MEASURING 3D FACE GEOMETRY FOR INTEGRATION WITH APPEARANCE MODELS

BY SIDDHARTH K. MADAN

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Dr. Kristin J. Dana

and approved by

____

____

____

New Brunswick, New Jersey

May, 2008

ABSTRACT OF THE THESIS

Measuring 3D Face Geometry for Integration with Appearance Models

by Siddharth K. Madan

Thesis Director: Dr. Kristin J. Dana

Capturing the appearance of skin is a nontrivial problem. The appearance of skin depends significantly on the illumination and viewing direction. The global and fine scale local geometry influence the appearance of skin. In this thesis, we work towards capturing the appearance of skin by acquiring the global geometry and modeling the skin texture.

We explore the use of structured light techniques to acquire global geometry. Two novel structured light techniques have been developed. The techniques use image ratios to obtain reliable reconstruction of low albedo regions. The use of laser scanner for geometry acquisition and it's advantages and disadvantages have also been discussed.

We model the interaction of light with the skin surface as a combination of surface and subsurface scattering. The reflectance due to subsurface scattering and the reflectance due to surface scattering have different properties and are modeled separately. The subsurface components lie on a lower dimensional linear subspace and the surface components lie on nonlinear manifolds. We look into the use of linear dimensionality reduction techniques to model the subsurface reflectance and nonlinear dimensionality reduction techniques to model the surface reflectance. In the initial results, separate modeling of the surface and the subsurface components preserved the specularities.

However, the specularities were lost when the reflectance were modeled without separating the surface and the subsurface components.

Acknowledgements

I would like to thank my advisor Dr. Kristin Dana for her invaluable guidance throughout the thesis. Countless discussion sessions with her have played an indispensable role in shaping my thesis and my outlook towards research. I would like to thank her for her time and patience throughout the research.

I would like to take this opportunity to thank my colleague Alexander Wu for initiating me to the work being done in the lab. It was great working with Alex during the initial stages of the structured light system. His constant optimism and search for new ideas are truly inspirational.

Table of Contents

List of Figures

Chapter 1

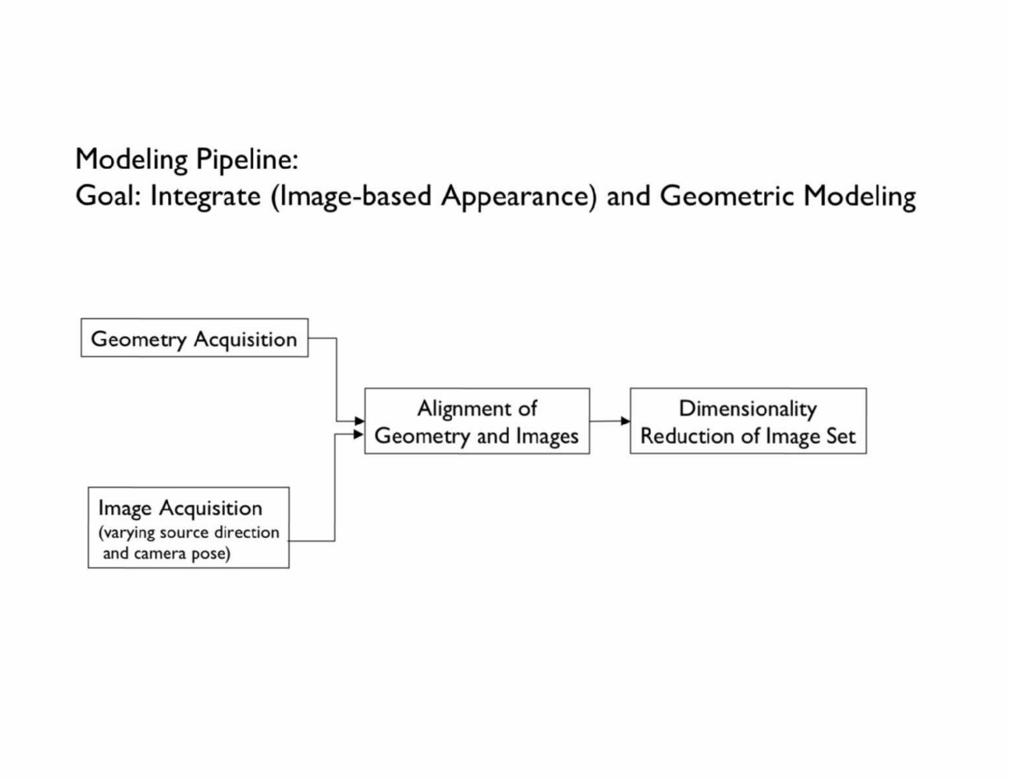
Geometry Acquisition Using Structured Light

1.1 Geometry for Appearance Modeling

Integrating appearance and geometry is an important goal for computer modeling of people, scenes and objects. A simple camera can be viewed as a scientific measurement device (assuming careful photometric calibration and no-loss compression) where the quantity being measured is appearance. Multiple images of the same object are samples of appearance that need to be coordinated in order to support analysis. Specifically, the images from multiple imaging parameters should be aligned so that the 3D spatial coordinates of every pixel-based measurement is known. The imaging parameters are source direction, camera pose and time (because objects may be monitored temporally). We seek a modeling approach such that every point on the surface is imaged under varying viewing and illumination directions by taking a series of images as the object, camera and light source are moved. The collection of images provides several measurement values for each point on the surface. By integrating appearance and geometry in this manner, we have a sampling of the BRDF (bidirectional reflecance distribution function) at every point on the surface. This framework is a powerful approach to modeling because it allows us to do methods of analysis that would not be possible on the raw images, e.g., locally modeling BRDF, clustering pixels based on appearance, and reducing dimensionality of the image set. The pipeline is shown in figure 1.1.

1.2 Active and Passive Stereo

Geometry can be acquired using active and passive stereo techniques. In passive stereo no interaction with the scene takes place. Two cameras are used to obtain the image



Modeling Pipeline:
Goal: Integrate (Image-based Appearance) and Geometric Modeling

Geometry Acquisition

Image Acquisition
(varying source direction
and camera pose)

Alignment of
Geometry and Images

Dimensionality
Reduction of Image Set

Figure 1.1: The goal is to integrate geometric modeling with image based appearance.

of the scene from two different viewpoints. As shown in Figure 1.2, a triangulation [16] problem is solved to determine the 3D coordinates of the scene points. Passive stereo relies on the extraction of a large number of corresponding feature points to obtain a reliable reconstruction of the scene. However, in many applications the scene to be reconstructed may contain sparse feature points. Figure 1.3 shows examples of scenes where very few features are available.

For scenes with few feature points, active stereo techniques are better suited to extract the 3D coordinates. In active stereo, interaction with the scene takes place. In particular, the scene is illuminated in a controlled way that creates features on the object. One way to obtain the 3D coordinates using active stereo techniques is to project light at one scene point at a time. The scene is imaged using two cameras, the point illuminated by the scanning system is detected and the 3D coordinates are extracted using triangulation. However, reconstruction by illuminating one scene point at a time is very slow. The reconstruction procedure can be accelerated by illuminating

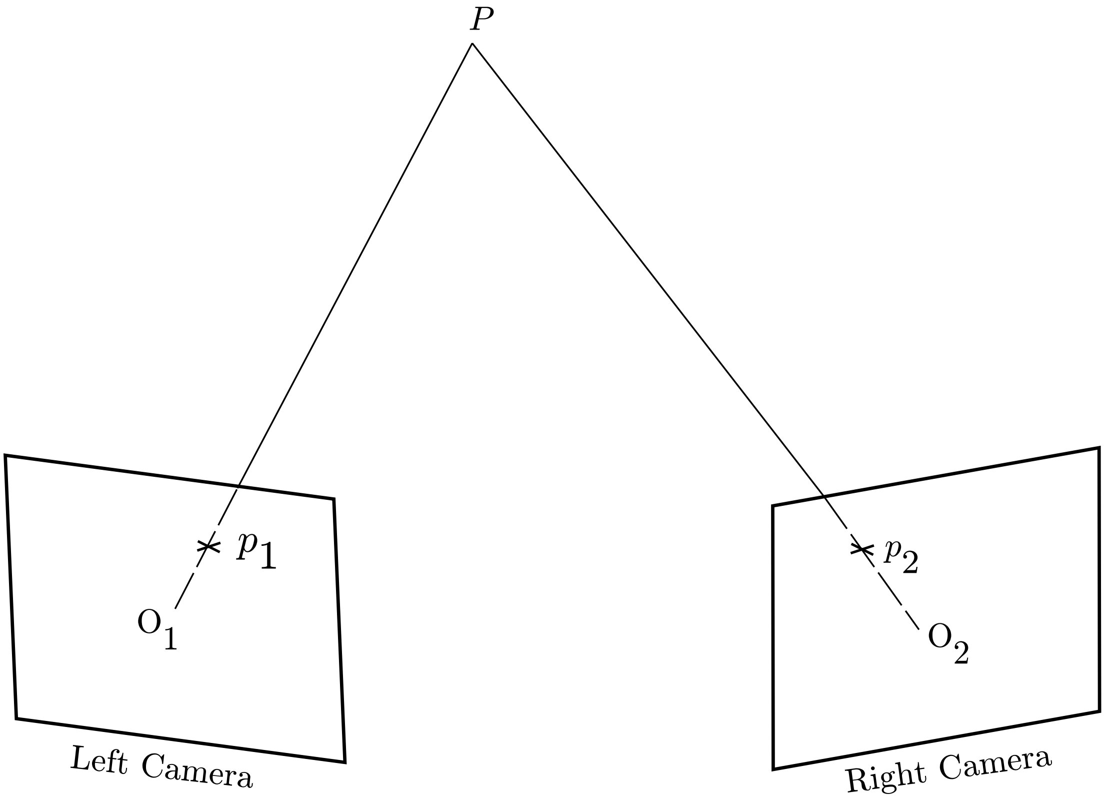


Figure 1.2: Determination of 3D coordinates by Triangulation. P is the world point. p_1 is the pixel location in the left camera. p_2 is the pixel location in the right camera.

multiple points simultaneously. This idea is embodied in structured light techniques. The 3D coordinates of the scene are obtained by triangulating between a camera and a projector [10]. Obtaining the 3D coordinates of the scene using structured light techniques consists of three main steps: calibration, encoding and spline smoothing. The calibration step determines the position of the camera and the projector with respect to the world coordinate frame. The encoding scheme determines the way in which the information about the projector coordinate is encoded in the projected patterns. The spline smoothing step is used to obtain sub–pixel accuracy from the recovered projector coordinates.

1.3 Reconstruction with Structured Light

Structured light techniques [31] use a camera and a light source to obtain the 3D coordinates of the points of the scene. In the imaging apparatus, the structured light

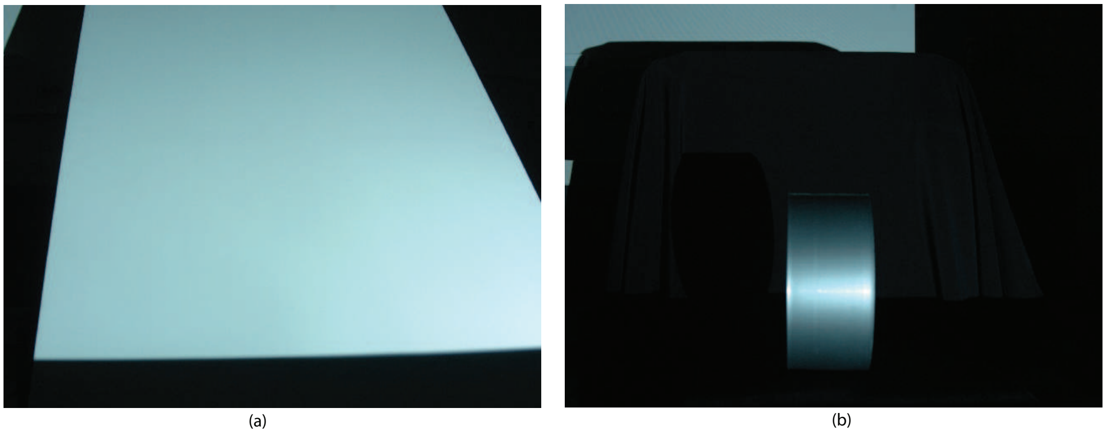


Figure 1.3: Scenes with sparse feature points. (a): Image of a white plane placed in an inclined position. (b): Image of a roll of packaging tape.

source is a projector. The projector projects a pattern of light on the scene to be reconstructed. The image of the scene under the projected light pattern is acquired by the camera. As shown in Figure 1.4 both the camera and the projector are modeled using a pin hole model. The world point lies on the ray joining the camera center and the pixel location of the world point. The world point also lies on the on a vertical plane in the projector coordinate system.

The vertical plane is uniquely defined by the x coordinate of the coordinate system associated with the projector's plane. Once the x coordinate in the projector reference frame and the pixel location are known the 3D coordinate can be obtained by simple triangulation. Ignoring the lens distortion, the transformation from the world coordinates to the pixel locations can be modeled as a linear transformation [5]. Let $\mathbf{C}_c$ be the 3×4 calibration matrix which transforms the homogeneous world coordinate $\mathbf{X}_w$ of the point to the homogeneous pixel location $\mathbf{X}_c$.

$$\mathbf{X}_c = \mathbf{C}_c \mathbf{X}_w, \tag{1.1}$$

where,

$$\mathbf{C}_c = \begin{bmatrix} f_x & kf_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \end{bmatrix}.$$

The parameters f_x and f_y are the focal length of the camera scaled by the pixel size in the x and y direction respectively, c_x and c_y are the origin of the coordinate system in

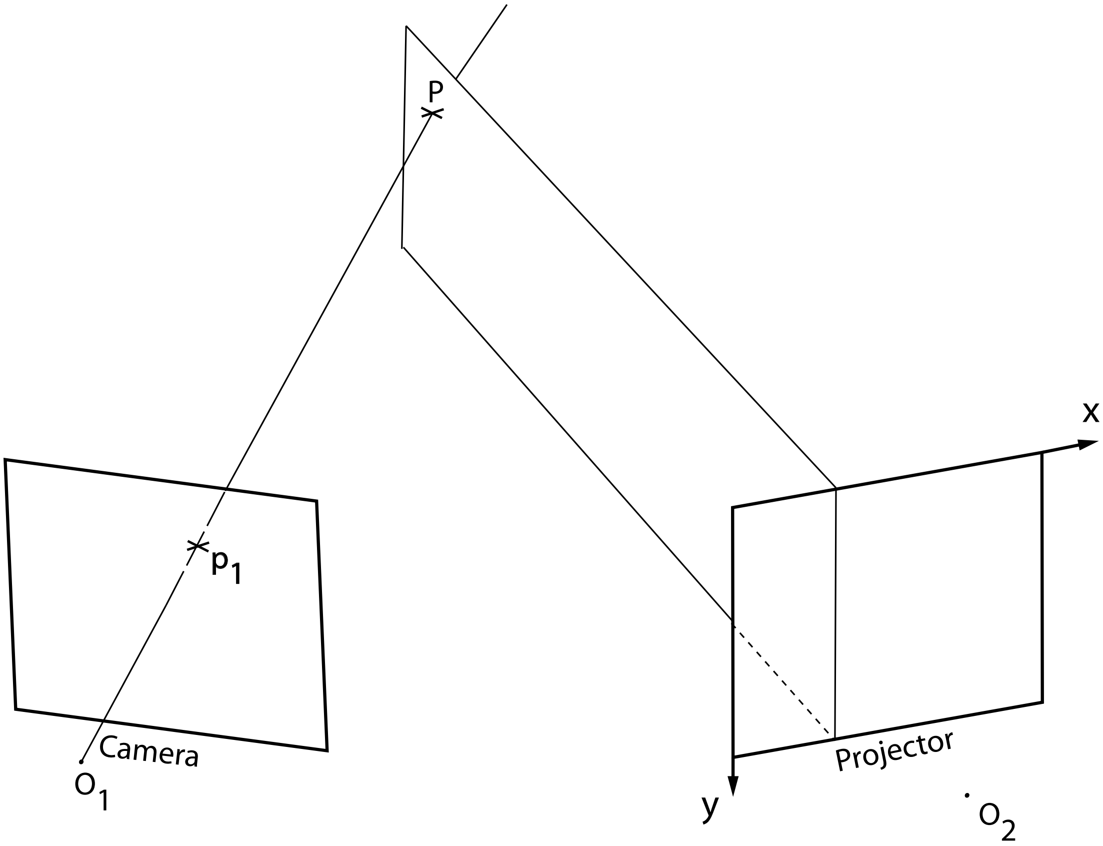


Figure 1.4: Triangulation between a camera and plane. P is the world point.

the image coordinate frame and k denotes the shear of the camera coordinate system. The transformation from the world coordinate system to the camera coordinate system is defined by 3×3 rotation matrix $\mathbf{R}_c$ and the 3×1 translation vector $\mathbf{t}_c$. Let $\mathbf{C}_p$ be the 2×4 linear transformation from the homogeneous world coordinates $\mathbf{X}_w$ to the homogeneous projector coordinate $\mathbf{X}_p$ given by

$$\mathbf{X}_p = \mathbf{C}_p \mathbf{X}_w, \tag{1.2}$$

where,

$$\mathbf{C}_p = \begin{bmatrix} f_p & 0 & c_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_p & \mathbf{t}_p \end{bmatrix}.$$

f_p is the projector focal length scaled by the LCD dimensions and c_p is the origin in the projector coordinates. The transformation between the world coordinate frame and the projector coordinate frame is defined by 3×3 rotation matrix $\mathbf{R}_p$ and 3×1 translation vector $\mathbf{t}_p$.

Let $\mathbf{X}_c$ denote the homogeneous pixel coordinates and $\mathbf{X}_p$ denote the homogeneous projector coordinates. where,

$$\mathbf{X}_c = [w_c x_c \; w_c y_c \; w_c]^T, \tag{1.3}$$

$$\mathbf{X}_p = [w_p x_p \; w_p]^T. \tag{1.4}$$

Here, $[x_c \; y_c]^T$ is the inhomogeneous pixel location and x_p is the inhomogeneous projector coordinate. Let $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$ denote the rows of the camera calibration matrix and $\mathbf{p}_1, \mathbf{p}_2$ denote the rows of the projector calibration matrix. Then equations 1.1 and 1.2 can be expanded as

$$\begin{aligned} w_c x_c &= \mathbf{c}_1 \mathbf{X}_w, && (1.5) \\ w_c y_c &= \mathbf{c}_2 \mathbf{X}_w, && (1.6) \\ w_c &= \mathbf{c}_3 \mathbf{X}_w. && (1.7) \end{aligned}$$

and

$$\begin{aligned} w_p x_p &= \mathbf{p}_1 \mathbf{X}_w, && (1.8) \\ w_p &= \mathbf{p}_2 \mathbf{X}_w. && (1.9) \end{aligned}$$

Substituting for w_c and w_p we get

$$\begin{aligned} x_c \mathbf{c}_3 \mathbf{X}_w = \mathbf{c}_1 \mathbf{X}_w, && (1.10) \\ y_c \mathbf{c}_3 \mathbf{X}_w = \mathbf{c}_2 \mathbf{X}_w, && (1.11) \\ x_p \mathbf{p}_2 \mathbf{X}_w = \mathbf{p}_1 \mathbf{X}_w, && (1.12) \end{aligned}$$

which can be written in the matrix form $\mathbf{Q}\mathbf{X}_w = \mathbf{0}$ where matrix $\mathbf{Q}$ is given by

$$\mathbf{Q} = \begin{bmatrix} x_c \mathbf{c}_3 - \mathbf{c}_1 \\ y_c \mathbf{c}_3 - \mathbf{c}_2 \\ x_p \mathbf{p}_2 - \mathbf{p}_1 \end{bmatrix}.$$

The matrix $\mathbf{Q}$ can be written in the block form as $\mathbf{Q} = [\mathbf{R} \; \mathbf{s}]$, where $\mathbf{R}$ is a 3×3 matrix and $\mathbf{s}$ is a 3×1 vector. Substituting, we get

$$\begin{bmatrix} \mathbf{R} & \mathbf{s} \end{bmatrix} \begin{bmatrix} w_w x_w \\ w_w y_w \\ w_w z_w \\ w_w \end{bmatrix} = \mathbf{R} \begin{bmatrix} w_w x_w \\ w_w y_w \\ w_w z_w \end{bmatrix} + w_w \mathbf{s} = 0.$$

Let $\mathbf{x}_w = [x_w \; y_w \; z_w]^T$ denote the inhomogeneous world coordinates. The inhomogeneous world coordinates $\mathbf{x}_w$ would be the solution to the linear systems of equations defined by

$$\mathbf{R}\mathbf{x}_w = -\mathbf{s}. \tag{1.13}$$

The world coordinates would be given by

$$\mathbf{x}_w = -\mathbf{R}^{-1}\mathbf{s}. \tag{1.14}$$

1.4 Stripe Encoding Problem

In structured light systems, the projector is modeled as a pin hole device and a coordinate system is defined on the projector plane. To obtain the 3D location of a scene point the x-coordinate of the scene point with respect to the projector coordinate system is required. In structured light systems, the x-coordinate is quantized as shown in Figure 1.5. The location of the scene points is defined with respect to the quantized coordinate system. As shown in Figure 1.6, due to quantization, the vertical planes would have a finite width.

Greater the quantization step, thicker would be the planes. The information about the location of the scene in the quantized projector coordinate frame is encoded in the light pattern projected by the projector. Depending upon how the information is encoded the coding method can be classified as spatial encoding or temporal encoding.

1.4.1 Spatial Coding Method

In spatial coding scheme the information about the location of the scene point in the projector coordinate frame is encoded spatially. The image of the scene under the projected pattern is acquired. The coordinate of the scene point in the projector coordinate

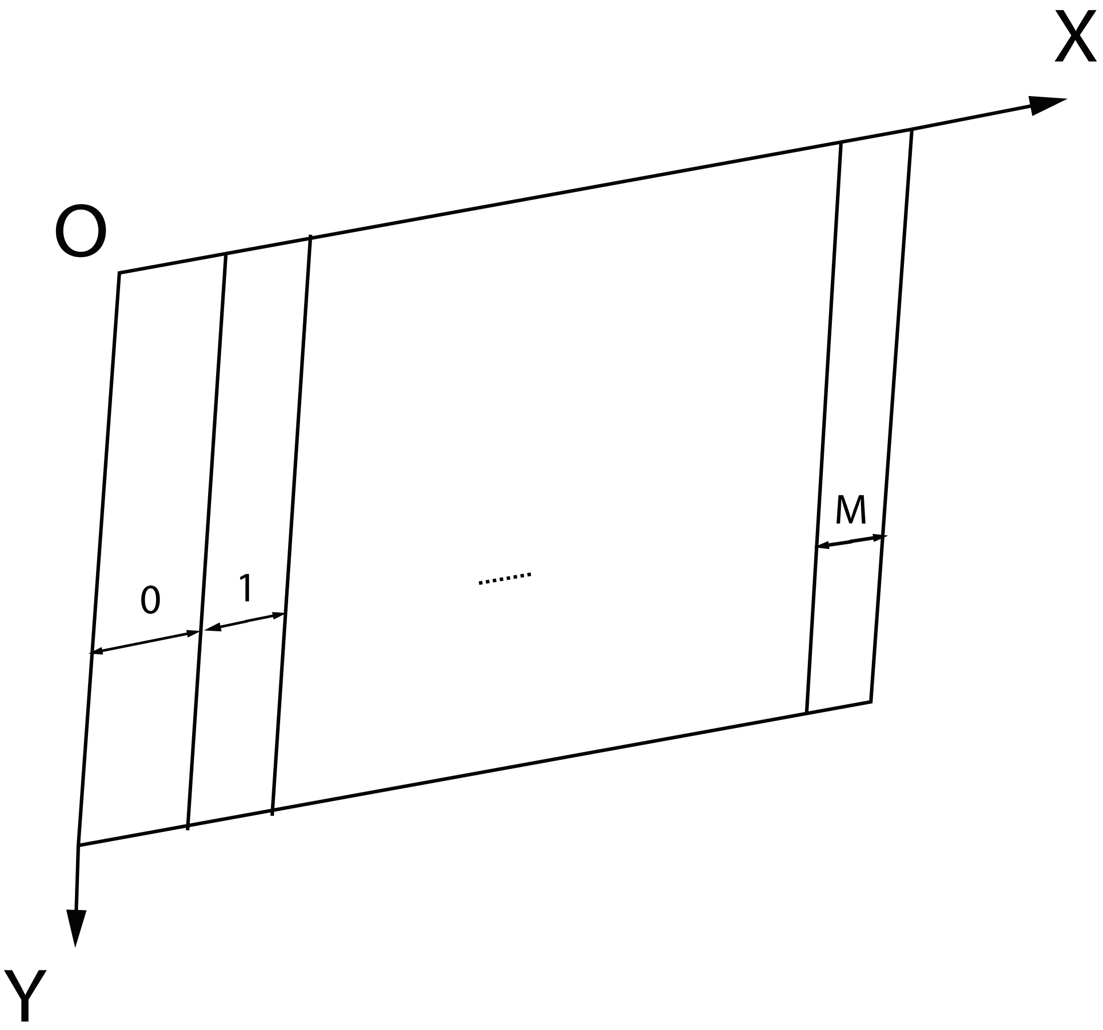


Figure 1.5: Quantization of projector coordinates.

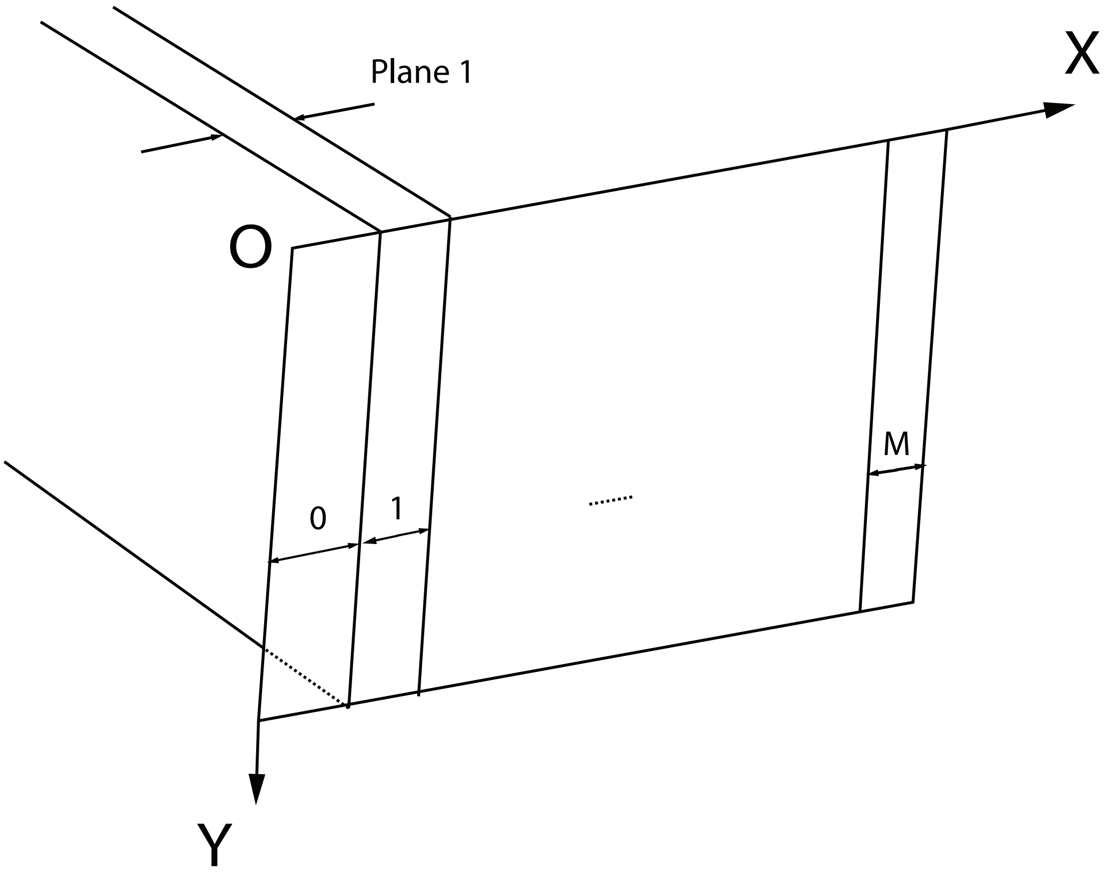


Figure 1.6: Figure shows thickness of projector planes due to quantization.

frame is contained in the surrounding pixels. For example, in [36] the authors project a pattern with vertical color stripes. The image of the scene illuminated by the pattern is acquired. As one moves from left to right horizontally, a change in the color in the captured scene indicates a change in the projector location. The increase or decrease in the red, green or blue intensities at the stripe transitions are noted and dynamic programming is used to obtain the coordinates in the projector's coordinate frame. The method assumes that the red, green and blue sensors of the camera respond only to the red, green and blue components of the light incident on the sensors. However, decoupling of the channels is not always a valid assumption. In [6], [13] the authors give the following analytic model for the relation between the red, green and blue intensities sensed by the camera and the red, green and blue intensities projected by the projector.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} a_{RR} & a_{RG} & a_{RB} \\ a_{GR} & a_{GG} & a_{GB} \\ a_{BR} & a_{BG} & a_{BB} \end{bmatrix} \begin{bmatrix} k_R & 0 & 0 \\ 0 & k_B & 0 \\ 0 & 0 & k_G \end{bmatrix} \phi \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{bmatrix} R_0 \\ G_0 \\ B_0 \end{bmatrix}. \qquad (1.15)$$

Here, $[R, G, B]^T$ denotes the red, green and the blue intensities sensed by the camera, $[R_0, G_0, B_0]^T$ is the intensity measured under ambient light conditions, $\phi(.)$ is the nonlinear transformation between the red, green and blue intensities input to the projector and the red, green and blue intensities projected by the projector. The nonlinear transformation $\phi(.)$ can be obtained offline by performing photometric calibration. Since the color channels of the projector are decoupled, the nonlinear transformation consists of three scalar transformation. All three scalar transformations are monotonic bijective functions. The k_R, k_G and k_B denote the albedos of the point for the red, green and the blue channels respectively. The diagonal matrix defined by these albedos can be obtained by acquiring an additional image under full white illumination. The a_{KL}, where K,L $\in \{R, G, B\}$ measures the response of the K^{th} channel to the L^{th} component of the light incident on the sensor. The projector–camera coupling matrix can be measured in advance with calorimetric calibration. The authors assign a different color code to each projected stripe. In the image of the scene acquired, the red, green and blue components of the pixel location of the scene point is noted. Using measured intensity values the projected color is estimated. Knowledge of the projected color gives the location of the grid point in the projector coordinate frame. Then, using the pixel location and the location in the projector coordinate frame the world coordinates of the point is obtained. Spatial encoding schemes are time efficient. However, they produce sparse depth maps and mediocre resolution. To obtain high resolution depth maps temporal encoding of the projector coordinate is preferred.

1.4.2 Temporal Encoding Method

In the temporal encoding method [29], multiple patterns are projected on the scene. For each projected pattern the image of the scene is acquired. The intensity of the scene point, in the multiple acquired images is measured. The set of intensities define the scene point's location in the projector coordinate frame. For example, suppose the projector plane has 256 quantization levels. An eight bit Gray code would represent each level uniquely. The first projected pattern would use black to represent quantization levels with zero in the most significant bit and white to represent quantization levels with one in the most significant bit. In the acquired image, whenever the pixel intensity is greater than a certain threshold, the corresponding scene point is located in the projector coordinate frame at a quantization level with the most significant bit as one. Whenever the pixel intensity is less than a certain threshold the scene point would be at a quantization level with most significant bit as zero. Similarly, the next projected pattern would contain information about the next most significant bit and so on and so forth. Apart from Gray code, other coding schemes like binary codes can also be used. However, the advantage of Gray code is that an error in the detection of any bit would contribute the same amount to the error in the detection of the scene point's projector coordinates. In binary or other codes the error would increase as the bit location moves from the least significant bit to the most significant bit. Temporal coding schemes generate much higher resolution depth maps than spatial encoding methods. However, the disadvantage is that the number of patterns to be projected on the scene is large. The use of large projection patterns increases the data acquisition time. The large data acquisition time may generate problems in scenes which exhibit motion over time.

1.5 Calibration

In the calibration step, the position of the camera and the projector relative to the world coordinate frame is determined. Figure 1.7 shows the experimental setup used in the lab for the calibration and the data acquisition process. In the lab, a Lego cube was used as the calibration target. The corner points of the Lego cube were detected using the Harris corner detector. The coordinates in the projector coordinate frame, of the corners, were extracted. The exact mechanism of obtaining the projector coordinates depends on the coding scheme used. The world coordinates of the corner points are assumed to be known.

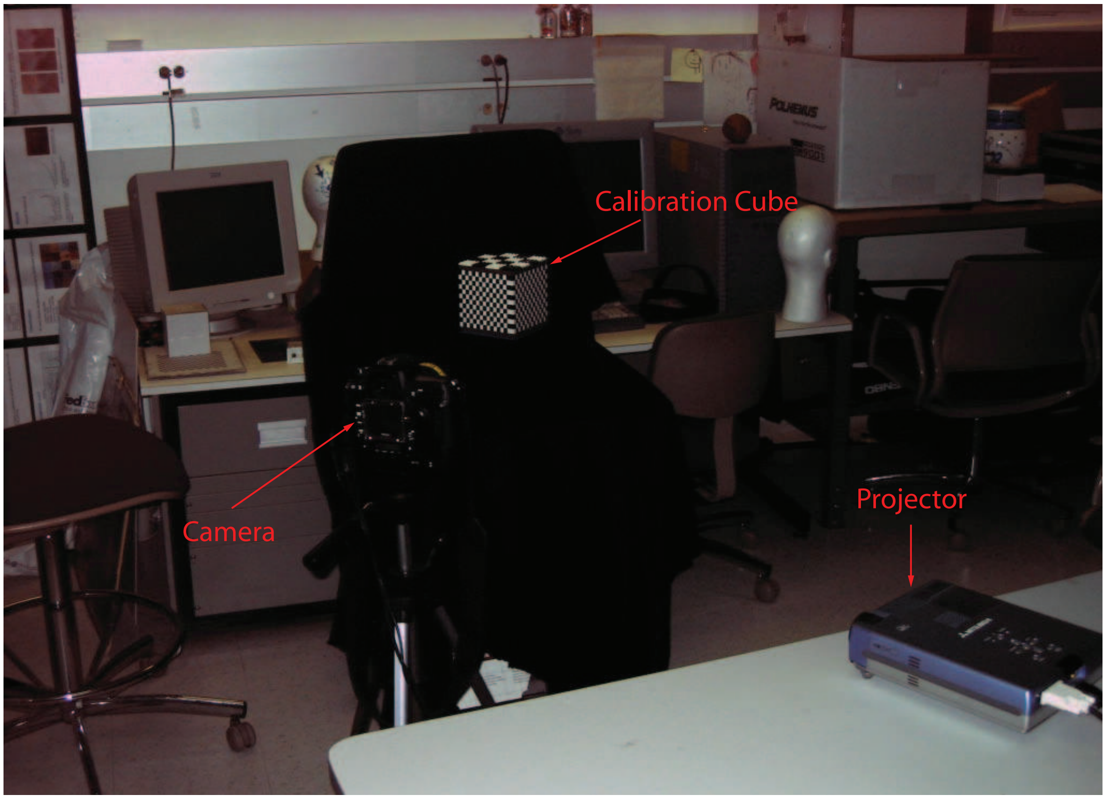


Figure 1.7: Experimental setup for the calibration and the data acquisition process.

Let $(\mathbf{X}_w)_k$, $(\mathbf{X}_c)_k$ and $(\mathbf{X}_p)_k$ be the homogeneous world coordinates, the pixel location and the projector coordinates of the k^{th} corner point. Let π_c be the function which projects the homogeneous world coordinates $(\mathbf{X}_w)_k$ to the corresponding inhomogeneous pixel locations $\left([x_c\ y_c]^T\right)_k$. Let π_p be the function which projects the homogeneous world coordinates $(\mathbf{X}_w)_k$ to the corresponding inhomogeneous projector coordinates $(x_p)_k$. The calibration problem is formulated as

$$\begin{aligned} \mathbf{C}_c &= \arg\min_{\mathbf{C_c}} \sum_{k=1}^{n} \|\pi_c\left((\mathbf{X}_w)_k\right) - \left([x_c\ y_c]^T\right)_k \|_2^2, && (1.16) \\ \mathbf{C}_p &= \arg\min_{\mathbf{C_p}} \sum_{k=1}^{n} \|\pi_p\left((\mathbf{X}_w)_k\right) - (\mathbf{x}_p)_k \|_2^2. && (1.17) \end{aligned}$$

Here n is the total number of corner points. The objective function can be minimized using the Levenberg–Marquardt algorithm [28]. The initial estimate can be obtained using the least squares estimate. The least square problem for the camera can be formulated as [33]

$$\left[\mathbf{B}_1^T \ldots \mathbf{B}_N^T\right]^T \theta_{\mathbf{c}} = \mathbf{0}, \tag{1.18}$$

where,

$$\mathbf{B}_k = \begin{bmatrix} (\mathbf{X}_w)_k & \mathbf{0} \\ \mathbf{0} & (\mathbf{X}_w)_k \\ -(\mathbf{X}_c)_k (\mathbf{X}_w)_k & -(\mathbf{Y}_c)_k (\mathbf{X}_w)_k \end{bmatrix}^T ,$$

$$\theta_{\mathbf{c}} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \mathbf{c}_3]^T .$$

$\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ are the first, second and the third rows of the camera calibration matrix $\mathbf{C}_c$. Since the calibration matrix is defined up to a scale factor, the constraint $\|\theta_{\mathbf{c}}\|_2 = 1$ is forced in the optimization process. Similarly the least square problem for the projector matrix can be set up as

$$\left[\mathbf{M}_1^T ... \mathbf{M}_N^T\right]^T \theta_{\mathbf{p}} = 0. \tag{1.19}$$

In case of the projector

$$\mathbf{M}_k = \begin{bmatrix} (\mathbf{X}_w)_k \\ -(\mathbf{x}_p)_k (\mathbf{X}_w)_k \end{bmatrix}^T ,$$

$$\theta_{\mathbf{p}} = [\mathbf{p}_1 \ \mathbf{p}_2]^T .$$

$\mathbf{p}_1$ and $\mathbf{p}_2$ are the first and second rows of the projector matrix $\mathbf{c}_p$. As in the case with the camera calibration, the constraint $\|\theta_p\|_2 = 1$ is enforced.

1.6 Spline Smoothing

The coordinates of the scene points in the projector coordinate are quantized. Due to the quantization process the projector coordinates have a step like structure. Once the coordinates have been recovered cubic spline approximation [3] can be used to remove the step like structure. In a spline approximation problem, a piecewise continuous curve approximating the data points is found. The piecewise continuous curve does not pass through the available data points but passes close to the available data points as shown in Figure 1.8. Spline approximation enables local control. Whenever the value of a data point is changed, only the portion of the curve near the data point changes. Since, each data point influences the characteristics of the approximation curve near it, the data points are called as control points.

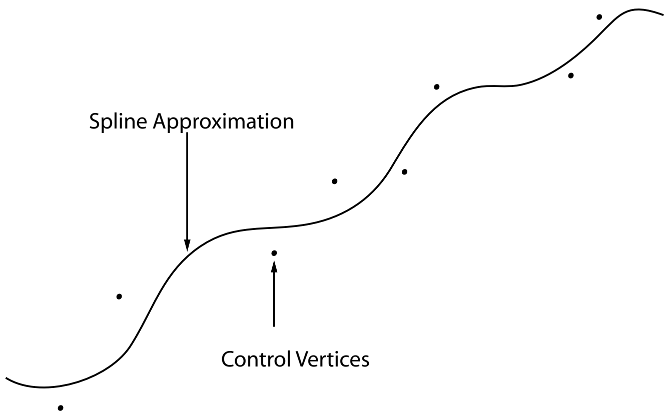


Figure 1.8: Spline approximation.

Consider the problem of spline approximation in one dimension. We denote the location of the data points by the scalar u. Suppose we are given a set of data points $\{Y_r\}$, where $r \in \{1, ..., R\}$. The data point Y_r is located at u_r. In the spline approximation problem, we find a piecewise continuous function $Q(u)$ which passes close the the data points. $Q(u)$ is given by,

$$Q(u) = \sum_i g_i B_i(u), \tag{1.20}$$

where, g_i are the weights and B_i's are called as the basis functions. Piecewise continuity can be achieved by ensuring that the basis functions themselves are piecewise continuous. Local control is obtained by ensuring that at any point only a few basis functions are nonzero. To achieve local control the basis functions have a compact support. Let (u_1^i, u_2^i) be the support of the basis function $B_i(u)$. In the interval (u_1^i, u_2^i), the basis functions $B_i(u)$ is a cubic polynomial given by

$$B_i(u) = a_i u^3 + b_i u^2 + c_i u + d_i. \tag{1.21}$$

To obtain a spline approximation the parameters a_i, b_i, c_i and d_i need to be estimated. To estimate the parameters we use the '*csaps*' function in the MATLAB spline toolbox. The function requires as input the data points $\{Y_r\}$ and the location of the data points $\{u_r\}$ to perform spline approximation.

1.7 Summary

We discussed the different steps involved in the use of structured light techniques for geometry acquisition. We first discussed the calibration step. The calibration problem required the pixel locations, coordinates in the projector coordinate frame and the world points of fiducial points. In the experimental sessions, a Lego cube was used as the calibration target and the corner points were used as the fiducial points. The use of spatial and temporal encoding schemes was discussed. The choice between spatial or temporal encoding schemes was a trade off between the resolution and the number of projected patterns. Finally, the use of spline smoothing to remove the step like structure of the scene coordinates in the projector coordinate frame was discussed.

Chapter 2

Novel Encoding Schemes for Structured Light

2.1 Introduction

As discussed in Chapter 1, the reconstruction using structured light techniques involves three steps: calibration, coding and smoothing. In the calibration step, the position of the camera and the projector with respect to the world coordinates is determined. In the coding step, the method of encoding the projector coordinate information in the projected patterns is determined. In the smoothing step, spline fits are used to remove the quantization noise. A major limitation of the the existing techniques is the dependency on the surface albedo. In the low albedo regions the projected patterns are not visible, making the reconstruction of the low albedo regions difficult. We will propose two new structured light techniques targeted towards the reliable reconstruction of the low albedo regions. Both techniques use image ratios to determine the coordinates in the projector coordinate frame. The invariance of image ratios to the surface albedo [23] enables the reliable reconstruction of the low albedo regions.

In the proposed temporal encoding scheme, the use of image ratios has been combined with high dynamic range imaging (HDR). To obtain reliable ratios in the low albedo regions the intensities measured by the camera in the low albedo regions should be reliable. The intensities measured by the camera are reliable only if the measured intensities lie in the linear range. The intensities measured in the low albedo regions would lie in the linear range only under the overexposed settings of the camera. The temporal encoding scheme is called as the 'HDR–Ratio Method'.

The proposed novel spatial encoding scheme requires three images to be projected by the projector. The projector coordinates are obtained by counting the stripes. It is assumed that the center stripe hits the scene and that no occlusions occur. The resolution obtained by using novel spatial encoding scheme is lower than the resolution of the HDR–Ratio method. However, the data acquisition is much faster with the spatial encoding scheme. The spatial coding scheme is called as the 'Edge Counting Method'.

From this point onwards, a scene point's projector coordinate would be referred as the stripe id.

2.2 HDR–Ratio Method

In HDR–Ratio method the focus has been on the reliable reconstruction of the low albedo regions. When the light from the projector hits the low albedo regions the light patterns are not clearly visible. The poor visibility of light patterns makes it difficult to extract the stripe id in the low albedo regions. For example, Figure 2.1 shows the image of a face illuminated by the projector's light pattern. It is clearly seen that the projected pattern is not visible in the low albedo hair region. This makes the reconstruction of the low albedo hair region extremely difficult. In the HDR–Ratio method the problem

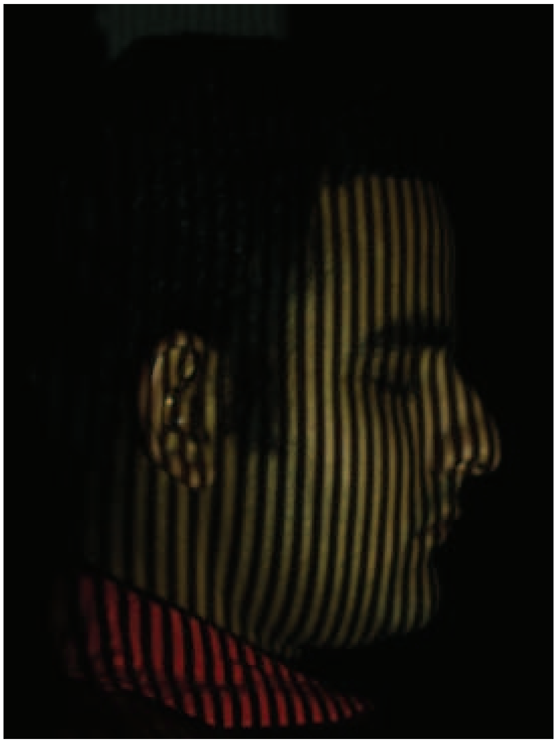


Figure 2.1: The projected stripes are not clearly visible in the low albedo regions.

of reliable reconstruction of the low albedo regions is addressed by using image ratios and high dynamic range imaging.

2.2.1 Image Ratios and the Gray Code Encoding Scheme

The quantization levels of the projector coordinate frame is coded using the Gray code encoding scheme. The advantage of Gray code encoding scheme is that an error in any bit causes the same amount of error in the inference of the stripe id. In other coding schemes, the error depends significantly on the position of the bit detected erroneously. For example, suppose an eight bit binary encoding scheme is used and the stripe id of a scene point has been detected as $b_7b_6b_5b_4b_3b_2b_1b_0 = 10101010$. Now, if the true value of b_2 is one instead of zero, the detected stripe id would have an error of $2^2 = 4$. If the true value of b_7 is zero instead of one, the detected stripe id would have an error of $2^7 = 128$. An erroneous detection of b_7, in the binary encoding scheme, generates a much higher error in the stripe id than an erroneous detection of b_2. In the Gray code encoding scheme, however, an error in b_7 or b_2 would lead to the same amount of error in the detected stripe.

In the HDR–Ratio method, to determine the most significant bit of a pixel's stripe id the ratio of the images acquired under the first two projected patterns is taken. We call the image obtained by taking the ratios as the ratio image. In the ratio image, pixels where the ratio is greater than one have the most significant bit of the Gray code as one, and the intensities at these pixel locations are set to one. Pixels where the ratio is less than one have the most significant bit of the Gray code as zero, and the intensities at these pixel locations are set to zero. Similarly the ratio of the next two projected patterns is used to determine the next most significant bit. In this way the ratio of successive projected patterns enable the identification of the stripe id of the scene points. The patterns projected by the projector is shown in Figure 2.2.

2.2.2 HDR

The dynamic range of a digital or film camera is very limited. With one camera setting it is not possible to capture all scene points reliably. For a particular camera setting,

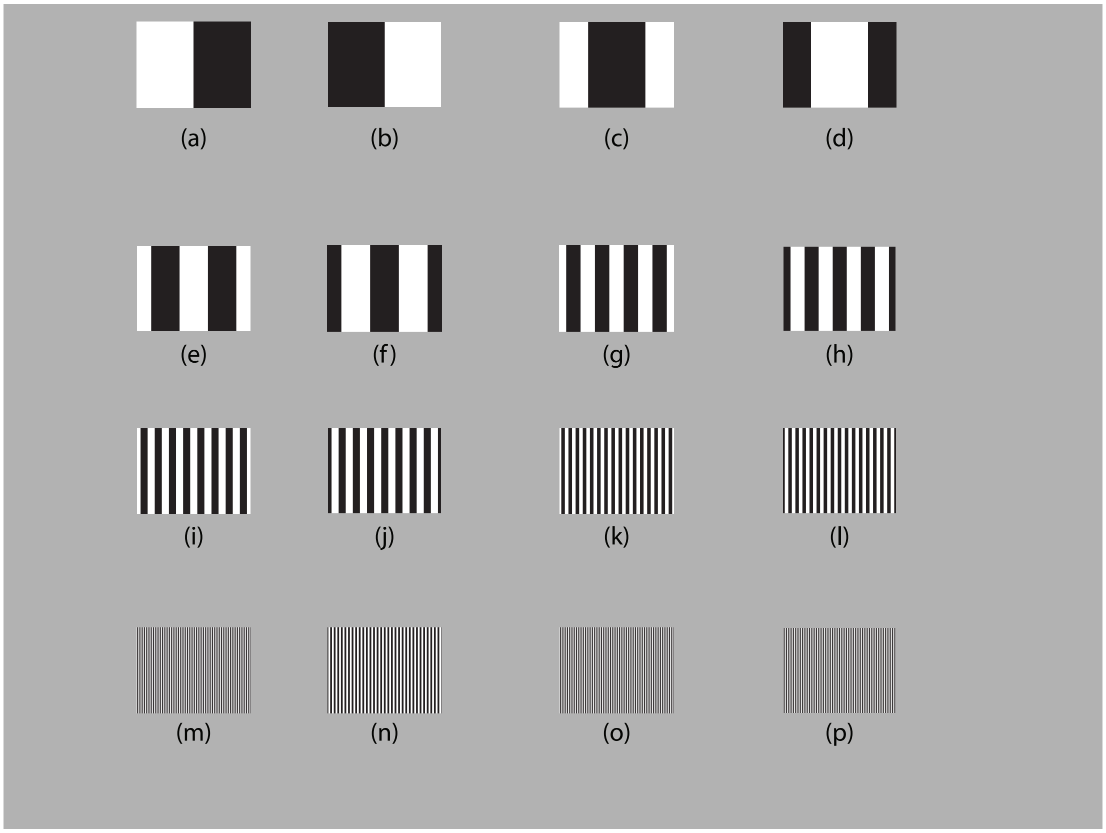


Figure 2.2: Patterns projected by the projector in the HDR–Ratio method.

some scene points may be too dark to be captured reliably, while some scene points may be too bright to be captured reliably. The intensities measured by the camera are reliable only if they lie in the linear range of the camera response. To reliably capture the intensities at all points it is necessary that the intensities measured at the scene points lie in the linear range of the camera. To ensure that that all the scene points lie in the linear range of the camera it is necessary to image the scene under multiple exposure settings. The technique of imaging the scene under multiple camera settings and combining the different images is called as HDR [9]. In the HDR–Ratio method, high dynamic range imaging has been used to obtain reliable ratios in the low albedo regions.

In high dynamic range imaging the image of the scene illuminated by the projected pattern is acquired under multiple exposure settings of the camera. The image is first acquired under normal exposure. Normal exposure settings enable the reliable capture of scene points with high albedos. However, scene points with low albedos are not reliably captured. To reliably capture scene points with low albedo the scene is overexposed. In the overexposed image the low albedo regions are within the dynamic range of the camera while the points with high albedos are saturated. Figure 2.3 shows the face image acquired under normal and overexposed conditions. It can be seen that under normal conditions the low albedo regions are not reliably captured but the high albedo regions are reliably captured. Under overexposed conditions the low albedo regions are reliably captured but the high albedo regions are saturated. In the low albedo regions the ratios obtained from the overexposed setting is more reliable. The ratios in the normally exposed regions are reliably captured under normal exposure settings and not under the overexposed conditions. To obtain the final ratio image we combine the ratio images obtained under the two exposure settings. A threshold is set. At pixel locations where the intensity is greater than a certain threshold the ratio value

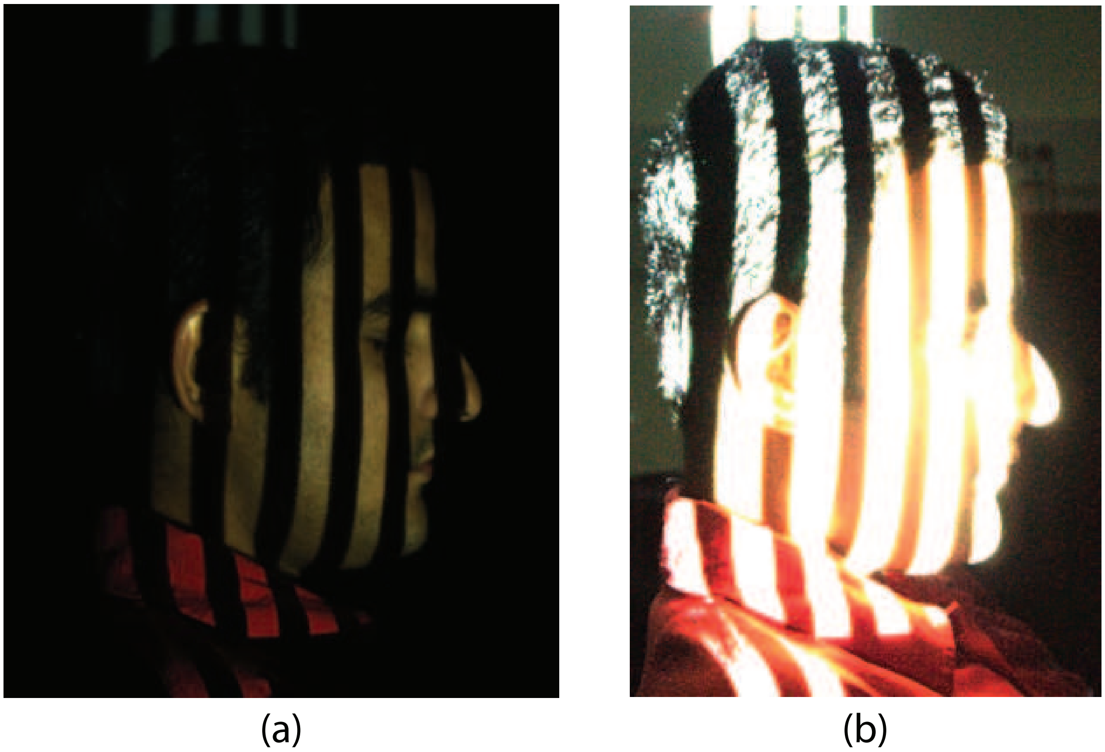


Figure 2.3: Images of the scene acquired under multiple exposures. (a): Image of the scene under normal exposure settings. (b): Image of the scene under overexposed settings. The stripes in the low albedo regions are clearly brought out in the overexposed image.

in the ratio image obtained from normal exposure settings are retained. Whenever the intensity is below a certain threshold the ratio value in the ratio image obtained from overexposed conditions is retained. The threshold is set at the intensity value below which the camera response becomes unreliable. In our experiments, for eight bit images, the threshold value is selected as fifty. If the intensity value at the pixel location is less than fifty, the ratio obtained using the overexposed images is selected.

Figures 2.4-2.7 shows the images of the side view of a face acquired under the normal exposure and overexposed settings.

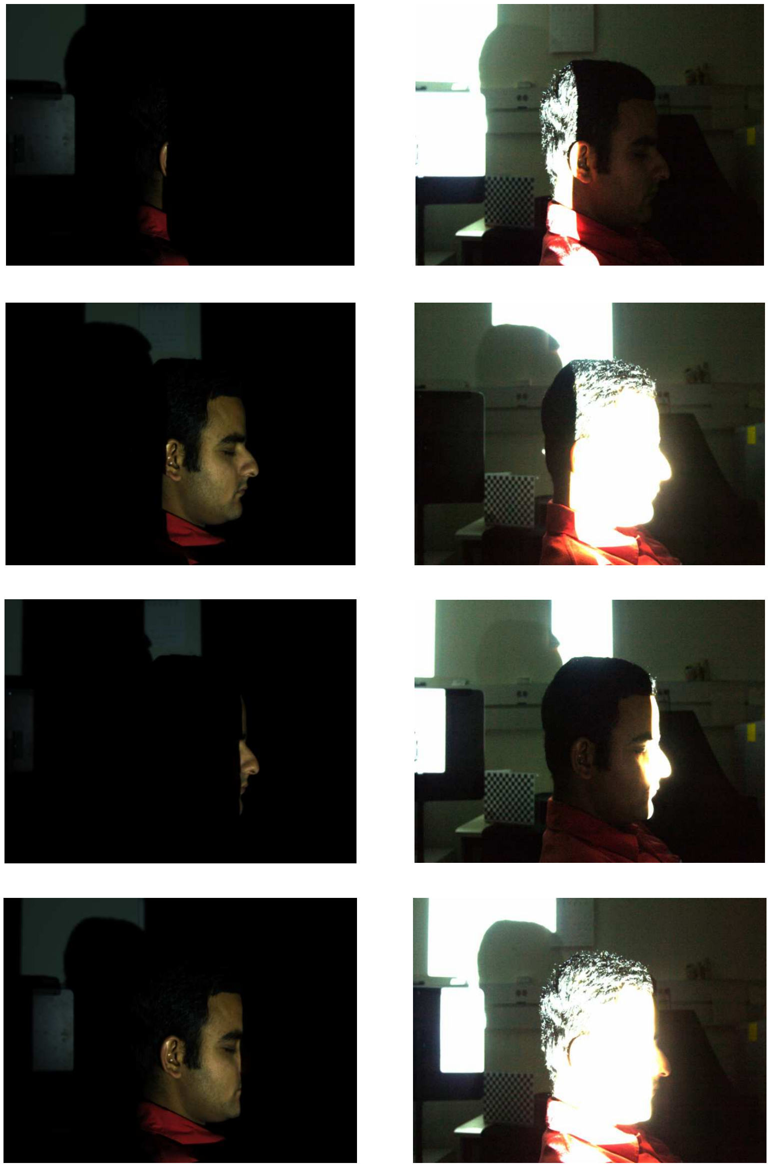


Figure 2.4: Sideview face images acquired by the camera (Images $1 - 8$).

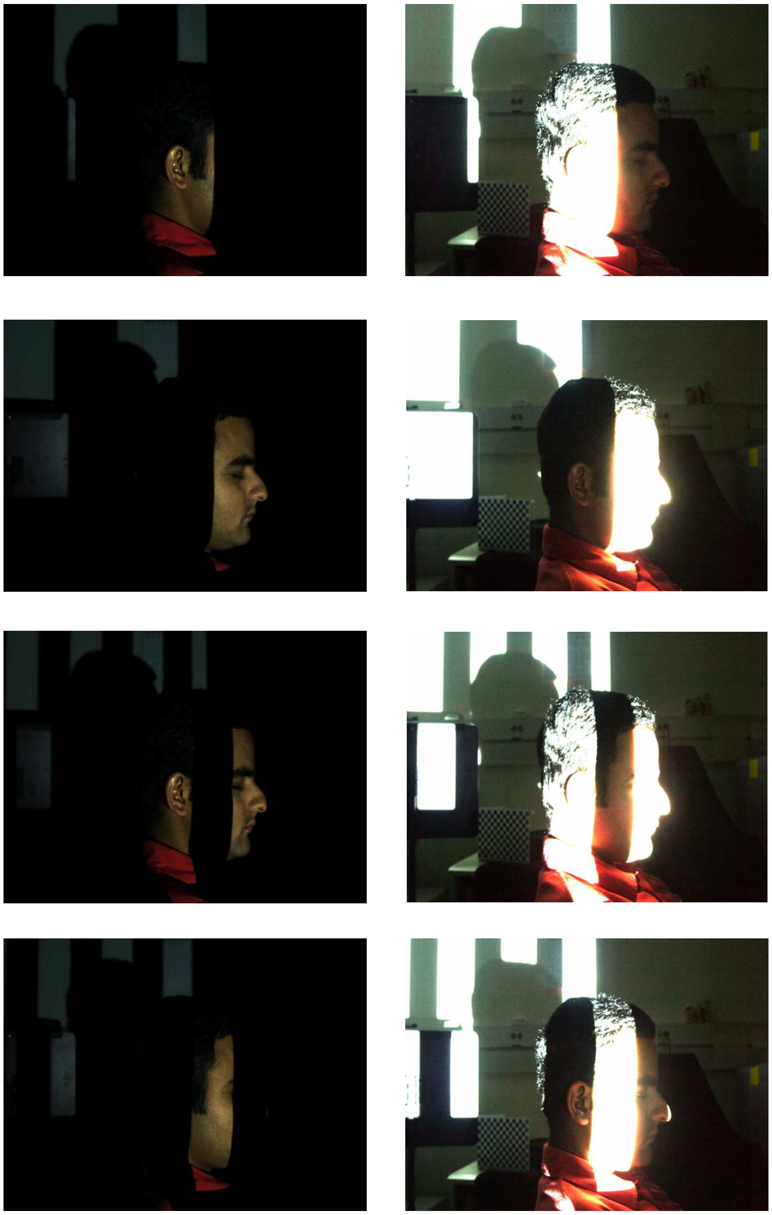


Figure 2.5: Sideview face images acquired by the camera (Images 9 – 16).

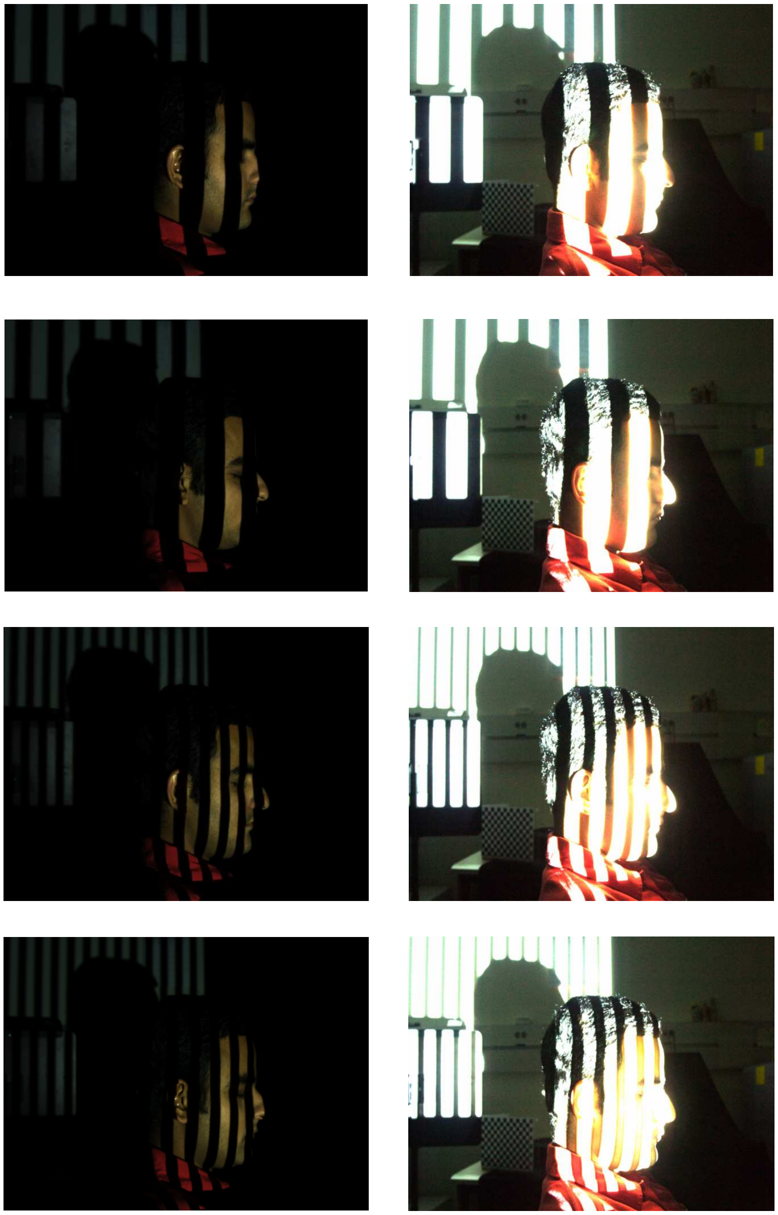


Figure 2.6: Sideview face images acquired by the camera (Images $17 - 24$).

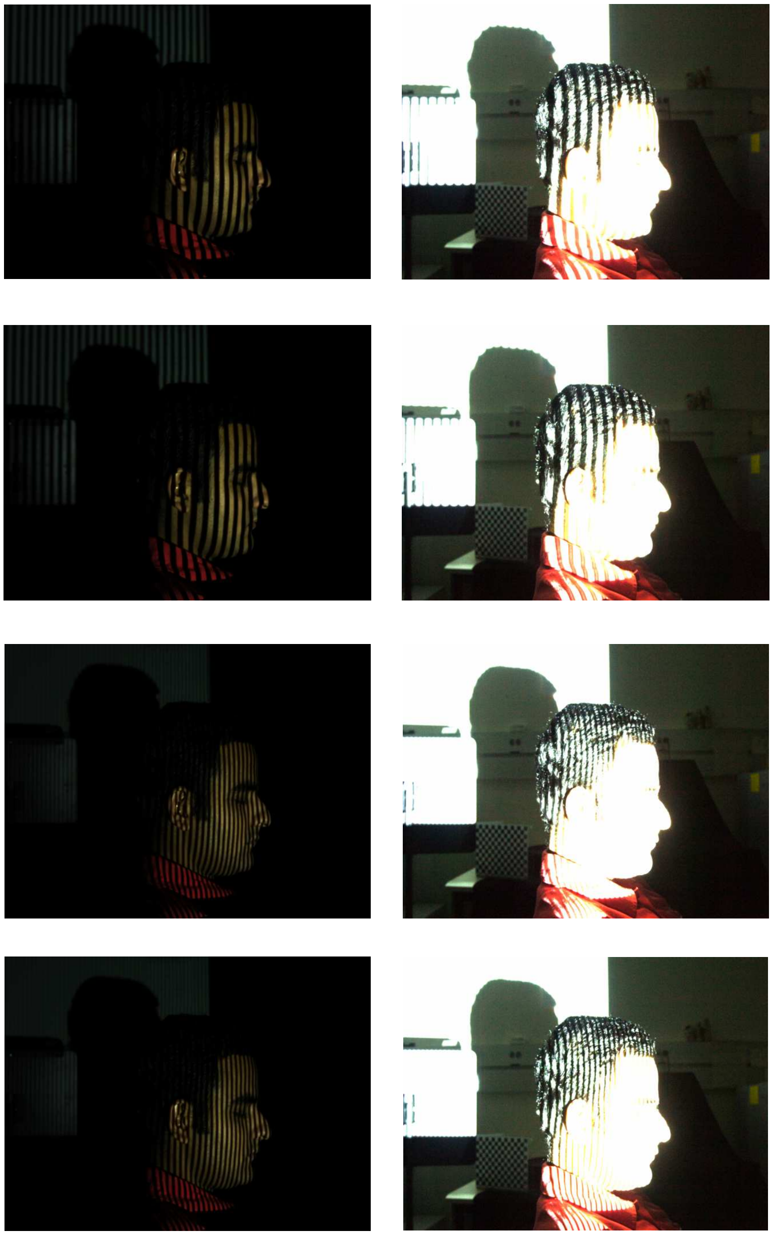


Figure 2.7: Sideview face images acquired by the camera (Images $25 - 32$).

The resulting ratio images are shown in Figure 2.8. Figure 2.9 shows the ratio image and the corresponding camera image. Note that the stripes are well brought out in the low albedo regions in the ratio images. Using the ratio images the stripe id of each pixel is determined. Figure 2.10 shows the stripe id of each pixel. In Figure 2.10 each pixel has been assigned an intensity equal to it's stripe id divided by 256. The camera images for the same face, when imaged from the front, is shown in Figures 2.11-2.14

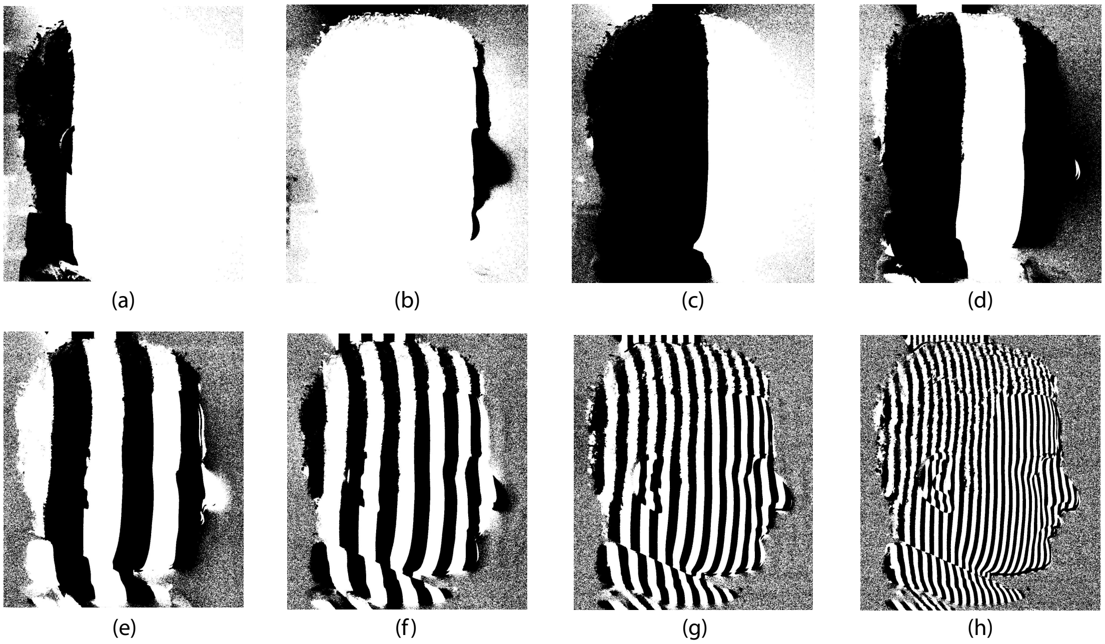
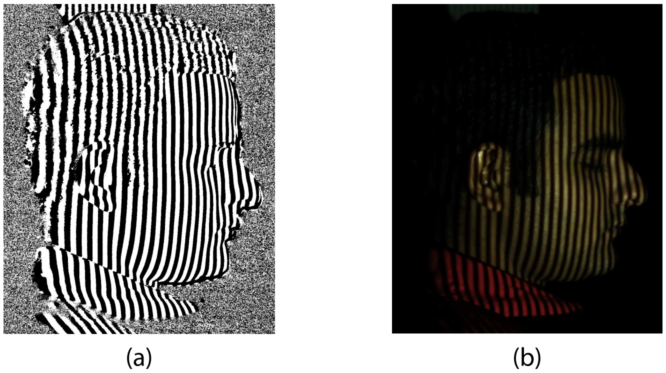


Figure 2.8: Ratio images obtained in the HDR–Ratio method. (a), (b),...(h) indicate the 1^{st}, 2^{nd},...8^{th} most significant bits of the Gray code. In any ratio image, at a particular pixel location, an intensity value of 1 indicates that the corresponding bit in the Gray code is 1. In any ratio image, at a particular pixel location, an intensity of value of 0 indicates that the corresponding bit in the Gray code is 0.

The ratio images for the front view are shown in Figure 2.15. The stripe image for the front view is shown in Figure 2.16.


(a)

(b)

Figure 2.9: (a): Ratio image HDR–Ratio Scheme. (b): Image of the scene under a light pattern projected by the projector. In the ratio image, the stripes are clearly visible in the low albedo regions. In the camera image the stripes are not clearly visible in the low albedo hair region.

Figure 2.10: Stripe image of the side view of the face. Each pixel has been assigned an intensity equal to it's stripe id divided by 256.

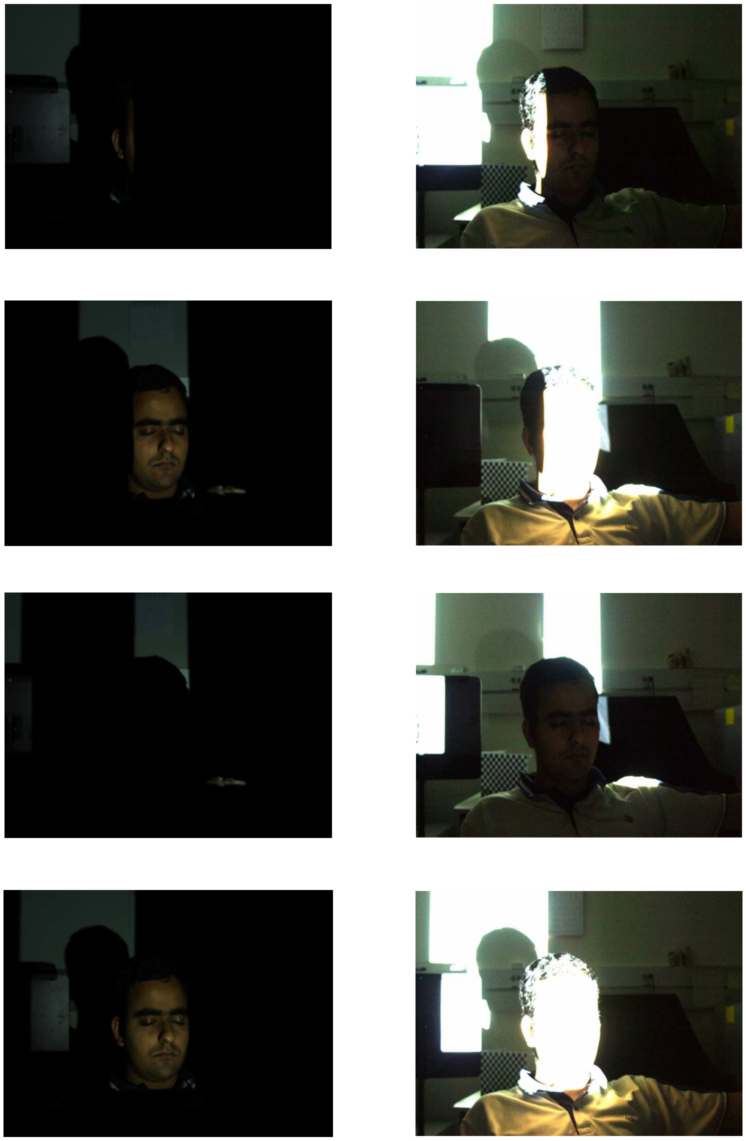


Figure 2.11: Front view face images acquired by the camera (Images $1 - 8$).

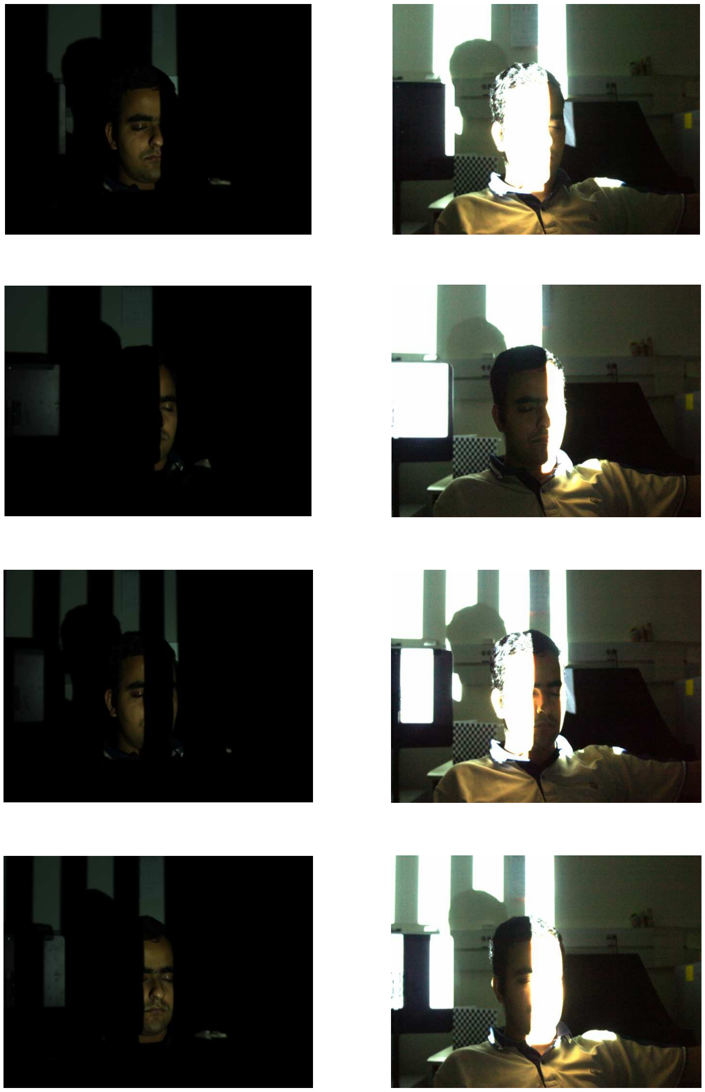


Figure 2.12: Front view face images acquired by the camera (Images $9 - 16$).

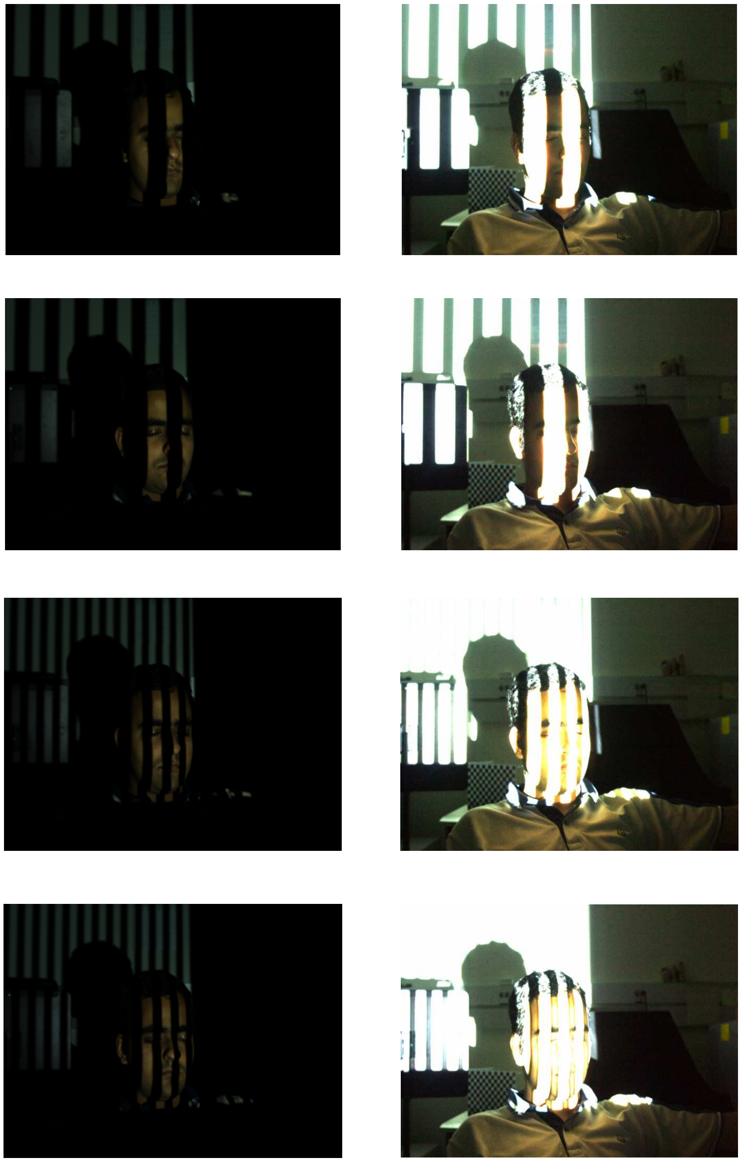


Figure 2.13: Front view face images acquired by the camera (Images $17 - 24$).

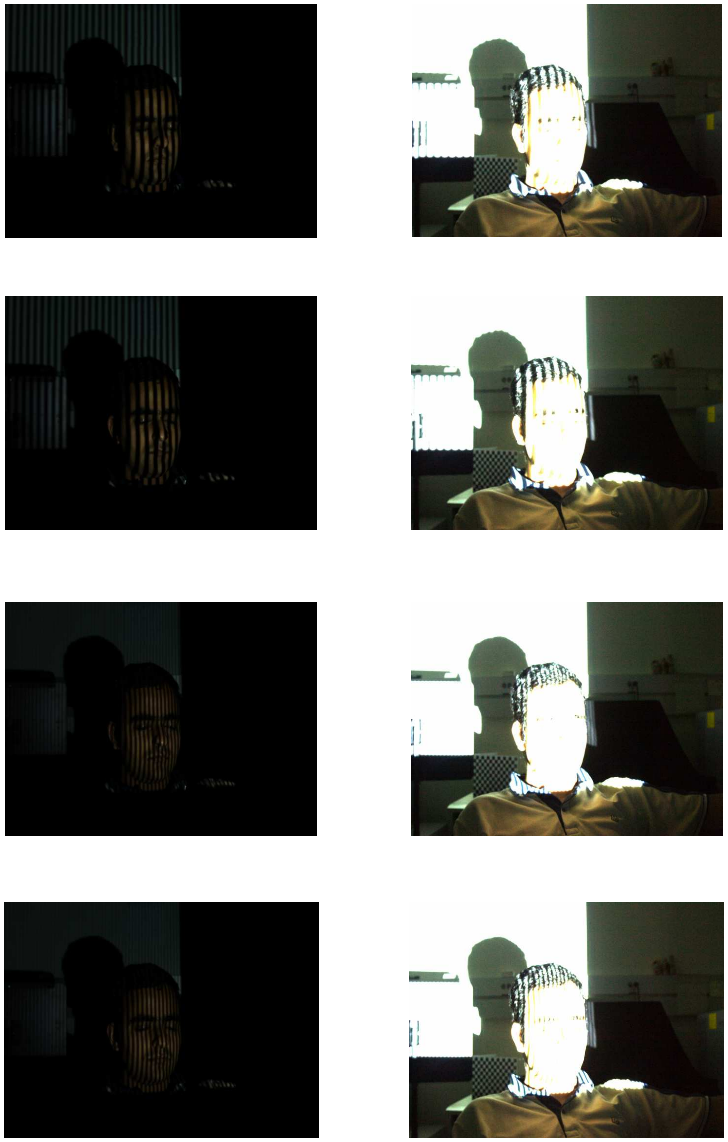


Figure 2.14: Front view face images acquired by the camera (Images $25 - 32$).

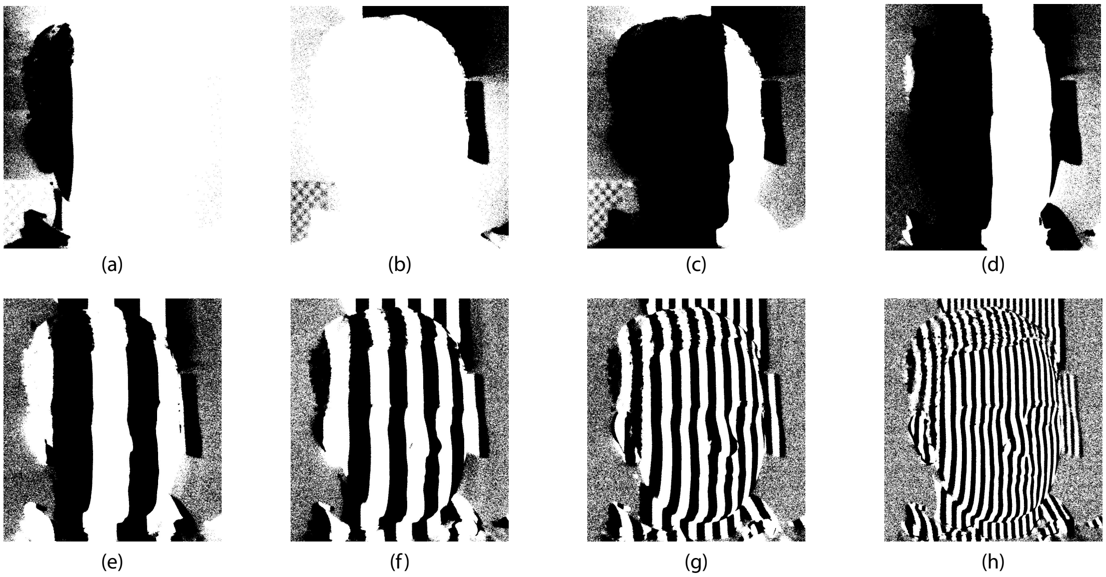


Figure 2.15: Ratio images obtained in the HDR–Ratio method. (a), (b),...(h) indicate the 1^{st}, 2^{nd},...8^{th} most significant bits of the Gray code. In any ratio image, at a particular pixel location, an intensity value of 1 indicates that the corresponding bit in the Gray code is 1. In any ratio image, at a particular pixel location, an intensity of value of 0 indicates that the corresponding bit in the Gray code is 0.

Figure 2.16: Stripe image of the front view of the face. Each pixel has been assigned an intensity equal to it's stripe id divided by 256.

2.2.3 Spline Smoothing

In Figure 2.10 each pixel location has been assigned an intensity value equal to its stripe id divided by the maximum resolution of 256. Figure 2.17 shows the plot of the stripe ids for pixels lying along a particular row in the camera image. Due to quantization the plot has a step like structure. The beginning of each step indicates a change in the quantization level and would be unaffected by quantization noise. For reconstruction purposes the beginning of each step is retained. The step points are detected by looking at the discrete gradient function. The beginning of the step would be accompanied by a significant increase or decrease in the gradient function depending upon whether the stripe id increases or decreases. Once the beginning of each step is detected, spline approximation is performed (see Section 1.6 on page 13 for discussion on spline smoothing).

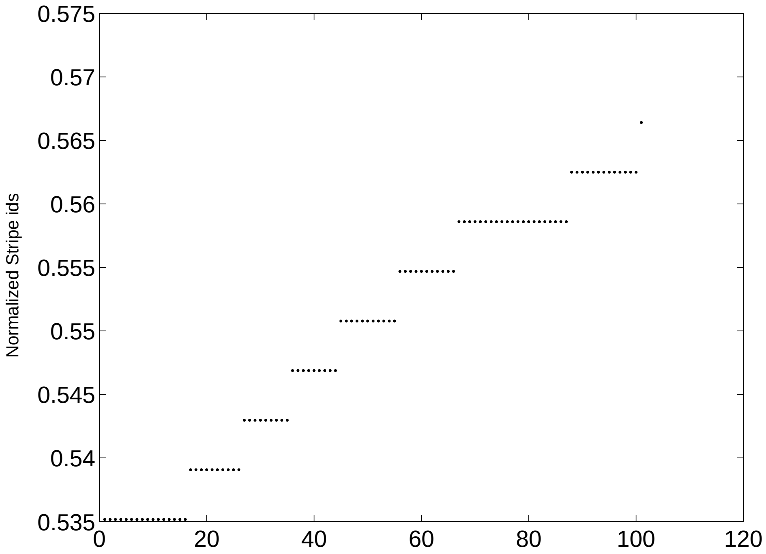


Figure 2.17: Stripe ids of pixels lying on a particular row. Due to quantization the plot has a step like structure.

2.2.4 Image Acquisition

Humans cannot remain perfectly still due to breathing and other involuntary movements. The HDR–Ratio method requires multiple patterns to be projected. The data acquisition time interval could be significant. If the human face is the scene to be imaged then movement of the face during the acquisition time interval could lead to erroneous results. Figure 2.18 shows the image of a face and the corresponding ratio image. Note the presence of two lips and two noses in the ratio image. The two lips and two noses are due to the movement of the face. To handle the scene movement, the acquisition speed must be accelerated. To achieve high acquisition speed the data acquisition process needs to be automated. The data acquisition process is automated using free open source utility called the autohotkey software [35, 1]. A screen shot of a script in autohotkey is shown in Figure 2.19. The autohotkey script synchronizes the camera and the projector. The projector patterns are automatically projected at regular intervals using the Irfan view software. The script automatically detects the position of the camera's GUI interface. It then triggers the 'capture' button of the interface and thus capturing the image of the scene. The speed at which the image is acquired is synchronized with the projector's projection speed, ie, immediately after the first pattern is projected the autohotkey script triggers the camera and the image is acquired, then the next pattern is projected and again the script triggers the camera and the image is acquired and this process repeats. With the automation process the entire acquisition can be completed before any significant movement of the scene being

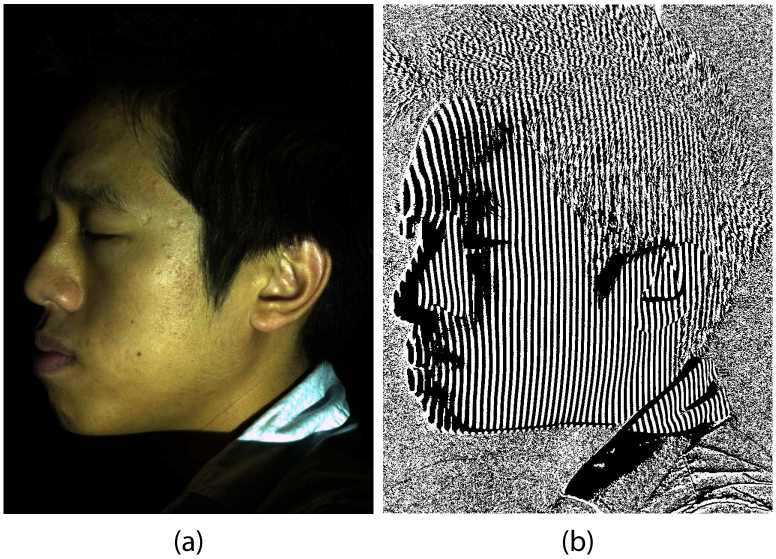
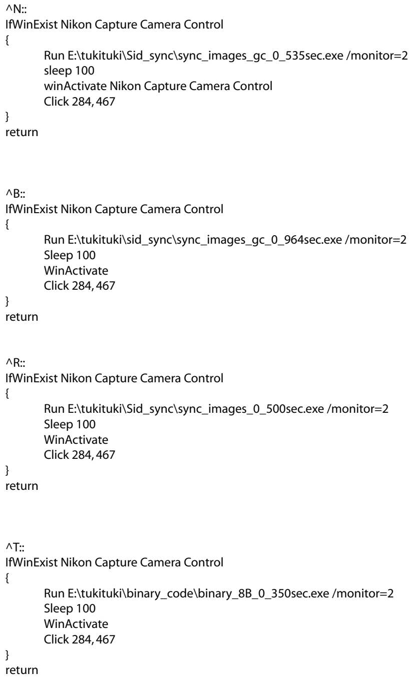


Figure 2.18: Incorrect measurement due to movement. (a): Actual face image. (b): Ratio image of the scene. Due to movement we see two lips and two noses.



```
^N::
IfWinExist Nikon Capture Camera Control
{
        Run E:\tukituki\Sid_sync\sync_images_gc_0_535sec.exe /monitor=2
        sleep 100
        winActivate Nikon Capture Camera Control
        Click 284, 467
}
return


^B::
IfWinExist Nikon Capture Camera Control
{
        Run E:\tukituki\sid_sync\sync_images_gc_0_964sec.exe /monitor=2
        Sleep 100
        WinActivate
        Click 284, 467
}
return

^R::
IfWinExist Nikon Capture Camera Control
{
        Run E:\tukituki\Sid_sync\sync_images_0_500sec.exe /monitor=2
        Sleep 100
        WinActivate
        Click 284, 467
}
return


^T::
IfWinExist Nikon Capture Camera Control
{
        Run E:\tukituki\binary_code\binary_8B_0_350sec.exe /monitor=2
        Sleep 100
        WinActivate
        Click 284, 467
}
return
```

Figure 2.19: Script to synchronize the camera and the projector.

imaged takes place.

2.2.5 Reconstruction Results

Figure 2.20 shows the reconstruction of the side view of the face image. Note the details in the ear and the side burns captured in the reconstruction. Also note the reliable reconstruction of the low albedo hair region. Figure 2.21 shows the camera image and the reconstruction of the same face, when imaged from the front.

2.3 Edge Counting Method

In the edge counting method the projector coordinate system is quantized to 256 different quantization levels. The stripe id of the imaged scene can be determined by projecting the patterns shown in Figure 2.22. The projected patterns have 256 vertical stripes which are alternately black and white. The first projected pattern starts with a white stripe and the second projected pattern starts with a black stripe. The ratio of the scene images acquired by the camera is taken. If the ratio of intensities is greater

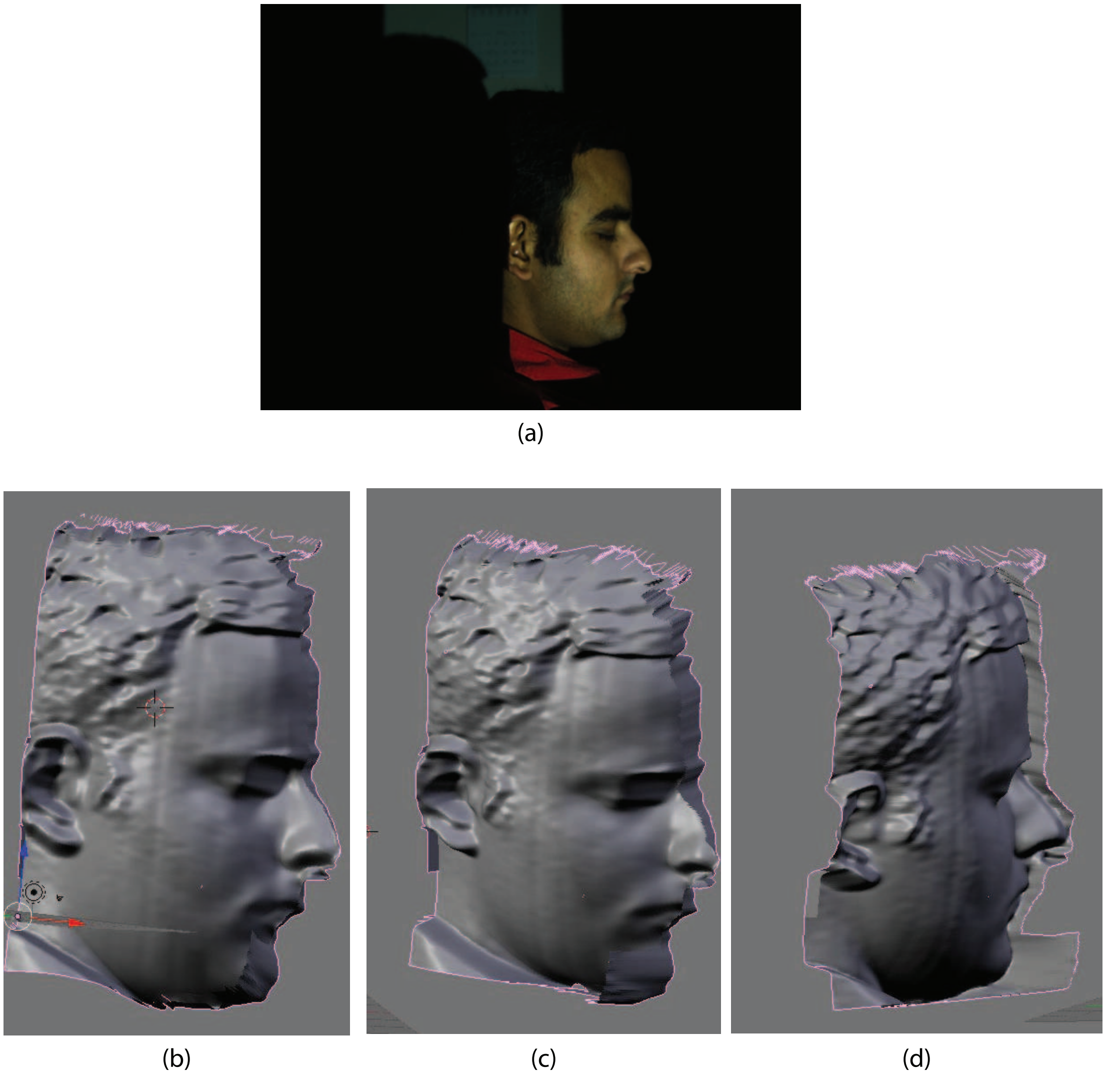


Figure 2.20: Reconstruction of the side view of a face. (a) shows the camera image. (b)-(d) shows the reconstruction of the side view from different viewpoints.

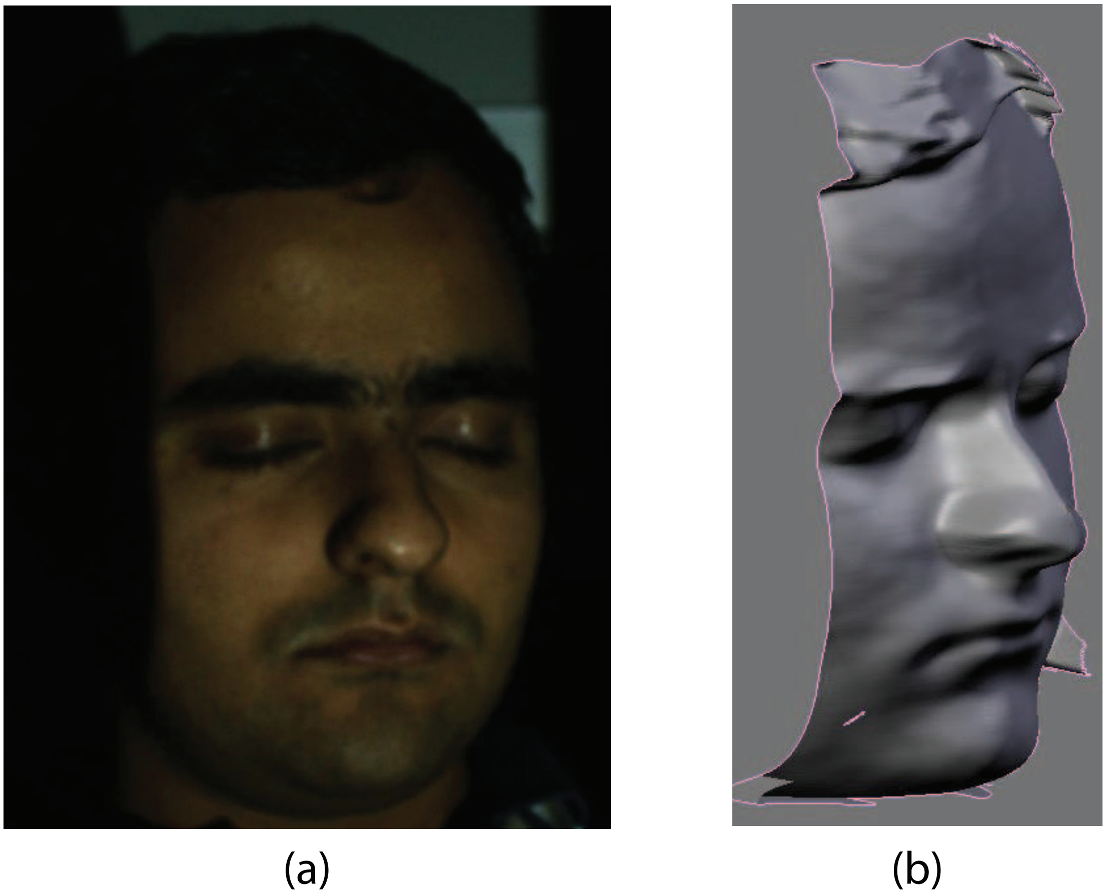


Figure 2.21: Reconstruction of the front view of a face. (a) shows the camera image of the face. (b) shows the reconstruction of the front view.

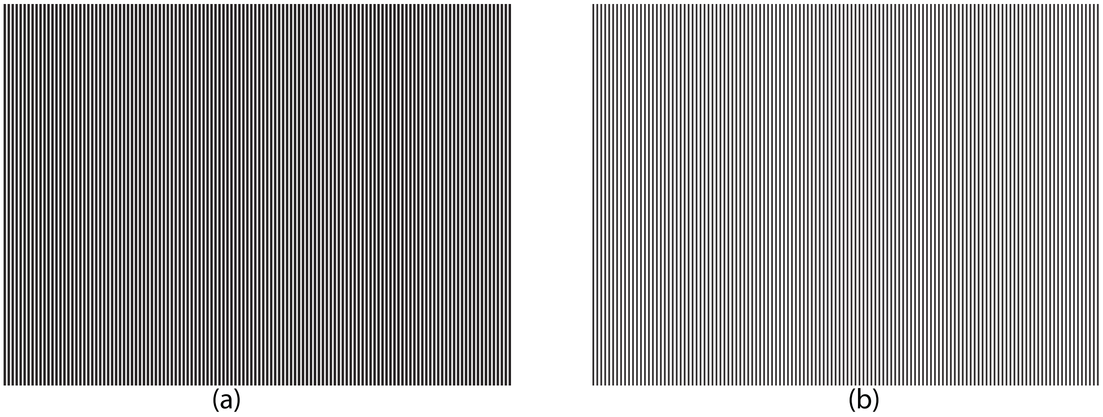


Figure 2.22: Projected patterns with 256 black and white stripes. (a) begins with a white stripe and (b) begins with a black stripe.

than one it is quantized to one and if it is less than one it is quantized to zero. The resulting ratio image is used to obtain the stripe ids of the scene points. Under the assumptions that the center stripe hits the scene and there is no occlusion, the stripe ids can be determined by counting from the center stripe. An edge detection is performed. All points from the center stripe up to the detected first edge would have a stripe id of 127. All points from the first detected edge to the next detected edge would have a stripe id of 128. Similarly all stripes from the center to the first edge on the left would have a stripe id of 127. All points from the first detected edge on the left to the second detected edge on the left would have a stripe id of 126. In this way the stripe ids of all the points in one scan line can be determined. The process is repeated for all rows. As the counting process moves form row i to row $i+1$ or row $i-1$ the reference center point at which the counting begins is changed to the average of the first detected edge locations on the left and the right. The averaging step is necessary because vertical stripes get deformed once they hit the scene to be imaged. In absence of the averaging step the center point may move out of the center stripe resulting in an error in the counting process.

When the high resolution 256 stripe pattern is projected, maintaining the reference point within the center stripe is a non–trivial problem. To address the problem of maintaining the reference point in the center stripe, in practice sixty four stripes are projected and the ratio of the scene is taken. When sixty four stripes are projected, starting from the center, the first stripe edge on the right coincides with what would have been the fifth stripe edge, from the center, if 256 stripes were projected. The second stripe edge, from the center, on the right coincides with what would have been the eighth stripe edge from the center if 256 stripes were projected. The first stripe edge on the left, starting from the center, coincides with what would would have been the third stripe edge if 256 stripes were projected. The second stripe edge on the left, starting from the center, coincides with what would have been the fifth stripe edge if 256 stripes were projected. When sixty four stripes are projected, detected edges on the right correspond to an increase of 5,3,5,3,... in the stripe ids, detected edges on the left correspond to a decrease of 3,5, 3, 5,... in the stripe ids. With 256 projected stripes, the center point has a stripe id of 127. With sixty four projected stripes, the detected stripe edges on the right of the center point would have stripe ids of $127 + 5 = 132$, $132 + 3 = 135$, $135 + 5 = 140$, $140 + 3 = 143$,...; the detected stripe edges on the left of the center point would have stripe ids of $127 - 3 = 124$, $124 - 5 = 119$, $119 - 3 = 116$,...

The steps involved in detecting the stripe ids, in the edge image, is summarized below:

1. Begin the counting at a point lying in the center stripe. Keep moving right, along the same row, till an edge is encountered. The pixel at which the edge occurs has a stripe id of $127 + 5 = 132$. Keep moving towards the right, along the same row, till the next edge is encountered. The edge point would have stripe id of $132 + 3 = 135$. The process is continued till the end of the row is encountered.

2. Using the same starting point as in step one, move towards the left, along the same row, till an edge point is detected. The pixel at which the edge was detected would have a stripe id of $127 - 3 = 124$. Keep moving towards the left, along the same row, till the next edge is encountered. The edge point would have stripe id of $124 - 5 = 119$. The process is continued till the end of the row is encountered.

3. Move to the next row and repeat Steps 1 and 2. The starting point is the average of the pixel locations of the first detected edge of step one and the first detected edge of step two. Repeat until the last row, in the image, has been traversed.

4. Move to the row before the first traversed row. Repeat Steps 1 and 2. Repeat until the first row has been traversed.

Figure 2.23 shows the images projected in the experimental sessions. The single red colored stripe corresponds to the location of the center stripe. Figure 2.24 shows the images acquired by the camera. By clicking in the red stripe region of the third image the location of the center stripe is found. The ratio image is shown in Figure 2.25. The edge detection and the reconstruction has been done within the boundaries indicated by the red line segments. Figure 2.26 shows the edge image of the region of interest. The counting is done in the edge image. Once the stripe ids of the points lying on the stripe edges are determined, a spline fit is done to obtain the stripe ids of all the scene points. The reconstruction of the scene after the spline fit is shown in Figure 2.27 The edge counting scheme has the advantage that it uses only three images to obtain the reconstruction. However the resolution obtained is much less than that for HDR–Ratio scheme.

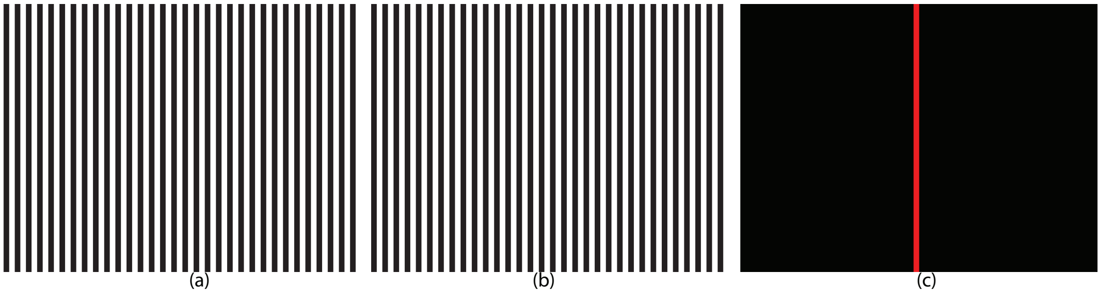


Figure 2.23: Patterns projected in the experimental session. The pattern with the single red stripe is used to identify the location of the center stripe.

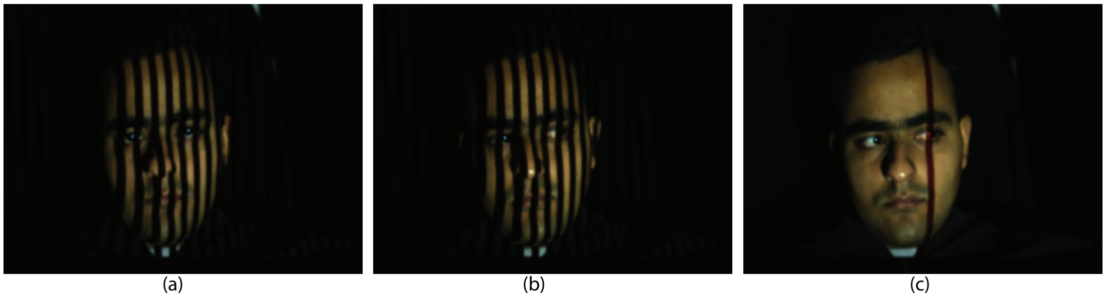


Figure 2.24: The images acquired by the camera in the edge counting method.

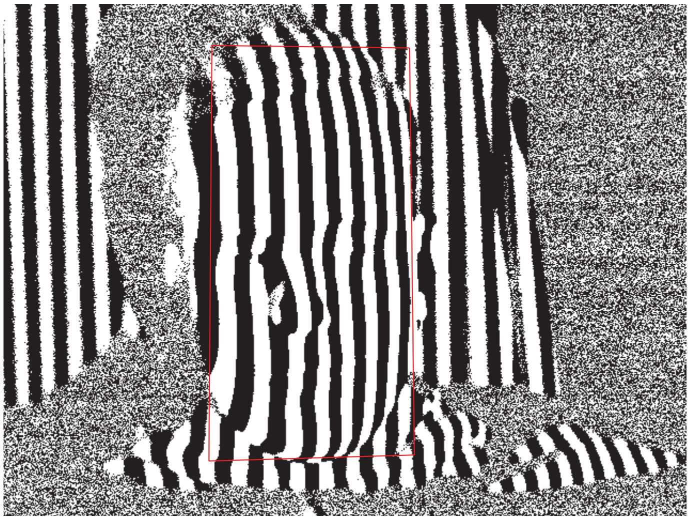


Figure 2.25: Image obtained by taking ratios of the camera images. Ratios greater than one have been set to one. Ratios less than one have been set to zero.

2.4 Summary

We proposed two novel encoding techniques, HDR–Ratio method and edge counting method. Both techniques used image ratios to obtain the stripe ids in the low albedo regions. In the HDR–Ratio method, the use of image ratios was combined with high dynamic range imaging. Using image ratios and high dynamic range imaging a very high resolution reconstruction of the scene was obtained. However, the HDR–Ratio method required multiple patterns to be projected and had a high acquisition time. The data acquisition time was accelerated using the autohotkey software. Compared to the HDR–Ratio method, the edge counting method required only three images to be projected. The acquisition time for the edge counting method was much lower, however, the resolution wasn't very high.

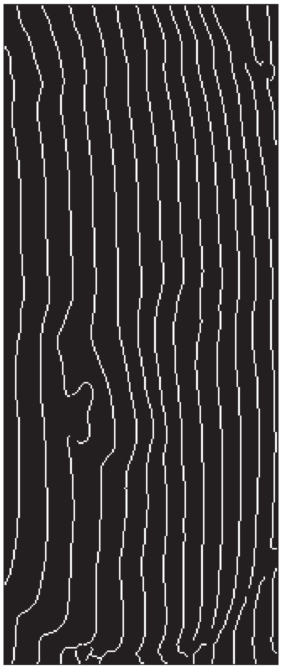


Figure 2.26: Image obtained by performing edge detection in the region of interest.

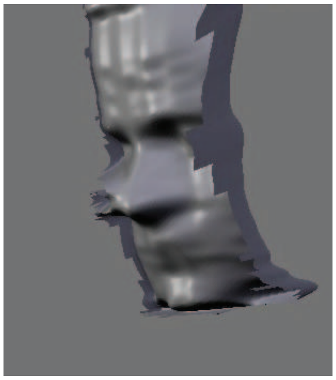


Figure 2.27: Reconstruction obtained using the Edge Counting scheme.

Chapter 3

Laser Scanner

3.1 Introduction

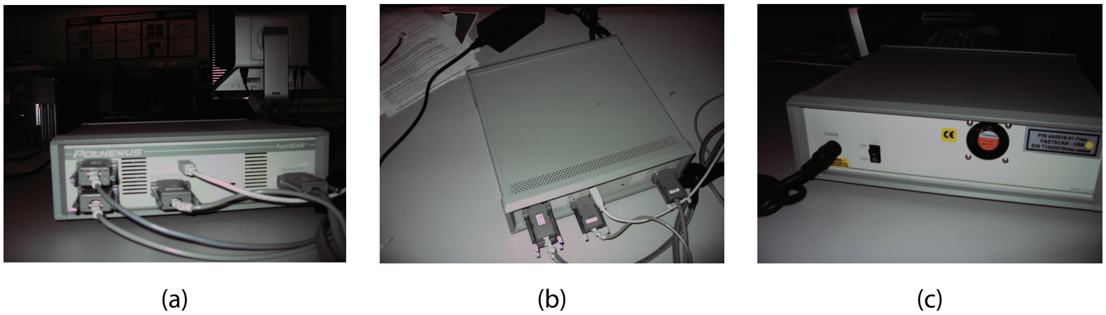
In Chapters 1 and 2, the use of structured light technique to obtain the 3D coordinates of the scene was discussed. The advantage of structured light techniques was that the coordinates were obtained in the camera coordinate frame. However, the disadvantage of structured light techniques was that the resolution obtained wasn't very high. A number of 3D scanners are available in the market. An alternate way to obtain geometry would be to use the 3D scanners available in the market. These scanners generate a very high resolution geometry of the scene. A popular hand held 3D laser scanner is the Polhemus fast scan. The Polhemus Fastscan [1] laser scanner was used in the lab to obtain the 3D coordinates of the scene points. The Polhemus Fastscan gives a very high resolution scan of the 3D coordinates. However, the disadvantage of the laser scanner is that the coordinates obtained aren't in the camera coordinate frame.

We will discuss the step by step procedure of using the laser scanner. We will begin with a description of the different components of the laser scanner, then discuss the steps involved, namely, setting up the scanner, calibrating the scanner, scanning the object and using the stylus. We will conclude with a discussion of obtaining the coordinates from the camera viewpoint. The coordinates in the camera viewpoint can be obtained by calibrating using fiducial markers. In the fiducial marker method, black round markers are placed on the scene being scanned. Using the 3D coordinates and the pixel location of the fiducial markers, scene coordinates in the camera viewpoint are obtained.

[1] Polhemus, 40 Hercules Drive, P.O. Box 560, Colchester, VT 05446

3.2 Laser Scanner

We have used the Polhemus Fastscan laser scanner. The Polhemus Fastscan generates a very high resolution geometry. The main components of the scanner are the processing unit, wand, transmitter, receiver and the stylus. The processing unit is shown in Figure 3.1. The processing unit houses the electronics for the scanner. The processor unit contains the magnetic tracker and the video processing unit. The processing unit has connections for the transmitter, the receiver and the wand.



(a)

(b)

(c)

Figure 3.1: Processing unit of the hand held scanner. (a), (b) and (c) show the processing unit from the front, top and back respectively.

The transmitter is shown in Figure 3.2 . The transmitter generates magnetic field in all three directions. The amplitude of the magnetic field is used to track the position of the wand with respect to the transmitter. Note the coordinate frame defined on the transmitter. Either the transmitter or the receiver can be used as the reference coordinate frame. When the transmitter is used as the reference, the world coordinates

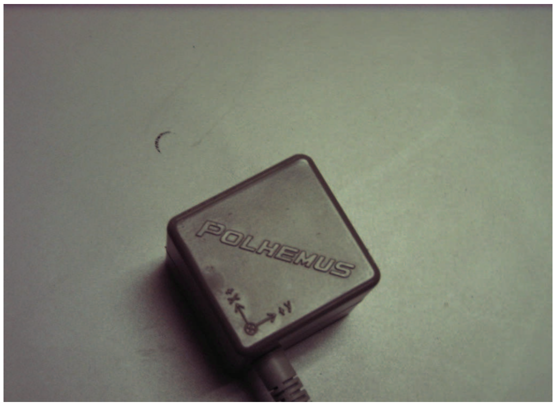


Figure 3.2: Transmitter of the laser scanner.

are obtained with respect to the coordinate frame defined on the transmitter. When the receiver is used as the reference frame the coordinates are obtained with respect to the receiver. Figure 3.3 shows the image of the receiver.

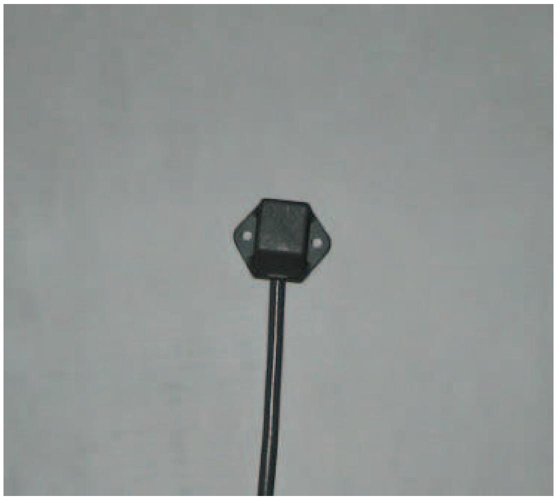


Figure 3.3: Receiver.

The wand is shown in Figure 3.4. The wand has two cameras attached at the two ends. When a region is scanned, the wand projects a strip of laser light on the region being scanned. The laser is detected by the wand and the two cameras triangulate to obtain the 3D coordinates of the scene points.

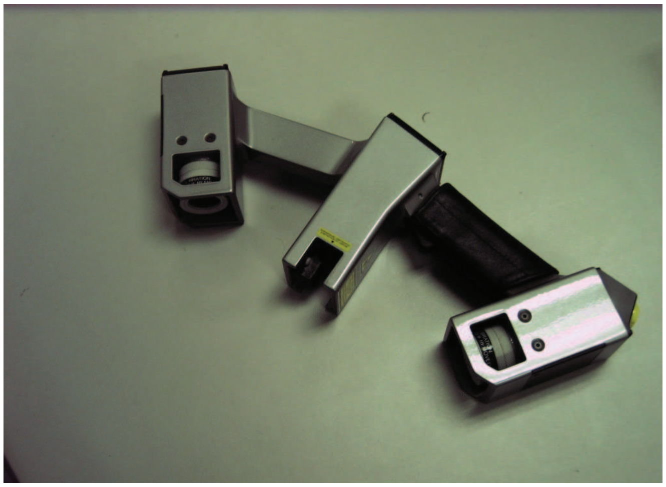


Figure 3.4: Wand of the laser scanner.

The stylus is used to obtain the 3D coordinates of specific scene points. Figure 3.5 shows the screen shot of a stylus. To obtain the 3D coordinates of a particular scene

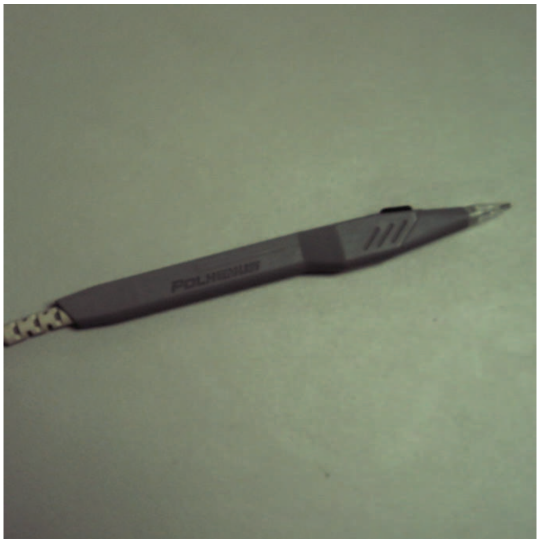


Figure 3.5: Stylus of the laser scanner.

point the stylus is pointed towards the 3D point and clicked. Once all the desired 3D locations are obtained, the points can be stored in a '.csv' file. The '.csv' file can be read using any text editing program.

3.3 Using the Laser Scanner

Using the Polhemus scanner to scan an object consists of three major steps: setting up the scanner, calibrating the scanner and scanning the object.

3.3.1 Setting up the Scanner

The following steps serve as a guideline to setup the scanner:

1. The processing unit is connected to the computer and to the power point. The processing unit has connections for the transmitter, receiver and the wand in the front (see Figure 3.1). The stylus and the receiver cannot be used simultaneously. When the stylus is used, it is connected in the receiver's position in the processor unit.

2. Once the wand and the reference device (transmitter or receiver) are connected to the processing unit the Fastscan software is started. The Fastscan software

can be started by clicking on the software's icon on the screen. Figure 3.6 shows the screen shot when the Fastscan software is started. The 'Scanner Offline' at the lower right indicates that the power switch of the scanner is off. The power switch of the scanner is located on the backside of the processor unit. When the power switch is turned on, 'scanner online' is displayed on the lower right.

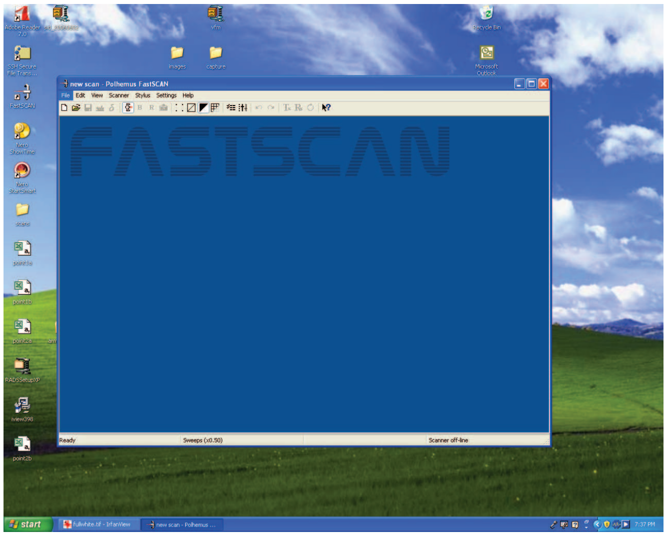


Figure 3.6: Screen shot of the Fastscan software.

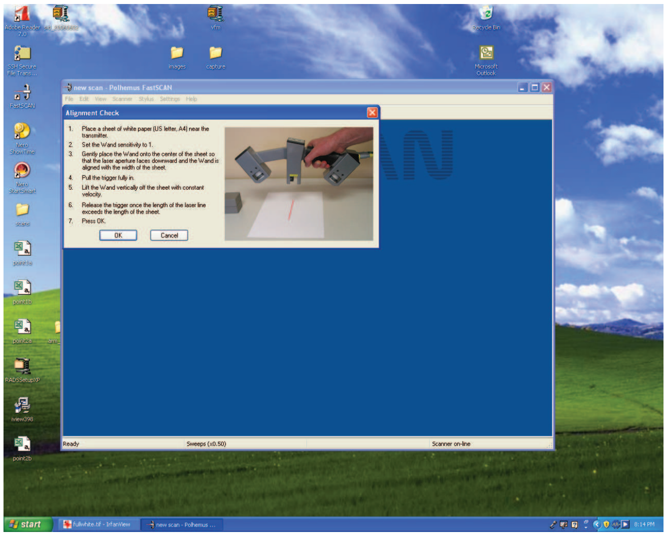
3. To register the position of the wand with respect to the reference (transmitter or receiver), the wand is pointed in the direction of the transmitter and clicked. If the transmitter is used as the reference, then, from this point onwards the transmitter must remain stationary.

The scanner is now ready to be calibrated.

3.3.2 Calibrating the Scanner

When the wand leaves the factory it is fully calibrated. However, for optimum performance it is necessary to check the calibration before using the scanner. The calibration can be checked using the following steps:

1. In the *scanner* menu, click on the *alignment check* option. When 'alignment check' is clicked the the instructions shown in Figure 3.7 gets displayed.



1. Place a sheet of white paper (US letter, A4) near the transmitter.
2. Set the Wand sensitivity to 1.
3. Gently place the Wand onto the center of the sheet so that the laser aperture faces downward and the Wand is aligned with the width of the sheet.
4. Pull the trigger fully in.
5. Lift the Wand vertically off the sheet with constant velocity.
6. Release the trigger once the length of the laser line exceeds the length of the sheet.
7. Press OK.

Figure 3.7: Alignment Check.

2. The wand is placed very close to a white paper. The trigger is pulled and the wand is slowly raised till the laser extends over the entire width of the white paper. When the laser extends over the entire width of the paper the trigger is released. On the screen a number is displayed. If the number is less than two the wand is properly calibrated. If the number is greater than two the wand needs to be calibrated.

The wand can be calibrated in the following steps

1. The calibration target is attached to the transmitter. The transmitter with the calibration target attached is shown in Figure 3.8

2. From *scanner* menu select the *calibration correction* option.

3. Holding the wand 100mm above the calibration target, each side of the calibration target is scanned. The computer beeps when the wand is held in the range $80 - 120$mm during the first scan of the calibration target. Each side of the calibration target is rescanned by holding the wand 200mm above the calibration target. The computer beeps when the wand is held in the range $180 - 220$mm during the second scan of the calibration target.

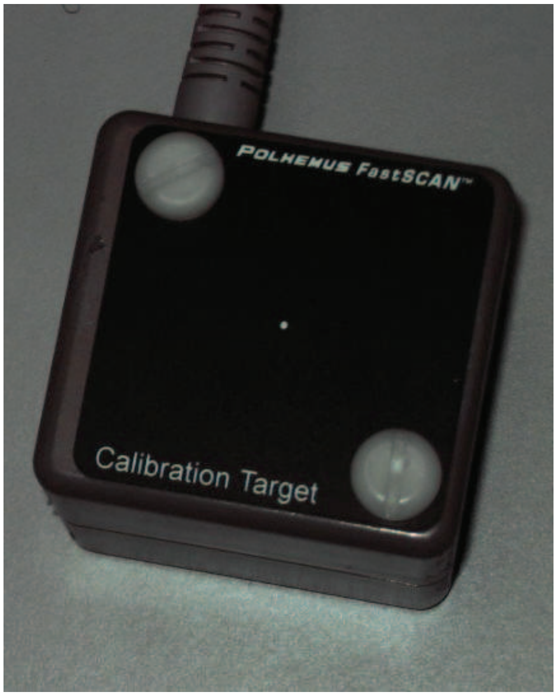


Figure 3.8: Calibration Target.

4. After the calibration step is performed, the alignment is rechecked. If the calibration is incorrect then the calibration process is repeated.

3.3.3 Scanning the Object

Once the wand is calibrated the scanner is ready to be used to scan an object. A typical scan session is shown in Figure 3.9. The wand projects laser light on the area to be scanned. A camera is attached to each end of the wand. The laser light is detected and the two cameras triangulate to estimate the 3D coordinates of the point with respect to the wand. The entire object is scanned by moving the wand to different parts of the object. The scanning procedure is completed in a few minutes.

3.4 Calibration

In the experimental sessions the transmitter has been used as the reference. The 3D coordinates obtained using the laser scanner are with respect to the transmitter coordinate frame. The transmitter coordinate frame is the world coordinate frame. When

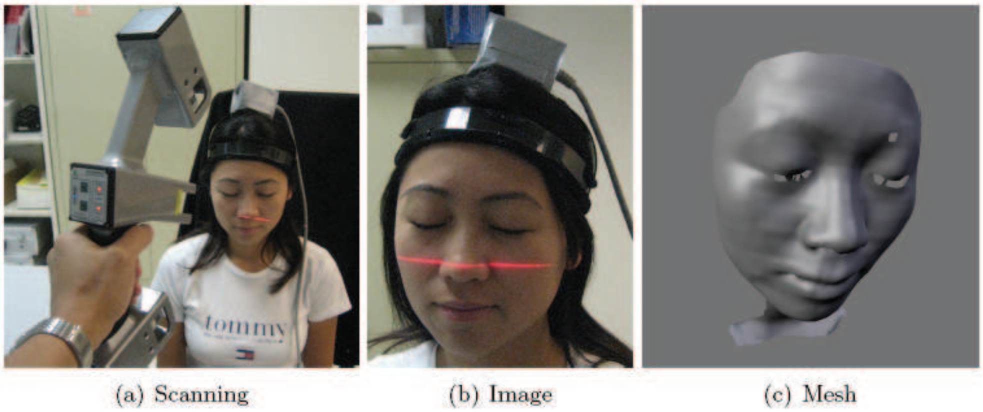

(a) Scanning (b) Image (c) Mesh

Figure 3.9: Illustration of a scan session with the laser scanner.

the image of the scene is acquired, the camera image gives the pixel location of each scene point. In the calibration step, the transformation from the transmitter coordinate frame to the pixel locations is determined. The transformation can be determined by using fiducial points described below.

3.4.1 Calibration Using the Fiducial Points

Round circular black markers are attached to the object being scanned. The markers serve as fiducial points for the registration of the geometry. Figure 3.10 shows the camera image and the laser scan of a foam head. To obtain the world coordinates

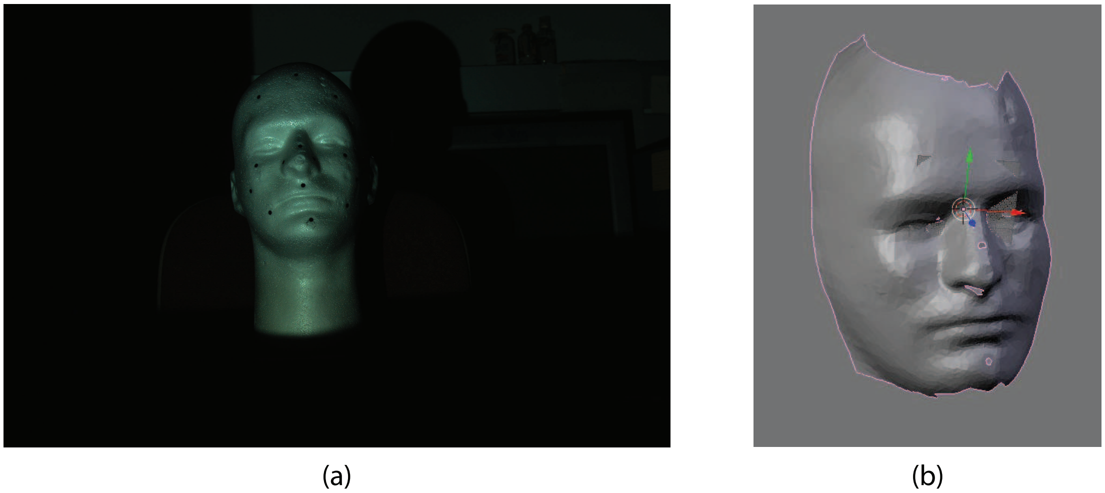
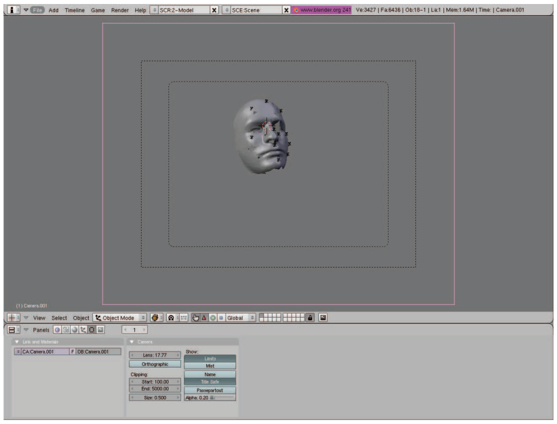

(a) (b)

Figure 3.10: Camera image and the laser scan of the foam head.

of the fiducial points the stylus is pointed to each fiducial point and clicked. The coordinates of the fiducial points are stored in a '.csv' file. The pixel locations of the fiducial points can be detected by clicking on the image. Once the corresponding world coordinates and camera coordinates is known calibration can be performed using Tsai's camera calibration algorithm [32]. Figure 3.11 shows the screen shot of the blender [2] interface once the calibration is performed. The square box indicates that the world is being seen from the camera viewpoint.



Ve:3427 | Fa:6436 | Ob:18-1 | La:1 | Mem:1.64M | Time: | Camera.001

Lens: 17.77

Start: 100.00

End: 5000.00

Size: 0.500

Alpha: 0.20

Figure 3.11: Blender interface showing the world from the camera viewpoint.

3.5 Summary

We have discussed the use of a hand held laser scanner in obtaining the geometry of the scene was discussed. We presented a step by step procedure to use the hand held Polhemus Fastscan laser scanner. The major steps in using the scanner were setting up the scanner, calibration, using the wand to scan the scene and using the stylus to obtain the 3D coordinates of specific points. Using the laser scanner, high resolution scans of the scene could be obtained. However, the disadvantage was that the world points were not in the camera coordinate frame. The use of fiducial markers to obtain the coordinates in the camera coordinate frame was also discussed.

[2] Blender is a free 3D content creation suite.

Chapter 4

Texture Unwrapping

4.1 Introduction

Using geometry acquisition techniques like structured light or the laser scanner, it is difficult to acquire fine scale geometry. However, the appearance of texture is greatly influenced by fine scale local geometry. The effect of fine scale geometry on the appearance of texture can be accounted for by unwrapping the mesh and mapping the texture on the unwrapped mesh. We have used the least square conformal mapping technique has been used to unwrap the mesh.

Mapping the reflectance data on the unwrapped mesh requires the solving of an interpolation problem. Various interpolation techniques, suitable for the texture mapping problem, are available. We will discuss the bilinear interpolation technique to interpolate the data.

4.2 Texture Mapping and Texture Unwrapping

Textures, in our case, are two dimensional images indexed by a two dimensional s–t coordinate system. The s–t coordinate system is called as the texture coordinates. The texture coordinates are commonly referred as texels. In texture mapping, the texture is mapped on curved surfaces like spheres, cylinders or face scans. The surface is defined in the world coordinate system. The surface is expressed in the parameterized form $(x(u,v), y(u,v), z(u,v))$. The texture mapping problem defines a mapping from texels to the u–v parameters.

$$u = u(s,t) \tag{4.1}$$

$$v = v(s,t) \tag{4.2}$$

The texture mapping problem is difficult because the surface is curved. Mapping from the planar texture to the curved surface results in some amount of distortion.

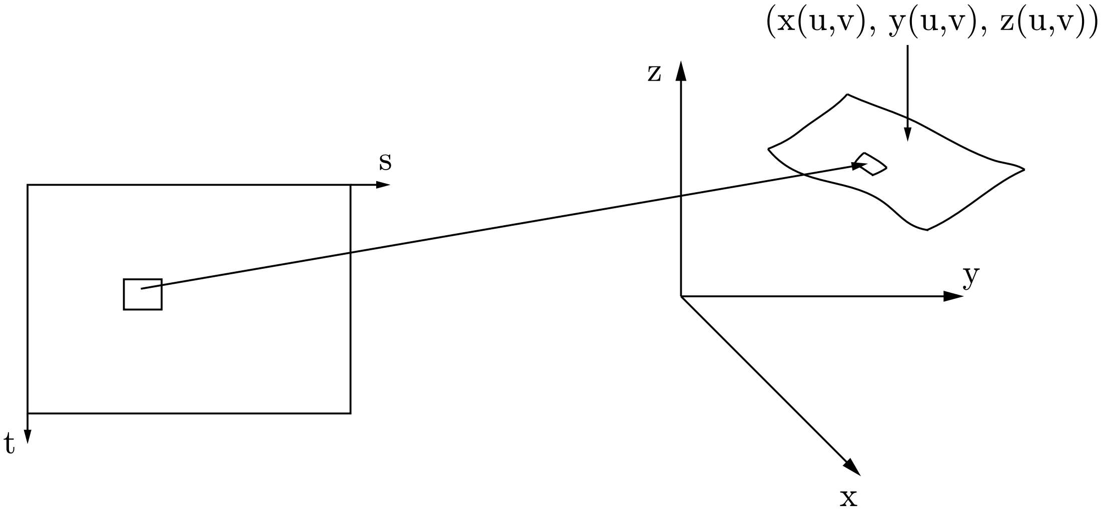


Figure 4.1: Mapping of texels to surface.

In the inverse texture mapping problem, the texture values at the integer u–v locations are evaluated. The resulting u–v image, formed by the texture values at integer u–v locations is called as the texture image. To obtain the texture image, we need to parameterize the mesh and perform interpolation. In the parameterization step we associate a unique u–v value with each mesh point. We parameterize the mesh using least squares conformal mapping (LSCM). Generally an integer u–v location gets mapped to a non integer s–t value. To obtain the texture at integer u–v locations, texture values at non integer s–t locations are required. Texture values at non integer s–t locations are obtained using interpolation.

Least Squares Conformal Mapping (LSCM)

In this section, we discuss least squares conformal mapping for unwrapping the mesh. The notations and equations used are consistent with those used in the paper [22]. We emphasize that we have not implemented the LSCM algorithm. We have used implementation in the 3D creation suite Blender to do the unwrapping of the mesh.

Consider the triangulation $G=\{1...n, \mathbf{T}, \mathbf{p}_j\}$. Here n is the number of vertices, $\mathbf{T}$ is the set of $n^{'}$ triangles and $\mathbf{p}_j$ is the geometric location of vertex j. With each triangle

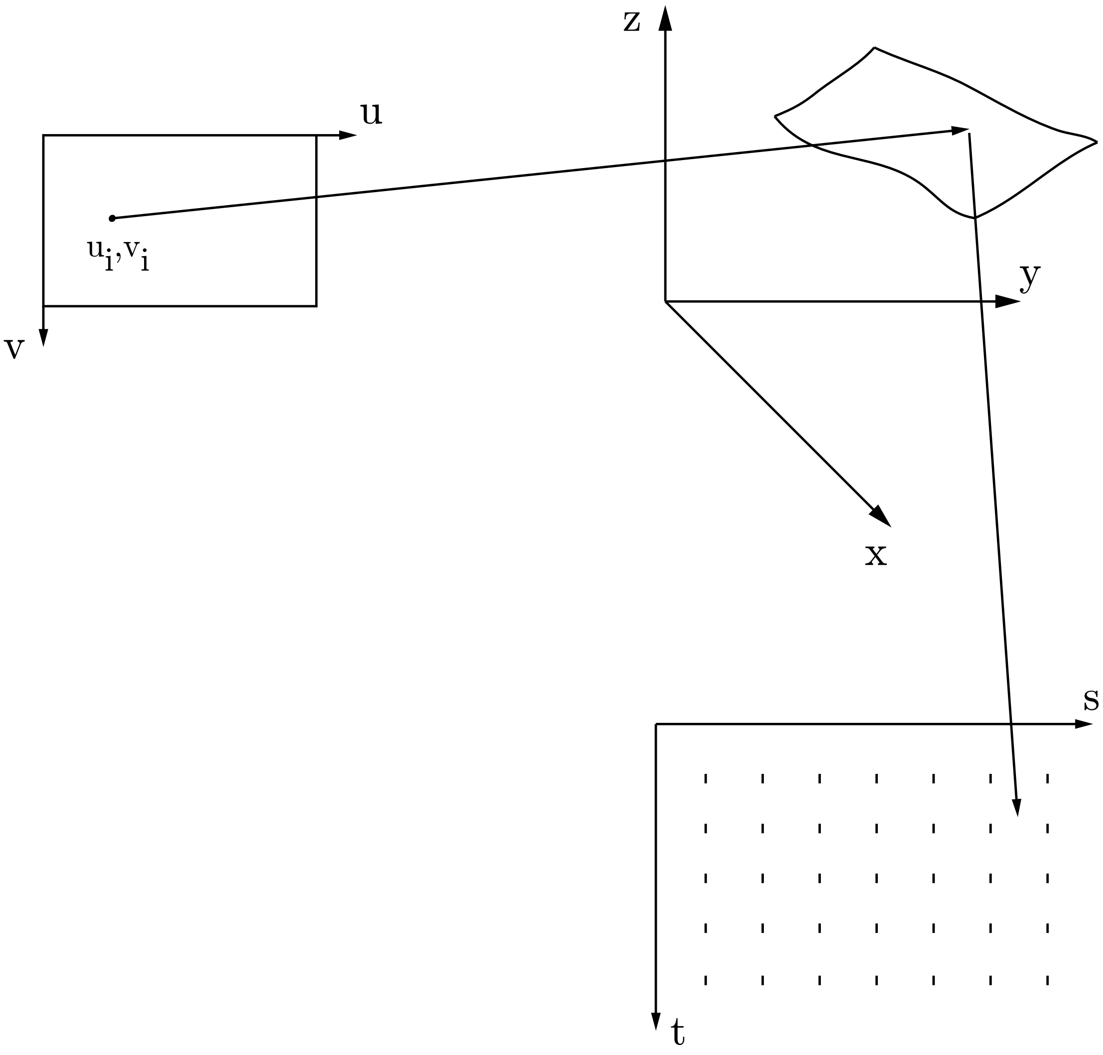


Figure 4.2: Integer u–v values are mapped to non integer s–t values.

in $T_{\hat{i}}$, $\hat{i} \in \{1, ..., n'\}$, an orthonormal basis is defined. Let (x_1, y_1), (x_2, y_2), (x_3, y_3) be the coordinates of the vertices of a triangle with respect to the local coordinate system. A conformal mapping $U{:}(x, y) \mapsto (u, v)$ has to be defined. The conformal constraint requires that

$$\frac{\partial U}{\partial x} + i\frac{\partial U}{\partial y} = 0, \tag{4.3}$$

where, $U = u + iv$. In a practice exact conformality cannot be achieved and is enforced in the least square sense. The least squares error for a particular triangle is:

$$C(T) = \int_T \left| \frac{\partial U}{\partial x} + i\frac{\partial U}{\partial y} \right|^2 dA = \left| \frac{\partial U}{\partial x} + i\frac{\partial U}{\partial y} \right|^2 A_T, \tag{4.4}$$

where, A_T is the area of the triangle. Summing over all the triangles the criterion is to minimize

$$C = \sum_T C(T). \tag{4.5}$$

Consider a triangle with local coordinates as (x_1, y_1), (x_2, y_2), (x_3, y_3). The partial derivatives of u with respect to x and y can be approximated as:

$$\begin{bmatrix} \partial u/\partial x \\ \partial u/\partial y \end{bmatrix} = \frac{1}{d_T} \begin{bmatrix} y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \tag{4.6}$$

where, d_T is twice the area of the triangle and u_1, u_2, and u_3 are the mappings of the triangle vertices. Equation 4.6 can be written as

$$\frac{\partial u}{\partial x} + i\frac{\partial u}{\partial y} = \frac{i}{d_T}[W_1\ W_2\ W_3][u_1\ u_2\ u_3]^T, \tag{4.7}$$

where,

$$W_1 = (x_3 - x_2) + i(y_3 - y_2),$$

$$W_2 = (x_1 - x_3) + i(y_1 - y_3),$$

$$W_3 = (x_2 - x_1) + i(y_2 - y_1).$$

Equation 4.3 can now be written as:

$$\frac{\partial U}{\partial x} + i\frac{\partial U}{\partial y} = \frac{i}{d_T}[W_1\ W_2\ W_3][U_1\ U_2\ U_3]^T = 0, \tag{4.8}$$

where, $U_j = u_j + v_j$. The error criterion C can be written as

$$C = \mathbf{U}^* \mathbf{Z} \mathbf{U}, \tag{4.9}$$

where, $\mathbf{U} = [U_1\ U_2 ... U_n]^T$ and $\mathbf{Z} = \mathbf{M}^* \mathbf{M}$. $\mathbf{M}$ is a sparse matrix such that

$$m_{ij} = \begin{cases} \frac{W_j}{\sqrt{d_{T_{\hat{i}}}}} & \text{if vertex j belongs to triangle } T_{\hat{i}} \\ 0 & \text{otherwise} \end{cases}$$

Substituting $\mathbf{Z} = \mathbf{M}^*\mathbf{M}$ in equation 4.9 we get,

$$C = \|\mathbf{MU}\|_2^2 \tag{4.10}$$

To obtain a non trivial solution the values of some $U_j's$ need to be assumed. The collection $\mathbf{U}$ can be split into $[\mathbf{U}_f \;\; \mathbf{U}_p]$. The $\mathbf{U}_f$ is the vector of $U_j's$ whose values haven't been assumed and $\mathbf{U}_p$ is the vector of $U_j's$ whose values have been assumed. On the similar lines, the matrix $\mathbf{M}$ can be split into $[\mathbf{M}_f \; \mathbf{M}_p]$. Equation 4.10 can now be written as,

$$C = \|\mathbf{M}_f\mathbf{U}_f + \mathbf{M}_p\mathbf{U}_p\|_2^2. \tag{4.11}$$

Rewriting with real vectors we get,

$$C = \|\mathbf{Ax} - \mathbf{b}\|_2^2 \tag{4.12}$$

where,

$$\mathbf{A} = \begin{bmatrix} \mathbf{M}_f^1 & -\mathbf{M}_f^2 \\ \mathbf{M}_f^2 & \mathbf{M}_f^1 \end{bmatrix}, \; \mathbf{b} = -\begin{bmatrix} \mathbf{M}_p^1 & -\mathbf{M}_p^2 \\ \mathbf{M}_p^2 & \mathbf{M}_p^1 \end{bmatrix} \begin{bmatrix} \mathbf{U}_p^1 \\ \mathbf{U}_p^2 \end{bmatrix}.$$

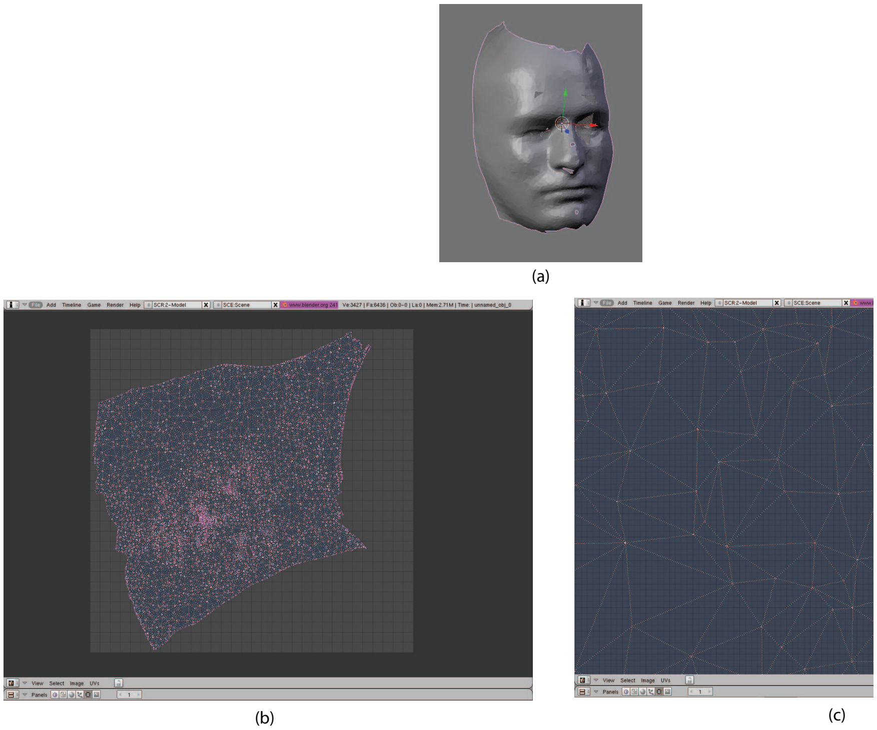
Here, $\mathbf{M}_f^1$, $\mathbf{M}_f^2$ denote the real and imaginary parts of $\mathbf{M}_f$ and $\mathbf{M}_p^1$, $\mathbf{M}_p^2$ denote the real and imaginary parts of $\mathbf{M}_p$. The minimization problem is a least squares problem [30]. The minimizing solution has a closed form solution and does not require the use of iterative techniques. The minimization problem has the following interesting properties:

1. The matrix A is full rank. As a result the minimization problem has a unique solution.

2. The solution is invariant to the resolution of the mesh. The solution, when restricted to the original mesh, is invariant if the density of the mesh is increased.

3. All triangles are consistently oriented, ie, triangle flips do not occur

Figure 4.3 shows the scanned mesh of a foam head and the unwrapped mesh. The mesh unwrapping was done in blender.


(a)

(b)

(c)

Figure 4.3: Unwrapping a mesh. (a) shows the laser scan of the foamhead. (b) shows the unwrapped mesh obtained using LSCM. (c) shows the unwrapping of a particular region of the mesh.

4.3 Mapping Reflectance Data on the Unwrapped Mesh

The unwrapped mesh gives the u–v coordinates of each vertex of the mesh. Each mesh vertex is colored with its u–v coordinates, ie, when the mesh is rendered with the camera parameters the red channel is colored with the u coordinate, the green channel with the v coordinate and the blue channel is assigned any arbitrary color. The mesh is rendered using the camera parameters obtained through the calibration process. The rendering process generates an image where each each pixel has its red and green intensity as the corresponding u and v values. The blue intensity is set to zero. The u–v value associated with each pixel location is now known. Equivalently, we know the pixel locations associated with non integer u–v values. To obtain a 2464 resolution texture image the red, green and blue intensity values associated with integer u–v values is required. However, rendering the mesh gives the intensity values at non–integer u–v locations. The intensity values at integer u–v locations is interpolated using the the intensity values at non–integer u–v values. The interpolation was done using bilinear interpolation. Figure 4.4 shows the camera image of a foam head and corresponding the texture image obtained using bilinear interpolation.

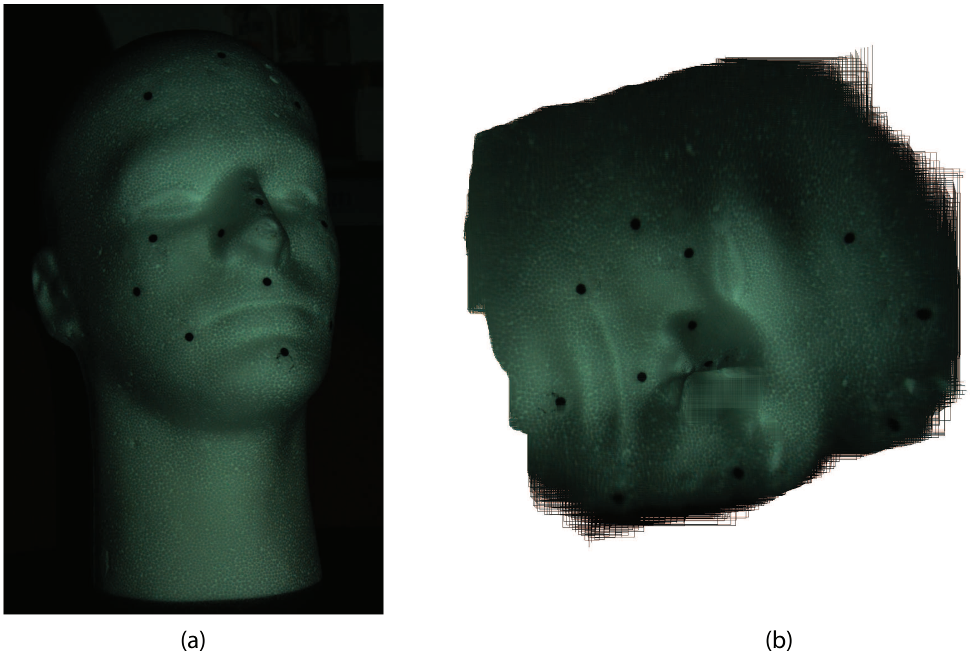


Figure 4.4: Camera image and texture image of a foamhead. (a) Shows the camera image of the foam head. (b) Shows the texture image obtained by bilinear interpolation.

4.4 Summary

We discussed the procedure for texture unwrapping. Texture unwrapping accounts for the effect of geometry on the appearance of texture. The different steps involved in the mapping procedure, geometry acquisition, mesh unwrapping and interpolation, were discussed in great detail. The geometry was acquired using the hand held laser scanner. The unwrapping of the mesh was done using least squares conformal mapping. Least squares conformal mapping had the advantages that minimizing solution was unique, the solution was invariant to similarity in texture space and triangle flips did not occur. The interpolation was done using bilinear interpolation.

Chapter 5

Surface–Subsurface Separation

5.1 Introduction

The interaction of light with the skin surface is a complex process. To understand the appearance of skin it is necessary to understand the different components of skin contributing to the appearance of the skin surface. The components influencing the appearance of the skin can be divided into the micro, meso and macro–scale. In this chapter the micro, meso and the macro–scale components we be discussed.

The modeling of the complex interaction of light with the skin surface is discussed in the chapter. The light reflected form the skin surface is divided into the surface and the subsurface components. The surface and the subsurface components have different properties and should be modeled separately. This chapter discusses the difference in properties of the surface and the subsurface components. Separation of the surface and the subsurface components using a polarizer are discussed in the chapter.

5.2 Skin

Skin is the topic of study in multiple areas like computer graphics, computer vision, dermatology and cosmeology surgery [18]. The goal and the aspects of skin studied in each field is different. In computer graphics and computer vision the study of skin finds applications in photo realistic rendering, animation and geometric modeling. In dermatology the goal is to study and treat skin disorders and in cosmetology the applications are cosmetic surgery and makeup development.

In many applications it is important to understand the appearance of skin. Depending on the size, the components which contribute to the appearance of skin can be classified as micro–scale, meso–scale and macro–scale.

Micro–Scale

The cellular level elements and the skin layers constitute the micro–scale components of the skin. The micro–scale components are barely visible to the human eye. The appearance of the micro–scale components depends upon the interaction of the components with the incident light. The dominant effect produce at this scale are scattering and absorption. These effects depends upon the size, shape and the refractive indices of the components.

Meso–Scale

Skin and skin features constitute the meso–scale components. The meso–scale components are clearly visible to the naked eye. The appearance of the meso–scale components depends upon the interaction of the micro–scale components with the incident light. Features like wrinkles, pores freckles are the meso scale components of skin.

Macro–Scale

Body parts and body regions constitute the macro–scale components of skin. The appearance of the different body parts is different. For example, the forehead and the nose are more glossy than the neck. Unfortunately there are no known physical models that describe appearance variation in a unified framework. In order to analyze and synthesize the appearance of skin it is necessary to computationally model the appearance of skin. The commonly used models in computer vision and computer graphics are bidirectional reflectance distribution function and bidirectional texture function.

Bidirectional Reflectance Distribution Function (BRDF)

The BRDF f_r [26] at point x is the ratio of the light reflected in direction $\vec{\omega}$ to the ratio ratio of light incident in direction $\vec{\omega'}$.

$$f_r(x, \vec{\omega}, \vec{\omega'}) = \frac{dL_r(x, \vec{\omega})}{dE_i(x, \vec{\omega'})}. \tag{5.1}$$

The function assumes that the light incident at a particular point on the skin surface would be reflected from the same point. However, when light is incident on the surface of the skin part of the incident light penetrates into the skin surface, scatters and then leaves the surface from a different point. Since part of the light incident to the skin surface leaves the surface from a different point the BRDF function is not an accurate model the interaction of light with the skin surface. However, since the BRDF function has low computational cost it is widely used for skin rendering.

Bidirection Texture Function (BTF)

Rendering obtained using BRDF modeling is not realistic. Realistic rendering of human faces requires the modeling of meso scale components like wrinkles, pores and freckles. The BTF can be used to model the bidirectional appearance of meso scale components. The BTF is a six–dimensional function which describes the dependency of texture on the viewing and illumination directions. Details of BTF can be found in [8].

5.3 Subsurface and Surface Components

When light is incident on the surface of the skin, part of the light is reflected off the surface of the skin and part of the incident light penetrates the skin's surface. The light which reflects off the surface of the skin is called the surface component. The light which penetrates into the surface of the skin is called the subsurface component [7]. The subsurface and the surface components have significantly different properties and should be modeled separately.

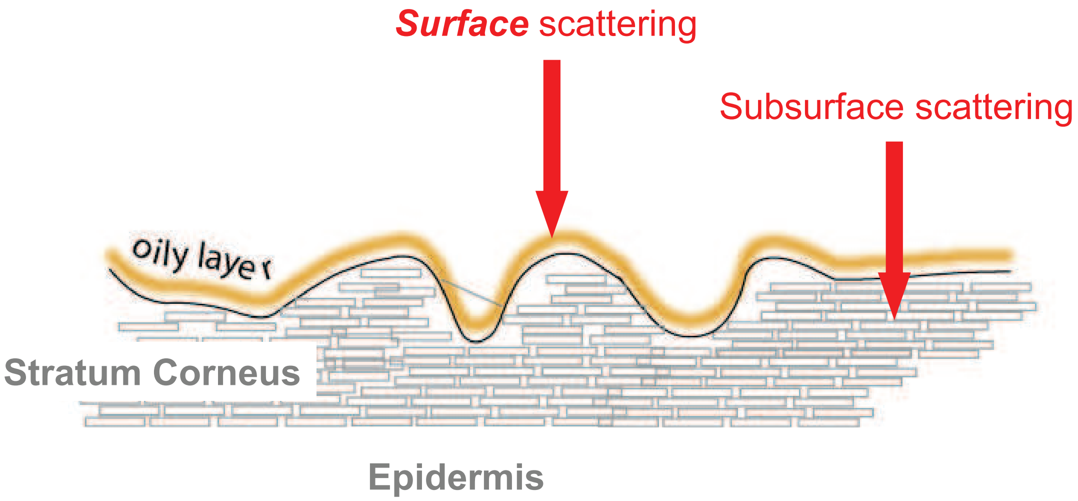


Figure 5.1: Surface and subsurface scattering.

5.3.1 Subsurface Component

Subsurface component is the part of the incident light which penetrates the skin surface. The subsurface component follows the linear reflectance model [4, 25]. In the linear reflectance model, if $n(\mathbf{x})$ denotes the normal vector scaled by the albedo of the surface at point $\mathbf{x}$ and $\mathbf{a}$ denotes the direction of the light source scaled by the intensity of the light source, then the brightness of the point $e(\mathbf{x})$ would be given by

$$e(\mathbf{x}) = n(\mathbf{x}) \cdot \mathbf{a}. \tag{5.2}$$

Let $\{\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}\}$ denote three source vectors. The brightness of a point due to the three source vectors is given by

$$e_1(\mathbf{x}) = n(\mathbf{x}) \cdot \mathbf{a_1}, \tag{5.3}$$

$$e_2(\mathbf{x}) = n(\mathbf{x}) \cdot \mathbf{a_2}, \tag{5.4}$$

$$e_3(\mathbf{x}) = n(\mathbf{x}) \cdot \mathbf{a_3}. \tag{5.5}$$

If $\mathbf{a}$ is any arbitrary source vector then $\mathbf{a}$ can be expressed as a linear combination of $\{\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}\}$ in the following way:

$$\mathbf{a} = \alpha_1\mathbf{a_1} + \alpha_2\mathbf{a_2} + \alpha_3\mathbf{a_3}. \tag{5.6}$$

The intensity of the point when illuminated by the source vector $\mathbf{a}$ is given by

$$
\begin{aligned}
e(\mathbf{x}) &= n(\mathbf{x}) \cdot \mathbf{a} && (5.7)\\
&= n(\mathbf{x}) \cdot (\alpha_1 \mathbf{a_1} + \alpha_2 \mathbf{a_2} + \alpha_3 \mathbf{a_3}) && (5.8)\\
&= \alpha_1 n(\mathbf{x}) \cdot \mathbf{a_1} + \alpha_2 n(\mathbf{x}) \cdot \mathbf{a_2} + \alpha_3 n(\mathbf{x}) \cdot \mathbf{a_3}. && (5.9)
\end{aligned}
$$

The intensity due to the source vector $\mathbf{a}$ is a linear combination of the intensities due to $\{\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}\}$. If the source vector $\mathbf{a}$ is distant and visible to all points the resulting image $\mathbf{I}$ would thus be a linear combination of the images $\mathbf{I}_1$,$\mathbf{I}_2$ and $\mathbf{I}_3$. The images of the subsurface component obey the linear reflectance model and hence lie on a linear manifold of dimensionality three. As shown in Figure 5.2 the subsurface component leaves the skin surface uniformly in all directions [11]. Since the subsurface component leaves the skin surface uniformly in all directions the appearance of the subsurface component images does not vary significantly with the viewing direction. The subsurface component does not retain the fine scale details of the skin surface. The fine scale details are captured in the surface component.

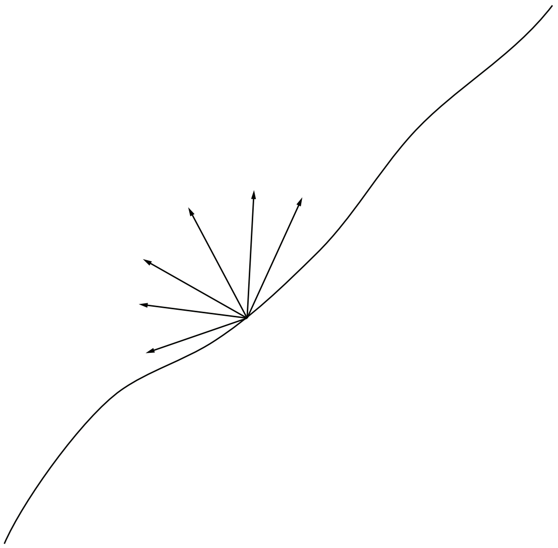


Figure 5.2: The subsurface component leaves the skin surface uniformly in all directions.

5.3.2 Surface Component

Surface component is the part of the incident light which reflects off the surface of the skin. As shown in Figure 5.3 the surface component reflects off the skin's surface in the direction of the spherical lobe [11, 27]. Since the reflection is concentrated in the direction of the spherical lobe the appearance of the surface component depends significantly on the viewing direction. The images of the surface component lie on a nonlinear manifold and are more difficult to model than the images of the subsurface component. However, fine scale details like pores, wrinkles, scars are well brought out in the surface component images. The modeling of the surface component images to preserve the fine scale details is a nontrivial but important problem. Preservation of fine scale details is important in applications like dermatology and face rendering.

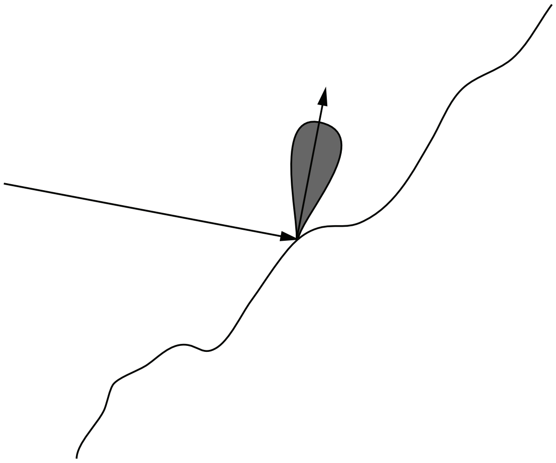


Figure 5.3: The surface reflection is concentrated around the specular lobe.

5.4 Separation of Surface and Subsurface Components

To separate the surface and the subsurface components, the skin surface is illuminated by polarized light and a polarizer filter is attached to the camera. We discuss the concept of polarized/unpolarized light and the use of a polarizer to separate the surface and subsurface components.

5.4.1 Polarized and Unpolarized Light

According to wave theory of light, the light is an electromagnetic wave characterized by electric field vector perpendicular to the direction of propagation of light [21]. In unpolarized light, at each spatial point, the the electric field is randomly oriented. In polarized light, the light field follows a certain pattern in the plane perpendicular to the direction of travel. The electric field can be split into two arbitrary components. In linear polarization the two components are in phase and the ratio of their amplitude is constant. In circular polarization the electric field component perpendicular to the direction of propagation is 90° out of phase with the electric field component parallel to the direction of propagation. The horizontal and the vertical components have the same amplitude. In elliptical polarization, the horizontal and the vertical components either do not have the same amplitude or are not 90° out of phase. Figure 5.5 shows the electric field vectors of a light ray from a viewpoint perpendicular to the direction of travel. Figure 5.5 (a) shows the electric field vectors of unpolarized light. We see that the electric field vectors are oriented in any random direction. Figure 5.5 (b) shows the electric field vectors in linearly polarized light. The electric field vectors are parallel to each other.

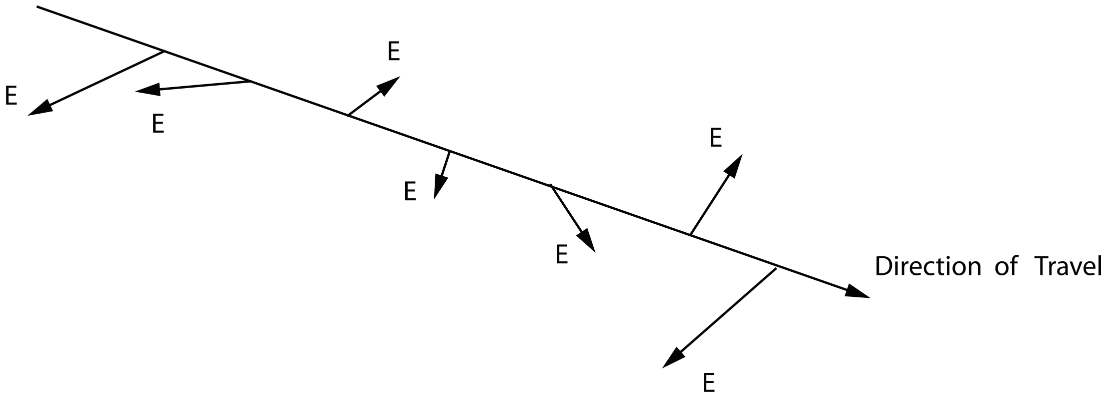


Figure 5.4: Unpolarized light. At each point, the electric field is randomly oriented.

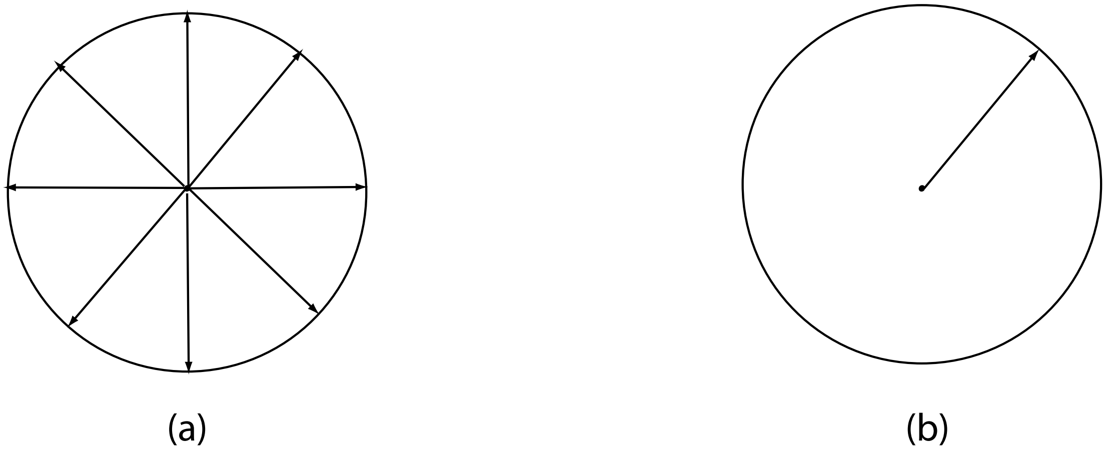


Figure 5.5: Electric field vectors of unpolarized and polarized light when viewed perpendicular to the direction of travel. (a) Shows the electric field vectors in unpolarized light. (b) Shows the electric field vectors in linearly polarized light.

5.4.2 Surface/Subsurface Separation Using the Polarizer

The light incident on the skin surface is polarized by passing it through a polarizer sheet. A Mitsubishi Electric $XD60U$ data projector was used as a light source. Figure 5.6 shows an image of the projector. The light passing through the polarizer sheet is linearly polarized. The light source direction can easily be changed by changing the position of the projector with respect to the skin surface. When the light is incident on the surface of the skin, a part of the incident light penetrates into the skin's surface and a part of the incident light reflects off the surface of the skin. The former is called

Figure 5.6: Mitsubishi Electric $XD60U$ data projector. (a), (b) show the projector from two different view points.

as the subsurface component while the latter is called as the surface component. The subsurface component gets depolarized while the surface component remains polarized. A polarizer filter is attached to the camera. The intensity measured by the camera [24] is given by:

$$I = I_c + I_{sv} \cos 2(\theta - \alpha), \tag{5.10}$$

where I is the intensity measured by the camera, I_c is the subsurface component incident at the camera, θ is the angle of the polarizer filter, α is the phase angle determined by the projection of the surface normal to the plane of the filter and $I_{sv} \cos 2(\theta - \alpha)$ is the surface component intensity sensed by the camera at the polarization angle θ. The intensity measured at any point varies sinusoidally with the polarizer angle. At any polarizer angle, the intensity at any pixel location is the sum of the intensities due to the surface component and the subsurface component. The intensity due to the subsurface component does not change with the polarizer angle. The intensity due to the surface

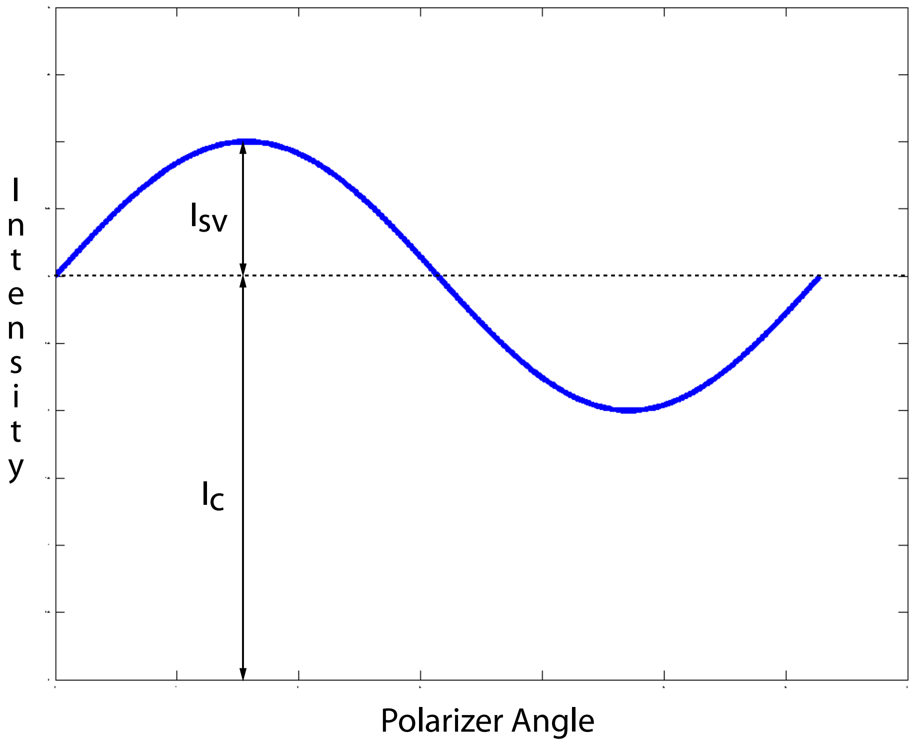


Figure 5.7: Pixel intensity vs polarizer angle.

component varies sinusoidally with the polarizer angle. The plot of the total intensity verses the polarizer angle would be a sinusoid with a constant offset. To separate the surface and the subsurface components images of the scene are acquired for polarizer angles. Using the acquired images the parameters I_c, I_{sv}, θ, α are estimated. Equation 5.10 is a nonlinear equation in the parameters I_c, I_{sv}, θ, α. One approach to estimate the parameters is to use iterative methods. However, the problem can be formulated as a linear estimation problem [24]. By expanding the cosine term in equation 5.10:

$$I_i = I_c + I_{sv} \cos 2\theta_i \cos 2\alpha + I_{sv} \sin 2\theta_i \sin 2\alpha, \tag{5.11}$$

where I_i is the intensity sensed by the camera for polarizer angle θ_i. Equation 5.11 can be written in the vector form:

$$I_i = \mathbf{f}_i^T \mathbf{v}, \tag{5.12}$$

where $\mathbf{v} = [I_c \; I_{sv} \; cos2\alpha \; I_{sv} \sin 2\alpha]^T$ is the vector of unknowns and $\mathbf{f}_i = [1 \;\; \cos 2\theta_i \;\; \sin 2\theta_i]^T$ is determined by the polarizer angle. Each polarizer angle gives one equation in the unknown vector $\mathbf{v}$. For $n \geq 3$ the vector $\mathbf{v}$ can be uniquely determined by using least squares estimation. Once vector $\mathbf{v}$ is uniquely determined the parameters I_c, I_{sv}, θ, α can easily be extracted from it.

5.4.3 Experimental Results

Figure 5.8 shows face images acquired at different positions of the polarizer filter. Note the variation in the appearance with the polarizer angle. Using the face images, acquired at different positions of the polarizer, the surface and the subsurface components were estimated. Figure 5.9 shows the plot, for a particular pixel location, of the red intensity

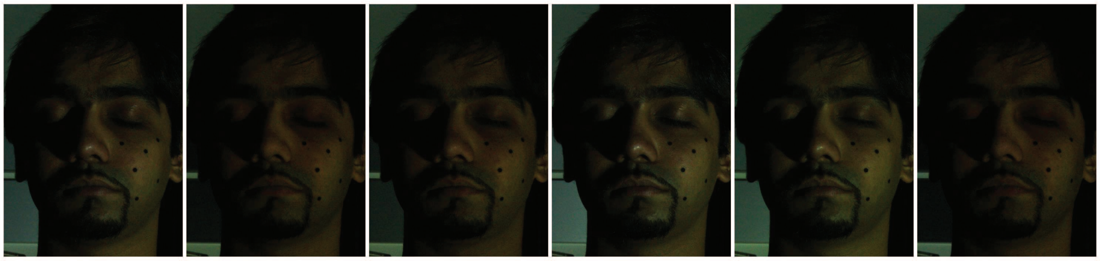


Figure 5.8: Images of the face at different positions of the polarizer.

verses the polarizer angle. Figure 5.10 shows the plot of the surface component and the

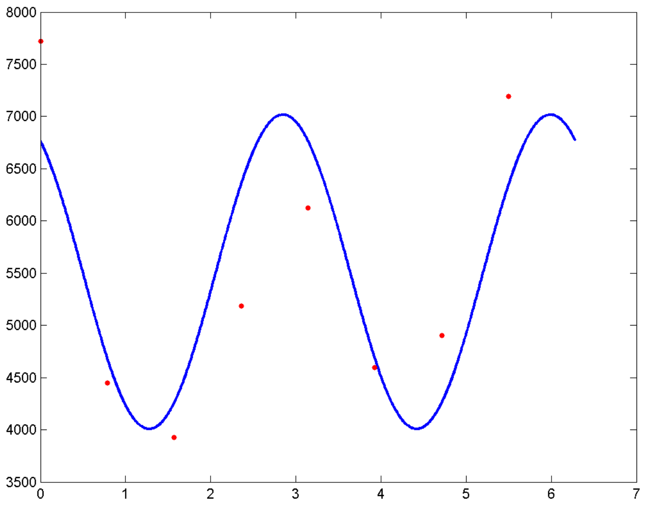


Figure 5.9: Intensity verses the polarizer angle. The red dots indicate the measured intensity values of the red channel at a particular pixel location. The blue line indicates the sinusoidal fit obtained with least squares estimate.

subsurface components of the face images.

Figure 5.11 shows the images of the orange acquired at different positions of the polarizer. The subsurface and the surface components of the orange are shown in Figure 5.12.

5.5 Summary

We discussed the different components of skin which contribute to the appearance of skin. We discussed the use of BRDF and BTF in modeling the appearance of skin. The interaction of light with the skin was modeled as a combination of subsurface and surface reflection. The difference in the properties of the subsurface and the surface components were discussed. The set of subsurface images, under constant pose, formed a linear subspace with dimensionality three. However, the subsurface components did not retain the fine scale details. The fine scale details are retained in the surface components. However, compared to the subsurface components, the surface components are much harder to model. The separation of the surface and the subsurface components

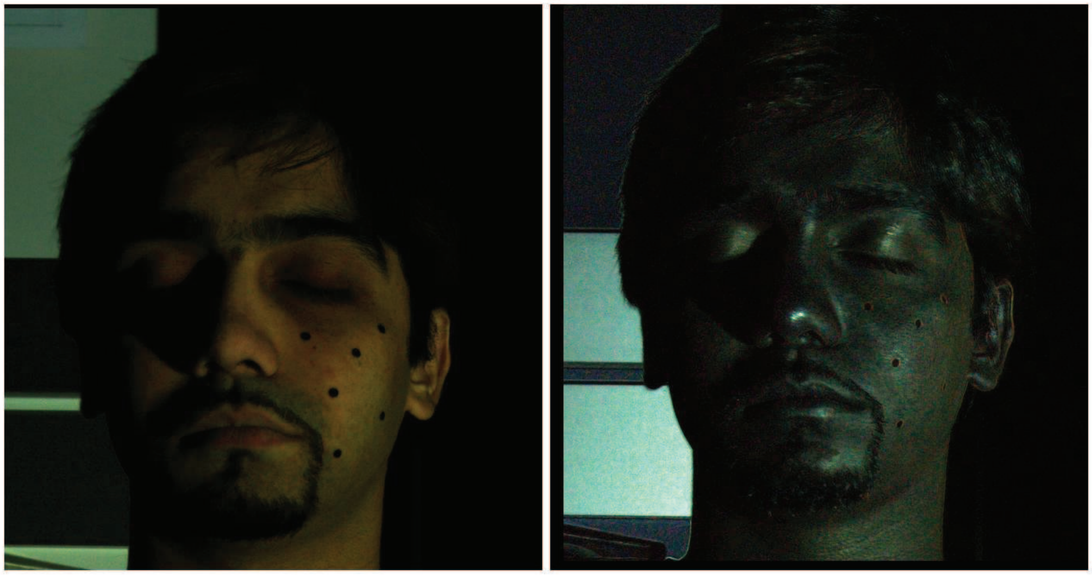


Figure 5.10: (Subsurface/Subsurface components of a face image. (a): Subsurface component of the face image. (b): Surface component of the face image.

using the polarizer was discussed. Experimental results obtained using a human face and an orange were presented at the end of the chapter.

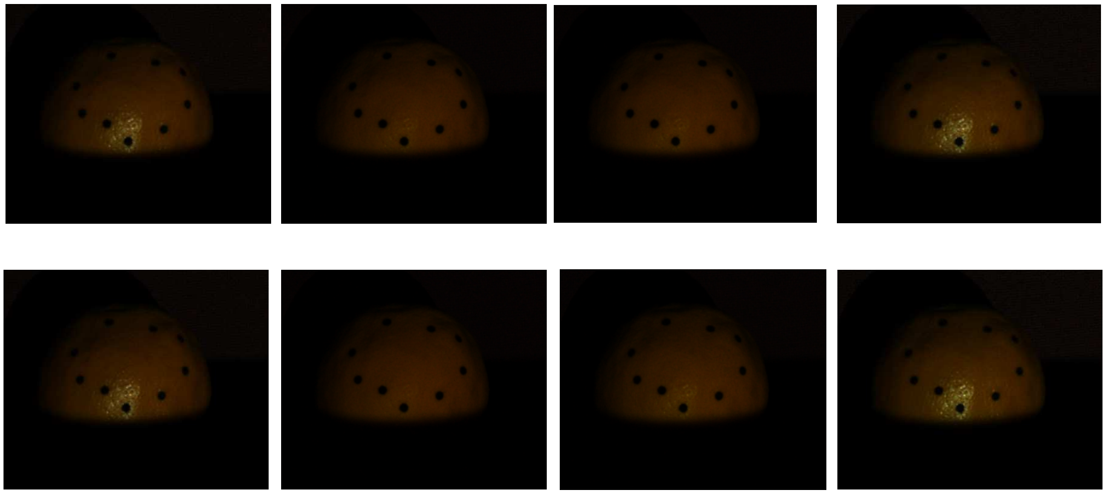


Figure 5.11: Images of the orange at different positions of the polarizer.

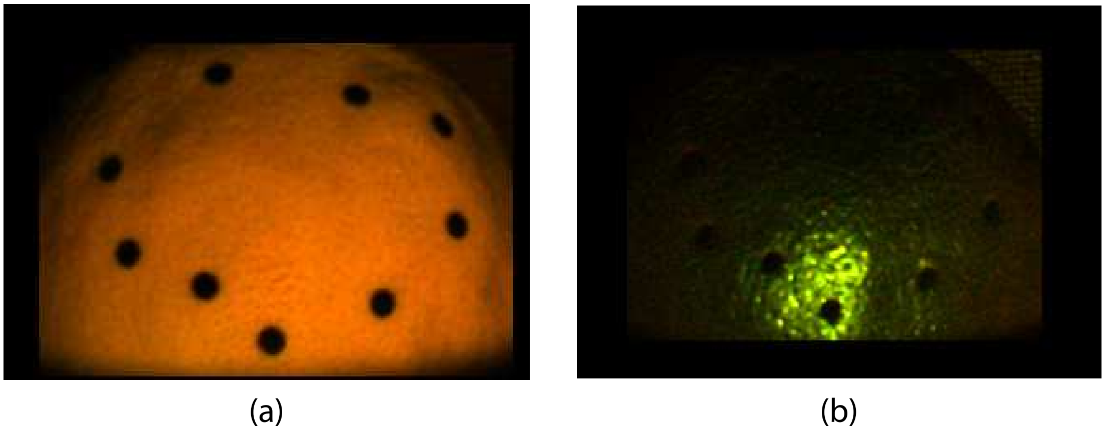


Figure 5.12: Surface/Subsurface components of an orange. (a): Subsurface component of the orange. (b): Surface component of the orange.

Chapter 6

Texture Modeling

6.1 Introduction

Face models have typically focused on appearance at a macroscopic scale which is consistent with human visual observations. However, with the widespread availability of low cost high resolution digital imagery, face details are of increasing interest. Facial details provide an additional fingerprint for recognition and add realism to computer graphics rendering. Our specific emphasis in face modeling is the capture of surface reflectance which encompasses skin roughness, wrinkles, pore distributions and skin glyphs. Many current skin reflectance models consider skin uniform and spatial variations are simply mapped with a flat texture. Such an approach makes the implicit assumption that all skin variations behave as a flat patterns like a tattoo, adding a multiplicative factor to the scattering path, but otherwise leaving the light path unchanged. Modeling pores, wrinkles, moles, scars as flat patterns is overly simplistic. Skin models which do consider the fine–scale geometry of skin include [19, 15]. Here the fine scale geometry is measured and then a reflectance model is applied to the geometry. However, the outer layer of skin is a complex structure. Just as a diffuse reflectance model applied to measured global geometry is not enough to capture subsurface scattering [20, 14], a specular reflectance model applied to measured fine–scale geometry is not sufficient for the surface reflection component. More recent work has addressed this topic of fine scale detail including, the pioneering large scale face measurement and modeling [34] which contain detailed geometry and reflectance measurements and [12] which develops a statistical model of facial detail.

Our contribution to this area is the development of a nonlinear PCA (principal components analysis) based compression scheme that uses polarization at the measurement stage to isolate the surface component texture. As in a typical kernel PCA methods, a higher dimensionality subspace is first constructed in order to create an appropriate subspace for linear PCA. In our method, the higher dimensional subspace is obtained by doubling the dimension of each image through separation of a subsurface reflectance image and an at–surface reflectance image. The at–surface reflection occurs at the boundary layer which often has relief texture or roughness. In particular, for human skin, the boundary layer has an oily layer on a fine–scale height profile as shown in Figure 6.1. The separation of subsurface and surface reflection components is a nonlinear preprocessing to traditional PCA. Consequently, the entire process can be cast in the framework of nonlinear PCA for dimensionality reduction. The higher dimensional subspace is obtained explicitly for visualization and comparison of the representative basis vectors. To remove variation in pose, conformal mapping is applied and all analysis takes place in a projected two dimensional space. The resulting model has implications for human skin modeling where surface texture is useful in recognition, classification and rendering.

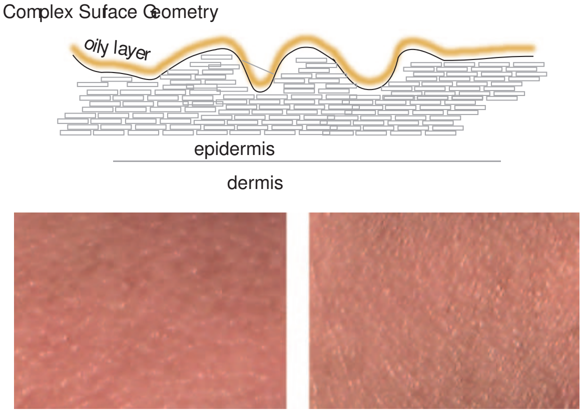


Figure 6.1: Skin structure includes complex geometry at the air–skin interface as well as layers of cells in the stratum corneus portion of the epidermis. The skin image examples shown illustrate how the significant contribution of the first interface, the oily layer, to appearance. Image–based capture of both the subsurface and surface reflection can is a useful modeling approach because the surface geometry can be very complex and difficult to measure.

6.2 PCA

Consider a set of data points $\{\mathbf{z}_i\}$, $i \in \{1, ..., N\}$ in R^n. In PCA, we project the points in a lower dimensional linear subspace of dimensionality $k < n$. The linear subspace is defined by the basis vectors $\{\mathbf{v}_j\}$, $j \in \{1, ..., k\}$. In PCA, we define $\mathbf{v}_1$ to be the direction in which, upon projection of data points, the variance is maximized . We define $\mathbf{v}_2$ to be the direction which is orthogonal to $\mathbf{v}_1$ and gives the second highest variance upon projection of the data points. The remaining vectors $\mathbf{v}_3$,...,$\mathbf{v}_k$ are defined in the similar way. It can be shown [2] that the vectors $\mathbf{v}_j$ are the eigenvectors corresponding to the decreasing eigenvalues of the covariance matrix $\mathbf{S}$ of the data points. The covariance matrix $\mathbf{S}$ is given by,

$$\mathbf{S} = \frac{\sum_{i=1}^{N}(\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^T}{N}. \tag{6.1}$$

Here, $\bar{\mathbf{z}}$ is the sample mean of the data points given by,

$$\bar{\mathbf{z}} = \frac{\sum_{i=1}^{N}\mathbf{z}_i}{N}. \tag{6.2}$$

If we define $\mathbf{V} = [\mathbf{v}_1...\mathbf{v}_k]$, the projection of the data point $\mathbf{z}_i$ in the lower dimensional subspace is given by $\mathbf{V}^T\mathbf{z}_i$.

6.3 Non–Linear PCA via Optical Separation

PCA is powerful representation which preserves major trends in the data, however, small variations can be ignored. When these small variations are important for reconstruction fidelity, traditional PCA can be problematic. A key insight in this approach is that specular reflection from surface texture are the type of small variation that are not well captured in PCA methods. But this problem is solved by optically separating the components and creating a higher dimensional subspace. After the separation, two distinct PCA analyzes are performed. For the subsurface components we expect a low dimensional subspace. For the at–surface reflection component, we have a fundamentally different situation. The surface normals change rapidly and the specular component is strong. Therefore the image as a function of illumination exhibits rapid change, unlike the slow changes typical in lambertian reflectance. Consider what happens as the illumination direction changes in a rough specular surface. Specularities appear and disappear abruptly in different places with the change in illumination direction. Consider the more extreme case of paints with metallic flecks which cause sparkle, the abrupt appearance of specularities in different spatial locations as the illumination changes. The surface reflectance of skin with the multitude of crevices, has a high frequency variation with respect to illumination changes as can be seen in the skin images. Many of the imperfections and realism of skin lie on the surface and are useful in characterizing both appearance and the difference of appearance between individuals. Using polarization–based separation, we are projecting our data into a higher dimensional space in order to perform dimensionality reduction. This is the intuition behind kernel PCA methods in our framework. Our approach to mapping the data to a higher dimensional space gives a substantial benefit over standard PCA. In particular we obtain basis functions that are tuned to either the subsurface or the surface variation in the measurement data.

6.4 Results with Non–Linear PCA

The first tests were done with a foam face object depicted in Figure 6.2. This test object was chosen because the separation of subsurface and surface reflection was surprising similar to human skin as depicted in Figure 6.2. Also, with an inanimate object, registration is not an issue, so it provides a good testbed. For the polarization based analysis, precise alignment at all high–resolution pixels is essential to avoid artifacts. Perfecting this alignment for breathing, moving humans is out of the thesis. Our experiment is to image multiple illumination directions and then create a lower dimensional subspace representation using nonlinear PCA. The nonlinear pre–processing consisted of fitting the sinusoidal variation for each pixel to separate a surface and subsurface component (as shown in Figure 6.2). After projecting to two dimensional texture coordinates, PCA for the diffuse and specular components were performed individually and the results were summed. The second row of Figure 6.3 shows the final result when six illumination directions (coplanar, separated by approximately 20 degrees) are

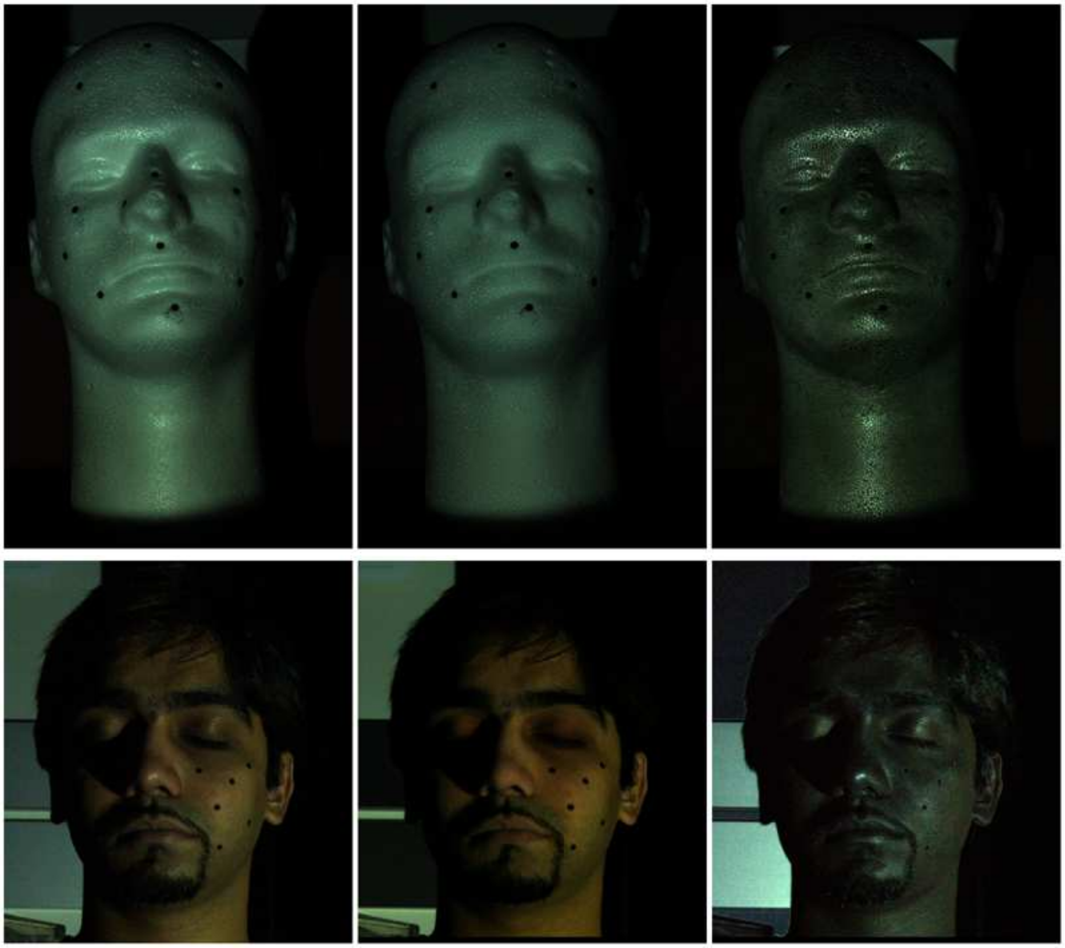


Figure 6.2: The foam head (top row) and real face (bottom row): (left) original image, (middle) diffuse subsurface reflectance, (right) specular surface reflectance. There is a similar pattern of subsurface and surface reflectance so the foam head is a reasonable test model to avoid registering high resolution skin images.

represented with one basis image for the diffuse component and one basis image for the specular component. The third row of Figure 6.3 shows the final result of PCA with the combined diffuse and specular reflection. Column 4 of these results are shown in a closer view in Figure 6.4. The basis images for the separated reflectance are better tuned to the large diffuse variation and the specular variation as shown in Figure 6.5.

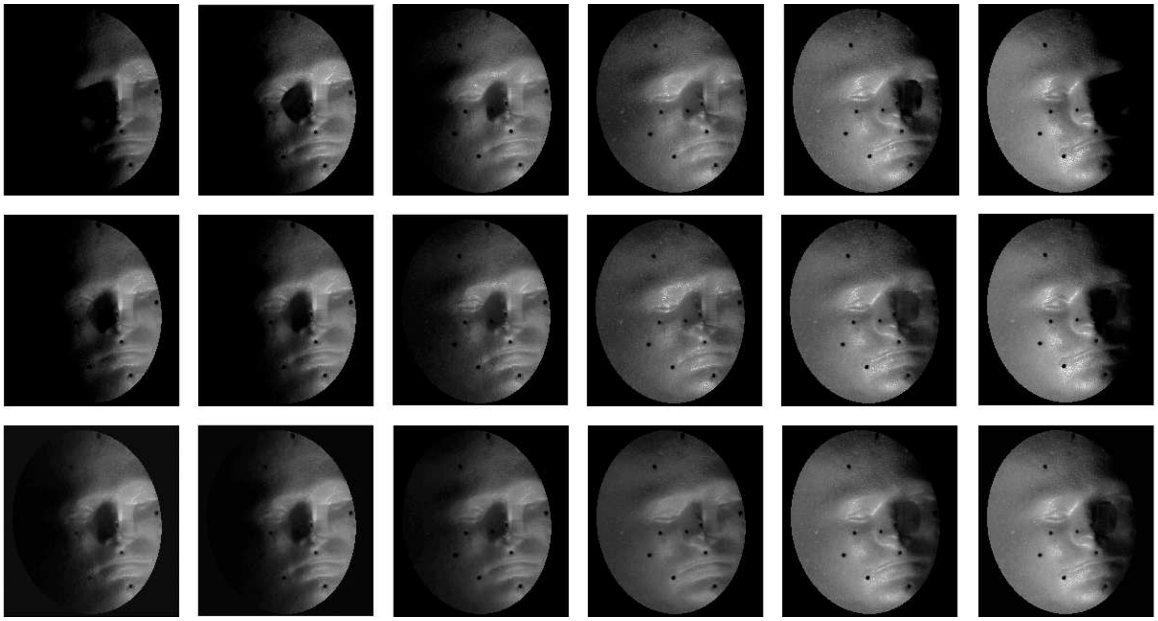


Figure 6.3: Top Row: Original foam head images (projected to texture coordinates) for six illumination directions. Note that all of the images have been projected to a 'texture plane' using least squares conformal mapping so that variations in pose are removed. Middle Row: Reconstructed images using nonlinear PCA. (The final result is obtained by summing the individual components). Bottom Row: Reconstructed images when standard PCA is applied to the original images (without surface/subsurface separation). Notice that the specularities are not well reconstructed in the bottom image set. All reconstructions are done using a single basis image to represent the 6 illumination directions.

6.5 Impact

Nonlinear PCA via optical separation provides a better representation for surfaces that exhibit surface texture (fine–scale height variation) as well as subsurface scattering. In addition to skin, numerous real world objects such as leaves, fruit, coated wood have reflections from the boundary layer at the surface and from subsurface scattering. The structure or texture at that boundary layer is a significant contributor to overall appearance. Our approach provides a computational representation tuned to capture this variation. Representations for surface detail with added accuracy have a direct impact in machine learning approaches to skin modeling. In particular local statistics of eigen images and the corresponding coefficients can be used in classification algorithms.

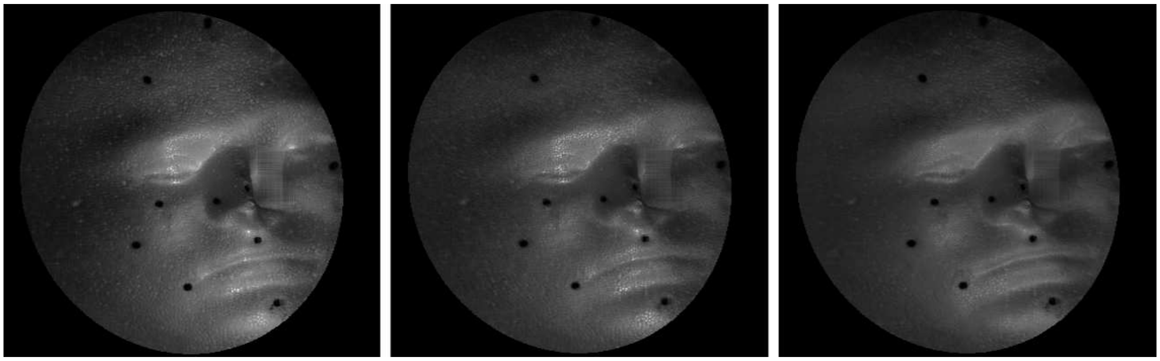


Figure 6.4: The differences in the reconstructions shown in Figure 6.3 are particularly noticeable in the fourth column. Here, these images are shown for comparison. (Left) Original image, (Middle) Reconstruction using nonlinear PCA, (Right) Reconstruction using standard PCA. Notice the specularities are not well reconstructed in the rightmost image.

6.6 Summary

We discussed the modeling of texture by separating the surface and the subsurface components and modeling them separately. For initial experiments, we used a foam head instead of human faces. Like human faces, the foam head had a significant surface and subsurface component. With the foam head, registration of the multiple images acquired was not an issue. The initial results obtained by modeling the surface and the subsurface components separately preserved the specularities in appearance. The specularities were not preserved when the texture was modeled without separating the surface and the subsurface components.

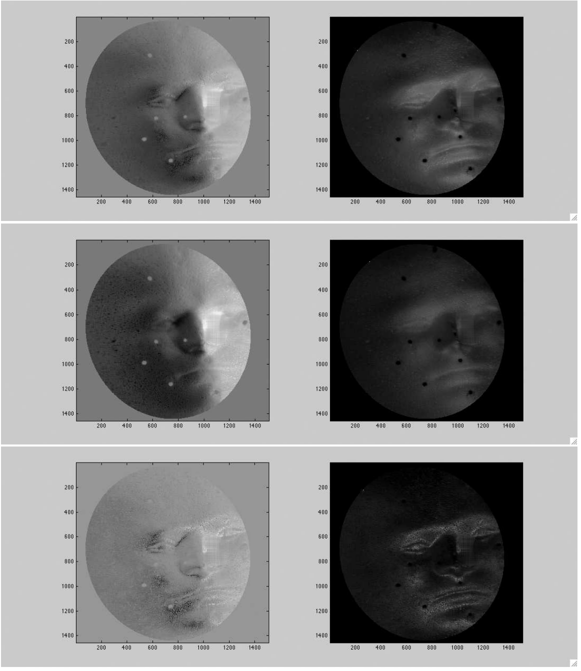


Figure 6.5: (Top) Basis image (left) and mean image (right) using standard linear PCA. Notice that the basis image must simultaneously encode the changes of the specularities as well as the diffuse shading. (Middle) Basis image (left) and mean image (right) of the diffuse subsurface component. Here the basis image is better tuned for the variation due to diffuse shading as can be seen from the overall structure of this basis image. (Bottom) Basis image (left) and mean image (right) of the specular surface component. By comparing this basis image to the top row basis image, it is clear that this basis is better tuned to the varying specularity in the image sequence that occurs when lighting changes.

Chapter 7

Conclusion and Future Work

7.1 Summary of Contributions

We have looked at the use of structured light techniques and the laser scanner to acquire the geometry of the scene. We proposed two new structured light techniques, the HDR–Ratio method and the edge counting method, targeted towards the reliable reconstruction of the low albedo regions of the scene. The geometry acquired using the structured light techniques was not as high resolution as desired. To obtain very high resolution geometry we used a hand held laser scanner. The coordinates obtained from the laser scanner were in the laser coordinate frame. To integrate the geometry for appearance modeling, we solved a calibration problem which gave us the transformation from the laser coordinates to the corresponding image pixels. We accounted for the influence of geometry on the appearance of texture by unwrapping the mesh and mapping the texture on the unwrapped mesh.

To model appearance we modeled the interaction of light with the surface as a sum of surface reflection and subsurface scattering. The surface reflection and the subsurface scattering have significantly different properties. So, we separated out the surface and the subsurface components and modeled them separately using principal component analysis. In the initial results obtained using principal component analysis we saw that separately modeling the surface reflectance and the subsurface components preserved the specularities in the images.

An important step in modeling appearance is to separate out the surface and the subsurface components. State of the art techniques separate the surface and the subsurface components by using a polarizer. In future, we would like to learn the appearance of the surface and subsurface components. Given a training set of unpolarized images and the corresponding surface and subsurface components we would like to learn the relation between the unpolarized images and the surface/subsurface components. Once the relation is learned, given any new unpolarized image we would be able to predict the corresponding the surface and the subsurface components. The learning of the surface/subsurface components is closely related to the image analogies [17] problem. In this chapter, we will discuss the image analogies problem and how it motivates us to learn the surface and the subsurface components.

7.2 Image Analogies

The problem of image analogies is as follows: Suppose we have a pair of images A and A'. Given an image B can we generate an image B' which is related to B in the way A' is related to A? We refer to A as the unfiltered source image and to A' as the filtered source image. B and B' are referred as the unfiltered and the filtered target images. Image analogies is a desirable goal for a number of applications. For example, let A be a blurry image and A' the corresponding high resolution image. Image analogies learn the transformation from the blurry image to the corresponding high resolution image. Once the transformation is learned, given any new blurry image B, we can predict the corresponding high resolution image B'. Another application is the learning of the traditional filters like Gaussian blurring. Given an image A and a Gaussian blurred image A', we learn the transformation from A to A'. Given any new image B we generate the Gaussian blurred version without actually applying the Gaussian filter.

Image analogies use local statistics to learn the transformation from A to A'. We use $A(\mathbf{p})$ and $A'(\mathbf{p})$ to denote the local statistics around pixel $\mathbf{p}$ in the unfiltered source image A and filtered source image A' respectively. We use $B(\mathbf{q})$ and $B'(\mathbf{q})$ to denote the local statistics around pixel $\mathbf{q}$ in the unfiltered target image B and filtered target image B'. Note that we have used $\mathbf{p}$ to denote the pixel locations of both the unfiltered and the filtered source images and $\mathbf{q}$ to denote the pixel locations of both the filtered and unfiltered target images. The choice of which features to use is an open problem. In the paper [17] the authors convert the RGB vector at each pixel location to the corresponding YIQ values and use the luminance values define the local statistics. The authors claim that the human visual system is more sensitive to changes in Y than to changes in I and Q channels.

To find the local statistics $B'(\mathbf{q})$ at pixel location $\mathbf{q}$ of the filtered target image, the entire images A, A' and B are used along with the partially generated image B'. The authors find the local statistics in A which is closest to $B(\mathbf{q})$. To find the closest local statistics a distance metric has to be defined. The definition of the distance metric is still an open problem. The distance metric must account for both statistical closeness as well as perceptual closeness. Perceptual closeness is required to obtain a smooth visually appealing image B' at the end of the procedure. Suppose the closest local statistics is found at pixel location $\mathbf{p_0}$. Then, the feature vector $B'(\mathbf{q})$ is defined to be $A'(\mathbf{p_0})$.

7.3 Learning the Surface/Subsurface Components

In future, we are interested in learning the appearance of surface/subsurface components using unpolarized images. Given a set of unpolarized images, we would like to find the transformation to the corresponding surface and subsurface components. The problem in many ways is similar to the image analogies problem. We are trying to learn the transformation from the unpolarized images to the surface/subsurface components and then predict the transformation for a new image. State of the art techniques use a polarizer to obtain the surface/subsurface separation. The use of a polarizer requires the acquisition of multiple images of the scene. Acquiring multiple images is undesirable when we are imaging humans. Small movements due to breathing gives rise to misalignments between the acquired images. Successfully learning of the surface/subsurface components would enable us to separate out the surface/subsurface components using a single unpolarized image of the scene. In this way, learning of the surface/subsurface components eliminates the need of using a polarizer and acquiring multiple images, resulting in a reduction of acquisition time and elimination of registration problems

References

[1] http://www.autohotkey.com/.

[2] E. Alpaydin. *Introduction to Machine Learning*, chapter 6, pages 108–115. The MIT Press, 2004.

[3] R. H. Bartles, J. C. Beatty, and B. A. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, chapter 4, pages 19–66. Morgan Kaufmann Publishers, INC, first edition, 1987.

[4] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible illumination conditions. *International Journal of Computer Vision*, 28(3):245–260, 1998.

[5] A. M. Bronstein, M. M. Bronstein, E. Gordon, and R. Kimmel. High resolution structured light range scanner with automatic calibration. Technical report, Technion Computer Science Department, 2003.

[6] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 20(5):470–480, 1998.

[7] O. G. Cula. *Bidirectional imaging and modeling for real world texture surfaces.* PhD thesis, Rutgers, The State University of New Jersey, Department of Computer Science, 2005.

[8] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture for real world surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 18, pages 1–34, 1997.

[9] P. E. Debevic and J. Malik. Recovering high dynamic range radiance maps from photographs. In *International Conference on Computer Graphics and Interactive Techniques*, pages 369–378, 1997.

[10] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*, chapter 21, pages 467–490. Prentice Hall, 2002.

[11] A. Glassner. *Digital Image Synthesis*, chapter 11, pages 563–567. Morgan Kaufmann, first edition, 1995.

[12] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser. A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics*, 25(3):1025–1034, 2006.

[13] M. D. Grossberg, S. K. Nayar, and P. N. Belhumeur. Making one object look like another: controlling appearance using projector-camera system. In *IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 452–459, 1995.

[14] P. Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of SIGGRAPH*, volume 27, pages 165–174, 1993.

[15] A. Haro, B. Guenter, and I. Essa. Real-time, photo-realistic, physically based rendering of fine scale human skin structure. *Proceedings 12th Eurographics Workshop on Rendering*, pages 53–62, June 2001.

[16] R. Hartley and A. Zisserman. *Multiview Geometry in Computer Vision*, chapter 10, pages 262–278. Cambridge University Press, second edition, 2004.

[17] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *International Conference on Computer Graphics and Interactive Techniques*, pages 327–340, 2001.

[18] T. Igarashi, K. Nishino, and S. K. Nayar. The appearance of human skin. Technical report, Department of Computer Science, Columbia University, 2005.

[19] H. Jensen. Digital face cloning. *SIGGRAPH 2003 Technical Sketch San Diego*, July 2003.

[20] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proceedings of SIGGRAPH*, pages 511–518, August 2001.

[21] J. D. Kraus and D. A. Fleisch. *Electromagnetics with Applications*, chapter 4, pages 211–214. McGraw-Hill, fifth edition, 1999.

[22] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics*, 21:362–371, 2002.

[23] S. Nayar and R. Bolle. Computing reflectance ratios from an image. *Pattern Recognition*, 26(10):1529–1542, 1993.

[24] S. K. Nayar, X.-S. Fang, and T. Boult. Separation of reflection components using color polarization. *International Journal of Computer Vision*, 21(3):163–186, 1997.

[25] S. K. Nayar and H. Murase. Dimensionality of illumination in appearance matching. In *International Conference on Robotics and Automation*, volume 2, pages 1326–1332, 1996.

[26] F. Nicodemus, J. Richmond, and J. Hsia. Geometrical considerations and nomenclature for reflectance. *National Bureau of Standards*, 1977. Monograph No. 160.

[27] M. Oren and S. K. Nayar. A theory of specular surface geometry. *International Journal of Computer Vision*, 24(2), 1996.

[28] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*, chapter 15, pages 683–688. Cambridge University Press, second edition, 1992.

[29] J. Salvi, J. Pagés, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004.

[30] G. Strang. *Linear Algebra and its Applications*, chapter 3, pages 160–170. Brooks Cole, fourth edition, 2005.

[31] M. Trobina. Error model of a coded light range sensor. Technical report, Communication Technology Laboratory, Image Science Group, ETH, 1995.

[32] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1986.

[33] R. J. Valkenberg and A. M. McIvor. Calibrating a structured light system. *Image and Vision Computing*, 16:99–110, 1998.

[34] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. W. Jensen, and M. Gross. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics*, 25:1013–1024, 2006.

[35] A. Wu. Capture appearance with geometry and image acquisition. Master's thesis, Rutgers, The State University of New Jersey, Department of Electrical and Computer Engineering, 2006.

[36] L. Zhang, B. Curless, and S. M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission*, pages 24–36, 2002.