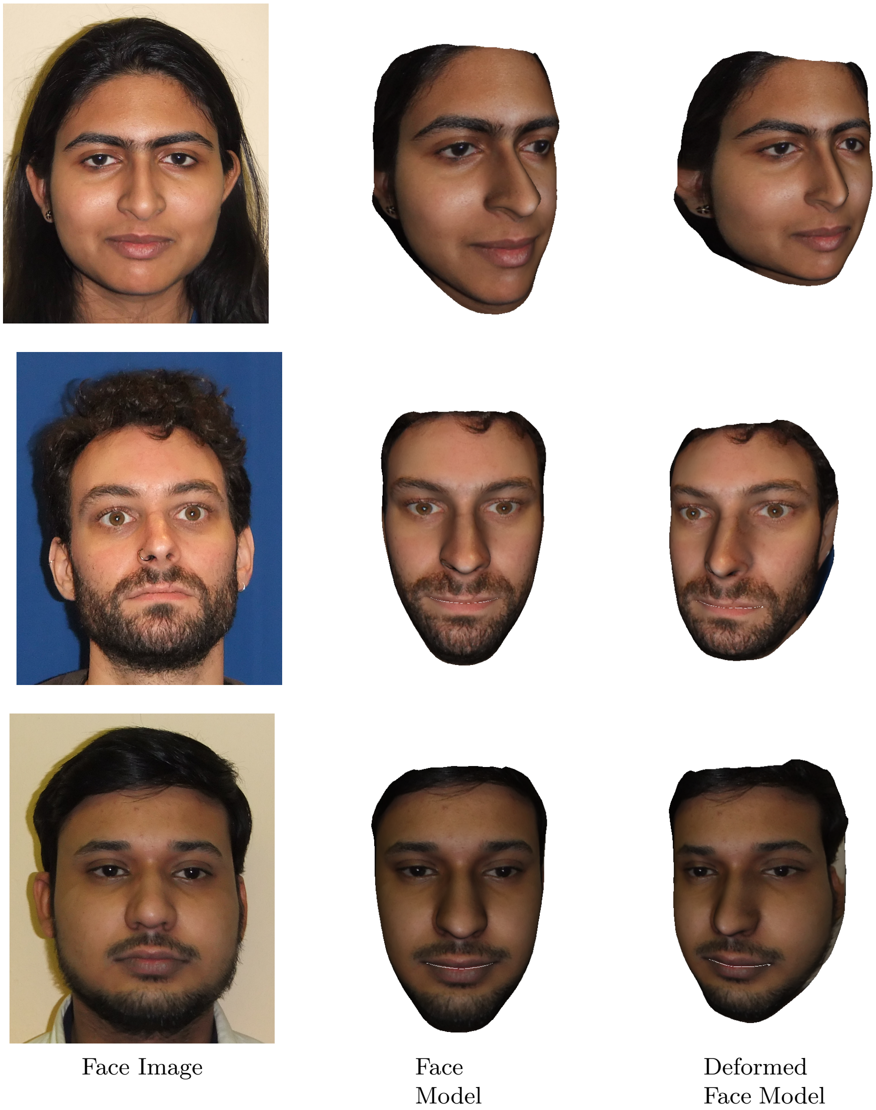


Figure 4.8: Comparative Visualization of face model and deformed face model for respective face images, faces numbers from top to bottom (Face IV - VI).

	Initial Alignment	Global Deformation	Global + Local Deformation
Face IV	5.0179	1.6681	1.0993
Face V	3.6053	1.6046	1.08
Face VI	3.3479	1.6357	1.1013

Table 4.2: Average cloud to cloud distance of the mesh from the stereo reconstruction, in all three stages.

4.4 3D Visualization on Cardboard Viewer

The resultant model is visually presented using cardboard viewer for inspection. A smartphone screenshot of android application in operation is shown in figure 4.9, where split screen view can be observed. To consider the radial distortion, the edges and lines are curved. The smartphone is placed in the groove provided on cardboard viewer. When the cardboard viewer (with smartphone) is mounted on head, the lens on cardboard focuses the split screen view on eyes. The viewer gives a 3D perception of the deformed face model.

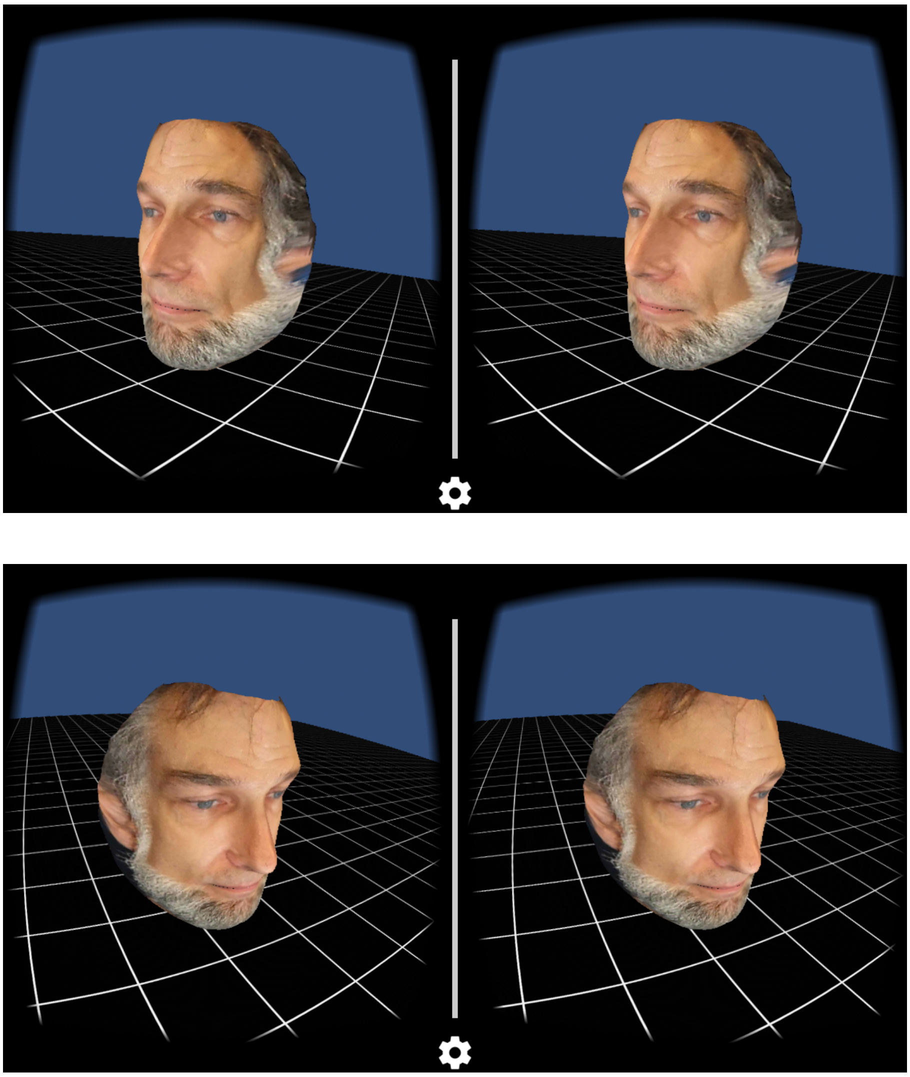


FIGURE 4.9: Screenshots when the android app is in operation

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis investigates 3D face reconstruction from a stereo image pair compared to 3D face reconstruction from single image making use of generic face shape information. Single image reconstruction has been considered for use with various models, BFM was found to be the most suitable and hence implemented in this work.

The algorithm for single-image based reconstruction uses local features at few landmark positions on the face image. Though generally showing noise-free human-like shape features, the resulting reconstruction lacks the correct geometric information of the individual face, which is difficult to obtain from a single image. Despite the fact that the stereo pair reconstruction has lots of local deficits and gives improper texture information, it possess correct geometric information about the individual person. This geometrically suitable reconstruction from stereo pair is improved by fusing information from the face model.

Approach identical to surface registration is used for shape information fusion. A two stage embedded deformation including global as well as local transformation has been exercised. The combined effect significantly adapts the stereo reconstruction to the face model, thereby eliminating the holes of stereo reconstruction and reducing the shape discrepancies.

Qualitative and quantitative analysis of the deformed face model show that this combination of stereo reconstruction with general shape information about human faces is geometrically superior to both the reconstruction based on single image from generic model as well as the stereo reconstruction without consideration of the generic model. The advantage of the method is improved surface reconstruction with minimal hardware requirement of stereo camera. The framework is useful for applications which requires accurate face model for visualization.

In combination with computer performance, the rendering of such perspective views, the quality of materials used for object texturing, and the lighting of the scenes play crucial roles in obtaining photo-realistic visualization. The Unity3D scene was developed, in which the resultant model was appropriately placed and illuminated by providing light source above the camera. Finally, the work highlights the modern trend of using smartphone as a head mounted display.

5.2 Scope for Future Work

The small contribution of thesis to the field of face reconstruction provide the foundation for a number of research directions in the future. This work could also be used for the reconstruction of various different objects or human organs, with better geometric information. Morphable models for different class of animal faces can be created and used for preservation and biological studies.

Moreover the smartphone Visualization environment can be improved by placing multiple faces in the game zone and instead of restricting the camera movements around the face, it could be made free to roam all over the space.

The future work also includes gaining permissions from the model owners to upload the android application on Google Play Store, motivating further improvement in the domain.

Publication

Hardik Jain, Olaf Hellwich, R S Anand, ”Improving 3D Face Geometry by Adapting Reconstruction From Stereo Image Pair to Generic Morphable Model”, in *2016 19th International Conference on Information Fusion (FUSION 2016)*, (Heidelberg, Germany), July 2016 (Accepted for Presentation).

Bibliography

[1] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 052162304, first ed., 2000.

[2] F. Bellocchio, N. A. Borghese, S. Ferrari, and V. Piuri, *3D Surface Reconstruction: Multi-Scale Hierarchical Approaches.* Springer Science & Business Media, 2012.

[3] P. Huber, Z.-H. Feng, W. Christmas, J. Kittler, and M. Rätsch, "Fitting 3D morphable models using local features," *arXiv preprint arXiv:1503.02330*, 2015.

[4] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187–194, ACM Press/Addison-Wesley Publishing Co., 1999.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 681–685, 2001.

[6] X. Liu, P. H. Tu, and F. W. Wheeler, "Face model fitting on low resolution images," in *BMVC*, vol. 6, pp. 1079–1088, 2006.

[7] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference On*, pp. 296–301, IEEE, 2009.

[8] R. van Rootseler, L. Spreeuwers, and R. Veldhuis, "Using 3D morphable models for face recognition in video," in *Proceedings of the 33rd WIC Symposium on Information Theory in the Benelux*, (Enschede, the Netherlands), pp. 235–242, Werkgemeenschap voor Informatie- en Communicatietheorie, WIC, May 2012.

[9] S. Romdhani, V. Blanz, and T. Vetter, "Face identification by fitting a 3D morphable model using linear shape and texture error functions," in *Computer Vision—ECCV 2002*, pp. 3–19, Springer, 2002.

[10] P. Huber, G. Hu, R. Tena, P. Mortazavian, W. P. Koppen, W. Christmas, M. Rätsch, and J. Kittler, "A multiresolution 3D morphable face model and fitting framework," in *11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, February 2016.

[11] S. Romdhani and T. Vetter, "Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 986–993, IEEE, 2005.

[12] J. R. Tena, R. S. Smith, M. Hamouz, J. Kittler, A. Hilton, and J. Illingworth, "2D face pose normalisation using a 3D morphable model," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pp. 51–56, IEEE, 2007.

[13] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1867–1874, IEEE, 2014.

[14] J. Tena Rodríguez, *3D Face Modelling for 2D+3D Face Recognition.* PhD thesis, University of Surrey, 2007.

[15] J. R. Tena, M. Hamouz, A. Hilton, and J. Illingworth, "A validated method for dense non-rigid 3D face registration," in *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pp. 81–81, IEEE, 2006.

[16] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[17] P. Zemcik, M. Spanel, P. Krsek, and M. Richter, "Methods of 3D object shape acquisition," *3-D Surface Geometry and Reconstruction: Developing Concepts and Applications: Developing Concepts and Applications*, p. 1, 2012.

[18] H. Li, R. W. Sumner, and M. Pauly, "Global correspondence optimization for non-rigid registration of depth scans," in *Computer graphics forum*, vol. 27, pp. 1421–1430, Wiley Online Library, 2008.

[19] K. Anjyo and J. Lewis, “RBF interpolation and gaussian process regression through an rkhs formulation,” *Journal of Math-for-Industry*, vol. 3, no. 6, pp. 63–71, 2011.

[20] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, “Reconstruction and representation of 3D objects with radial basis functions,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 67–76, ACM, 2001.

[21] Z. Levi and D. Levin, “Shape deformation via interior rbf,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 7, pp. 1062–1075, 2014.

[22] K. Anjyo, J. P. Lewis, and F. Pighin, “Scattered data interpolation for computer graphics,” in *ACM SIGGRAPH 2014 Courses*, p. 27, ACM, 2014.

[23] G. E. Fasshauer, *Meshfree approximation methods with MATLAB*, vol. 6. World Scientific, 2007.

[24] M. D. Buhmann, “Radial basis functions: theory and implementations,” *Cambridge monographs on applied and computational mathematics*, vol. 12, pp. 147–165, 2004.

[25] R. W. Sumner, J. Schmid, and M. Pauly, “Embedded deformation for shape manipulation,” *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 80, 2007.

[26] D. G. Kendall, “A survey of the statistical theory of shape,” *Statistical Science*, pp. 87–99, 1989.

[27] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 397–403, 2013.

[28] D. E. King, “Max-margin object detection,” *arXiv preprint arXiv:1502.00046*, 2015.

[29] P. J. Besl and H. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, Feb 1992.