Measuring and compensating for transport delay in real-time interactive driving simulation

1993

Joseph D. Dumas II
Joe-Dumas@utc.edu

Find similar works at: http://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Electrical and Computer Engineering Commons

STARS Citation

http://stars.library.ucf.edu/rtd/3649

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
SHAPE RECONSTRUCTION FROM SHADING USING LINEAR APPROXIMATION

By
Ping-Sing Tsai

1995

UCF
SHAPE RECONSTRUCTION FROM SHADING USING LINEAR APPROXIMATION

BY
PING-SING TSAI
B.S. Chung-Yuan Christian University 1985

DISSERTATION
Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science in the graduate studies program of the College of Arts and Sciences University of Central Florida Orlando, Florida

Summer Term
1995

Major Professor: Mubarak Shah
Shape from shading (SFS) deals with the recovery of 3D shape from a single monocular image. This problem was formally introduced by Horn in the early 1970s. Since then it has received considerable attention, and several efforts have been made to improve the shape recovery.

In this thesis, we present a fast SFS algorithm, which is a purely local method and is highly parallelizable. In our approach, we first use the discrete approximations for surface gradients, $p$ and $q$, using finite differences, then linearize the reflectance function in depth, $Z(x, y)$, instead of $p$ and $q$. This method is simple and efficient, and yields better results for images with central illumination or low-angle illumination. Furthermore, our method is more general, and can be applied to either Lambertian surfaces or specular surfaces. The algorithm has been tested on several synthetic and real images of both Lambertian and specular surfaces, and good results have been obtained.

However, our method assumes that the input image contains only single object with uniform albedo values, which is commonly assumed in most SFS methods. Our algorithm performs poorly on images with nonuniform albedo values and produces incorrect shape for images containing objects with scale ambiguity, because those images violate the basic assumptions made by our SFS method. Therefore, we extended our method for images with nonuniform albedo values. We first estimate the albedo
values for each pixel, and segment the scene into regions with uniform albedo values. Then we adjust the intensity value for each pixel by dividing the corresponding albedo value before applying our linear shape from shading method. This way our modified method is able to deal with nonuniform albedo values.

When multiple objects differing only in scale are present in a scene, there may be points with the same surface orientation but different depth values. No existing SFS methods can solve this kind of ambiguity directly. We also present a new approach to deal with images containing multiple objects with scale ambiguity. A depth estimate is derived from patches using a minimum downhill approach and re-aligned based on the background information to get the correct depth map. Experimental results are presented for several synthetic and real images.

Finally, this thesis also investigates the problem of the discrete approximation under perspective projection. The straightforward finite difference approximation for surface gradients used under orthographic projection is no longer applicable here, because the image position components are in fact functions of the depth. In this thesis, we provide a direct solution for the discrete approximation under perspective projection. The surface gradient is derived mathematically by relating the depth value of the surface point with the depth value of the corresponding image point. We also demonstrate how we can apply the new discrete approximation to a more complicated and realistic reflectance model for SFS problem.
ACKNOWLEDGEMENTS

The author is thankful to Prof. A. Pentland at MIT, Prof. K. Ikeuchi and Prof. S. Shafer at CMU, Prof. C. C. J. Kuo and Dr. K. M. Lee at USC, Prof. L. B. Wolff at Johns Hopkins University, Prof. Xin Li at UCF, Dr. Y. G. Leclerc at SRI International, and Dr. Q. Zheng at Martin Marietta for providing useful information and images. The author would also like to thank Miss M. Worley for not only reading this thesis but shaping it into what is here before you.

I was fortunate enough to have Prof. Mubarak Shah as my advisor. This work would not have been possible without his guidance and support over the past six years.
# TABLE OF CONTENTS

LIST OF FIGURES ........................................... vii

1 INTRODUCTION ............................................. 1

2 BACKGROUND KNOWLEDGE AND RELATED WORK ...... 6
   2.1 Image Formation ....................................... 6
      2.1.1 Lambertian Model ................................ 6
      2.1.2 Specular Model ................................... 8
      2.1.3 Hybrid Model .................................... 10
      2.2 Related Work ....................................... 11
         2.2.1 Source Estimation .............................. 11
         2.2.2 Albedo Estimation .............................. 17
         2.2.3 Shape From Shading ............................ 21

3 A LINEAR SHAPE FROM SHADING METHOD ............ 26
   3.1 Lambertian Surfaces .................................. 26
   3.2 Specular Surfaces .................................... 30
   3.3 Comparison With Pentland's Method ................ 31
   3.4 Implementation Details .............................. 34
      3.4.1 The Kalman Filter Relevance .................. 34
      3.4.2 Parallel Implementation ......................... 37
   3.5 Experimental Results ............................... 39
### 3.5.1 Results for Lambertian Surfaces

### 3.5.2 Results for Specular Surfaces

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>SFS FOR IMAGES WITH NONUNIFORM ALBEDO VALUES</td>
<td>60</td>
</tr>
<tr>
<td>4.1</td>
<td>Albedo Adjustment</td>
<td>62</td>
</tr>
<tr>
<td>4.2</td>
<td>Segmentation</td>
<td>63</td>
</tr>
<tr>
<td>4.3</td>
<td>Proposed Method</td>
<td>65</td>
</tr>
<tr>
<td>4.4</td>
<td>Experimental Results</td>
<td>65</td>
</tr>
<tr>
<td>5</td>
<td>SFS FOR IMAGES WITH SCALE AMBIGUITY</td>
<td>73</td>
</tr>
<tr>
<td>5.1</td>
<td>Objects with Scale Ambiguity</td>
<td>73</td>
</tr>
<tr>
<td>5.2</td>
<td>Proposed Method</td>
<td>77</td>
</tr>
<tr>
<td>5.3</td>
<td>Experimental Results</td>
<td>79</td>
</tr>
<tr>
<td>6</td>
<td>SFS UNDER PERSPECTIVE PROJECTION</td>
<td>82</td>
</tr>
<tr>
<td>6.1</td>
<td>Perspective Projection and Discrete Approximation</td>
<td>83</td>
</tr>
<tr>
<td>6.2</td>
<td>SFS Using A Sophisticated Reflectance Model</td>
<td>87</td>
</tr>
<tr>
<td>6.3</td>
<td>Experiment</td>
<td>90</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSIONS</td>
<td>93</td>
</tr>
</tbody>
</table>

REFERENCES 96
## LIST OF FIGURES

2.1 Lambertian Reflection Geometry. ........................................ 7  
2.2 Specular Reflection Geometry. ........................................ 9  
3.1 The results for Sphere Image using our method. ................... 35  
3.2 The time-processor curve for the sphere image. ................... 38  
3.3 Synthetic images of Vase generated using different light sources. . 41  
3.4 Results for Synthetic images of Vase. ............................... 42  
3.5 Synthetic images of Mozart generated using different light sources. . 43  
3.6 Results for Synthetic images of Mozart. ............................ 44  
3.7 Results (with smoothing) for Synthetic images of Mozart. ........ 45  
3.8 Synthetic images of Penny generated using different light sources. . 46  
3.9 Results for Synthetic images of Penny. ............................. 47  
3.10 Results (with smoothing) for Synthetic images of Penny. .......... 48  
3.11 The results for Halloween Mask Image. ............................. 49  
3.12 The results for Lenna Image. ....................................... 50  
3.13 The results for Mannequin Image. .................................. 51  
3.14 The results for Yowman Image. ..................................... 52  
3.15 The results for Part Image. ........................................ 53  
3.16 The results for a synthetic specular sphere Image. ............... 57  
3.17 The results for a specular cylinder Image. ....................... 58  
3.18 The results for a specular tomato Image. ......................... 59
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Results for SFS methods.</td>
<td>61</td>
</tr>
<tr>
<td>4.2</td>
<td>Results for a synthetic image of two spheres with different albedo values.</td>
<td>67</td>
</tr>
<tr>
<td>4.3</td>
<td>Results for real image of a foam ball with two different albedo values.</td>
<td>70</td>
</tr>
<tr>
<td>4.4</td>
<td>Results for a real image of a flower with two different albedo values.</td>
<td>71</td>
</tr>
<tr>
<td>4.5</td>
<td>Results for a real image of a flower with three different albedo values.</td>
<td>72</td>
</tr>
<tr>
<td>5.1</td>
<td>Two hemispheres example.</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>Results for a synthetic image of three spheres.</td>
<td>80</td>
</tr>
<tr>
<td>5.3</td>
<td>Results for a synthetic image of Mozarts with different scales.</td>
<td>81</td>
</tr>
<tr>
<td>6.1</td>
<td>Perspective projection model.</td>
<td>83</td>
</tr>
<tr>
<td>6.2</td>
<td>Results for Synthetic images of Sphere (under perspective projection).</td>
<td>91</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Generally speaking, the goal of computer vision is: from one or a sequence of images of a moving or stationary object or a scene, taken by a monocular (one eye) or polynocular (many eyes) moving or stationary observer, to develop methods for understanding the object or the scene and its three-dimensional properties [1]. One way to understand an image (or images) is to reconstruct the physical parameters of the visual world, such as the depth or orientation of surfaces, the boundaries of objects, the direction of light sources and the like. Shape recovery is one of the classic problems in the reconstruction school. The recovered shape can be expressed in several ways: depth $Z$, surface normal $(n_x, n_y, n_z)$, surface gradient $(p, q)$, and surface slant, $\phi$, and tilt, $\theta$. The depth is the relative height of the surface. The surface normal is the orientation of a vector perpendicular to the tangent plane on the object surface, which is usually a unit vector. The surface gradient, $(p, q) = \left( \frac{\partial z}{\partial x}, \frac{\partial z}{\partial y} \right)$, is the rate of change of depth in the $x$ and $y$ directions. The surface slant, $\phi$, and tilt, $\theta$, are related to the surface normal as $(n_x, n_y, n_z) = (l \sin \phi \cos \theta, l \sin \phi \sin \theta, l \cos \phi)$, where $l$ is the magnitude of the surface normal.

In Computer Vision, the techniques to recover shape are called shape-from-X
techniques, which include shape from motion, shape from texture, shape from stereo, shape from shading... etc. The major differences between them are the number of input images. Shape from shading (SFS) extracts the 3-D shape information from a single shading image, and can be viewed as an inversion problem of the image formation process. This problem was formally introduced by Horn [14] in the early 1970s. Since then it has received considerable attention, and several efforts have been made to improve the shape recovery [3], [5], [8], [20], [21], [24], [28], [41], [43]... etc.

Shape from shading techniques can be basically divided into two groups [42]: global approaches and local approaches. Global approaches can be further divided into minimization approaches and propagation approaches. Minimization approaches obtain the solution by minimizing an energy function. Propagation approaches propagate the shape information from a set of surface points (e.g., singular points) to the whole image. Local approaches derive shape only from the intensity information of the surface points in a small neighborhood.

In SFS, the imaging model for expressing the relationship between surface shape and image brightness is specified through a proper reflectance map. Most traditional SFS algorithms use three assumptions: a single object with Lambertian surface and uniform albedo, point light source, and orthographic projection. Under these assumptions, the reflectance equation can be written as

\[ I = R(p, q) = \rho(\mathbf{N} \cdot \mathbf{L}), \]  

(1.1)
where $I$ is the image brightness, $R$ is the reflectance function with $p$ and $q$ being the surface gradient, $\rho$ is the composite albedo, $\vec{N}$ is the surface normal, and $\vec{L}$ is the light source direction. These assumptions are simple but restrictive.

In this research, we introduce a linear and highly parallelizable technique to solve the SFS problem. The basic idea of our approach is to employ a linear approximation of the reflectance function, as used by others. However, the major contribution is that we first use the discrete approximations for surface gradient, $p$ and $q$, using finite differences, then linearize the reflectance function in term of depth, $Z(x, y)$, instead of $p$ and $q$. This method is simple and efficient, and yields better results for images with central illumination or low-angle illumination. Furthermore, our method is more general, and can be applied to either Lambertian surface or specular surface.

Most SFS methods, including our method, are developed under several basic assumptions about surface properties and imaging geometry to simplify the problem. When input images do not match their assumptions, those SFS methods will break down. We extend our SFS method for images with nonuniform albedo values by cancelling the effect of albedo variation before applying our linear SFS method. In order to cancel the effect of albedo variation, we first estimate the albedo values for each pixel, and segment the scene into regions with uniform albedo values. However, the estimated albedo values are not accurate at the points near edges, we apply a modified median filter to improve the result. Then we adjust the intensity value for each pixel by dividing the corresponding albedo value before applying our linear
shape from shading method.

When multiple objects differing only in scale are present in a scene, there will be points having the same surface orientation but different depth values. No existing SFS method can solve this kind of ambiguity directly. We also present a new approach to deal with images containing multiple objects with scale ambiguity. A depth estimate is derived from patches using a minimum downhill approach and re-aligned based on the background information to get the correct depth map.

The problem of the discrete approximation for surface gradients under the perspective projection is also discussed, and a direct solution for the discrete approximation under the perspective projection is provided in this thesis. The surface gradient is derived mathematically by relating the depth value of the surface point with the depth value of the corresponding image point. We also demonstrate how we can apply the new discrete approximation to a more complicated and realistic reflectance model for SFS problem.

The organization of this dissertation is as follows: In chapter 2, we introduce background knowledge related to image formation and reflectance models, and present a survey of related work in source and albedo estimation, and shape from shading techniques. Chapter 3 contains our linear SFS method for both Lambertian and specular surfaces. We also present a parallel implementation of our method. In chapter 4, we extend our method for images with nonuniform albedo values. Chapter 5 presents a new approach to deal with images containing multiple objects with
scale ambiguity. Chapter 6 discusses the issue of SFS under perspective projection, and provides a direct solution for the discrete approximation under the perspective projection. Finally, conclusions and future research directions are presented in chapter 7.
Chapter 2

BACKGROUND KNOWLEDGE AND RELATED WORK

This chapter provides background knowledge of the relation between surface shape and image brightness, i.e., reflectance models which are directly determined by surface properties. An overview of related work in source and albedo estimation, and shape from shading is then presented.

2.1 Image Formation

Depending on their physical properties, surfaces can be classified as pure Lambertian, pure specular, or hybrid. In this section, I will briefly describe all three reflectance models and discuss their properties related to shape from shading.

2.1.1 Lambertian Model

Lambertian surfaces are surfaces having only diffuse reflectance, i.e., surfaces which reflect light in all directions. The brightness of a Lambertian surface is proportional to the energy of the incident light. The amount of light energy falling on a surface element is proportional to the area of the surface element as seen from the light source position (the foreshortened area). The foreshortened area is a cosine
function of the angle between the surface orientation and the light source direction. Therefore, the Lambertian surface can be modelled as the product of the strength of the illumination $\lambda$, the reflectivity coefficient of the surface $\rho$, and the foreshortened area $\cos \theta_i$ as follows:

$$I_L = R = \lambda \rho \cos \theta_i,$$

where $R$ is the reflectance map and $\theta_i$ is the angle between the surface normal $\vec{N} = (n_x, n_y, n_z)$ and the source direction $\vec{S} = (s_x, s_y, s_z)$ (See Figure 2.1). The product of illumination and reflectivity is called the composite albedo or loosely called albedo.

If we let the surface normal and the light source direction both be unit vectors, we can rewrite the above formula as:

$$I_L = \lambda \rho \vec{N} \cdot \vec{S},$$

(2.2)
where '·' represents dot product.

2.1.2 Specular Model

Specularity occurs only when the incident angle of the light source is equal to the reflected angle. It is formed by two components: the specular spike and the specular lobe. The specular spike is zero in all directions except for a very narrow range around the direction of specular reflection. The specular lobe spreads around the direction of specular reflection.

The simplest model for specular reflection is described by the following delta function:

$$I_S = B\delta(\theta_s - 2\theta_r),$$  \hspace{1cm} (2.3)

where $I_S$ is the specular brightness, $B$ is the strength of the specular component, $\theta_s$ is the angle between the light source direction and the viewing direction, and $\theta_r$ is the angle between the surface normal and the viewing direction. This model assumes that the highlight caused by specular reflection is only a single point, but in real life this assumption is not true. Another model was developed by Phong [31]. It represents the specular component of reflection as powers of the cosine of the angle between the perfect specular direction and the viewing direction. This model is capable of predicting specularities which extend beyond a single point; however, the parameters have no physical meaning. A more refined model, the Torrance-Sparrow model [36], assumes that a surface is composed of small, randomly oriented, mirror-
like facets. It describes the specular brightness as the product of four components: energy of incident light, Fresnel coefficient, facet orientation distribution function, and geometrical attenuation factor adjusted for foreshortening. On the basis of the Torrance-Sparrow model, Healey and Binford [12] derived a simplified model by using the Gaussian distribution as the facet orientation function, and considering the other components as constants. It can be described as:

$$I_s = K e^{-\left(\frac{\alpha}{m}\right)^2},$$  \hspace{1cm} (2.4)$$

where $K$ is a constant, $\alpha$ is the angle between the surface normal $\vec{N}$ and the bisector $H$ of the viewing direction and source direction, and $m$ indicates the surface roughness (Figure 2.2).
2.1.3 Hybrid Model

Most surfaces in the real world are neither purely Lambertian, nor purely specular; instead, they are hybrid surfaces with properties of both. That is, they are hybrid surfaces. One straightforward equation for a hybrid surface is:

\[ I = (1 - \omega)I_L + \omega I_S, \]  

where \( I \) is the total brightness for the hybrid surface, \( I_S, I_L \) are the specular brightness and Lambertian brightness respectively, and \( \omega \) is the weight of the specular component.

Nayar, Ikeuchi and Kanade [25] proposed a reflectance model which consists of three components: diffuse lobe, specular lobe, and specular spike. The Lambertian model was used to represent the diffuse lobe, the specular component of the Torrance-Sparrow model was used to model the specular lobe, and the spike component of the Beckmann-Spizzichino model was used to describe the specular spike. The resulting hybrid model is given as:

\[ I = K_{dl} \cos \theta_i + K_{sl} e^{-\frac{\beta^2}{2\sigma^2}} + K_{ss} \delta(\theta_i - \theta_r) \delta(\phi_r). \]  

where \( K_{dl}, K_{sl}, K_{ss} \) are the strengths of the three components, \( \beta \) is the angle between the surface normal of a micro-facet on a patch and the mean normal of this surface patch, and \( \sigma \) is its standard derivation. If we consider the surface normal in
the $Z$ direction, then, $(\theta_i, \phi_i)$ is the direction of incidence light in terms of the slant and tilt in 3-D, $(\theta_r, \phi_r)$ is the direction of reflected light.

2.2 Related Work

2.2.1 Source Estimation

Most SFS algorithms require known light source direction. Since the light source is usually assumed to be at infinity, the light source orientation is constant for all of the surface points in the image, and one image can provide enough information to estimate the source. There are two ways to describe a light source direction: One uses a 3-D vector, the other uses the two angles – slant and tilt. If the image plane is parallel to the X-Y plane, slant is the angle the illuminant vector makes with the $Z$-axis, and tilt is the angle the image plane component of the illuminant vector makes with the X-axis.

Several techniques to estimate source orientation have been developed. The first one, by Pentland [29], estimates the light source direction from the distribution of image derivatives. By assuming an umbilical surface and isotropic surface normal, a maximum-likelihood analysis was performed to estimate the slant and tilt angles of the light source. The basic idea underlying this approach is simple, since surface orientation can be considered as a random variable over the whole image for most scenes, so both surface normal $\hat{N}$ and the change of the surface normal $d\hat{N}$ are isotropically distributed. This means that if we consider any direction $(dx_I, dy_I)$ on
the image plane, the $z$ component of the expected value of $d\vec{N}$, $dn_z$, is zero. Therefore, the derivative of the brightness equation,

$$I = B \vec{N} \cdot \vec{S},$$

(here, $B$ is a constant including the albedo term), gives:

$$E[dI] = dI = B(s_x \, dn_x + s_y \, dn_y),$$

where $E$ indicates the expected value.

For a sphere, $Z(x, y) = \sqrt{r^2 - x^2 - y^2}$, $Z_x = -\frac{x}{Z}$, $Z_y = -\frac{y}{Z}$, hence $(n_x, n_y, n_Z) = -\frac{1}{r}(x, y, Z)$. Consequently, it can be shown that the $x$ and $y$ components of the derivative of normal in any direction $\theta$ are given by $dn_x = -\frac{1}{r} \cos \theta$, and $dn_y = -\frac{1}{r} \sin \theta$. Let $(\cos \theta, \sin \theta) = (dx_1, dy_1)$, then $\bar{k} \, dx_1 = d\vec{n}_x$, and $\bar{k} \, dy_1 = d\vec{n}_y$, where $\bar{k} = \frac{1}{r}$ is the mean projected surface curvature, which is the same in all directions using the locally spherical assumption. Repeating the above process in $m'$ different directions $(dx_i, dy_i) (i = 1, ..., m')$, the regression model can be described as:

$$
\begin{pmatrix}
  d\bar{I}_1 \\
  d\bar{I}_2 \\
  \vdots \\
  d\bar{I}_{m'}
\end{pmatrix}
= \begin{pmatrix}
  dx_1 & dy_1 \\
  dx_2 & dy_2 \\
  \vdots & \vdots \\
  dx_{m'} & dy_{m'}
\end{pmatrix}
\begin{pmatrix}
  \bar{s}_x \\
  \bar{s}_y
\end{pmatrix},
$$
where $dI_i$ is the average of the intensity change along the image direction $(dx_i, dy_i)$, $\hat{s}_x = Bks_x$ and $\hat{s}_y = Bks_y$. A typical choice for the $(dx_i, dy_i)$ are the eight directions in the image plane: two in the horizontal direction, two in the vertical direction, and four along the diagonals.

Solving the above system by least squares, we get:

$$
\begin{pmatrix}
\hat{s}_x \\
\hat{s}_y
\end{pmatrix} = (\beta^T \beta)^{-1} \beta^T
\begin{pmatrix}
dl_1 \\
dl_2 \\
\vdots \\
dl_{m'}
\end{pmatrix},
$$

(2.7)

where $\beta$ is the matrix of directions $(dx_i, dy_i)$.

The tilt, $\tau_S$, of the light source direction is given by:

$$
\tau_S = \arctan\left(\frac{\hat{s}_y}{\hat{s}_x}\right),
$$

(2.8)

and the slant, $\sigma_S$, of the light source direction is:

$$
\sigma_S = \arccos \sqrt{1 - \hat{s}_x^2 - \hat{s}_y^2}.
$$

Taking the expected value of the square of intensity derivative $E[dI^2]$, and cancelling out the common terms between $E[dI]^2$ and $E[dI']^2$ by subtracting one from the other, we have the relation $E[dI^2] - E[dl] = B^2 k^2$. Since $\hat{s}_x = Bks_x$, $\hat{s}_y = Bks_y$,
by introducing \( k = B \hat{k} = \sqrt{E[dI^2] - E[dI]^2} \), the equation for the slant of the light source can be simplified to:

\[
\sigma_S = \arccos \sqrt{1 - \frac{s_x^2 + s_y^2}{k^2}}.
\] (2.9)

Instead of taking intensity derivatives along a number of directions, Lee and Rosenfeld [24] considered only the derivatives along the \( x \) and \( y \) directions. They approximated the surface geometry by a spherical patch in a local region, so their method was also based on an isotropic distribution of the surface orientation. Since the image of a sphere is symmetric about the projection of the light source vector in the image plane, the average direction of the intensity gradient must be parallel to this projection. This gives \( \frac{E(I_y)}{E(I_x)} = \frac{s_y}{s_x} \), so we have,

\[
\tau_S = \arctan \left( \frac{E(I_y)}{E(I_x)} \right),
\] (2.10)

where the expectations are taken over the given image region.

Considering the sampling distribution for the slant, and expected values of intensity and intensity squared, the following equation can be used to solve for the slant, \( \sigma_S \):

\[
\frac{E(I)}{\sqrt{E(I^2)}} = \frac{8(\pi - \sigma_S) \cos \sigma_S + \sin \sigma_S}{3\pi(1 + \cos \sigma_S)^\frac{3}{2}}.
\]

The expectation here is taken over the whole image.
Zheng and Chellappa [43] modified Lee and Rosenfeld's method by considering not only the area of the illuminated portion in the integral, but also the area of the portion in shadow. Although the shadow does not contribute to the total intensity, it does contribute to the area computation in order to correctly calculate the mean intensity value over the whole image. After the modification, the computation for slant, \( \sigma_S \), became:

\[
\frac{E(I)}{\sqrt{E(I^2)}} = \frac{4\sqrt{2}(\pi - \sigma_S) \cos \sigma_S + \sin \sigma_S}{3\pi (1 + \cos \sigma_S)}.
\]

Under the assumption that the orientations of the surfaces are uniformly distributed in 3-D space, they also proposed two methods to estimate the tilt of the light source. One was the local voting method, which assumes that each surface point and its neighbors can be locally approximated by a spherical patch. If we consider small increments in the various image directions, and intensities along these directions, the tilt of the light source is:

\[
\tau_S = \arctan\left(\frac{E\left(\frac{\tilde{s}_x}{\sqrt{\tilde{s}_x^2 + \tilde{s}_y^2}}\right)}{E\left(\frac{\tilde{s}_y}{\sqrt{\tilde{s}_x^2 + \tilde{s}_y^2}}\right)}\right),
\]

where \((\tilde{s}_x, \tilde{s}_y)\) is the same as given by Pentland's method.

The other was the contour-based method, which uses shading information along image contours. Under the assumption that the slant of the surface normals along the boundary are constant, the tilt angle of a boundary pixel, \( \alpha \), is just the tilt angle
of the boundary contour in the image plane, and the summations $\sum \cos \alpha$ and $\sum \sin \alpha$ over the closed boundary are zero. This yields:

$$\gamma_S = \arctan \left( \frac{x_2}{x_1} \right),$$

where

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (\beta^T \beta)^{-1} \beta^T \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_{m'} \end{pmatrix},$$

and $\beta$ is the same as in equation 2.7.

The slant of the light source $\sigma_S$ is estimated using the following equation:

$$\frac{E(I)}{\sqrt{E(I^2)}} = 0.5577 + 0.6240 \cos \sigma_S + 0.1881 \cos^2 \sigma_S - 0.6514 \cos^3 \sigma_S - 0.5350 \cos^4 \sigma_S + 0.9282 \cos^5 \sigma_S + 0.3476 \cos^6 \sigma_S - 0.4984 \cos^7 \sigma_S. \tag{2.11}$$

where $E(I)$ and $E(I^2)$ are the ensemble averages of the image intensities, and the square of the image intensities. Since the seventh-order polynomials in $\cos \sigma_S$ is a monotonically decreasing function of $\sigma_S$, $\sigma_S$ can be uniquely determined.

In [43], Zheng and Chellappa tested the above methods on a set of three different images. The results showed that for the estimation of tilt, all algorithms work almost perfectly for a sphere without background. However, background and noise
will degrade the performance of both Lee and Rosenfeld's and Pentland's methods. Consequently, Zheng and Chellappa's method is more robust with background and noise in most of the cases. For the estimation of slant, Pentland's method is very sensitive to noise, but Lee and Rosenfeld's and Zheng and Chellappa's methods are robust with respect to Gaussian noise. If a uniform background is included, results for all three methods are degraded.

2.2.2 Albedo Estimation

Two statistical approaches for estimating albedo were reported by Lee & Rosenfeld [24] and Zheng & Chellappa [43]. The major difference between these two methods lies in their assumptions about the distribution of surface normals. Lee and Rosenfeld [24] assumed the surface patches to be locally spherical and used a Gaussian sphere to derive the probability density function of the surface normal tilt (τ) and slant (σ), \( f_{τσ} \) as

\[
 f_{τσ} = \frac{1}{2π} sin2σ. \quad (2.12)
\]

By considering the sampling distribution for the slant of the surface normal, and expected values of intensity and intensity squared, the following equation was used to solve for the slant of the light source, \( σ_S \):

\[
 \frac{E(I)}{E(I^2)} = 8(\pi - σ_S)\cos σ_S + sin σ_S \cdot \frac{1}{3π(1 + cos σ_S)^{\frac{3}{2}}}. \quad (2.13)
\]
After $\sigma_s$ is estimated, the albedo of the surface, $\rho$, can be determined by

$$\rho = \frac{1}{a^2 + b}(aE(I) + (bE(I^2))^{\frac{1}{2}})$$

(2.14)

where

$$a = \frac{4((\pi - \sigma_s)\cos \sigma_s + \sin \sigma_s)}{3\pi(1 + \cos \sigma_s)}, \text{ and}$$

$$b = \frac{1 + \cos \sigma_s}{4}.$$  

Zheng and Chellappa [43] assumed the surface patches to be locally flat, and the tilt, $\tau$, and slant, $\sigma$, are independent to each other. They used

$$f_\tau = \frac{1}{2\pi}$$

as the distribution of $\tau$ because the range of $\tau$ is between 0 and $2\pi$, and there is no preference for the tilt angle. They assumed that $\sigma$ is uniformly distributed in 3-D space, and the distribution of $\sigma$ in the image plane is

$$f_\sigma = \cos \sigma.$$  

Therefore, the statistical model for the distribution of surface normals is

$$f_{\tau\sigma} = f_\tau \cdot f_\sigma = \frac{1}{2\pi} \cos \sigma.$$  

(2.15)
The first two moments of image intensity,

\[ I(\tau, \sigma) = \max\{\rho(\cos(\tau - \tau_s) \sin \sigma \sin \sigma_s + \cos \sigma \cos \sigma_s), 0\}, \]

were evaluated using (2.15), and the following equation was used to solve for the slant of the light source, \( \sigma_s \):

\[
\frac{E(I)}{\sqrt{E(I^2)}} = 0.5577 + 0.6240 \cos \sigma_s + 0.1881 \cos^2 \sigma_s - 0.6514 \cos^3 \sigma_s - 0.5350 \cos^4 \sigma_s + 0.9282 \cos^5 \sigma_s + 0.3476 \cos^6 \sigma_s - 0.4984 \cos^7 \sigma_s. \tag{2.16}
\]

After \( \sigma_s \) was estimated, the albedo of the surface, \( \rho \), was computed by

\[
\rho = \frac{1}{f_1(\sigma_s) + f_2(\sigma_s)} \cdot (E(I) \cdot f_1(\sigma_s) + \sqrt{E(I^2)} \cdot f_2(\sigma_s)), \tag{2.17}
\]

where

\[
f_1(\sigma_s) = 0.1615 + 0.3959 \cos \sigma_s + 0.3757 \cos^2 \sigma_s - 0.0392 \cos^3 \sigma_s - 0.3977 \cos^4 \sigma_s + 0.1174 \cos^5 \sigma_s + 0.1803 \cos^6 \sigma_s - 0.0984 \cos^7 \sigma_s,
\]

\[
f_2(\sigma_s) = 0.0834 + 0.2169 \cos \sigma_s + 0.2487 \cos^2 \sigma_s + 0.1836 \cos^3 \sigma_s + 0.0048 \cos^4 \sigma_s - 0.1086 \cos^5 \sigma_s - 0.0043 \cos^6 \sigma_s + 0.0424 \cos^7 \sigma_s.
\]

Lee and Rosenfeld [23] presented another albedo estimation method for the scene.
segmentation. They computed the composite albedo value, $\rho$, for each pixel using only the local intensity information. Let $I_P$ denote the image intensity at point $P$, $Q$ be a point near $P$ in the gradient direction at $P$ and $R$ be a point on the opposite side of $P$. Let

\begin{align}
I_P &= \rho(N_P \cdot l) = \rho \cos \theta, \quad (2.18) \\
I_Q &= \rho(N_Q \cdot l) = \rho \cos(\theta + \delta \theta_1), \quad (2.19) \\
I_R &= \rho(N_R \cdot l) = \rho \cos(\theta - \delta \theta_2), \quad (2.20)
\end{align}

where $N_P$, $N_Q$, and $N_R$ are the surface normals respectively at $P$, $Q$, and $R$; $\theta$ is the angle between $N_P$ and the light source direction, $l$; and $\delta \theta_1$, $\delta \theta_2$ are positive and small. They approximated $\delta \theta_1 \approx \delta \theta_2 = \delta \theta$. Therefore, this yields a system (equations 2.18-2.20) consisting of 3 equations with 3 unknowns $\rho$, $\theta$, and $\delta \theta$. After some algebraic manipulation, the albedo value for $P$ is given by

$$
\rho^2 = \frac{I_P^2 - I_R I_Q}{I_P^2 - ((I_R + I_Q)/2)^2} I_P^2. \quad (2.21)
$$

(Note that if $P$, $Q$, and $R$ lie on a planar surface, this method breaks down because the denominator becomes zero.)
2.2.3 Shape From Shading

The shape from shading problem is one of the classic problems in computer vision. Since the first SFS technique was developed by Horn [14] in the early 1970s, different approaches have been emerging in the past two decades.

Minimization Approaches

Several SFS methods are based on the variational formulation, in which the surface normals (or surface gradient and depth) are determined by minimizing an energy function over the entire image. Ikeuchi and Horn [17] introduced two constraints: the brightness constraint and the smoothness constraint. The brightness constraint requires that the reconstructed shape shall produce the same brightness as the input image at each surface point, while the smoothness constraint forces the gradient of the surface to change smoothly. The shape was computed by minimizing an energy function which consists of the above two constraints. Also using these two constraints, Brooks and Horn [5] minimized the same energy function, in terms of the surface normal. Frankot and Chellappa [11] enforced the integrability in Brooks and Horn’s algorithm in order to recover integrable surfaces (surfaces for which \( z_{xy} = z_{yx} \)). Surface slope estimates from the iterative scheme were expressed in terms of a linear combination of a finite set of orthogonal Fourier basis functions. The enforcement of integrability was done by projecting the nonintegrable surface slope estimates onto the nearest (in terms of distance) integrable surface slopes. This projection was ful-
filled by finding the closest set of coefficients which satisfy integrability in the linear combination. Their results showed improvements in both accuracy and efficiency. Later, Horn also [15] replaced the smoothness constraint in his approach with an integrability constraint. The major problem with Horn's method is its slow convergence. Szeliski [35] sped it up using a hierarchical basis pre-conditioned conjugate gradient descent algorithm. Based on the geometrical interpretation of Brooks and Horn's algorithm, Vega and Yang [39] applied heuristics to the variational approach so that the stability of Brooks and Horn's algorithm was improved.

In place of the smoothness constraint, Zheng and Chellappa [43] introduced an intensity gradient constraint, which requires that the intensity gradient of the reconstructed image be close to the intensity gradient of the input image. Their energy function contains the brightness constraint, the intensity gradient constraint and the integrability constraint. The Euler equations were simplified by taking the Taylor series of the reflectance map and representing the depth, gradient and their derivatives in discrete form. Then an iterative scheme, which updates depth and gradients simultaneously, was derived. The algorithm was implemented using a hierarchical structure (pyramid) in order to speed up the computation.

Lee and Kuo's approach [21] involves only the brightness constraint and the smoothness constraint. In their approach, surfaces were approximated by the union of triangular surface patches. The vertices of the triangles were called nodal points, and only nodal depths were recovered. Depths at the pixels which are not nodal
points were obtained through interpolation. For each triangular patch, the intensity of the triangle was taken as the average intensity of all pixels in the triangle, and the surface gradient of the triangle was approximated by the cross product of any two adjacent edges of the triangle. This established a relationship between the triangle's intensity and the depth at its three nodal points. Linearizing the reflectance map in terms of the surface gradient \((p, q)\), a linear relationship between the intensity and depth at the nodal points was derived. The surface depths at the nodal points were computed using optimization. The optimization problem was reduced to the solution of a sparse linear system, and a multigrid computational algorithm was applied to solve the depth.

**Propagation Approaches**

Oliensis and Dupuis [10, 26] introduced a different approach, which takes full advantage of the singular point constraints. Their method is based on a connection with a calculus of variations and optimal control problem. Following the main idea of Oliensis and Dupuis, Bichsel and Pentland [3] developed an efficient minimum downhill approach which recovers depth and guarantees a continuous surface. Given initial values at the singular points (brightest points), the algorithm looks in eight discrete directions in the image and propagates the depth information away from the light source to ensure the proper termination of the process. Since slopes at the surface points in low brightness regions are close to zero for most directions (except the directions which form a very narrow angle with the illumination direction), the image was
initially rotated to align the light source direction with one of the eight directions. The inverse rotation was performed on the resulting depth map in order to get the original orientation back.

**Local Approaches**

There are several simple and efficient SFS approaches which assume the local surface type to derive shape. Pentland [30] recovered shape information from the intensity, and its first and second derivatives. He used the assumption that the surface is locally spherical at each point. Under the same spherical assumption, Lee and Rosenfeld [24] computed the slant and tilt of the surface in the light source coordinate system through the first derivative of the intensity. This makes it less sensitive to noise. However, the local spherical assumption of the surface limits its application.

**Linear Approaches**

The reflectance function is non-linear in nature. Linear approaches simplify the non-linear problem into a linear one through the linearization of the reflectance function. The idea is based on the assumption that the lower order components in the reflectance function are dominating. Pentland [28] linearized the reflectance function in terms of the surface gradient, and applied a Fourier transform to the linear function to get a closed form solution for the depth at each point. However, when the light source direction and the viewing direction are close to each other (images with central illumination), the quadratic terms of the surface gradient ($p^2$ and $q^2$) will become
dominating in the reflectance function, and the Fourier transforms of $p^2$ and $q^2$ will have a doubling effect in the frequency domain.

In this thesis, we present a different linear approach. In our approach, we first apply the discrete approximation for the surface gradient using finite differences of the depth, then linearize the reflectance function in term of the depth instead of the surface gradient, and solve for the depth directly. Our approach provides better results for images with central or low-angle illumination.
Chapter 3

A LINEAR SHAPE FROM SHADING METHOD

In this chapter, I will present an extremely simple linear approximation algorithm for shape from shading, which can be implemented in 25 lines of C code. The algorithm is very fast, taking 0.2 seconds on a Sun Sparc Station-1 for a 128 x 128 image, and the computation is purely local and highly parallelizable. Our method is more general, and can be applied to both Lambertian and specular reflectance models. I will also present the comparison of proposed method with Pentland’s linear method [28], and provide implementation details.

3.1 Lambertian Surfaces

The reflectance function for the Lambertian surfaces is modelled as follows:

\[ E(x, y) = R(p, q) \]
\[ = \frac{1 + pp + qq}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p^2 + q^2}} \]
\[ = \frac{\cos \sigma + p \cos \tau \sin \sigma + q \sin \tau \sin \sigma}{\sqrt{1 + p^2 + q^2}} \]
where \( E(x, y) \) is the gray level at pixel \((x, y)\), 
\[
p = \frac{\partial Z}{\partial x} \quad q = \frac{\partial Z}{\partial y} \quad p_s = \frac{\cos \tau \sin \sigma}{\cos \sigma},
\]
\[
q_s = \frac{\sin \tau \sin \sigma}{\cos \sigma}, \quad \tau \text{ is the tilt of the illuminant and } \sigma \text{ is the slant of the illuminant.}
\]
Using the following discrete approximations for \( p \) and \( q \)

\[
p = \frac{\partial Z}{\partial x} = Z(x, y) - Z(x - 1, y) \quad (3.3)
\]
\[
q = \frac{\partial Z}{\partial y} = Z(x, y) - Z(x, y - 1), \quad (3.4)
\]

the above reflectance equation can be rewritten as:

\[
0 = f(E(x, y), Z(x, y), Z(x - 1, y), Z(x, y - 1))
\]
\[
= E(x, y) - R(Z(x, y) - Z(x - 1, y), Z(x, y) - Z(x, y - 1)). \quad (3.5)
\]

For a fixed point \((x, y)\) and a given image \( E \), a linear approximation (Taylor series expansion up through the first order terms) of the function \( f \) (equation 3.5) about a given depth map \( Z^{n-1} \) is

\[
0 = f(E(x, y), Z(x, y), Z(x - 1, y), Z(x, y - 1)) \quad (3.6)
\]
\[
\approx f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x - 1, y), Z^{n-1}(x, y - 1))
\]
\[
+ (Z(x, y) - Z^{n-1}(x, y)) \frac{\partial}{\partial Z(x, y)} f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x - 1, y), Z^{n-1}(x, y - 1))
\]
\[
+ (Z(x - 1, y) - Z^{n-1}(x - 1, y)) \frac{\partial}{\partial Z(x - 1, y)} f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x - 1, y), Z^{n-1}(x, y - 1))
\]
\[
+ (Z(x, y - 1) - Z^{n-1}(x, y - 1)) \frac{\partial}{\partial Z(x, y - 1)} f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x - 1, y), Z^{n-1}(x, y - 1))
\]

The above equation can be written as follows:

\[
\frac{\partial}{\partial Z(x, y - 1)} f(E(x, y), Z^{n-1}(x, y), Z^{n-1}(x - 1, y), Z^{n-1}(x, y - 1)) \ast Z(x, y - 1)
\]
\[ \begin{align*}
&+ \frac{\partial}{\partial Z(x-1,y)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) * Z(x-1,y) \\
&+ \frac{\partial}{\partial Z(x,y)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) * Z(x,y) \\
&= - f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) \\
&+ Z^{-1}(x,y) * \frac{\partial}{\partial Z(x,y)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) \\
&+ Z^{-1}(x-1,y) * \frac{\partial}{\partial Z(x-1,y)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) \\
&+ Z^{-1}(x,y-1) * \frac{\partial}{\partial Z(x,y-1)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)),
\end{align*} \]

or in vector form as follow:

\[
(0, \ldots, a_{x,y-1}, 0, \ldots, a_{x-1,y}, a_{x,y}, 0, \ldots) \begin{pmatrix}
Z_{1,1} \\
\vdots \\
Z_{x,y} \\
\vdots \\
Z_{N,N}
\end{pmatrix} = b_{x,y}, \quad (3.7)
\]

where

\[
a_{x,y} = \frac{\partial}{\partial Z(x,y)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1))
\]

and

\[
b_{x,y} = - f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) \\
+ Z^{-1}(x,y) * \frac{\partial}{\partial Z(x,y)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) \\
+ Z^{-1}(x-1,y) * \frac{\partial}{\partial Z(x-1,y)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1)) \\
+ Z^{-1}(x,y-1) * \frac{\partial}{\partial Z(x,y-1)} f(E(x,y), Z^{-1}(x,y), Z^{-1}(x-1,y), Z^{-1}(x,y-1))
\]
For a $N \times N$ image, there are $N^2$ such equations which will form a linear system, $A \cdot Z = B$, where $A$ is a $N^2 \times N^2$ matrix, $Z$ and $B$ are $N^2 \times 1$ vectors. This linear system is difficult to solve directly, since it will involve the inverse of a huge matrix, $A$. However, it can be solved easily using the Jacobi iterative method. Now, let us carefully examine the Jacobi iterative method. For a given initial approximation $Z^0$, each depth value is solved sequentially in an iteration. For example, the depth value $Z(x, y)$ at the $n^{th}$ iteration can be solved using the previous estimates, $Z^{n-1}(i, j)$, for all $Z(i, j)$ with $i \neq x$ and $j \neq y$. When $Z^{n-1}(x-1, y)$ and $Z^{n-1}(x, y-1)$ are respectively substituted for $Z(x-1, y)$ and $Z(x, y-1)$ in equation 3.6 the third and fourth terms on the right hand side vanish. Therefore, equation 3.6 reduces to a surprisingly simple form given in the following equation:

$$0 = f(Z(x, y))$$

$$\approx f(Z^{n-1}(x, y)) + (Z(x, y) - Z^{n-1}(x, y)) \frac{\partial}{\partial Z(x, y)} f(Z^{n-1}(x, y)).$$

(3.8)

Then for $Z(x, y) = Z^n(x, y)$, the depth map at the $n$-th iteration, can be solved directly as follows:

$$Z^n(x, y) = Z^{n-1}(x, y) + \frac{-f(Z^{n-1}(x, y))}{\frac{\partial}{\partial Z(x, y)} f(Z^{n-1}(x, y))}$$

(3.9)
Now, assuming an initial estimate of $Z^0(x, y) = 0$ for all pixels, the depth map can be iteratively refined using Equation 3.9. We have observed that applying Gaussian averaging and median filtering to the final depth map $Z^n(x, y)$ results in a much better depth map.

### 3.2 Specular Surfaces

The reflectance function for a specular surface using Torrance-Sparrow model and the terminology in equation (3.1) can be modelled as follows:

$$E(x, y) = R(p, q) = I_S = Ke^{-\left(\frac{\cos^{-1}(N \cdot H)}{m}\right)^2}.$$ 

Using the discrete surface normal as before, and applying the same technique as in the case of a Lambertian surface, we have

$$0 = f(Z(x, y))$$

$$= E(x, y) - R(Z(x, y)) = Z(x - 1, y), Z(x, y) = Z(x, y - 1))$$

$$= E(x, y) - I_S$$
\[\approx f(Z^{n-1}(x,y)) + (Z(x,y) - Z^{n-1}(x,y)) \frac{\partial f}{\partial Z(x,y)}(Z^{n-1}(x,y)), \quad (3.11)\]

where

\[
\frac{\partial f(Z^{n-1}(x,y))}{\partial Z(x,y)} = -2.0 \cdot K \cdot e^{-\left(\frac{\cos^{-1}(N \cdot H)}{m}\right)^2} \cdot \frac{\cos^{-1}(N \cdot H)}{m^2 \cdot \sqrt{1 - (N \cdot H)^2}} \cdot ((H_x + H_y + H_z) - \frac{(pH_x + qH_y + H_z) \cdot (p + q)}{p^2 + q^2 + 1}) \cdot (p^2 + q^2 + 1)^{-1/2}.
\]

Then the depth information can be recovered by the above formula with function \(f\), as in the case of a Lambertian surface.

### 3.3 Comparison With Pentland’s Method

Our method is similar to Pentland’s [28] linear shape from shading method in some aspects; therefore, we will compare these two methods here. Pentland uses the linear approximation of the reflectance map in \(p\) and \(q\). By taking the Taylor series expansion of the reflectance function \(R\), given in equation (3.1), about \(p = p_0, q = q_0\), up through the first order terms, we have

\[E(x,y) = R(p_0, q_0) + (p - p_0) \frac{\partial R}{\partial p}(p_0, q_0) + (q - q_0) \frac{\partial R}{\partial q}(p_0, q_0). \quad (3.12)\]
For Lambertian reflectance, the above equation at \( p_0 = q_0 = 0 \), reduces to

\[
E(x, y) = \cos \sigma + p \cos \tau \sin \sigma + q \sin \tau \sin \sigma.
\]

Next, Pentland takes the Fourier transform of both sides of the equation. Since the first term on the right is a DC term, it can be dropped. Using the identities:

\[
\frac{\partial}{\partial x} Z(x, y) \leftrightarrow F_Z(\omega_1, \omega_2)(-i\omega_1)
\]

\[
\frac{\partial}{\partial y} Z(x, y) \leftrightarrow F_Z(\omega_1, \omega_2)(-i\omega_2),
\]

where \( F_Z \) is the Fourier transform of \( Z(x, y) \), we get,

\[
F_E = F_Z(\omega_1, \omega_2)(-i\omega_1) \cos \tau \sin \sigma + F_Z(\omega_1, \omega_2)(-i\omega_2) \sin \tau \sin \sigma,
\]

where \( F_E \) is the Fourier transform of the image \( E(x, y) \). The depth map \( Z(x, y) \) can be computed by rearranging the terms in the above equation, then taking the inverse Fourier transform.

The major difference between Pentland’s method and our method is that instead of linearizing the reflectance in \( p \) and \( q \), we use the discrete approximations for \( p \) and \( q \) in terms of \( Z \), then linearize the reflectance in \( Z(x, y) \). In this way, we have the following advantages.

First, we feel that the linearization of reflectance in \( Z \) is better than the lineariza-
tion in \( p \) and \( q \). For instance, it produces a better depth estimate for the spherical surface than Pentland’s method. Figure 3.1(a) shows the gray level image of sphere without using any approximation. Figure 3.1(b) shows the image generated using linear approximation of reflectance map in \( Z \), Figure 3.1(c) shows the histogram of the difference in gray levels in 3.1(a) and 3.1(b), and Figure 3.1(d) shows the reconstructed depth by our method.

Second, when the light source direction and the viewing direction are similar (images with central illumination), as pointed out by Pentland, the quadratic terms of the surface normal \( (p^2\text{ and } q^2) \) will become dominant in the reflectance function, and the Fourier transforms of \( p^2 \) and \( q^2 \) will have a doubling effect in the frequency domain. Since we do not use the Fourier transform, we have not frequency doubling effect, and our method is more general, as it can apply to both low-angle illumination and central illumination.

Third, note that the Fourier components exactly perpendicular to the illuminant cannot be seen in the image data, and must be obtained from the boundary conditions, or simply assumed to be zero. In our method, the depth is obtained from the intensity domain rather than the Fourier domain; therefore, no boundary conditions are required.

Another advantage of our method is that, computationally, it is very simple. Each operation is purely local, so the method is highly parallelizable. In Pentland’s method, one must compute the Fourier and inverse Fourier transform of the whole
image, which is time-consuming.

### 3.4 Implementation Details

#### 3.4.1 The Kalman Filter Relevance

Our iterative algorithm can be implemented in a very straightforward manner. Assuming an initial estimate of $Z_0(x, y) = 0$ for all pixels, we only need to compute the function $f(Z^{n-1}(x, y))$, and the first derivative of the function $f'(Z^{n-1}(x, y))$ at each iteration. The formula in equation (3.9) will refine the depth map at each step. However, recall that the first derivative of the function $f'(Z^{n-1}(x, y))$ in Equation (3.10) guarantees a nonzero value only for the first step. Depending on the surface shape of the object, Equation (3.10) could become zero, which causes division by zero in equation (3.9). For example, when the surface normal is directly facing the light source ($p = p_s$ and $q = q_s$), or when $p = q$ and $p + q = p_s + q_s$, the derivative of the function $f'(Z(x, y))$ becomes zero. In order to solve this problem, we must make some modifications. Let us rewrite the Equation (3.9) as follows:

$$Z^n(x, y) = Z^{n-1}(x, y) + K^n(-f(Z^{n-1}(x, y))),$$

where $K^n$ needs to satisfy three constraints. First, $K^n$ is approximately equal to the inverse of $\frac{\partial f}{\partial Z(x, y)}(Z^{n-1}(x, y))$. Second, $K^n$ equals zero when $\frac{\partial f}{\partial Z(x, y)}(Z^{n-1}(x, y))$ approaches zero. And third, $K^n$ becomes zero when $Z^n(x, y)$ approaches to true...
Figure 3.1. The results for Sphere Image using our method.
(a) Gray level image without any approximation. The light source direction is (0.01, 0.01, 1).
(b) Gray level image using linearity in $Z$.
(c) Histogram of difference in (a) and (b).
(d) A 3-D plot of the depth map computed by our algorithm.
Now, we define $K^n$ as follows:

$$
K^n = \frac{S^n_{x,y} M_{x,y}}{W_{x,y} + S^n_{x,y} M^2_{x,y}},
$$

where

$$
M_{x,y} = \frac{\partial f}{\partial Z(x,y)}(Z^{n-1}(x,y)),
$$

$$
S^n_{x,y} = E[(Z^n(x,y) - Z(x,y))^2],
$$

$E$ is the expectation operator, and $W_{x,y}$ is small, but non-zero. Since $W_{x,y}$ is a small value, $K^n$ is approximately equal to $\frac{1}{M_{x,y}}$, which is the inverse of $\frac{\partial f}{\partial Z(x,y)}(Z^{n-1}(x,y))$. When $M_{x,y}$ approaches the zero, $K^n$ becomes zero. When $Z^n(x,y)$ approaches to true $Z(x,y)$, $S^n_{x,y}$ (the expected value of $(Z^n(x,y) - Z(x,y))^2$) will become zero. Therefore, $K^n$ will be zero. We can clearly see that the definition of $K^n$ satisfies all three constraints.

Many people in the vision community will recognize this as an example of Kalman filtering, which has been applied to many problems in the lower level vision. This can be considered as Extended Kalman filtering because, in general, the equations for both the state vector and a measurement vector in Kalman filter are nonlinear. However, if good estimates of these vectors are available, a linear approximation in a small neighborhood can be considered. That is precisely what is being done here. In the Kalman filtering terminology, $K$ is known as the Kalman gain, and $W_{x,y}$ and
$S_{x,y}$ are the standard deviations associated with the input and the state variables, respectively.

### 3.4.2 Parallel Implementation

Since the computation of our algorithm is purely local, it is very suitable for the parallel implementation. Currently, the algorithm takes .2 seconds for a $128 \times 128$ image on the Sun Sparc station-1. The algorithm can be made real-time by using a parallel machine. We have experimented with two preliminary versions of this algorithm on the BBN GP1000 parallel machine. In the first version, the input image was treated as common memory and was shared by all processors. In the second version, each processor was supplied with a copy of the input image in order to reduce the memory contention. For the first version the single processor of GP1000 took about 17 seconds, due to slow processors in the BBN machine. However, the speedup of 34 with 59 processors was achieved. With the second version, the single processor took about 6 seconds, and a speedup of 29 was obtained with the 32 processors. Figure 3.2 shows the time-processor curve for the sphere image using the second version of our parallel algorithm.

The architecture of GP1000 (MIMD machine) is not very suitable for our problem, and in general it is not appropriate for problems involving image and matrix data structures. In the GP1000 a whole row of an image or matrix is assigned to one process which results in memory contention. Note that due to the local nature of our algorithm, a SIMD machine will be more suitable for parallel implementation. The
MasPar, a SIMD parallel machine, has some features to facilitate parallelization of algorithms for solving our problem. First, the MasPar machine has a 2-D PE array of $64 \times 64$. Second, each processor has its own distributed local memory. Third, each processor can easily communicate with its neighbors. Therefore, an image can be naturally mapped to this PE array, in which each processor performs the same instructions simultaneously on one or more pixels.
3.5 Experimental Results

3.5.1 Results for Lambertian Surfaces

Results for synthetic images

The method was first tested on the synthetic Vase images. The synthetic Vase was generated using the formula provided by Ascher and Carter [2] as follows:

\[ Z(x, y) = \sqrt{f(y)^2 - x^2}, \]

where

\[ f(y) = 0.15 - 0.1 \times y \times (6y + 1)^2 \times (y - 1)^2 \times (3y - 2), \]

\[-0.5 \leq x \leq 0.5, \text{ and } 0.0 \leq y \leq 1.0.\]

Note that this yields a maximum depth value of approximately 0.29. In order to generate a depth map with the proper size and scale, we map the \( x \) and \( y \) ranges to \([0, 127]\), and scale \( Z \) by a factor of 128. The ground truth depth map of the synthetic Vase is shown in Figure 3.3(a), and the synthetic images generated from the ground truth with different light source directions \((0, 0, 1)\), \((5, 5, 7)\) and \((1, 0, 1)\) are shown in Figure 3.3(b)-(d). The estimated results are shown in Figure 3.4. We can clearly see that our method works very well on smooth objects with central illumination (as shown in Figure 3.4(a)). However, it does not work well when the light source is from
one side (as shown in Figure 3.4(c)).

The range data of Mozart and Penny (as shown in Figure 3.5(a) and Figure 3.8(a)) were obtained from a laser range finder, which were provided by Professor Kuo of University of Southern California. Images generated from the ground truth with different light source directions \((0,0,1), (5,5,7)\) and \((1,0,1)\) are shown in Figure 3.5(b)-(d) and Figure 3.8(b)-(d), respectively. The results from our method without any smoothing are shown in Figure 3.6 and 3.9. The results after Gaussian averaging and median filter smoothing are shown in Figure 3.7 and 3.10. As we can see, some peaks occurred at the boundary of the object surface were removed, and we have a much smoother surface.

**Results for real images**

We have applied our algorithm to several real images, and have obtained quite encouraging results. The results are shown in Figures 3.11–3.15. In all of these figures, the depth map is shown after two iterations. In these experiments, either the direction of the light source was computed by using Lee and Rosenfeld’s method [24], or the results for illuminant direction quoted in Zheng and Chellappa [43] were directly used.

The results for the Halloween mask image are shown in Figure 3.11. The picture of the plastic Halloween mask was taken by a standard camcorder, then digitized. The depth map computed by our algorithm is shown in Figure 3.11(b). The nose, eyes, and mouth are clearly visible. Also, the surface variations around the forehead and
Figure 3.3. Synthetic images of Vase generated using different light sources.  
(a) The ground truth depth map.  (b) Vase (0, 0, 1).  (c) Vase (5, 5, 7).  (d) Vase (1, 0, 1).
Figure 3.4. Results for Synthetic images of Vase.
(a) The estimated depth map for Vase (0,0,1). (b) The estimated depth map for Vase (5,5,7). (c) The estimated depth map for Vase (1,0,1).
Figure 3.5. Synthetic images of Mozart generated using different light sources. (a) The ground truth depth map. (b) Mozart (0, 0, 1). (c) Mozart (5, 5, 7). (d) Mozart (1, 0, 1).
Figure 3.6. Results for Synthetic images of Mozart.
(a) The estimated depth map for Mozart (0,0,1). (b) The estimated depth map for Mozart (5,5,7). (c) The estimated depth map for Mozart (1,0,1).
Figure 3.7. Results (with smoothing) for Synthetic images of Mozart.  
(a) The estimated depth map for Mozart (0, 0, 1). (b) The estimated depth map for Mozart (5, 5, 7). (c) The estimated depth map for Mozart (1, 0, 1).
Figure 3.8. Synthetic images of Penny generated using different light sources. (a) The ground truth depth map. (b) Penny (0,0,1). (c) Penny (5,5,7). (d) Penny (1,0,1).
Figure 3.9. Results for Synthetic images of Penny.
(a) The estimated depth map for Penny (0,0,1). (b) The estimated depth map for Penny (5,5,7). (c) The estimated depth map for Penny (1,0,1).
Figure 3.10. Results (with smoothing) for Synthetic images of Penny.
(a) The estimated depth map for Penny (0,0,1). (b) The estimated depth map for Penny (5,5,7). (c) The estimated depth map for Penny (1,0,1).
Figure 3.11. The results for Halloween Mask Image.
(a) The input image. The light source parameters estimated by Lee & Rosenfeld’s method are: \( \text{slant} = 39.27^\circ, \text{tilt} = 97.3^\circ \). (b) A 3-D plot of the depth map computed by our algorithm. (c) A reconstructed gray level image using depth map in (b) and constant \( \text{albedo} = 255 \) with the estimated light source direction \( (\text{slant} = 39.27^\circ, \text{tilt} = 97.3^\circ) \). (d) A reconstructed gray level image using depth in (b) and constant \( \text{albedo} = 255 \) with the light source direction \( (\text{slant} = 45^\circ, \text{tilt} = 0^\circ) \).
Figure 3.12. The results for Lenna Image.
(a) The input image. The light source parameters estimated by Zheng & Chellappa’s method are: $slant = 52.46^\circ$, $tilt = 11.73^\circ$. (b) A 3-D plot of the depth map computed by our algorithm. (c) A reconstructed gray level image using depth map in (b) and constant $albedo = 255$ with the estimated light source direction ($slant = 52.46^\circ$, $tilt = 11.73^\circ$). (d) A reconstructed gray level image using depth map in (b) and constant $albedo = 255$ with the light source direction ($slant = 45^\circ$, $tilt = 0^\circ$).
Figure 3.13. The results for Mannequin Image.
(a) The input image. The light source parameters estimated by Lee & Rosenfeld’s method are: slant = 42.2°, tilt = 14.4°. (b) A 3-D plot of the depth map computed by our algorithm. (c) A reconstructed gray level image using depth map in (b) and constant albedo = 255 with the estimated light source direction (slant = 42.2°, tilt = 14.4°). (d) A reconstructed gray level image using depth in (b) and constant albedo = 255 with the light source direction (slant = 135°, tilt = 0°).
Figure 3.14. The results for Yowman Image.
(a) The input image. The light source parameters estimated by Lee & Rosenfeld's method are: \( \text{slant} = -45.75^\circ, \text{tilt} = 62.14^\circ \). (b) A 3-D plot of the depth map computed by our algorithm. (c) A reconstructed gray level image using depth map in (b) and constant \( \text{albedo} = 255 \) with the estimated light source direction (\( \text{slant} = -45.75^\circ, \text{tilt} = 62.14^\circ \)). (d) A reconstructed gray level image using depth map in (b) and constant \( \text{albedo} = 255 \) with the light source direction (\( \text{slant} = 45^\circ, \text{tilt} = 0^\circ \)).
Figure 3.15. The results for Part Image.
(a) The input image. The light source parameters estimated by Lee & Rosenfeld’s method are: \( \text{slant} = 65.28^\circ \), \( \text{tilt} = 237.94^\circ \). (b) A 3-D plot of the depth map computed by our algorithm. (c) A reconstructed gray level image using depth map in (b) and constant \( \textit{albedo} = 255 \) with the estimated light source direction (\( \text{slant} = 65.28^\circ \), \( \text{tilt} = 237.94^\circ \)). (d) A reconstructed gray level image using depth map in (b) and constant \( \textit{albedo} = 255 \) with the light source direction (\( \text{slant} = 45^\circ \), \( \text{tilt} = 0^\circ \)).
cheeks of the mask are present in the depth map. The gray level images generated from the recovered depth map using two different light sources are shown in Figures 3.11(c)–(d). These images look very similar to the original gray level image shown in 3.11(a).

The results for the Lenna image are shown in Figure 3.12. This image has been used as a test case in several papers on image compression and shape from shading. In this example, the nose, eyes, and lips are recovered quite reasonably, as shown in the 3D plot of the depth map in Figure 3.12(b). The surface area around the cheeks also appears nice and smooth. Two gray level images generated from the recovered depth map using two different light sources are shown in Figures 3.12(c)–(d).

Next, the results for the Mannequin image are shown in Figure 3.13. In this example, the head and the surface area around cheeks are recovered reasonably, as shown in the 3D plot of the depth map in Figure 3.13(b). Two gray level images generated from the recovered depth map using two different light sources are shown in Figures 3.13(c)–(d).

The results of the Yowman image are shown in Figure 3.14. This is a line drawing of a famous underground cartoon character named Zippy. This image was taken from Pentland’s [28] paper using a standard camcorder, and was then digitized. The recovered 3D surface shown in Figure 3.14(b) is amazingly good. The ears, nose, eyes, and lips are very clear. These results appear to be slightly better than the results shown by Pentland on the same image. Most parts of Pentland’s 3D plot appear
almost flat. Two gray level images generated from the recovered depth map using two different light sources are shown in Figures 3.14(c)–(d), which appear very similar to the original gray level image shown in Figure 3.14(a).

Finally, the results for the Part image are shown in Figure 3.15. This is the image of an automobile part. The recovered 3D depth map in Figure 3.15(b) clearly shows various surfaces. The round surface in the center appears at a higher depth than the four surface areas shown outside the center.

3.5.2 Results for Specular Surfaces

The results for the synthetic specular sphere image are shown in Figure 3.16. The input image, Figure 3.16(a), was generated based on Healey and Binford’s reflectance model. The reconstructed gray level image generated from the recovered depth map using the same light source and reflectance model is shown in Figure 3.16(b). The scaled needle map of the center area of the image is shown in Figure 3.16(c). The needle map clearly shows the shape of a sphere.

The results for the cylinder image are shown in Figure 3.17. The input image, Figure 3.17(a), was taken by a camcorder in our lab. Since we do not have an exact point light source, the image has one wide bright strip instead of a thin line. However, the reconstructed gray level image, Figure 3.17(b), looks very similar to the original gray level image.

The results for a tomato image are shown in Figure 3.18. The input image, Figure 3.18(a), was obtained from Carnegie Mellon University. The scaled needle map of the
center area of the image is shown in Figure 3.18(b).
Figure 3.16. The results for a synthetic specular sphere Image.
(a) The input image. The light source direction is (0.01,0.01,1). (b) A reconstructed gray level image using the estimated depth map with same light source direction. (c) The needle map.
Figure 3.17. The results for a specular cylinder image.
(a) The input image. The light source direction is approximate (0,0,1). (b) A reconstructed gray level image using the estimated depth map with same light source direction.
Figure 3.18. The results for a specular tomato Image.
(a) The input image obtained from CMU. The light source direction is \((-0.059, -0.039, 0.997)\). (b) The needle map of the center area.
Chapter 4

SFS FOR IMAGES WITH NONUNIFORM ALBEDO VALUES

Based on the simple Lambertian model, changes in image intensity are a function of changes in light source direction, surface orientation and composite albedo value at each pixel. The linear SFS method described in the previous chapter (and the majority of other SFS methods) assume that objects in a scene have uniform albedo value. Consequently, most SFS methods including ours cannot be applied directly to images containing surfaces with nonuniform albedo values. For example, if we apply our linear method (as described in the previous chapter) to the image of a flower with three different albedo values (as shown in Figure 4.1.a), we will obtain a depth map (as shown in Figure 4.1.b) showing a difference in height between the petals and the central region of the flower, which is incorrect. (All the petals and the central region of the flower should have approximately the same height.) This is due to the albedo variation. Similarly, if we apply Pentland’s method [28], Bichsel and Pentland’s method [3], and Zheng and Chellappa’s method [43] to the same image, we obtain wrong depth estimations as shown in Figure 4.1.c-e. Lee and Kuo’s method [21] does not even converge on this image.
Figure 4.1. Results for SFS methods.

(a) A real image of a flower with three different albedo values. (b) Results obtained from the linear SFS method. (c) Results obtained from Pentland’s method. (d) Results obtained from Bichsel and Pentland’s method. (e) Results obtained from Zheng and Chellappa’s method.
However, real images usually contain nonuniform albedo values. In this chapter, I will present a method to deal with images with nonuniform albedo values. We first estimate the albedo values for each pixel using Lee and Rosenfeld's local method [23], and segment the scene into regions with uniform albedo values. Then we adjust the intensity value for each pixel by dividing the corresponding albedo value before applying the linear shape from shading method.

### 4.1 Albedo Adjustment

In order to get the correct result for the SFS problem, we must cancel the effect of albedo variation before applying any SFS algorithm. To cancel the effect of albedo variation, we must compute the albedo value for each pixel. The two statistical approaches [24] and [43], as mentioned in section 2.2.2, are no longer applicable here because they attempt to estimate a constant albedo value for all pixels in the scene. The local estimate method by Lee and Rosenfeld [23] (see section 2.2.2), which assumes only the neighboring points have same albedo values, is more appropriate for our purpose.

As discussed in section 2.2.2, the albedo value $\rho$ for a point $P$ is estimated as

$$\rho^2 = \frac{I_P^2 - I_R I_Q}{I_P^2 - ((I_R + I_Q)/2)^2 I_P^2},$$

where $I_P$, $I_Q$ and $I_R$ are the image intensity respectively at point $P$ and its two
neighboring points Q and R. The above equation can be rewritten as

$$\rho^2 = \frac{4(I_P^2 - I_R I_Q)I_P^2}{(2I_P)^2 - (I_R + I_Q)^2}$$

$$= \frac{4(I_P^2 - I_R I_Q)I_P^2}{(2I_P - I_R - I_Q)(2I_P + I_R + I_Q)}.$$ 

We can see clearly that this method breaks down when $\Delta = 2I_P - I_R - I_Q = 0$; their method only works when $\Delta \neq 0$ (but small). If the images are digitized using 8 bits gray scale, the neighboring pixels may have similar intensity value. Due to this $\Delta$ may become zero. Therefore, we use the 24 bits RGB image format when we digitize the image and convert it to gray level image using floating point arithmetic.

### 4.2 Segmentation

The estimated albedo value for each pixel using Lee and Rosenfeld’s method [23] is not very accurate for real images, especially for points near edges. This is because neighboring points do not have the same albedo values at edge points, which violates their assumption. Therefore, we begin by segmenting the scene into different regions, based on the estimated albedo values, and assign a single albedo value for each region. The segmentation is performed using a simple histogram-based approach. We compute the histogram from the estimated albedo values, and apply the peakiness test to find the proper albedo value for each region.

The peakiness test is used to find genuine peaks in the histogram which correspond
to the albedo regions. A peak is a good peak if it is sharp and deep. The sharpness of a peak can be defined as the ratio of area of a rectangle enclosing the peak to the number of pixels $N$ under the peak, i.e., $\frac{N}{(W \times P)}$ where $W$ is the width of the peak from valley to valley, and $P$ is the height of the peak. The depth of a peak is the relative height of the peak, which can be defined as the ratio of the height of the valleys to the height of the peak, $\frac{(V_a + V_b)}{2P}$, where $V_a$ and $V_b$ are the two valley points to each side of the peak. The actual peakiness test will be the product of these ratios:

$$Peakiness = \left(1 - \frac{(V_a + V_b)}{2P}\right) \times \left(1 - \frac{N}{(W \times P)}\right).$$

If the peakiness is greater than some threshold, then that peak will be used for segmentation.

Due to the inaccuracy in the albedo estimate, some small regions near the edges may still be misclassified. Therefore, these small regions will have incorrect albedo values. Lee and Rosenfeld [23] used the median filter to avoid interactions across edges. However, the standard median filter, which replaces the albedo value of each pixel by the median of the albedo values in a neighborhood of that pixel, can deal with sharp albedo values in the scene, but not a small region. We will apply a modified median filter, in which we replace the albedo value of each pixel by the median of the non-zero albedo values (pixel with zero albedo value corresponds to the background) in a predefined neighborhood of that pixel. The segmentation can be improved by applying this modified median filter several times.
4.3 Proposed Method

The idea for our approach is very simple. We want to cancel the effect of albedo variation before applying the SFS algorithm. After we obtain the albedo map for the scene, we can adjust the intensity value for each pixel by dividing the corresponding albedo value. Since the intensity value at each point is just the composite albedo value multiplied by the dot product of the surface normal and the illumination direction \(I = \rho(\vec{N} \cdot \vec{L})\), the effect of albedo variation will be canceled by the intensity adjustment. Afterward, we can apply any SFS method on the adjusted intensity image to get the correct depth map.

The proposed method is summarized as follows:

- Estimate the albedo value for each pixel using Lee & Rosenfeld's local method.
- Segment scene into regions with uniform albedo values using a simple histogram-based approach.
- Apply a modified median filter to improve the result of segmentation.
- Adjust the intensity value for each pixel by dividing the corresponding albedo value.
- Apply the linear shape from shading method to the adjusted intensity values.

4.4 Experimental Results
Figure 4.2.: Results for a synthetic image of two spheres with different albedo values. (a) Input image. (b) Results obtained from the linear SFS method. (c) Estimated albedo map using Lee and Rosenfeld's method. (d) Histogram for the estimated albedo map. (e) Albedo map after histogram-based segmentation. (f) Albedo map after applying the modified median filter twice. (g) Results obtained from linear SFS method after intensity adjustment.
Our first experiment used a synthetic image of spheres (as shown in Figure 4.2.a). These two spheres have the same depth values. However, the sphere in the left of the image was generated with albedo value 200, and the sphere in the right was generated with albedo value 255. Figure 4.2.b shows the result obtained by applying the linear SFS method to the image without any intensity adjustment. We can see that the estimated depth map is incorrect, since the two spheres should have the same height. Figure 4.2.c shows the estimated albedo map (which was obtained from Lee and Rosenfeld’s method), and Figure 4.2.d shows the histogram. Figure 4.2.e shows the albedo map after applying the histogram-based segmentation, and Figure 4.2.f shows the albedo map after applying the modified median filter twice. Figure 4.2.g shows the correct depth estimate (two spheres with same height) which was obtained from the linear SFS method after the intensity adjustment using the estimated albedo map (Figure 4.2.f).

Figure 4.3 shows a real image containing a foam ball with two different albedo values. The image was taken by a standard camcorder, and digitized on a SUN workstation. The light source direction is approximately \((0,0,1)\), and the input image is shown in Figure 4.3.a. The depth map computed by the linear SFS method is shown in Figure 4.3.b. Figure 4.3.c shows the estimated albedo map, and Figure 4.3.d shows the histogram. Figure 4.3.e shows the albedo map after applying the histogram-based segmentation, and Figure 4.3.f shows the albedo map after applying the modified median filter. The depth map computed by linear SFS method after
intensity adjustment using the estimated albedo (Figure 4.3.f) is shown in Figure 4.3.g. We can see that the correct depth values can be obtained easily after the intensity adjustment.

The results for a real image containing a flower with two albedo values are shown in Figure 4.4. The image was taken by a SONY CCD camera with a single fiber optic illuminator, and digitized on a SUN workstation. The advantages of using a fiber optic illuminator are that the fiber optic cable can be mounted on the top of the camera or other location easily, and with proper lenses one can control intensity and focus of the light. The albedo value for petals is higher than the central region of the flower, and the central region of the flower should have approximately the same height as the petal region. The input image is shown in Figure 4.4.a, and the light source direction is approximately \((0,0,1)\). The linear SFS method produced an estimate with a large hole in the center (as shown in Figure 4.4.b). This is due to the fact that the albedo value of the central region is lower than the albedo value of the petals. Figure 4.4.c shows the estimated albedo map, and the histogram is shown in Figure 4.4.d. Figure 4.4.e shows the albedo map after applying the histogram-based segmentation, and Figure 4.4.f shows the albedo map after applying the modified median filter. A better depth estimate (as shown in Figure 4.4.g) was obtained by the proposed method.

Figure 4.5 shows the results for a real image of a three-albedo flower. The petals of the flower have two different albedo values, and the central region has the low-
est albedo value. All the petals and the central region of the flower should have approximately the same height. The linear SFS method produced a depth map (as shown in Figure 4.5.b) with different heights for those regions. Figure 4.5.c-d show the estimated albedo map, and the histogram. Figure 4.5.e shows the albedo map after applying the histogram-based segmentation, and Figure 4.5.f shows the albedo map after applying the modified median filter. The depth map produced by the proposed method (as shown in Figure 4.5.g) is much better than that without intensity adjustment.
Figure 4.3. Results for real image of a foam ball with two different albedo values. (a) Input image. (b) Results obtained from the linear SFS method. (c) Estimated albedo map using Lee and Rosenfeld’s method. (d) Histogram for the estimated albedo map. (e) Albedo map after histogram-based segmentation. (f) Albedo map after applying the modified median filter. (g) Results obtained from the linear SFS method after intensity adjustment.
Figure 4.4. Results for a real image of a flower with two different albedo values. (a) Input image. (b) Results obtained from the linear method. (c) Estimated albedo map using Lee and Rosenfeld’s method. (d) Histogram for the estimated albedo map. (e) Albedo map after histogram-based segmentation. (f) Albedo map after applying the modified median filter. (g) Results obtained from the linear method after intensity adjustment.
Figure 4.5. Results for a real image of a flower with three different albedo values. (a) Input image. (b) Results obtained from the linear SFS method. (c) Estimated albedo map using Lee and Rosenfeld’s method. (d) Histogram for the estimated albedo map. (e) Albedo map after histogram-based segmentation. (f) Albedo map after applying the modified median filter. (g) Results obtained from the linear SFS method after intensity adjustment.
Chapter 5

SFS FOR IMAGES WITH SCALE AMBIGUITY

As mentioned in the introduction, most SFS methods assume that only one object is presented in the scene. When multiple objects differing only in scale are present in a scene, there may be points with the same surface orientations but different depth values. None of the existent SFS methods including ours can solve this kind of ambiguity. In this chapter, I will present a new approach to deal with images containing multiple objects with scale ambiguity. The scale ambiguity problem is solved using the assumption that objects in a scene lie on the same background. In our approach, an image is divided into patches, and the depth estimate is derived from patches and re-aligned based on the background information to get the correct depth map.

5.1 Objects with Scale Ambiguity

Scale ambiguity may happen when multiple objects are present in a scene. For example, if we have two hemispheres differing in scale in a scene, in which the illumination direction is the same as the viewing direction, the two brightest points (the centers of the spheres in the scene) will have the same surface orientation but different depth
values. Actually, we can easily show that for any point on the first sphere, there exists a point on the second sphere with the same surface orientation but different depth values.

The depth map, $Z(x, y)$, for the image containing two hemispheres with different scales can be written as follows.

$$Z(x, y) = \begin{cases} \sqrt{r^2 - x^2 - y^2} & \text{if } r^2 - x^2 - y^2 \geq 0 \\ \sqrt{(sr)^2 - (x - a)^2 - (y - b)^2} & \text{if } (sr)^2 - (x - a)^2 - (y - b)^2 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $r$ is the radius of the first sphere, $s$ is a positive scale factor, $(a, b)$ is the center of the second sphere, and $r + sr < \sqrt{a^2 + b^2}$. The surface orientation, $(p(x, y), q(x, y))$, can be written as

$$ (p(x, y), q(x, y)) = \begin{cases} \left( \frac{-x}{\sqrt{r^2 - x^2 - y^2}}, \frac{-y}{\sqrt{r^2 - x^2 - y^2}} \right) & \text{if } r^2 - x^2 - y^2 \geq 0 \\ \left( \frac{-x - a}{\sqrt{(sr)^2 - (x - a)^2 - (y - b)^2}}, \frac{-y - b}{\sqrt{(sr)^2 - (x - a)^2 - (y - b)^2}} \right) & \text{if } (sr)^2 - (x - a)^2 - (y - b)^2 \geq 0 \\ 0 & \text{otherwise} \end{cases} $$

The point $P1 = (x, y)$ on the first sphere satisfies the condition $r^2 - x^2 - y^2 \geq 0$. We can find a point $P2 = (x', y') = (sx + a, sy + b)$, and

$$ (sr)^2 - (x' - a)^2 - (y' - b)^2 = (sr)^2 - ((sx + a) - a)^2 - ((sy + b) - b)^2 $$

$$ = s^2(r^2 - x^2 - y^2) $$

$$ \geq 0, $$
which is on the second sphere. We can see that \( P1 \) and \( P2 \) have the same surface orientation,

\[
(p(x, y), q(x, y)) = \left( \frac{-x}{\sqrt{r^2 - x^2 - y^2}}, \frac{-y}{\sqrt{r^2 - x^2 - y^2}} \right)
\]

and

\[
(p(x', y'), q(x', y')) = \left( \frac{-(x' - a)}{\sqrt{(sr)^2 - (x' - a)^2 - (y' - b)^2}}, \frac{-(y' - b)}{\sqrt{(sr)^2 - (x' - a)^2 - (y' - b)^2}} \right)
\]

\[
= \left( \frac{-(sx + a) - a}{\sqrt{(sr)^2 - ((sx + a) - a)^2 - ((sy + b) - b)^2}}, \frac{-(sy + b) - b}{\sqrt{(sr)^2 - ((sx + a) - a)^2 - ((sy + b) - b)^2}} \right)
\]

\[
= \left( \frac{-x}{\sqrt{r^2 - x^2 - y^2}}, \frac{-y}{\sqrt{r^2 - x^2 - y^2}} \right).
\]

However, \( P1 \) and \( P2 \) have different depth values,

\[
Z(x, y) = \sqrt{r^2 - x^2 - y^2}
\]

and

\[
Z(x', y') = \sqrt{(sr)^2 - (x' - a)^2 - (y' - b)^2}
\]

\[
= \sqrt{(sr)^2 - ((sx + a) - a)^2 - ((sy + b) - b)^2}
\]

\[
= s\sqrt{r^2 - x^2 - y^2}
\]
This kind of ambiguity can not be solved using the existent SFS methods. For example, if we apply Bichsel and Pentland's method [3] to a two hemispheres image (as shown in Figure 5.1a), without a priori information, their method will assign the same depth values for the two singular points, and start the propagation from these two points. This will result in an incorrect depth map (as shown in Figure 5.1b). The two spheres should not have the same maximum height. The major problem for Bichsel and Pentland's minimum downhill approach [3] (see section 2.2.3) is the multiple singular points problem. Without a priori knowledge, these singular points will be initialized with the same depth values, and will have identical depth values at the end. This is incorrect for a scene containing objects with scale ambiguity.
5.2 Proposed Method

In order to solve the scale ambiguity case, we assume that **objects in a scene lie on the same background.** With this assumption, we can solve the scale ambiguity as follows. First, we partition the scene into smaller patches (2x2 or 3x3...), and estimate the depths from each patch separately by using Bichsel and Pentland’s method. There can be three different types of patches: patches which contain singular points, patches which do not contain singular points, and patches containing background only. For patches containing singular points, the initial depth values for singular points are still the same. For patches which do not contain singular points, the SFS method will use the brightest point in the patch as the starting point. We will assign the initial depth value for the brightest point in the patch according to the ratio between the brightest point and the singular point in the scene. For patches containing only background, we simply assign zero depth values to all points in the patch. In the next step, we align the estimated depth map in each patch with the same background depth values. With the same patch size and initial depth values for singular points, the minimum downhill approach will have higher background depth values for patches containing smaller objects. After aligning the background value in all patches with the same value, we can obtain the correct depth map for the scene containing multiple objects with scale ambiguity.

The proposed method is summarized as follows:

- Partition the scene into smaller patches (2x2 or 3x3...).
- Estimate the depths for each patch using a minimum downhill approach.
  
  - For patches containing singular points, assign the same initial depth value for singular points.
  
  - For patches which do not contain singular points, assign the initial depth value for the brightest point in the patch according to the ratio between the brightest point and the singular point in the scene.
  
  - For patches containing only background, assign zero depth values to all points in the patch.

- Align the background depth value in all patches with the same value.

The technique of partitioning an image into patches has been used in motion analysis (such as [40, 41]). Traditional approaches to motion analysis assume that the optical flow is smooth. These methods have difficulty dealing with occlusion boundaries. Wang and Adelson [40] decompose the image sequence into a set of overlapping layers, where the motion of each layer is described by a smooth flow field. The layered image motion representation provides an accurate description of motion discontinuities and motion occlusion. Bober and Kittler [4] develop a block-based algorithm for estimating optical flow. The locality of using block made it possible for them to model the effect of changing illumination. One common problem in these approaches is how to choose the patch size. For our application, a larger patch size increases the probability of multiple singular points in a patch. Therefore we must
choose a smaller patch size. However, if the patch size is too small, there will be patches containing only object surface points (without any background information); the estimated depth map of such a patch cannot be aligned with other patches. In this research, we will partition a 128x128 image into either (2x2) patches or (3x3) patches.

5.3 Experimental Results

The proposed method was initially tested on a synthetic image of spheres. Figure 5.2a shows the input image, which was generated using 3 spheres of different sizes. The depth map computed by Bichsel and Pentland’s method is shown in Figure 5.2b. We see that all three spheres have the same maximum depth value, which is incorrect. Figure 5.2c shows the correct estimated depth map by our method. The number of patches used for this example is nine (3x3).

Figure 5.3 shows another example, an image of two Mozarts with only scale difference. The input image (as shown in Figure 5.3a) was generated using the ground truth depth map (as shown in Figures 5.3d) with light source direction (0, 0, 1). Figure 5.3b shows the estimated depth map computed by Bichsel & Pentland’s SFS method. The depth map have the same maximum depth value for the two Mozarts, which is incorrect. The estimated depth map obtained from the proposed method is shown in Figure 5.3c. The number of patches used for this example is four(2x2).
Figure 5.2. Results for a synthetic image of three spheres.
(a) Input image. (b) Estimated results by Bichsel and Pentland’s method. (c) Estimated results by the proposed method.
Figure 5.3. Results for a synthetic image of Mozarts with different scales. (a) Input image. (b) Results obtained from Bichsel and Pentland's method. (c) Results obtained from the proposed method. (d) Ground truth.
Chapter 6

SFS UNDER PERSPECTIVE PROJECTION

Most SFS methods assume orthographic projection, which simplifies the problem by allowing us to apply the discrete approximation of the surface gradient using a straightforward finite difference method. Because, in orthographic projection, the image and world coordinates are assumed to be the same. However, for real images, perspective projection is a more realistic model than orthographic projection. In perspective projection, the world and image coordinates are not the same. The simple finite difference discrete approximation for surface gradients used under orthographic projection, \( p_{x,y} = \frac{\partial Z(x,y)}{\partial x} = Z(x,y) - Z(x-1,y) \) and \( q_{x,y} = \frac{\partial Z(x,y)}{\partial y} = Z(x,y) - Z(x,y-1) \), is no longer applicable under perspective projection, because the image position components \((x,y)\) are in fact functions of the depth \(Z\). In this chapter, I will present a direct solution for the discrete approximation under perspective projection. The surface gradient is derived mathematically by relating the depth value of the surface point with the depth value of the corresponding image point. We also demonstrate how we can apply the new discrete approximation to a more complicated and realistic reflectance model for SFS problem.
Figure 6.1. Perspective projection model.

6.1 Perspective Projection and Discrete Approximation

Perspective projection (as shown in Figure 6.1), which models an ideal pinhole camera, maps a surface point $P = (X, Y, Z(X, Y))$ to an image point $p = (x, y)$ by the following relationship,

$$
x = -f \frac{X}{Z(X, Y)}, \quad \text{and} \quad y = -f \frac{Y}{Z(X, Y)}
$$

where $f$ is the focal length. With this model, the surface gradients $p$ and $q$ should be defined as

$$
p = \frac{\partial Z(X, Y)}{\partial X} \quad \text{and} \quad q = \frac{\partial Z(X, Y)}{\partial Y}.
$$
Here \((X, Y)\) are the world coordinates as compared to \((x, y)\) which are the image coordinates. The discrete approximations of \(p\) and \(q\) by a straightforward finite difference method using \(Z(x, y)\) (recall the equations (3.3) and (3.4) in chapter 3) is no longer applicable here since the image position components \((x, y)\) for a point \((X, Y, Z)\) are in fact functions of the depth \(Z\). In this section, we present a different discrete approximation of \(p\) and \(q\) for images with the perspective projection model as follows.

The depth value for an image point \((x, y)\) is defined as

\[
Z(x, y) = Z(X, Y),
\]

where

\[
X = \frac{x Z(x, y)}{-f}, \quad Y = \frac{y Z(x, y)}{-f},
\]

\(f\) is the focal length and \((X, Y, Z(X, Y))\) are the world coordinates of the corresponding surface point.

Consider the derivative of \(Z(X, Y)\), we have

\[
dZ(X, Y) = \frac{\partial Z(X, Y)}{\partial X} dX + \frac{\partial Z(X, Y)}{\partial Y} dY = p \cdot dX + q \cdot dY
\]
From equation (6.2) and (6.3), we have

\[ dX = \frac{1}{-f} \left( Z(x, y) + x \frac{\partial Z(x, y)}{\partial x} \right) dx + \frac{1}{-f} \left( Z(x, y) + y \frac{\partial Z(x, y)}{\partial y} \right) dy \]

\[ = \left( \frac{Z(x, y) + x \cdot \hat{p}}{-f} \right) dx + \frac{(x \cdot \hat{q})}{-f} dy \]

and

\[ dY = \frac{1}{-f} \left( y \frac{\partial Z(x, y)}{\partial x} \right) dx + \frac{1}{-f} \left( y \frac{\partial Z(x, y)}{\partial y} \right) dy \]

\[ = \frac{(y \cdot \hat{p}) \, dx + (Z(x, y) + y \cdot \hat{q})}{-f} dy \]

where \( \hat{p} = \frac{\partial Z(x, y)}{\partial x} \) and \( \hat{q} = \frac{\partial Z(x, y)}{\partial y} \). Note that, the \( \hat{p} \) and \( \hat{q} \) can be approximated using the straightforward finite difference as before, i.e.

\[ \hat{p} = \frac{\partial Z(x, y)}{\partial x} = Z(x + 1, y) - Z(x, y) \]

\[ \hat{q} = \frac{\partial Z(x, y)}{\partial y} = Z(x, y + 1) - Z(x, y). \]

By substituting (6.5) and (6.6) into (6.4), we can represent the derivative of \( Z(X, Y) \) as

\[ dZ(X, Y) = p \cdot \left( \frac{Z(x, y) + x \cdot \hat{p}}{-f} dx + \frac{x \cdot \hat{q}}{-f} dy \right) + q \cdot \left( \frac{y \cdot \hat{p}}{-f} dx + \frac{Z(x, y) + y \cdot \hat{q}}{-f} dy \right) \]

\[ = \left( \frac{Z(x, y) + x \cdot \hat{p} \cdot f - f}{-f} + q \cdot \frac{y \cdot \hat{p} \cdot f - f}{-f} \right) dx + \left( \frac{x \cdot \hat{q} \cdot f - f}{-f} + q \cdot \frac{Z(x, y) + y \cdot \hat{q} \cdot f - f}{-f} \right) dy. \]
From equation (6.1), we have

\[ dZ(X, Y) = dZ(x, y) \]
\[ = \frac{\partial Z(x, y)}{\partial x} dx + \frac{\partial Z(x, y)}{\partial y} dy \]
\[ = \hat{p} dx + \hat{q} dy \tag{6.8} \]

By comparing equation (6.7) and (6.8), we have

\[ p \cdot \frac{Z(x, y) + x \cdot \hat{p}}{-f} + q \cdot \frac{y \cdot \hat{p}}{-f} = \hat{p} \tag{6.9} \]
\[ p \cdot \frac{x \cdot \hat{q}}{-f} + q \cdot \frac{Z(x, y) + y \cdot \hat{q}}{-f} = \hat{q}, \tag{6.10} \]

and we can solve for \( p \) and \( q \) in terms of \( Z(x, y) \) directly as follows:

\[ p = \frac{B \cdot F - C \cdot E}{B \cdot D - A \cdot E} \tag{6.11} \]
\[ q = \frac{C \cdot D - A \cdot F}{B \cdot D - A \cdot E}, \tag{6.12} \]

where

\[ A = \frac{Z(x, y) + x \cdot \hat{p}}{-f} \]
\[ B = \frac{y \cdot \hat{p}}{-f} \]
\[ C = \hat{p} \]
SFS Using A Sophisticated Reflectance Model

Now let us consider a more complicated reflectance function: a reflectance map with Lambertian surface, a nearby point light source, and perspective projection. When an object is illuminated by a nearby point light source, the inverse square law for illumination must be considered, because the distance information is included in the law. This model was used by Iwahori et al. [18] for their photometric method, and the reflectance map can be written as:

\[
D = \frac{x \cdot \hat{q}}{-f}
\]

\[
E = \frac{Z(x,y) + y \cdot \hat{q}}{-f}
\]

\[F = \hat{q}.
\]

\[E(x,y) = R(p,q)\]

\[= C \cdot \frac{p \cdot l + q \cdot m - n}{(l^2 + m^2 + n^2)^{3/2}(p^2 + q^2 + 1)^{1/2}}\]

where \(E(x,y)\) is the gray at pixel \((x,y)\), which is mapped to a surface point \((X,Y,Z)\),

\[l = Xs - X,\]

\[m = Ys - Y,\]
\[ n = Z_s - Z, \]
\[ p = \frac{\partial Z(X,Y)}{\partial X}, \]
\[ q = \frac{\partial Z(X,Y)}{\partial Y}, \]
\[ X = x \frac{Z}{f}, \]
\[ Y = y \frac{Z}{f}, \]

\( C \) is the illumination constant and \((X_s, Y_s, Z_s)\) is the point light source location.

For a given image \( E \), light source location and illumination constant, the reflectance function (equation 6.14) is a nonlinear function of surface gradient \( p \) and \( q \). By applying the discrete approximation derived in the previous section for the surface gradient (equation 6.11 and 6.12), the reflectance function becomes a nonlinear function of depth, \( Z \), i.e.

\[
E(x, y) = R(p, q) = R(Z(x, y), Z(x + 1, y), Z(x, y + 1)).
\]

This reflectance equation can be rewritten as:

\[
0 = f(Z) = \sum_{x,y} (E(x, y) - R(Z))^2. \tag{6.15}
\]

We can directly solve for the depth \( Z \) using a conjugate gradient technique.
The objective here is, given an image $E$ and the function $f(Z)$ that depends on depth variable $Z$, find the proper values for $Z$ which will minimize the function. This is known as the \textit{multivariable nonlinear optimization problem}. In the case where we are able to calculate, at a given point $P$, not just the value of the function $f(P)$ but also the gradient (vector of first partial derivatives) $\nabla f(P)$, the Steepest Descent method \cite{6} can be used. The idea of the Steepest Descent method is simple, and can be summarized as follows: start at an initial estimated point $P_0$. As many times as needed, move from point $P_i$ to the point $P_{i+1}$ by minimizing along the line from $P_i$ in the direction of the local downhill gradient, $-\nabla f(P_i)$.

However, the problem with the Steepest Descent method is that it may take too many tiny steps to get to the minimum. As pointed out in \cite{32}, we really want a way of proceeding not down the new gradient direction, but rather in a direction that is somehow constructed to be \textit{conjugate} to the old gradient, and, insofar as possible, to all previous directions traversed. (Two vectors ($u$ and $v$), are said to be \textit{conjugate} to each other, if the condition $0 = u \cdot A \cdot v$ holds, where $A$ is the Hessian matrix of the function at $P$.) Methods that accomplish this construction are called \textit{conjugate gradient} methods. The conjugate gradient procedure \cite{32} is an iterative minimization technique. The algorithm requires the construction of only two functions: one that computes the value of the function, $f(Z)$, and another that computes the gradient value of the function with respect to the unknown vector, $Z$. However, by solving directly for depth, one can use the result from stereo or range data to provide initial
and boundary conditions. For a given image, $E$, and an initial depth map, $Z_0$, the value of the function $f(Z_0)$ (equation 6.15) is easy to compute. The derivative of the function with respect to the depth are lengthy and complex which can be derived using a symbolic mathematical package, such as MACSYMA. The conjugate gradient method is much more efficient for optimizing functions of many variables as compared with simple gradient decent methods.

### 6.3 Experiment

Figure 6.2a shows a gray level image of a synthetic sphere which is generated using the reflectance model described in the previous section (equation 6.14). The center of the lens is placed at the origin $(0,0,0)$, and a point source is located at the coordinate $(150,150,0)$. The center of the sphere is located at $(0,0,512)$, and the focal length is set to 256. Figure 6.2b shows the results obtained from the linear SFS method (described in chapter 3). We can clearly see that the upper right corner of the background is higher than the left bottom corner, and the shape of the sphere is tilted. This is mainly due to the modeling error and the self-shadow area in the scenes. Figure 6.2c shows a much better result, which is obtained by solving the depth directly using conjugate gradient method as described in the previous section. (We used the ground truth depth map with 4% uniform random noise as the initial estimate.) As we can see that, by applying the discrete approximation for the surface gradient, we can solve for the depth directly using a more efficient numerical method.
Figure 6.2. Results for Synthetic images of Sphere (under perspective projection).
(a) A gray level image of a synthetic sphere generated under perspective projection with light source location (150, 150, 0).
(b) The estimated depth map obtained from the linear SFS method (described in section 3).
(c) The estimated depth map obtained from the method proposed in this chapter.
(conjugate gradient method), and we can use other source of information (such as stereo processing or range data) to provide initial and boundary conditions.
Chapter 7

CONCLUSIONS

The recovery of surface shape from a single shaded image is a very important problem in computer vision. In this thesis, we have presented a very simple and fast algorithm for computing the depth map from a single monocular image. Our approach uses a linear approximation of the reflectance function in $Z$. We first employ discrete approximations of the surface gradient ($p$ and $q$), and then compute the Taylor series of the reflectance function up to the first order term. This results in a simple iterative scheme for computing the depth map from an intensity image when the light source direction is known. The algorithm works quite well and yields better results for images with central illumination or low-angle illumination.

Our algorithm similar to most traditional SFS algorithms perform poorly on images with nonuniform albedo or images containing multiple objects with scale ambiguity, because those images violate the basic assumptions made by our SFS method. Therefore, we extended our method for images with nonuniform albedo values. In order to cancel the effect of albedo variation, we first estimate the albedo values for each pixel, and segment the scene into regions with uniform albedo values. However, the estimated albedo values are inaccurate at the points near edges, so we apply
a modified median filter to improve the result. We then adjust the intensity value for each pixel by dividing the corresponding albedo value before applying the linear shape from shading method. We also present a new approach to deal with images containing multiple objects with scale ambiguity. A depth estimate is derived from patches using a minimum downhill approach and re-aligned based on the background information to get the correct depth map.

Perspective projection, which models an ideal pin-hole camera, is a more realistic model than orthographic projection, which is commonly used in most SFS methods. As pointed out by Lee and Kuo [22], the straightforward finite difference approximation for surface gradients used under orthographic projection is no longer applicable here, because the image position components are in fact functions of the depth. In this thesis, we provide a direct solution for the discrete approximation under perspective projection. The surface gradient is derived mathematically by relating the depth value of the surface point with the depth value of the corresponding image point. We also demonstrate how we can apply the new discrete approximation to a more complicated and realistic reflectance model for SFS problem.

Future work in shape reconstruction may proceed along three major lines:

1. More sophisticated reflectance models. As we noted, the reflectance models used in SFS methods are overly simplistic. More sophisticated models, such as the model proposed by Oren and Nayar [27], Clark [7], Hougen and Ahuja [16], or Langer and Zucher [19], can be used to reduce modeling error in SFS
problem.

2. Combination of shading with other cues (such as the recent works by Leclerc and Bobick [20], Pien and Gauch [33] and Cryer et al. [8, 9]). One can use the results of stereo or range data to improve the results of SFS or use the results of SFS or range data to improve the results of stereo. Another approach is to directly combine results from shading and stereo (such as [8]).

3. Multiple images. When more images are available, shape reconstruction can be greatly improved. Multiple images can also be employed by moving either the viewer (such as the work by Heel [13]) or the light source (such as the work by Zhang et al. [41] and Shah et al. [34]) in order to successively refine the shape. Successive refinement can improve the quality of estimates by combining estimates between image frames, and reduce the computation time since the estimates from the previous frame can be used as the initial values for the next frame, which may be closer to the correct solution. By using successive refinement, the process can be easily started at any frame, stopped at any frame, and restarted if new frames become available. The advantage of moving the light source over moving the viewer is the elimination of the mapping of the depth map (warping) between image frames.
REFERENCES


