ABSTRACT

PATHAK, CHETNA. Shape Recovery from Brightness Images of 2-Dimensional Specular Reflectors. (Under the direction of Dr. Wesley Snyder.)

This thesis proposes two methods to reconstruct the shape of 2-dimensional specular reflectors from multiple brightness images of the reflectors. Several images of a reflector are taken as the camera and a mounted light source move along the x-axis. From the specularities present in the images, reflectance information and camera locations are extracted. This information is used for reconstruction. The first method is an optimization algorithm that determines the surface shape by minimizing an objective function. The second method employs relaxation labeling. Reflector points are declared labels while the camera positions are considered objects. Object-label associations are investigated and the most consistent among these are retained to give the reflector shape.

SHAPE RECOVERY FROM BRIGHTNESS IMAGES OF 2-DIMENSIONAL SPECULAR REFLECTORS

by

CHETNA PATHAK

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

ELECTRICAL ENGINEERING

Raleigh

2003

APPROVED BY:

_____ _____

_____

Chair of Advisory Committee

To my parents

Biography

Chetna Pathak was born in Ahmedabad, India on 31st May, 1979. She did her schooling in Ahmedabad at Little Flower School and St. Xavier's High School, Loyola Hall. After graduating from High School in 1996, she joined the Electronics and Communications Engineering Department at L.D. College of Engineering, Gujarat University. She graduated with a Bachelor of Engineering Degree in 2000.

She then moved to Bombay for a year to work as a Research Engineer in the Indian Institute of Technology. In August 2001, she enrolled in the Master of Science program of the Electrical and Computer Engineering Department at North Carolina State University. She carried out her MS thesis research in the area of Computer Vision.

Acknowledgements

I would like to thank my advisor, Dr. Wesley Snyder for his invaluable guidance which has shaped this thesis. I believe that what I have learnt from working with him will stand me in good stead throughout my academic life. I am grateful to Dr. Hamid Krim and Dr. Griff Bilbro for serving on my advisory committee. I am also thankful to my lab-mates for their help.

I want to thank my parents for the unconditional love and support they have extended to me all my life. I owe my every acheivement to them. This gratitude extends to my sisters , Seemantini and Medha, not only for being wonderful, fun siblings, but also for being the rock-steady pillars of support I could always turn to. I would like to thank Abhiram, for being the adorable brother he is. I am grateful to my brother-in-law, Prasad, for all his help.

Thanks go out to each of my friends from Ahmedabad and Bombay simply for being there. My life is all the richer to have you in it. I am grateful to my friends in Raleigh for the crazy, fun times that helped make my stay at NCSU memorable.

And finally, I thank Abheek for his constant love and support.

Table of Contents

List of Figures vii
List of Tables ix

Chapter 1 *Introduction* *1*

1.1 Motivation 1
1.1.1 Potential Applications *2*
1.2 Problem Description 3
1.2.1 Problem Statement *3*
1.2.2 Simulation Set-up *3*
1.2.3 Creation of Simulated Reflectance Images *6*
1.2.4 Simulation Results *7*
1.2.5 Mapping of Special Surface Features into Reflectance Image Domain . . *8*

Chapter 2 *Literature Review* *14*

2.1 Reflectance Models 15
2.1.1 Lambertian Surfaces *16*
2.1.2 Specular Surfaces *19*
2.1.3 Hybrid Sufaces *22*
2.2 Shape Recovery for Specular Reflectors 24

Chapter 3 *Approach* *27*

3.1 Extracting information from Special Surface Features: Corners 27
3.1.1 About the Hough Transform: *29*
3.1.2 Applying Hough Transforms to find corners: *32*
3.2 Specular Shape Recovery Approaches 34
3.2.1 Optimization Approach *34*
3.2.2 Using Derivatives to Solve for the Surface *43*
3.3 Proposed Algorithms 47

Chapter 4 *Algorithm 1: Shape Recovery through Optimization* *49*

4.1 Algorithm Description 49
4.1.1 The Objective Function *51*
4.1.2 Calculation of Estimated Camera Positions *55*
4.1.3 Choice of Candidate Solutions *55*

4.1.4 Algorithm - Shape Recovery through Optimization (SA) *56*
4.2 Algorithm Variations 57
4.3 Algorithm Drawbacks 58
4.3.1 Requires Corner Points to be Input *58*
4.3.2 Requires Length of Reflector to be known *58*

Chapter 5 *Algorithm 2: Shape Recovery through Relaxation Labeling* *59*

5.1 Algorithm Description 59
5.1.1 Relaxation Labeling - Theoretical Background *60*
5.1.2 Application of RL to this problem *62*
5.1.3 Algorithm - Shape Recovery using Relaxation Labeling *66*
5.2 Algorithm Variations 69
5.2.1 Selection of Final Reflector Points *69*
5.3 Algorithm Drawbacks 71
5.3.1 High Computational Complexity *71*
5.3.2 Requires Corner Points to be Input *71*
5.3.3 Requires Length of Reflector to be known *71*

Chapter 6 *Results and Conclusion* *72*

6.1 Corner Extraction 75
6.2 Algorithm 1 Results 80
6.2.1 Problem with Processing Experimental Data *81*
6.3 Algorithm 2 Results 82
6.4 Summary of Shape Reconstruction Results 85
6.5 Conclusion 87
6.6 Further Work 88

Bibliography *89*

List of Figures

Figure 1.1 Simulation set-up 4

Figure 1.2 Image representation of reflectance information 8

Figure 1.3 Corners in specular reflectors 9

Figure 1.4 Reflectance pattern of corners. 10

Figure 1.5 Linear segments in specular reflectors 12

Figure 1.6 Reflectance pattern of linear segments 13

Figure 2.1 Lambertian surface geometry 17

Figure 2.2 Effect of Lambert's Law and Cosine Law 18

Figure 2.3 Specular surface geometry 19

Figure 3.1 Corners in specular reflectors. 28

Figure 3.2 Hough Transform for detection of straight lines. 31

Figure 3.3 Indexing problem in minimization algorithm 39

Figure 3.4 Sample surfaces meeting criterion of equation (3.18) 41

Figure 3.5 Surface not meeting criterion of equation (3.18) 42

Figure 3.6 Problem in using Parametric Transform to Estimate Derivatives 45

Figure 3.7 Detection of linear segments. 46

Figure 4.1 Original and blurred reflectance images 52

Figure 5.1 Compatibility calculation for 2 non-corner points 64

Figure 5.2 Selecting a maximum probability labeling for every x. 70

Figure 6.1 Reflector 'S1' - 1st reflector used in Simulations 73

Figure 6.2 Reflector 'S2' - 2nd reflector used in Simulations 73

Figure 6.3 Reflector 'R1' - reflector used to gather experimental data 74

Figure 6.4 Accumulator image for Reflector 'S1' 76

Figure 6.5 Accumulator image for Reflector 'S2' 76

Figure 6.6 Accumulator image for Reflector R1(experiment 1) 77

Figure 6.7 Accumulator image for Reflector R1 (experiment 2) 78
Figure 6.8 Reconstruction for reflector ‘S1’ using Algorithm 1 80
Figure 6.9 Reconstruction for reflector ‘S2’ using Algorithm 1 80
Figure 6.10 Reflectance image for reflector R1 (experiment 1) 81
Figure 6.11 Reconstruction of reflector ‘S1’ using Algorithm 2, Variant 1 83
Figure 6.12 Reconstruction of reflector ‘S1’ using Algorithm 2, Variant 2 83
Figure 6.13 Reconstruction of reflector ‘S2’ using Algorithm 2, variant 1 84
Figure 6.14 Reconstruction of reflector ‘S2’ using Algorithm 2, variant 2 84

List of Tables

Table 1 Results of the Corner Extraction Algorithm 79

Table 2 Summary of Algorithm 1 and 2 Results. 86

Chapter 1 Introduction

1.1 Motivation

Shading greatly assists our perception of the world. A clear indication of this is that a photograph of a sphere would be indistinguishable from that of a circle if the shading information were not considered. Shading provides us an estimate of object shape in the absence of other cues and augments this estimate when other cues are present. Several techniques have been developed to recover the shape of a surface from its brightness images, each technique making certain assumptions about illumination conditions and surface properties. Depending on their reflectance characteristics, surfaces can be classified as Lambertian, Specular and Hybrid. Lambertian surfaces have a diffused reflectance, i.e., they reflect light in all directions. Specular surfaces reflect virtually all of the incident light along one direction. This specularity occurs when the angle of incidence is equal to the angle of reflectance. The reflectance properties of hybrid surfaces

are a combination of specular and Lambertian and will be discussed in more detail in section 2.1.3. Most shape-from-shading algorithms use the reflectance property of diffuse surfaces to assume that pixel brightness depends only on the position of the light-source relative to the surface. This does not hold true for specular reflectors, where the position of the observer is of equal importance. In this thesis, two algorithms are proposed for shape recovery of a specular object from multiple images taken when both the light source and the observer are moving. This thesis deals with the 2-D case. The algorithm results for nearly-2-dimensional specular objects (objects with very narrow width along the third dimension) are presented.

1.1.1 Potential Applications

Many practical applications in industrial computer vision require interpretation of specular surfaces. Often, the inspection and handling of shiny industrial parts such as turbine blades [5] involve the automated shape recovery of the part. Assembly lines for the automobile industry and junk-yards requiring separation of metallic objects are two ready examples which can benefit from algorithms such as the one presented in this thesis, when applied to the 3-D case. Modifi-

cations might make it possible for the algorithm applications to be extended to solder joints and electronic components mounted on PC boards.

1.2 Problem Description

1.2.1 Problem Statement

To develop and evaluate an algorithm for determining the shape of (approximately)2-D specular reflectors using images taken when both light source and observer are moving.

1.2.2 Simulation Set-up

In this section, the problem set up is described for 2-D reflectors:

A camera with a mounted point-light source is utilised to capture images of the specular reflector. The camera and light source are allowed to move along a linear path, henceforth referred to as the ‘camera axis’. As convention, the camera axis is considered to be parallel to the x-axis. The 2-dimensional specular reflector is aligned so that the it lies in the same plane as the camera axis, with its longest side more or less parallel to the axis. The camera and mounted light

source move along the camera axis capturing images of the specular reflector at regular intervals along the axis. The set up is as shown in Figure 1.1.

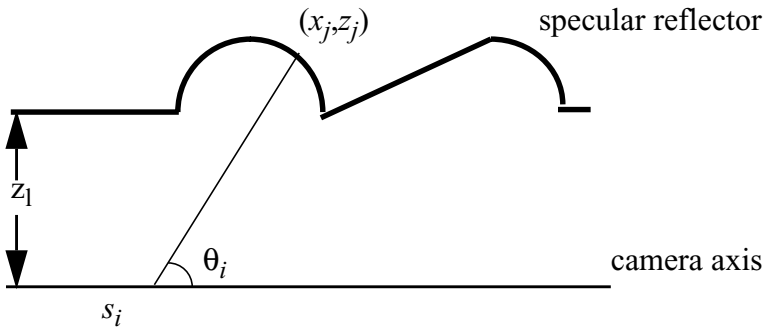


Figure 1.1 Simulation set-up

In Figure 1.1,

z_l = distance of the camera axis from the reflector

s_i = i^{th} position of camera on camera axis

θ_i = orientation of the camera line-of-sight at i^{th} position

(x_j,z_j) = j^{th} point of the specular reflector

At each position, s_i, the camera scans over the two quadrants containing the reflector, i.e. it scans over $\theta_i \in [0,\pi]$.

For a given $s \in S$, where S is the set of all camera positions, $S=\{s_1,s_2, \ldots s_N\}$, the angle of camera line-of-sight lies between: $0° < \theta < 180°$, measured w.r.t. the positive x-direction.

By the nature of reflectance of specular surfaces, from a particular combination of camera position, s_i, and viewing angle, θ_i, a maximum of one specular point is observed. Then, each such combination, represented by the ordered pair (s_i,θ_i), completely defines the ray containing the specular point. This combination (s_i,θ_i), not s_i alone, will be referred to as the "i^{th} camera position".

1.2.3 Creation of Simulated Reflectance Images

The following algorithm was used for the creation of the simulated reflectance images:

1)Enter shape information of the reflector

2)Precompute normals at each point in reflector

3)For each camera position $s \in S$,

For each orientation of camera line of sight $0° < \theta < 180°$

3.1 Use ray tracing to locate reflector point, if it exists, lying on line of sight

3.2 Look up angle of precomputed normal, α_n for the located reflector point

3.3 if (($|\theta - \alpha_n| \leq$ a threshold) or (the reflector point is a corner))

Mark the point as specular or visible to the camera: Assign value 255 to pixel at row=s and column=θ (scaled)

This algorithm simulates the reflectance pattern of a perfect specular reflector. Reconstructing the shape of the reflector, given the reflectance pattern in the form of images (like the one shown in Figure 1.2), is the problem addressed by this thesis.

1.2.4 Simulation Results

On carrying out the simulation of reflectance images, a set of S_N images is obtained, each of which contains the specularities visible from the corresponding camera positions, s. This information is organised into a matrix structure:

$$\begin{bmatrix} e_{(s_1,\theta_1)} & e_{(s_1,\theta_2)} & \cdots & e_{(s_1,\theta_n)} \\ e_{(s_2,\theta_1)} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ e_{(s_m,\theta_1)} & \cdots & \cdots & e_{(s_m,\theta_n)} \end{bmatrix}$$

where $e_{(s_m,\theta_n)}$ denotes element corresponding to angle θ_n at position s_m

$e_{(s_m,\theta_n)}$ =1, if a specularity is observed along ray defined by s_m and θ_n

= 0, if no specularity is observed along ray defined by s_m and θ_n

This information is stored as an image, with the rows and columns corresponding to s-values and θ-values respectively, both in ascending order. As an illustration, the image representation of the reflectance information for the reflector of Figure 1.1 is shown in Figure 1.2.

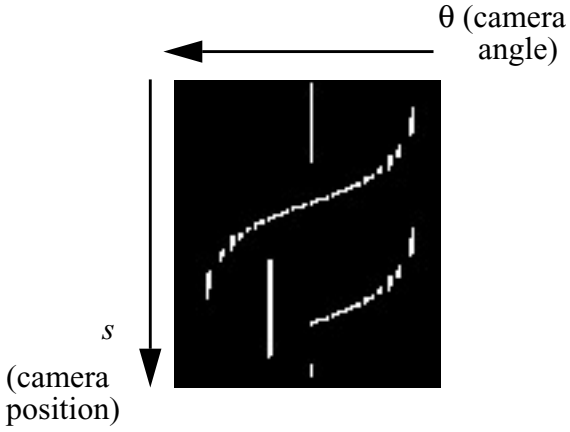


Figure 1.2 Image representation of reflectance information

1.2.5 Mapping of Special Surface Features into Reflectance Image Domain

A relationship was found between certain surface features and their manifestation in the reflectance image.

Corners

In reality, perfect corners do not exist - corners on physical objects are rounded at the microscopic level. Since a corner can be approximated as a microscopic cylinder(radius, $R \to 0$. R is less than one pixel), regardless of the position it is

viewed from, the line of sight coincides with the normal. Due to this reason, in the case of a specular reflector, a corner appears bright independent of the angle from which it is viewed. This is illustrated in Figure 1.3.

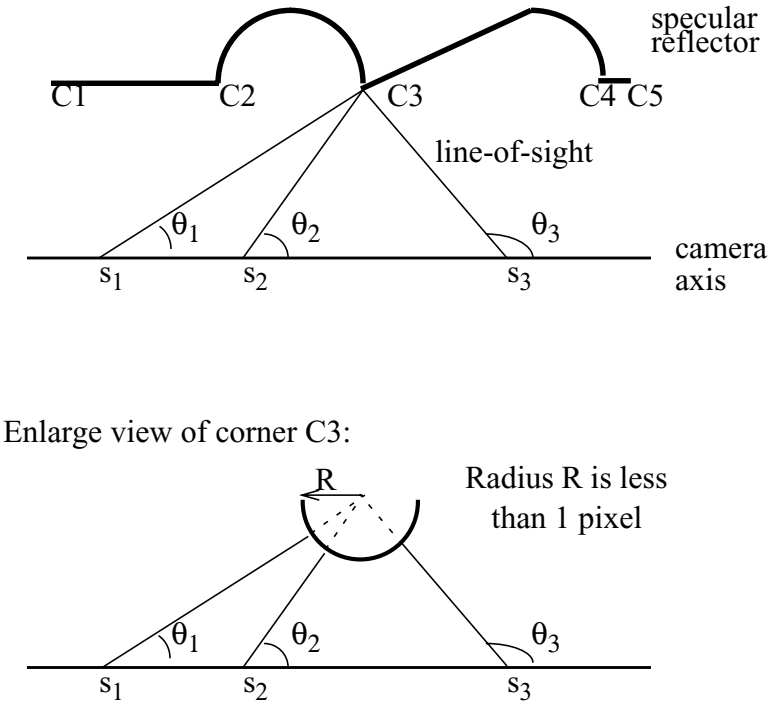


Figure 1.3 Corners in specular reflectors

A particular corner appears as a specular point, for all values of *s*, with the value of θ changing gradually with changes in *s*. In the reflectance image, the reflectance information corresponding to a corner is represented by a bright curve stretching from the zeroth row (s=0) to the last row (s=S). This reflectance pattern of the corners for the reflector in Figure 1.3 is presented in Figure 1.4. The reflector has 5 corners and so the reflectance image contains 5 distinct curves corresponding to each of these corners. Corners C4 and C5 are very close together; this manifests itself in the overlap between their corresponding curves.

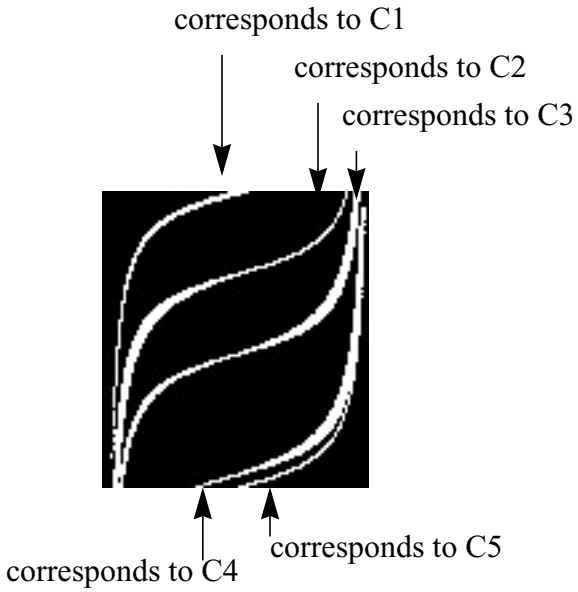


Figure 1.4 Reflectance pattern of corners

For a particular s (row), the column containing the bright pixel corresponding to a particular corner, (x_j,z_j), is derived from the line-of-sight orientation, θ, given by:

$$\theta = \operatorname{atan}\frac{(z_l + z_j)}{(x_j - s)} \tag{1.1}$$

The position of the bright pixel varies as the inverse tangent. Since s is the only variable in this equation, the mapping of a corner in the physical domain to the reflectance image domain is that of an inverse tangent function of some transformation of s.

Linear segments

The linear segments of the reflector are labeled as L1, L2 and L3 in Figure 1.5. Let the inclination of a linear segment w.r.t the positive x-axis be α. Then, for a particular point on the segment to be bright, the line-of-sight orientation should be:

$$\theta = \pi/2 + \alpha \tag{1.2}$$

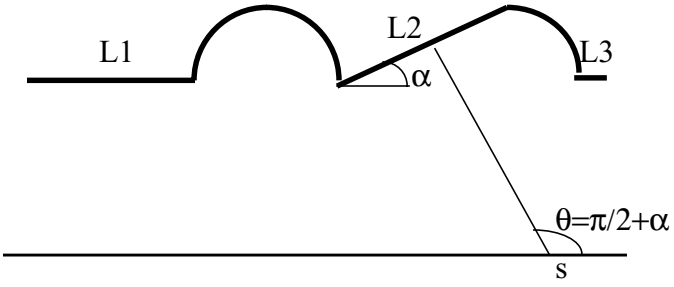


Figure 1.5 Linear segments in specular reflectors

Thus a linear segment in the physical domain maps to a vertical line segement in the reflectance image, located in the column corresponding to $\theta=\alpha$, with length l_r equal to

$$l_r = 2l_p \cos\alpha \tag{1.3}$$

where l_p is the actual length of the linear segment. This mapping is demonstrated in Figure 1.6. There are three line segments in the reflectance image, each corresponding to one of the three linear segments of the reflector.

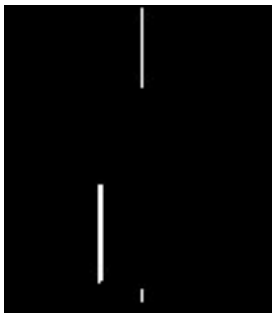


Figure 1.6 Reflectance pattern of linear segments

The special geometry that features such as corners and linear segments produce in the reflectance domain can possibly be employed to constrain the solution of the reflector shape. This will be discussed further in section 3.1.

Chapter 2 Literature Review

Shape recovery is one of the basic problems in computer vision. Techniques for recovering shape from brightness images are referred to as “Shape-from-X”, where X could be shading[9], texture[7], perspective[12] etc. Shape-from-shading is concerned with recovering surface orientation from local variations in measured brightness.The shape-from-shading problem was first formulated in computer vision by B.P.K.Horn [9].

The recovered shape can be represented by the surface normals, the depth of the surface, surface gradient surface slant and surface tilt. The surface normal (n_x,n_y,n_z) at a point is the orientation of the vector normal to the plane tangent to the surface at that point. The surface depth can be interpreted as the distance from the camera to the surface points or the distance of the surface points from the x-y plane. The surface gradient $(p,q)=(\frac{\partial z}{\partial x},\frac{\partial z}{\partial y})$ is the rate of change of depth

in the x and y directions. Surface slant,ϕ, and surface tilt,θ, are related to the surface normal as follows:

$$(n_x, n_y, n_z) = (l\sin(\theta)\cos(\phi), l\sin(\phi)\sin(\theta), l\cos(\phi)) \quad (2.1)$$

where l = the magnitude of the surface normal

2.1 Reflectance Models

Several techniques have been developed to recover the shape of a surface from images. Each of these techniques makes certain assumptions about the optical properties of the surface. Nayar et al. [15] showed the importance of understanding the reflectance properties of a surface before selecting a shape recovery technique. Various reflectance models have been developed for different kinds of surfaces. The most common of these are for Lambertian and specular surfaces.

2.1.1 Lambertian Surfaces

A Lambertian surface is a surface of perfectly matte properties, which means that it adheres to Lambert's cosine law[13]. Lambert's Cosine Law holds that the radiation per unit solid angle (the radiant intensity) from a flat surface, in a particular direction, varies with the cosine of the angle between that direction and the surface normal.

So, a Lambertian surface can be modeled as

$$I_L = A\rho\cos(\theta_s - \theta_n) \tag{2.2}$$

where A = strength of the light source

ρ = albedo of the surface

θ_s = the source direction

θ_n = the surface orientation

$\cos(\theta_s - \theta_n)$ represents the foreshortened area

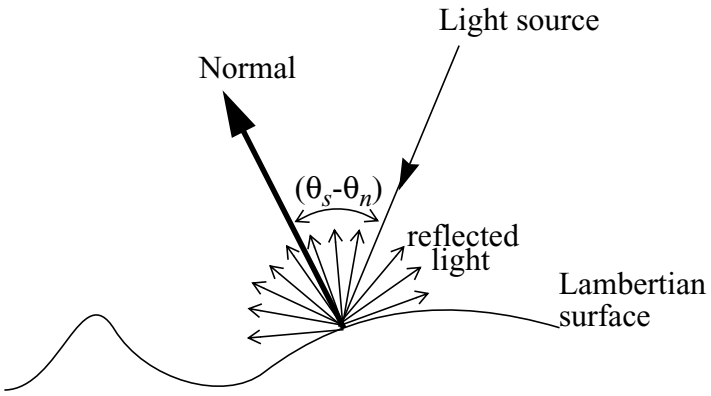


Figure 2.1 Lambertian surface geometry

As shown in Figure 2.1, for Lambertian surfaces, the reflected intensity is independent of the viewing direction (in the figure, intensity is indicated by the length of the rays denoting reflected light). This is a direct consequence of Lambert's law: Let θ_o be the angle between the surface normal and direction of observation. Although the emitted radiation per unit area falls off with cos θ_o according to Lambert's Law, the observed surface area is growing at the same rate. Figure 2.2 displays these two phenomena which act in tandem to produce the result that the radiance of a Lambertian surface is constant with respect to θ_o.

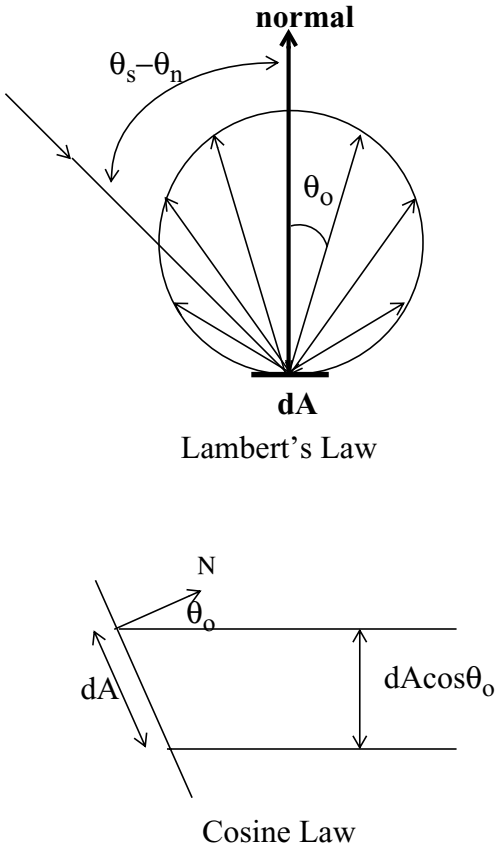


Figure 2.2 Effect of Lambert's Law and Cosine Law

The intensity does however depend on the orientation of the light source relative to the surface normal, as expressed in the equation (2.2).

2.1.2 Specular Surfaces

A specular surface is one which reflects light in a directional manner, where the angle of the reflected ray with respect to the surface normal at the point is equal to the angle between the normal and the incident ray. Thus, the normal, the incident ray and the reflected ray lie in the same plane. There are two characteristics associated with specular reflectors: the specular spike and the specular lobe. The *specular spike* is zero everywhere except in a very narrow range around the direction of specular reflection. The *specular lobe* spreads around the specular direction.

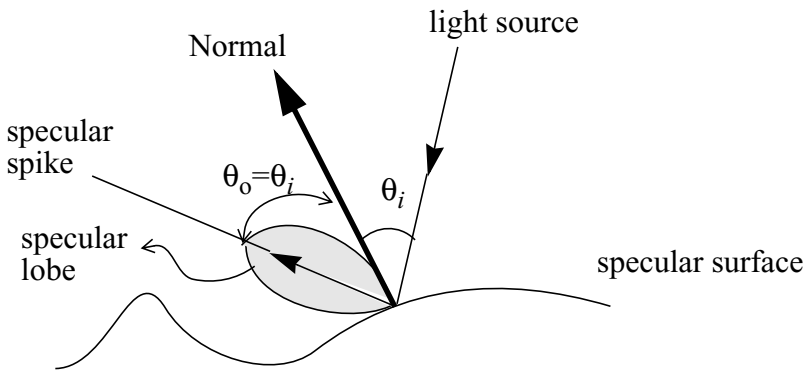


Figure 2.3 Specular surface geometry

Phong [19] presented a model for specular reflection, in which the specular component of reflection is represented as powers of the cosine of the angle between the specular direction and the viewing direction. Phong observed that for very shiny surfaces the specular highlight was small and the intensity fell off rapidly, while for duller surfaces it was larger and fell off more slowly.

$$I_S = W(i)[\cos(s)]^n \tag{2.3}$$

where, *W(i)*= function giving ratio of specular reflected light and incident light as a function of the incident angle *i*.

s = angle between direction of reflected light and the line of sight

n = power which models the specular reflected light for each material.

The function *W(i)* and the power *n* express the specular reflection characteristics of a material. For a highly reflective material, the values of both *W(i)* and *n* are large.

However, although Phong's model describes observed brightness, it does not have a physical interpretation.

Nayar et al. [16] explored the following two models based on physical optics and geometrical optics respectively. By studying the reflectance curves predicted by the two models, they proposed the reflectance framework described in 2.1.3.

Beckmann and Spizzichino[1] proposed a model based on physical optics theory. The surface height is modeled as a continuous stationary random process with mean value zero and standard deviation σ_k that represents the physical roughness of the surface. Maxwell's equations are used to determine how incident light waves are scattered by the surface in various directions. Beckmann and Spizzichino found that the spike component is mathematically represented by the square of a very sharp sinc function that tends to be symmetric with respect to the specular angle $\theta_s = 2\theta_n$. For simplicity, the spike component can be denoted by the delta function:

$$I_S = B\delta(\theta_s - \theta_n) \tag{2.4}$$

where, θ_s= source direction

θ_n= surface orientation direction

Torrance and Sparrow [23], in their geometrical optics model assumed that a surface is composed of small, randomly oriented, mirror-like facets. Their model describes 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. Based on the Torrance-Sparrow model, Healey and Binford [8] derived the following simplified model, which can be used to model the distribution of specular intensities for regions of high surface curvature:

$$I_S = K \exp\left(-\frac{\alpha^2}{\sigma_\alpha^2}\right) \tag{2.5}$$

where α = slope of individual micro-facets

σ_α = roughness parameter

K = constant

2.1.3 Hybrid Sufaces

Most real-world surfaces are neither purely specular nor purely Lambertian, but possess properties which are a combination of both. Nayar et al. [16] proposed a reflectance model consisting of three components: the diffuse lobe, the spec-

ular lobe and the specular spike. They used the Lambertian model for the diffuse lobe, the Torrance-Sparrow model for the specular lobe and the Beckmann Spizzichino model for the specular spike, and came up with the following model for hybrid surfaces:

$$I = K_{dl}\cos(\theta_i) + K_{sl}\exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2}\right) + K_{ss}\delta(\theta_i - \theta_r)\delta(\phi_r) \tag{2.6}$$

where K_{dl}, K_{sl}, K_{ss}= strengths of diffuse lobe, specular lobe & specular spike components respectively.

θ_i = direction of incident light

(θ_i,ϕ_i)= direction of reflected light in terms of slant and tilt in 3D

2.2 Shape Recovery for Specular Reflectors

Two major issues are associated with specular reflection. The first is detection of specularity - determining whether an image feature corresponds to an acual scene point or whether it is the specular reflection of another scene point. The second problem is shape recovery of specular surfaces. It is possible that specular highlights may be misidentified as curved surface features such as ridges and domes. In other words, even with a known light source, the shape cannot be uniquely recovered in general.

Since specular highlights strictly constrain imaging parameters such light source direction, surface normal and viewing direction, if accurately located, they can provide important cues for the recovery of surface shape. However, highlights also introduce problems for algorithms requiring correspondence between images such as stereo or motion. Highlights present in one image may be missing from other images due to their great dependence on imaging geometry[6]. Specular highlights also pose problems for edge detection algorithms, since edge detectors respond not only to real edges, but also to highlights.

In a comprehensive treatment of specular shape-from-shading Brelstaff and Blake [2],[3] have analysed the geometric constraints provided by specularities, and have shown how to detect specularities using Lambertian irradiance constraints. Ikeuchi [11] has employed photometric stereo by locating three light sources in different positions and taking three images of specular objects and then determining surface orientations from image triples at each point using photometric stereo. Nayar, Ikeuchi and Kanade[16] extended this method by using a sphere illuminated with many point light sources located around it. Their method determines surface orientation and reflection parameters.

Park and Tou presented a normal vector equalization method for separating probable highlights and obtaining surface orientation of 3-dimensional specular objects[18]. Based on the Torrance-Sparrow model, their method uses a set of monochromatic images taken from different illumination directions. They then separate probable hightlights and generate separate sets of Lambertian and specular images. Surface orientations are then obtained using the Lambertian images by solving an image equation for a Lambertian Surface. The information from highlights is then used to augment the knowledge of surface vector normals.

Franke and Snyder [5] proved a theorem stating that the position and orientation of a specular reflecting object may be determined using a single camera and an unconstrained light source. The strategy is a variation on radiometric stereopsis. However, instead of three pictures, they take a large (theoretically infinite) number of pictures as the camera and light source are moved along the x-axis. Each picture has several specular peaks. The part pose is determined from the changes in the location of the peaks and knowledge of the surface function. They find the position and orientation of a known point on the surface. Given this and the equation of the surface, the part pose is determined.

In this thesis, a similar experiment is carried out: as a camera and mounted light source are moved along the x-axis, several images are taken of the specular reflector. Through only the behaviour of the specularities in the image plane, the shape of the reflector is recovered.

In conclusion, while specularities provide important geometric constraints for shape from shading, their reliable modelling and detection is a difficult and elusive problem.

Chapter 3 Approach

As described in section 1.2.4, a matrix structure (an image for the 2-D case) is used to store the reflectance information captured by the camera. The task of reconstructing the surface from this reflectance information is accomplished by finding the set of points (x,z) that would give rise to specularities visible from the set of camera positions and orientations, (s,θ).

3.1 Extracting information from Special Surface Features: Corners

As discussed in section 1.2.5, a corner is a special kind of surface feature that appears specular regardless of observer position. This property of corners can be used to solve for their coordinates. Once the corners of a reflector are known (assuming they exist), they can be used as seed elevation points to be fed into the shape recovery algorithm.

For a sharp corner located at (x,z), both x and z are unknown. If this point is observed from position s, at angle θ, we know that the parameters s,θ and variables x,z are related by:

$$z_j = (x_j - s_i)\tan\theta_i \tag{3.1}$$

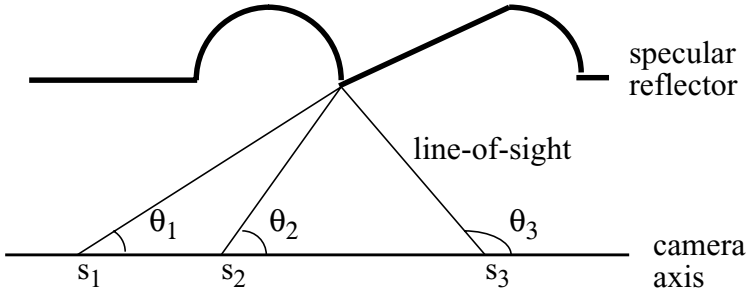


Figure 3.1 Corners in specular reflectors

Since the same corner is visible from all s along the camera line, a Hough Transform approach can be used to solve for the x and z coordinates of a corner point.

3.1.1 About the Hough Transform:

The Hough Transform, a special kind of parametric transform, was introduced as a point-to-line transformation by P.Hough in 1962 [10]. It was further developed into a point-to-curve transformation by Duda and Hart [4]. The Hough transform is a technique that can be used to isolate features of a particular shape within an image. This approach requires that the object being searched in the image be described by a mathematical expression, which in turn is represented by a set of parameters. Due to this, the *Classical Hough Transform* is most commonly used for the detection of regular curves such as lines, circles, ellipses etc. A *generalized Hough Transform* can be employed in applications where a simple analytical description of a feature is not possible.

Hough Transform for finding Straight Lines:

Consider an isolated edge point (x,y). There could be an infinite number of lines that pass through this point, each of which can be defined as the solution to the equation:

$$y = mx + b \tag{3.2}$$

where m= the slope of the line

b = the y-intercept of the line

So, each of the possible lines passing through point (x,y) of the image space, can be characterized as having coordinates (m,b) in some slope-intercept space (parameter space). In fact, for all the lines passing through a given point, (x,y), there is a unique value of b for m such that:

$$b = y - mx \tag{3.3}$$

From Equation (3.3) we see that the set of (m,b) values corresponding to all possible lines passing through the point (x,y), forms a line in (m,b) space. So, every point in image space (x,y) corresponds to a line in parameter space, (m,b), and each point in (m,b) space corresponds to a line in image space, (x,y), as shown in Figure 3.2.

The Hough Transform works by letting each feature point (x,y) 'vote' in (m,b) space for each possible line passing through it. These votes are totaled in an 'accumulator'. Suppose that a particular (m,b) has one vote - this means that there is one feature point through which the line characterized by that particular (m,b) passes. An (m,b) with n votes would mean that n feature points lie on that line.

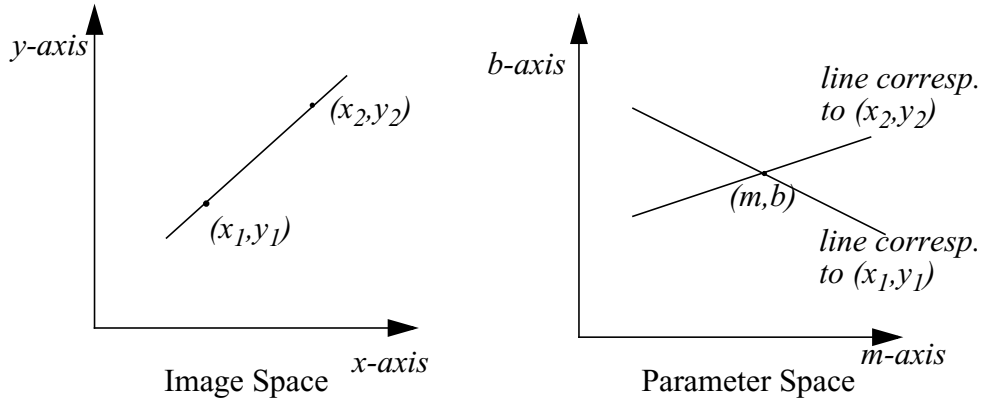


Figure 3.2 Hough Transform for detection of straight lines

3.1.2 Applying Hough Transforms to find corners:

Since a particular corner, (x_j, z_j) can be seen from all s on the camera axis, there are a number of lines-of-sight, completely defined by (s_i,θ_i), that pass through the corner point. The relationship between (x_j, z_j) and (s_i,θ_i) is given by:

$$z_j = (x_j - s_i)\tan\theta_i \tag{3.4}$$

Henceforth, the line-of-sight defined by (s_i,θ_i) will be referred to as $L(s_i,\theta_i)$.

Since the lines-of-sight corresponding to a corner will intersect at the corner alone, just as the Hough Transform is used to find an (m,b) pair, in the parameter space, corresponding to a line in the image space, a similar approach can be used to locate, in the physical domain, the corner (x,z) corresponding a set of (s,θ)'s in the reflectance domain. This application differs from the original Hough Transform in that no transformation of parameters is carried out. However, the principle of 'voting' for points in an accumulator is the same.

The following algorithm based on the Hough Transform was used to locate corners of the specular reflector:

1) Create an all black accumulator image of same size as reflectance image.

2) Count the bright pixels in the reflectance image. Each bright pixel represents an (s,θ) from which a specularity is visible.

3) For every bright pixel (s,θ) in the reflectance image

   For every accumulator pixel i.e. (x,z) coordinate

   If $(x,z) \in L(s,\theta)$

   Increment brightness of neighborhood of accumulator pixel corresponding to (x,z)

4) Threshold the accumulator image. Pixels with brightness above the threshold represent the corner points.

5) Obtain coordinates of the each of corners (x_j,z_j), by setting x_j equal to the column and z_j equal to row of bright accumulator pixel.

The performance of this algorithm will be discussed through simulations and experimentally in section 6.1.

3.2 Specular Shape Recovery Approaches

This section details the preliminary work done in exploring specular shape recovery approaches. While the approaches detailed here have some inherent drawbacks which render them unsuitable except in special cases, they provided the groundwork for the two shape-recovery algorithms presented in Chapter 4 and Chapter 5.

3.2.1 Optimization Approach

In performing shape recovery through optimization, an attempt was made to solve for the reflector shape by finding the set of ordered pairs $(x,z)_i$ which satisfy equation (3.5), provided

$$\frac{z_i}{x_i - s_i} - \tan\theta_i = 0 \tag{3.5}$$

Here, $(x, z)_i \leftrightarrow (s, \theta)_i$. Also, $z_i \in (x, z)_i$ and $x_i \in (x, z)_i$.

This is accomplished by setting up an optimization problem. The above equation represents the reflectivity constraint for the surface. The minimal curve which satisfies this reflectivity constraint can be obtained by minimizing the objective function of equation (3.6) subject to the constraints shown in (3.7).

$$E_L = \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 + (z_i - z_{i+1})^2 \tag{3.6}$$

where n = total number of points on the reflector

$$\lambda_i \left(\frac{z_i}{x_i - s_i} - \tan\theta_i \right) \tag{3.7}$$

where λ_i= lagrange multiplier

The problem with performing a constrained minimization is that each point in the solution is constrained to precisely satisfy the measurements. This is not a viable technique in the presence of noise and measurement errors. So, instead of a constrained minimization, the objective function of equation (3.8) is introduced in addition to E_L.

$$E_n = \sum_{i=1}^{n} \left(\frac{z_i}{x_i - s_i} - \tan\theta_i \right)^2 \tag{3.8}$$

This objective function, E_n or noise energy, takes into account the reflectivity of the surface.

Minimizing the sum of the noise energy E_n and the energy E_L will lead to a solution that is constrained by the reflectivity of the surface and also takes into ac-

count neighbourhood relationships. The E_L term in the energy minimization leads to the shortest line solution. Minimizing a different form for E_L will lead to a line with minimal slope change. Both these cases are discussed below.

For a Shortest-Line Solution

The set of points (x_i,z_i) which minimize E_L and E_n simultaneously result in a smooth curve which is consistent with the measurements. Gradient descent may be used to minimize the objective function $E_I = E_L + E_n$, resulting in the x and z coordinates of the n surface points. The derivatives for E_L and E_n are given below. In the following equations, the substitution $Q_i \equiv \tan\theta_i$ is used for notational convenience.

Derivatives of E_n with respect to x_k and z_k respectively:

$$\frac{\partial E_n}{\partial x_k} = 2\left(\frac{z_k}{x_k - s_k} - Q_k\right)\left(\frac{-z_k}{(x_k - s_k)^2}\right) \tag{3.9}$$

and

$$\frac{\partial E_n}{\partial z_k} = 2\left(\frac{z_k}{x_k - s_k} - Q_k\right)\left(\frac{1}{x_k - s_k}\right) \tag{3.10}$$

Similarly derivatives of E_L with respect to x_k and z_k are:

$$\frac{\partial E_L}{\partial x_k} = 2(x_k - x_{k+1}) - 2(x_{k-1} - x_k) \tag{3.11}$$

and

$$\frac{\partial E_L}{\partial z_k} = 2(z_k - z_{k+1}) - 2(z_{k-1} - z_k) \tag{3.12}$$

Now, the surface points are obtained by applying gradient descent separately for the x and z coordinates as follows:

$$\begin{aligned} x_k^m &\Leftarrow x_k^{m-1} - \alpha_L \frac{\partial E_L}{\partial x_k} - \alpha_n \frac{\partial E_n}{\partial x_k} \\ z_k^m &\Leftarrow z_k^{m-1} - \alpha_L \frac{\partial E_L}{\partial z_k} - \alpha_n \frac{\partial E_n}{\partial z_k} \end{aligned} \tag{3.13}$$

For a Smooth-Line Solution

Minimizing the function E_L produces the shortest line solution. An alternate objective function, E_P, which requires a minimal slope change between consecutive surface points, is presented in equation (3.14).

$$E_p = \sum_{i=1}^{n-1} \left[\left(\frac{z_{i+1} - z_i}{x_{i+1} - x_i} \right) - \left(\frac{z_i - z_{i-1}}{x_i - x_{i-1}} \right) \right]^2 . \tag{3.14}$$

Minimizing E_P results in solutions that have smooth lines with minimal slope change.

The derivatives of E_P are presented in equations (3.16) and (3.17) respectively. The variables x_i and z_i occur in three terms of the summation each. Hence, before taking the derivative, the following substitition is performed for notational convenience:

$$\Upsilon_r = \frac{(z_{k+r} - z_{k+r-1})}{(x_{k+r} - x_{k+r-1})} \tag{3.15}$$

Now, the derivatives of E_P with respect to x_k and z_k respectively are:

$$\frac{\partial E_p}{\partial z_k} = 2\Big[(\Upsilon_0 - \Upsilon_1)\Big(\frac{1}{(x_k - x_{k-1})} + \frac{1}{(x_{k+1} - x_k)}\Big) - (\Upsilon_1 - \Upsilon_2)\Big(\frac{1}{x_{k+1} - x_k}\Big) - (\Upsilon_{-1} - \Upsilon_0)\Big(\frac{1}{x_k - x_{k-1}}\Big)\Big] \tag{3.16}$$

and

$$\frac{\partial E_p}{\partial z_k} = 2\Big[-(\Upsilon_0 - \Upsilon_1)\Big(\frac{\Upsilon_0}{(x_k - x_{k-1})} + \frac{\Upsilon_1}{(x_{k+1} - x_k)}\Big) \text{"" } + (\Upsilon_1 - \Upsilon_2)\Big(\frac{\Upsilon_1}{x_{k+1} - x_k}\Big) + (\Upsilon_{-1} - \Upsilon_0)\Big(\frac{\Upsilon_0}{x_k - x_{k-1}}\Big)\Big] \tag{3.17}$$

Algorithm Drawbacks

There is an indexing problem with this algorithm. The objective function E_n requires that the indices of (s,θ) and (x,z) be the same, i.e. the i^{th} measurement of (s,θ) correspond to the i^{th} coordinate pair (x,z). However in reality, it is possible that $(x,z)_i$ may not be adjacent to $(x,z)_{i+1}$, as illustrated below:

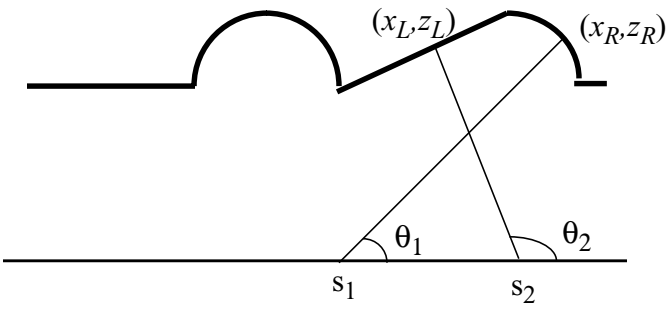


Figure 3.3 Indexing problem in minimization algorithm

In the Figure 3.3, the leftmost coordinate pair, (x_L,z_L) corresponds to the 2^{nd} measurement, (s_2,θ_2) and the rightmost coordinate pair (x_R,z_R) corresponds to the 1^{st} measurement, (s_1,θ_1). Since the (s,θ) are indexed according to the camera

motion (in this case from left to right), there may not always be a correspondence between (s_i,θ_i) and (x_i,z_i) measured.

Class of surfaces solvable by Algorithm

This algorithm can only be applied to surfaces satisfying the following property:

Let v_i *be the reflector point* (x_i,z_i) *and*

c_i *be the camera position* (s_i,θ_i)

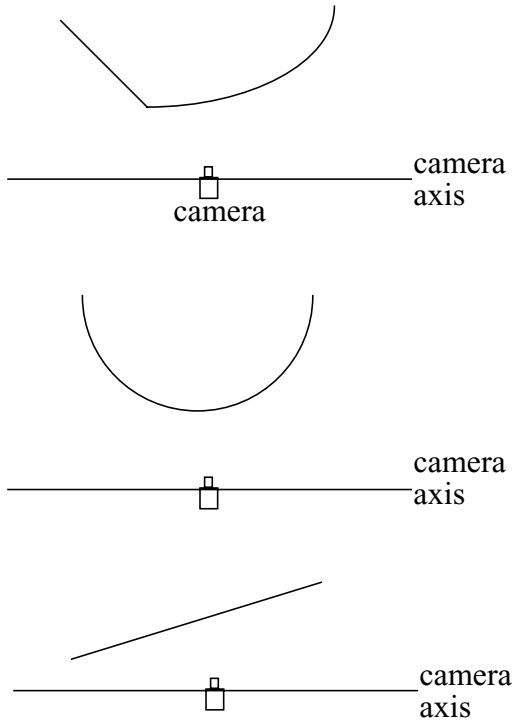
Let $v_i \leftrightarrow c_i$ *denote a correspondence between* v_i *and* c_i*. This algorithm can be applied to those and only those surfaces that satisfy the following property:*

$$v_i \leftrightarrow c_i \Leftrightarrow v_{i+1} \leftrightarrow c_{i+1} \tag{3.18}$$

i.e., The algorithm can be applied to any surface for which there exists a one-to-one and sequential correspondence between reflector points (or (x,z) pairs) and camera positions ((s,θ) pairs). Some examples of such surfaces are presented below.

Sample Surfaces

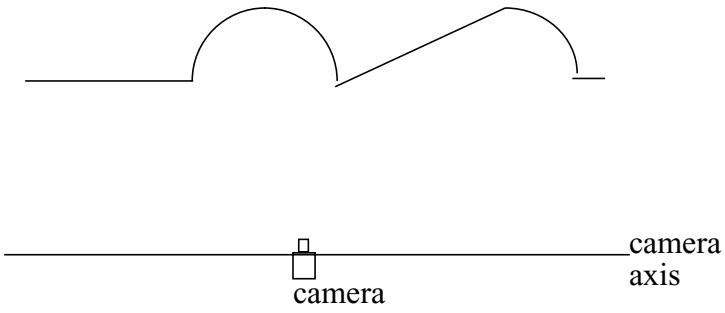
Convex surfaces satisfy the property stated above. Some examples of reflectors which can be solved using the minimization algorithm are given in Figure 3.4.



Linear and Convex surfaces can be solved by the minimization aglortihm.

Figure 3.4 Sample surfaces meeting criterion of equation (3.18)

As shown in Figure 3.4, the reflectors need to be convex w.r.t. the camera axis. An example of a suface that does not satisfy this property and hence cannot be solved by the minimization algorithm is given in Figure 3.5.



This surface cannot be solved by the minimization algorithm since it contains concavities w.r.t the camera line

Figure 3.5 Surface not meeting criterion of equation (3.18)

Summary

While this approach has drawbacks which renders it ineffective in solving a general class of surfaces, it has contributed towards formulating an algorithm (Chapter 4) to accurately reconstruct specular reflectors.

3.2.2 Using Derivatives to Solve for the Surface

For the set-up of Figure 1.1, representing $\tan\theta_i$ by m_i (for slope), equation (1.1) becomes

$$\frac{z_j}{x_j - s_i} = m_i \tag{3.19}$$

Taking the derivative yields:

$$\frac{z_j}{(x_j - s_i)^2} = \frac{dm_i}{ds_i} \tag{3.20}$$

Then applying equation (3.19) to equation (3.20):

$$\frac{(x_j - s_i) \cdot m_i}{(x_j - s_i)^2} = \frac{dm_i}{ds_i} \tag{3.21}$$

Upon rearranging:

$$x_j = m_i \cdot \frac{ds_i}{dm_i} + s_i \tag{3.22}$$

Which appears to yield a solution. However, there are the following problems with this approach:

1. It fails for points at which camera angle is 90 degrees (looking up) since m_i, the slope, is then infinity

2. It fails for linear segments since the $\frac{ds_i}{dm_i}$ will not have a finite value.

Variation: Using a Parametric Transform

As an attempt to solve the above mentioned problems, a parametric transform is employed. Using the (ρ,α) representation of the camera line of sight instead of the (s,θ) involves the following transformations:

$$\alpha_i = |90 - \theta_i| \tag{3.23}$$

and

$$\rho_i = s_i \cdot \cos\alpha_i \tag{3.24}$$

The expression of reflector points (x_j,z_j), then is:

$$\rho_i = x_j \cdot \cos\alpha_i + z_j \cdot \sin\alpha_i \tag{3.25}$$

Differentiation and some algebraic manipulation yields:

$$\frac{d\rho_i}{d\alpha_i} = \frac{\rho_i \cdot \cos\alpha_i - x_k}{\sin\alpha_i} \tag{3.26}$$

From which, the following expression for x is obtained:

$$x_j = \rho_i \cdot \cos\alpha_i + \frac{d\rho_i}{d\alpha_i} \cdot \sin\alpha_i \tag{3.27}$$

Implementing the solution presented in equation (3.27), first for a monotonic surface:

At a given *(s,θ)* it is necessary to find the corresponding *(ρ,α)* and then approximate the derivative using

$$\frac{\Delta\rho}{\Delta\alpha} = \frac{\rho - \rho_{prev}}{\alpha - \alpha_{prev}} \tag{3.28}$$

These estimated derivative values are then substituted into equation (3.27)

There is a problem with this solution: While using the parametric transform tackles the problem of expressing vertical camera lines of sight effectively, it fails for general linear surface segments. This case is illustrated by Figure 3.6.

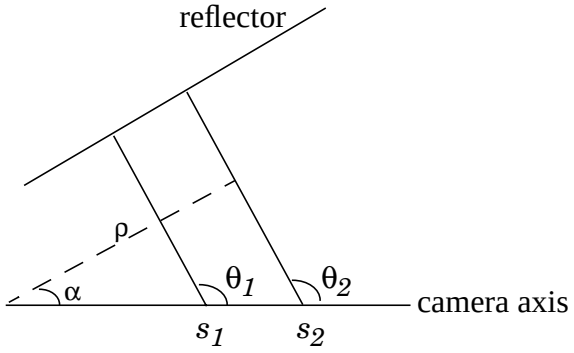


Figure 3.6 Problem in using Parametric Transform to Estimate Derivatives

Thus, since α does not change for linear segments, $\frac{\Delta\rho}{\Delta\alpha}$ cannot be computed. Using $\frac{d\alpha_i}{d\rho_i}$ instead of $\frac{d\rho_i}{d\alpha_i}$ does not yield a suitable solution either.

To detect linear segments separately, the following approach could be employed: A histogram for θ can be obtained by summing the instances of θ across the '*s*' values or the rows. Histogram peaks indicate probable linear segments.

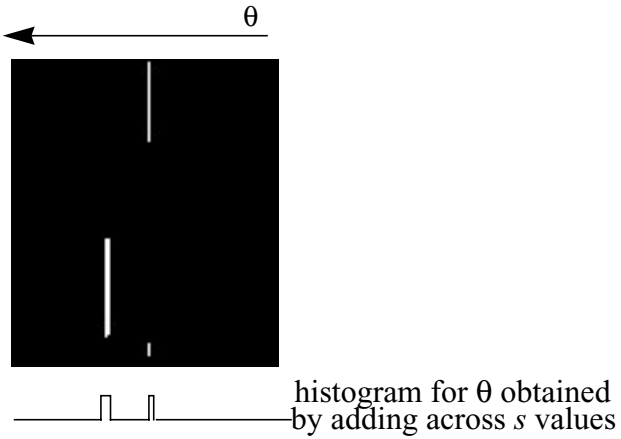


Figure 3.7 Detection of linear segments

Summary

To conclude, the approach of using derivatives to solve for the surface is not applicable to this problem of reconstructing the shape of a specular reflector.

3.3 Proposed Algorithms

Two approaches were developed for solving the inverse problem of extracting reflector shape from the reflectance information.

The first of these is an optimization approach, that determines the surface shape by minimizing an objective function. It differs from the optimization presented in section 3.2.1 in that it meets the reflectivity constraint by a correlation between measured and estimated reflectance images, rather than through minimizing an energy function. This algorithm is described in Chapter 4.

The second approach employs relaxation labeling, to find the reflector point, (x_j,z_j) that is visible from each (s_i,θ_i). The (s,θ)'s are considered to be objects, and all the candidate (x,z) pairs are declared labels. Initially, object-label pairs (labelings) are made with each object having one or more than one label associated with it. Labelings that are found inconsistent with the others are iteratively discarded. In the end, the labelings that are most consistent with each other i.e. the ones with the greatest probability are retained. The solution surface is made up of the (x,z) pairs that are present in the final list of labelings. This ap-

proach is discussed in Chapter 5. It directly recovers the depth of the surface, rather than solving for surface normals.

Chapter 4 Algorithm 1: Shape Recovery through Optimization

4.1 Algorithm Description

As discussed in section 3.2.1, there is an inherent difficulty, known as the indexing problem in solving for the reflector shape by finding the set of ordered pairs (x_i,z_i) which satisfy equation (4.1).

$$\frac{z_i}{x_i - s_i} - \tan\theta_i = 0 \tag{4.1}$$

The problem is caused due to the indexing of the camera positions as (s_i,θ_i), since it then becomes necessary to find a correspondence between the indexing of the camera positions and indexing of the reflector points (x_j,z_j). Instead, another approach to the optimization is presented here.

The camera positions (S,Θ) are not extracted from the input reflectance image.

The reflector points, (x_j,z_j) are initialized to give a reflector of an arbitrary shape. The camera position, (s_j,θ_j), corresponding to each reflector point is approximated from (x_j,z_j) and its neighbours. These estimates of (S,Θ) are expressed in the form of a reflectance map similar to the input reflectance image. Plotting (s_j,θ_j) the reflectance image space is equivalent to a discretization of s and θ values and is achieved as follows:

$$\text{row} = s_j \tag{4.2}$$

$$\text{col} = (N_{col} - 1) - \left(\frac{N_{col} - 1}{\pi}\right) \cdot \theta_j \tag{4.3}$$

where N_{col} = Number of columns in reflectance image

The correlation between the calculated reflectance image and the input reflectance image is computed. The optimization process then involves iteratively perturbing the (X,Z) values by small amounts in order to maximize the correlation between the calculated and input reflectance images. Simulated Annealing is used for the optimization process.

4.1.1 The Objective Function

The objective function being minimized involves the correlation between the calculated and the input reflectance images. The correlation is found to have very low values for most *(X,Z)*s and increases very sharply for *(X,Z)*s faithful to the actual solution. Even *(X,Z)*s close to the actual solution can lead to low correlation values. This makes it difficult for the optimization to converge. In order to make the search space more monotonic, the input image is blurred. This ensures that there is a steady increase in the correlation as the candidate *(X,Z)* values approach the actual solution.

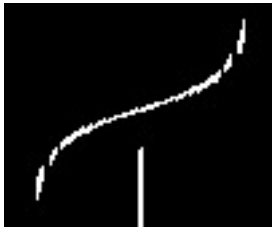
Blurring of Input Reflectance Image

Guassian blurring kernels were experimented with. However, a problem was encoutered with Gaussian blurring. The algorithm requires that the blurred image possess brightness peaks of equal magnitude at all (s_j, θ_j). This is not the case with the output of a Gaussian blur. To counter this, 'manual blur' was employed. If pixel p corresponds to an (s_j, θ_j), and hence has brightness 255 in the original image, then the brightness of pixel q is set at:

$$\text{brightness of pixel q} = 255 - 50 \cdot d(p, q) \tag{4.4}$$

Where $d(p,q)$ = Distance between pixels p and q

This difference is shown in Figure 4.1.



Original Image

Using 5x5 Gaussian blurring kernel

Using manual blur

Figure 4.1 Original and blurred reflectance images

Correlation Energy

If pixel at (s,θ) in the calculated reflectance image is bright (i.e. it corresponds to an observed specularity, (x,z)) AND the pixel at (s,θ) in the blurred input image is non-zero, then let I be the set of pixels located at (s,θ) of the blurred image.

The correlation between the calculated and input reflectance images is calculated as per equation (4.5). The multiplication by 1 denotes that the normalized brightness of the calculated reflectance image is 1.

$$\text{Corr} = \sum_{i \in I} \frac{\text{Brightness of pixel i of blurred image}}{\text{Maximum image brightness}} \times 1 \tag{4.5}$$

Since Simulated Annealing (SA) is traditionally used for minimization, a function of the correlation (such as the inverse) is minimized.

$$E_{corr} = \frac{N_{\text{cam}} \cdot 10}{\text{corr}} - 10 \tag{4.6}$$

where $\text{corr} \neq 0$.

and N_{cam} = number of bright pixels in input reflectance image, i.e., number of camera positions

Using an objective function that involves only the correlation causes the algorithm to converge very slowly. This is because while perturbing the *(X,Z)* values, there is a very wide range over which they can vary. Since Simulated Annealing, allows all possible solutions to be reached (even those that do not lead to a decrease in the objective function), it takes a long time before the effect of candidate *(X,Z)* values on the objective function is examined.

So the following two constraints are imposed on the *(X,Z)* values:

Shortest Line Energy

The *(X,Z)* values are constrained to lead to the shortest line solution. This leads to the following objective function:

$$E_{SL} = (x_i - x_{i+1})^2 + (z_i - z_{i+1})^2 \tag{4.7}$$

Since the difference between two consecutive x values is always 1, the x value term can be excluded from the expression of E_{SL} to give

$$E_{SL} = (z_i - z_{i+1})^2 \tag{4.8}$$

Smoothness Energy

The *(X,Z)* values are also constrained to yield a smooth solution. The smoothness constraint is imposed by the following objective function:

$$E_{SM} = (z_{i-1} - 2 \cdot z_i + z_{i+1})^2 \tag{4.9}$$

The x value term has been excluded here as well, since it always has a value of zero.

The three objective functions in equations (4.6), (4.8) and (4.9) are combined:

$$E_{comb} = E_{\text{corr}} + E_{SL} + E_{SM} \tag{4.10}$$

E_{comb} is the objective function that is minimized using Simulated Annealing.

4.1.2 Calculation of Estimated Camera Positions

To calculate the camera position corresponding to a particular (x_j,z_j), first the slope of the normal passing through that point is estimated. This is done by estimating the slope of the tangent through that point, mt_j as the average of the slopes of the two line-segments, mt_{j+1} and mt_{j-1} connecting (x_j,z_j) to its two neighbours. The normal slope, m_j is obtained from the tangent slope and used in equation (4.1) to approximate the camera position (s_j,θ_j).

$$s_j = x_j - \frac{z_j}{m_j} \tag{4.11}$$

4.1.3 Choice of Candidate Solutions

Each candidate solution is chosen as follows: an index value j is randomly chosen, and the associated z-value, z_j is perturbed by a random amount.

4.1.4 Algorithm - Shape Recovery through Optimization (SA)

1) Blur the input reflectance image as discussed in 4.1.1

2) Calculate the coordinates of the corners of the reflector using the algorithm to Extract corners using the Hough Transform, presented in section 3.1.2.

3) Initialize the reflector points (x_j,z_j) to straight lines between corners. The number of reflector points is taken to be the same as the number of bright points in the input reflectance image, i.e., the number of camera positions.

4) Compute the energy, E_{old}, for this set of (X,Z) values as per equation (4.10).

5) Set $T = T_{initial}$

6) While $T > T_{final}$

6.1 Call subroutine *purb* to slightly perturb (X,Z) to give a candidate set, (X_{can},Z_{can}), as described in section 4.1.3.

6.2 Compute the energy, E_{new}, for (X_{can},Z_{can})

6.3 Calculate $\Delta E = E_{new} - E_{old}$

6.4 If $\Delta E < 0$

Set $X = X_{can}$ and $Z = Z_{can}$

Go to 6.1

6.5 Calculate $V = \exp(-\Delta E / T)$

6.6 If V>(a randomly produced number between 0 and 1)

Set $X = X_{can}$ and $Z = Z_{can}$

Go to Label

6.7 Decrease the value of T by a small amount, $T = \alpha \cdot T$ where through experimentation, α has been chosen to be a number in the range [0.8,1)

6.8 Go to 6.1

4.2 Algorithm Variations

A variation in the objective function is introduced. Instead of the correlation objective function of equation (4.6), the sum of squared differences may be minimized. In this case,

$$E_{corr} = \sum_{i \in I} \left| 1 - \frac{\text{brightness of pixel i of blurred image}}{\text{maximum brightness of image}} \right| \tag{4.12}$$

The combined energy function, then, is as per equation (4.10). The performance of the algorithm with this variation is detailed in section 6.1

4.3 Algorithm Drawbacks

4.3.1 Requires Corner Points to be Input

As discussed in section 4.1, the algorithm requires seed points to constrain the solution and choice of candidate solutions. Corners have been identified as the seed points to be fed into the algorithm. This is vital to the functioning of the algorithm, and hence it can reconstruct only those surfaces that possess sharp, specular corners.

4.3.2 Requires Length of Reflector to be known

The length of the reflector is an input to the algorithm. The reflectors discussed in this thesis have corners as end points. Hence, once the coordinates of the corners have been extracted as per section 3.1, they can be used to calculate the length of the reflector. However, for reflectors not ending in corners, the length of the reflector will have to be declared.

Chapter 5 Algorithm 2: Shape Recovery through Relaxation Labeling

5.1 Algorithm Description

This algorithm works by assigning labels to objects given some relational constraints. It links the (x,z) pairs to the (s,θ) pairs, iteratively evaluating the probability of each such link based on certain criteria and eliminating the links with low probability. This process is continued till the algorithm reaches convergence. This algorithm produces a set of (x,z) pairs, each pair corresponding to a camera location, (s,θ).

The linking of (x,z)'s and (s,θ)'s with each other and the evaluation of probabilities is done in accordance with Relaxation Labeling. Section 5.1.1 gives an overview of the theory behind Relaxation Labeling.

5.1.1 Relaxation Labeling - Theoretical Background

The problem of assigning labels to objects given some relational constraints occurs quite often in the field of pattern recognition. Of the solutions proposed, *Relaxation Labeling* is a particularly successful one. It was first proposed by Rosenfeld et al in 1976 [21].

Let $\{a_1, a_2, ... a_m\}$ be the set of m objects

and $\{\lambda, \mu, ...\}$ be the set of n labels

The pairing of an object with a label forms a labeling and will henceforth be denoted by $[a_i,\lambda]$.

$p_i(\lambda)$ is the *confidence* that object a_i has label λ. It is a measure of the extent to which the labeling $[a_i,\lambda]$ is accepted and can take on any value between 0 and 1. It is also referred to as the 'probability' of $[a_i,\lambda]$; however, it is not strictly a probability since there is no random process involved.

During the initialization, object-label pairs, referred to as *labelings*, are made and each labeling is assigned a certain probability. These probabilities are iteratively updated using the revision formula shown in equation (5.1) for the $k+1^{th}$ iteration, using values from the k^{th} iteration.

$$p_i^{k+1}(\lambda) = \frac{p_i^k(\lambda)[1 + q_i^k(\lambda)]}{\sum_{\mu} p_i^k(\mu)[1 + q_i^k(\mu)]} \tag{5.1}$$

where $q_i(\lambda)$ is the *consistency*, a measure of how consistent labeling $[a_i,\lambda]$ is with other labelings.

$q_i(\lambda)$ can take on positive and negative values. A negative value of $q_i(\lambda)$ implies that the labeling $[a_i,\lambda]$ is incompatible with most other labelings. $q_i(\lambda)$ is evaluated as shown in equation (5.2).

$$q_i^k(\lambda) = \sum_j C_{ij}\left[\sum_{\mu} r_{ij}(\lambda,\mu)p_j^k(\mu)\right] \tag{5.2}$$

where $r_{ij}(\lambda,\mu)$ is the *compatibility* of labeling $[a_i,\lambda]$ with labeling $[a_j,\mu]$. $r_{ij}(\lambda,\mu)$ can take on any value, v, in the range [-1,1].

$r_{ij}(\lambda,\mu) = 1 \Rightarrow [a_i,\lambda]$ and $[a_j,\mu]$ are completely consistent with each other.

$r_{ij}(\lambda,\mu) = -1 \Rightarrow [a_i,\lambda]$ and $[a_j,\mu]$ are completely inconsistent with each other.

$r_{ij}(\lambda,\mu) = 0 \Rightarrow [a_i,\lambda]$ and $[a_j,\mu]$ do not affect each other.

C_{ij} weights the influence that object j has on object i. It can be incorporated into the compatibility function.

5.1.2 Application of RL to this problem

Let S be the set of camera positions and Θ be the set of camera orientations.

Then, for a given $s_i \in S$ and $\theta_i \in \Theta$, $(s,\theta)_i$ is an object, i.

The (x,z) coordinate pairs within a certain user-defined rectangle are the candidate reflector points, $(x,z)_j$. These constitute the labels.

Labelings are defined as object-label pairs, $[(s,\theta)_i,(x,z)_j]$, such that $(x,z)_j \in L(s,\theta)_i$, where $L(s,\theta)_i$ is the line intersecting the camera line at s_i and having slope equal to $\tan\theta_i$.

Labeling $[(s,\theta)_i,(x,z)_j]$ will also be referred to as labeling $[i,j]$ for short.

Calculation of compatibility

The compatibility between two labelings $[i,j]$ (i.e., $[(s_i,\theta_i),(x_j,z_j)]$) and $[k,l]$ (i.e. $[(s_k,\theta_k),(x_l,z_l)]$) is referred to as $r([i,j],[k,l])$ and calculated as per the following:

- Two labelings both involving corners are always compatible with each other.
- If one labeling contains a corner and the second labeling contains a non-corner point, (x_j,z_j)
  - if the orientation of the line segment connecting the corner and the point is consistent with the input reflectivity information, the labelings are compatible
  - else the labelings are neither compatible nor incompatible.
- If neither labeling contains a corner
  - if the labelings they lie on same $L(s_i,\theta_i)$ they are incompatible
  - else
    - if they are neighbours , they're compatible according to the extent to which their orientation agrees with input reflectivity information.
    - else, they are neither compatible nor incompatible.

These considerations can expressed in the form of the following equations:

$r([i,j],[k,l])$ = 1, if $i=k$ and $j=l$

= 1, if j and l are both corners points

= $\cos(N(j,l)-\theta_l)$, if j is a corner, l is a non-corner point & j is in the neighborhood of l

= $\cos(N(j,l)-\theta_j)$, if l is a corner, j is a non-corner point & l is in the neighborhood of j

= -1, if neither j nor l are corners and if they lie on the $i=k$

= $\cos\left(N(j,l)-\frac{\theta_j+\theta_l}{2}\right)$, if neither j nor l are corners, but they are neighbours and $i \neq k$.

= 0, otherwise.

Where $N(j,l)$ is the angle of the normal to the line-segment connecting points j and l.

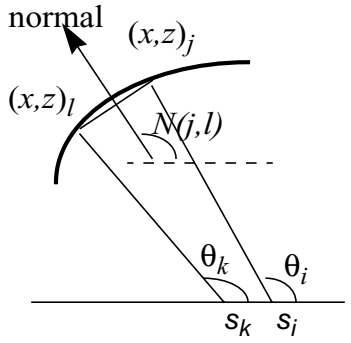


Figure 5.1 Compatibility calculation for 2 non-corner points

Updating Probabilities

$P[(s_i,\theta_i),(x_j,z_j)]$ denotes the probability of labeling $[(s_i,\theta_i),(x_j,z_j)]$. The initial probability assignments are made as per:

$$P([(s_i,\theta_i),(x_j,z_j)]) = \frac{1}{n_i} \tag{5.3}$$

where n_i= the number of labels assigned to object i

The probabilities are then updated iteratively as per the revision formula in equation (5.1). Each time, the probabilities are normalized so that

$$\sum_{j \in L_i} P([(s_i,\theta_i),(x_j,z_j)]) = 1 \tag{5.4}$$

where L_i= the set of labels assigned to object i

In each iteration, labelings whose probabilities are below a certain threshold (i.e. with probability close to 0) are discarded. So the set of labelings shrinks as the algorithm converges. When the algorithm has achieved convergence, some labelings have probabilities close to 1, while the rest have probabilities very close to 0. For each object, the label from the labeling with highest probability is chosen as being the coordinates of the reflector point.

5.1.3 Algorithm - Shape Recovery using Relaxation Labeling

1) Call subroutine *camrays* to extract camera positions (s_i,θ_i) from input reflectance image.

2) Use the Hough Transform approach from section 3.1.2 to calculate the coordinates of corners of the reflector.

3) For every iteration,

If it is not the first iteration,

3.1 If for a particular value of $x_j \in X$, there is no labeling with probability above the threshold,

Retain all the labelings involving *x*.

3.2 If for a particular (s_i,θ_i), there is no labeling with probability above the threshold,

Retain all labelings involving (s_i,θ_i).

3.3 For every labeling,

If the labeling has probability below the threshold,

Delete the labeling from the set of labelings.

3.4 Normalize the probabilities so that the probabilities for all the labels associated with a particular (s_i,θ_i) sum to 1.

3.5 Call subroutine *revision* which updates the probability values.

4) Call the subroutine *maxprobs* which declares the final reflector points.

Description of the subroutines is giving below:

Camrays

1) For every bright point, *i*, in the input reflectance image,

1.1 Calculate s_i and θ_i as per

$$s_i = row$$
$$\theta_i = \pi - (col \cdot \pi)/N_{col}$$

2) For every camera position (s_i,θ_i)

2.1 Calculate all possible labelings $[(s_i,\theta_i),(x_j,z_j)]$

Revision

1) Call subroutine *qvalcal* to calculate the consistency, *q*

2) Call subroutine *denvalcal* to calculate the denominator of equation (5.1)

3) Apply equation (5.1) to compute the probability for each labeling, for the current iteration.

qvalcal

1) For every labeling [*i*,*j*]

1.1 For every labeling [*k*,*l*]

1.1.1 Calculate $r_{[i,j][k,l]}$, the compatibility of labeling [*i*,*j*] with labeling [*k*,*l*], as per 5.1.2

1.2 Calculate $q_{[i,j]}$ as per

$$q^{n}_{[i,j]} = \sum_{[k,l]} r^{n}_{[i,j][k,l]} \times P^{n-1}_{([k,l])}$$

5.2 Algorithm Variations

5.2.1 Selection of Final Reflector Points

The subroutine '*maxprobs*' selects the highest probability labelings that determine the coordinate points of the reconstructed reflector. This can be done in two ways:

1. Selecting maximum probability labelings over each (s_i,θ_i)

For each (s_i,θ_i), the maximum probability labeling is chosen from all labelings associated with (s_i,θ_i). The (x,z)'s in these maximum probability labelings make up the reconstructed reflector. This method has been detailed in the algorithm presented in section 5.1.3. The reconstructed reflectors produced by this method have the same number of coordinate points as the number of camera positions (s_i,θ_i).

2. Selecting maximum probability labelings over each x

For each $x \in X$, the maximum probability labeling is chosen among all labelings containing x. The final points of the reconstructed reflector are the (x,z)'s contained in the set of maximum probability labelings.

For every $x_j \in X$

For every labeling [*i*,*j*]

If the labeling contains x_j

If it is the maximum probability labeling

Assign the x_j and z_j of the labeling to the set of points constituting the re-constructed reflector

This way of selecting the maximum probability labeling has the disadvantage of producing results in which the length of the reflector depends on the length of the user-defined rectangle *X*. Hence, for accuracy, it would require the user to have knowledge of the length of the reflector beforehand.

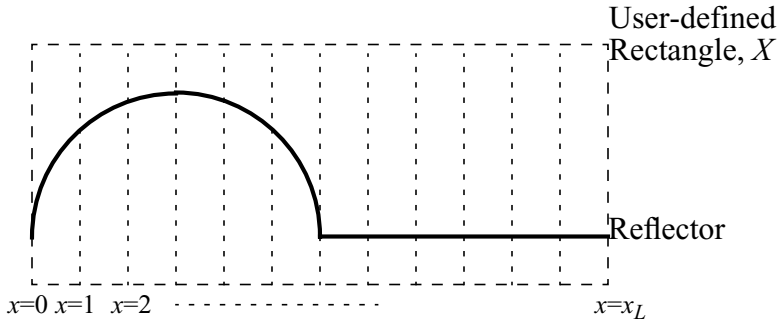


Figure 5.2 Selecting a maximum probability labeling for every *x*

5.3 Algorithm Drawbacks

5.3.1 High Computational Complexity

The program has a computational complexity of $O(2n^2+n)$ where n is the number of labelings. However, the input to the algorithm is the camera position and not the number of labelings. Let N be the number of camera positions. $n = a \cdot N$, i.e. there is an average of a labelings per camera position. The order of complexity therefore is $O(2a^2N^2+aN)$, where N is the number of (s,θ) or inputs. Typical values of a for the relectors dealt with in this thesis are around 100.

5.3.2 Requires Corner Points to be Input

This algorithm requires that the corner points of the surface be known, in order to satisfactorily reconsturct the surface. Hence it has to be used in conjunction with the Corner Extraction Algorithm presented in section 3.1.2.

5.3.3 Requires Length of Reflector to be known

As Discussed in section 4.3.2, for reflectors not ending in corners, the length of the reflector will need to be entered as an input to the algorithm.

Chapter 6 Results and Conclusion

The simulation and experimental results for the surface reconstruction methods proposed in this thesis are presented here. Sections 6.1, 6.2 and 6.3 contain the results for the Corner Extraction algorithm (section 3.1), Optimization algorithm (Chapter 4) and Relaxation Labeling algorithm (Chapter 5) respectively. In each section, the simulation results for two 2-D reflectors and where possible, experimental results for one approximately 2-D reflector are considered. The first simulated reflector is the one presented in Figure 1.1. The second simulated reflector is the simulation of the actual reflector used to obtain experimental data. This enables comparison of the algorithm performance under simulation and experiment conditions. The three reflectors are presented in the figures 6.1 to 6.3.

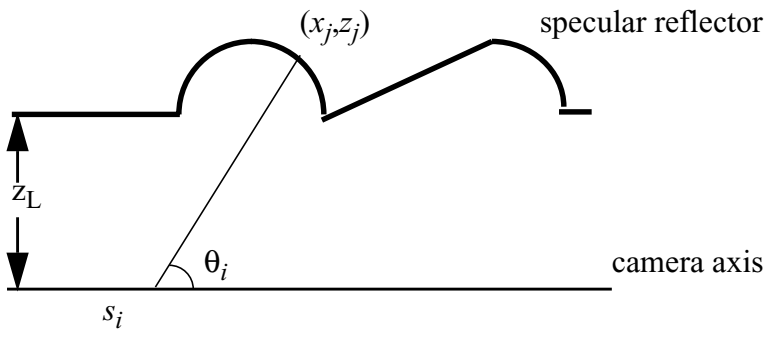


Figure 6.1 Reflector 'S1' - 1st reflector used in Simulations

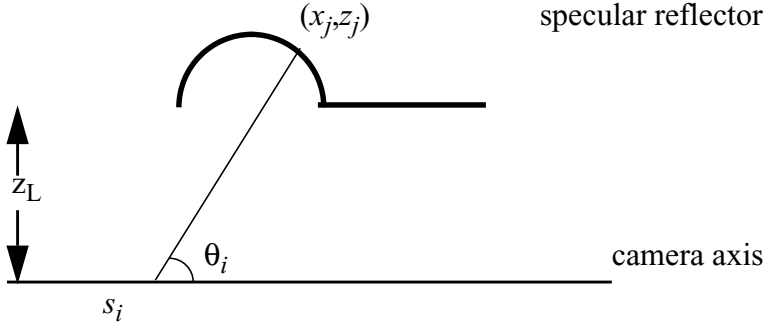


Figure 6.2 Reflector 'S2' - 2nd reflector used in Simulations

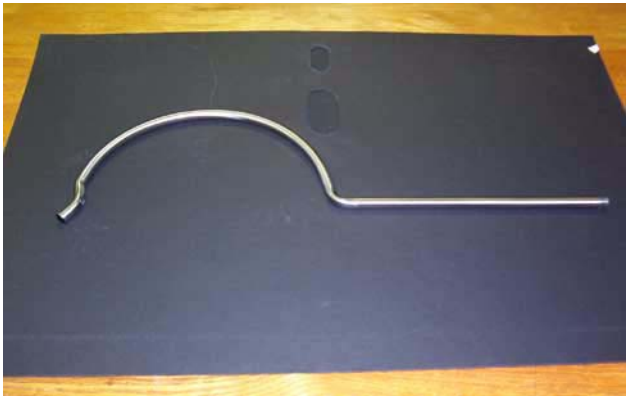


Figure 6.3 Reflector ‘R1’ - reflector used to gather experimental data

6.1 Corner Extraction

Corner extraction is performed by an algorithm modeled after the Hough transform approach, as described in section 3.1.

The accumulator images obtained by running the corner extraction on the reflectance data of the two simulation reflector, S1 and S2 are presented in Figures 6.4 and 6.5. For reflector R1, two experiments were carried out to demonstrate repeatability. The difference in the two experiments was in the distance,z_L, of the camera line from the reflector. In experiment 1, z_L=62 and in experiment 2, z_L=69. The results of the corner extraction algorithm for both sets of experimental data are presented in Figure 6.6 and Figure 6.7 respectively. The bottom row of the accumulator image represents the position of the camera line. z_L, the distance of the camera line from the reflector is also shown. From the accumulator images, the corner coordinates were obtained as follows:

1. The clusters of bright pixels in the region $z>z_L$ were identified.

2. In each cluster, the pixel of maximum brightness was located.

The resulting corner coordinates are marked in each accumulator image as x_c and z_c and the results are summarized in Table 6.1.

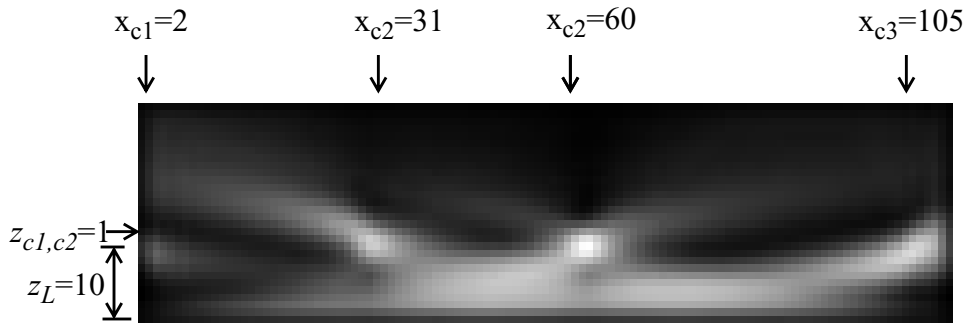


Figure 6.4 Accumulator image for Reflector 'S1'

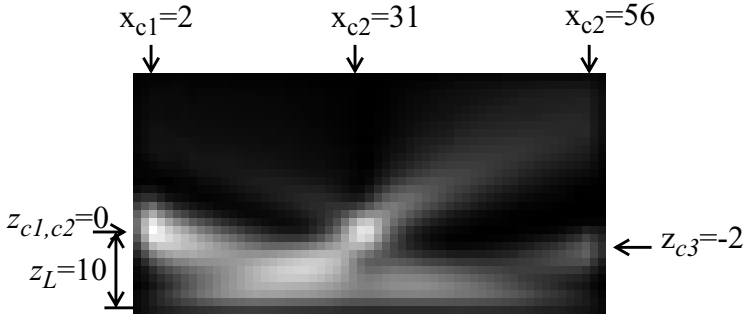


Figure 6.5 Accumulator image for Reflector 'S2'

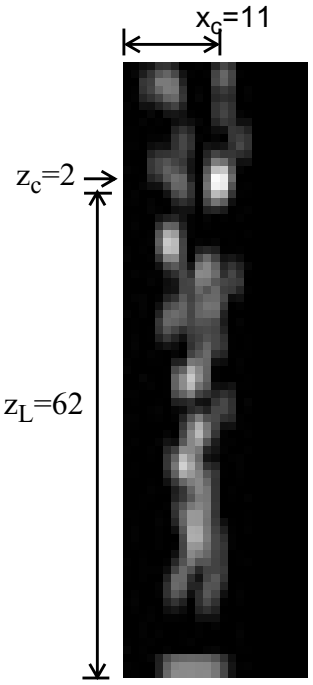


Figure 6.6 Accumulator image for Reflector R1(experiment 1)

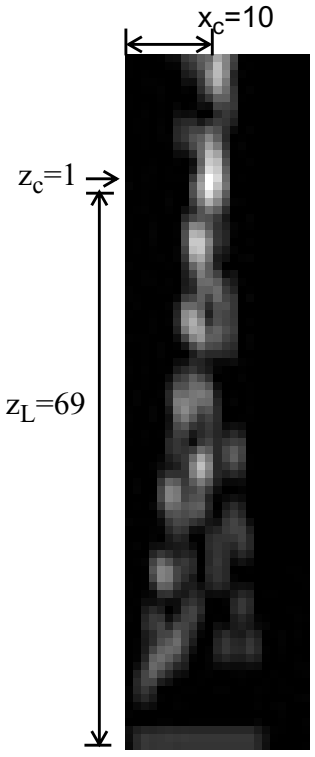


Figure 6.7 Accumulator image for Reflector R1 (experiment 2)

Table 1Results of the Corner Extraction Algorithm

Reflector	Number of Corners Present	Coordinates of Acual Corners	Number of Corners Detected	Coordinates of Detected Corners
S1	5	C1=(0,0)	4	c1=(2,1)
		C2=(30,0)		c2=(31,1)
		C3=(60,0)		c3=(60,0)
		C4=(105,0)		c4=(105,0)
		C5=(110,0)		
S2	3	C1=(0,0)	3	c1=(2,0)
		C2=(28,0)		c2=(28,0)
		C3=(58,0)		c3=(56,-2)
R1 - Expt 1	1	C1=(11,0)	1	c1=(11,2)
R1 - Expt 2	1	C1=(11,0)	1	c1=(10,1)

As seen from Table 1, for reflector S1, the algorithm detects only 4 out of 5 corners. This is due to the proximity of corners C4 and C5. The lines of sight corresponding to C4 and C5, being very close together, lead to only one, not two, cluster of bright points in the accumulator. Therefore the corner corresponding to C4 is detected, while the corner corresponding to C5 is not. For the remaining corners of S1, S2 and R1(both, Expt 1 and 2), the coordinates of the detected corners are within and error of two pixels.

6.2 Algorithm 1 Results

Algorithm 1, Shape Recovery through Optimization, is presented in Chapter 4. The simulation results of this algorithm for reflectors S1 and S2 are presented in Figure 6.8 and 6.9.

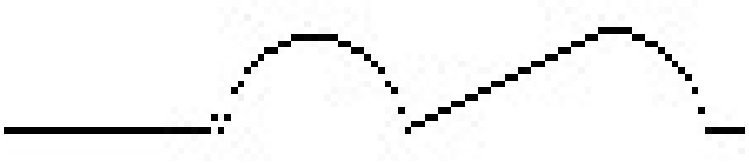


Figure 6.8 Reconstruction for reflector 'S1' using Algorithm 1

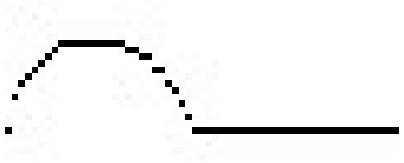


Figure 6.9 Reconstruction for reflector 'S2' using Algorithm 1

As seen from the figures and a comparison with Figure 6.1 and Figure 6.2, reflectors S1 and S2 are successfully reconstructed by Algorithm 1. This accuracy is quantified in section 6.4.

6.2.1 Problem with Processing Experimental Data

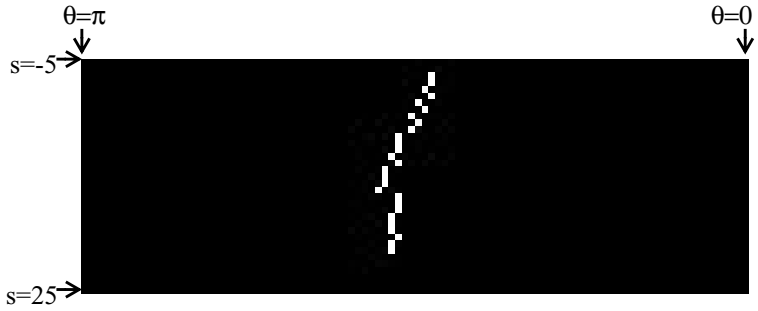


Figure 6.10 Reflectance image for reflector R1 (experiment 1)

The experimental reflectance data for reflector R1 is shown in Figure 6.10. As seen in the figure, all the values of θ, the angle of the line of sight, are very close to π/2. This is because, z_L, the distance between the camera and the reflector, is high compared to the length of the reflector. Here, $z_L \approx 3 \times \text{Length of reflector}$. Since the values of θ are close to π/2, the reconstruction of the reflector is just a straight line between corners. Though not an accurate reconstruction, it is in tune with the experimental data. Obtaining data with a wider range of θ*s*, would require a z_L to the order of $z_L \approx \frac{1}{2} \times \text{Length of reflector}$, which is not feasible, owing to narrow viewing angle range of available cameras.

6.3 Algorithm 2 Results

Algorithm 2, Shape Recovery using Relaxation Labeling, is presented in Chapter 5. In this section the results of the algorithm are presented for the two simulated reflectors. Section 5.2.2 discussed a variation in selection of the final reflector points, once the algorithm has converged satisfactorily. The two variants were:

1. For each token, (s_i,θ_i), select, among its associated labels, $(x_j,z_j)$the label with maximum probability. This way, the number of points in the reconstructed reflector will equal the number of camera positions. So, there need not be a one-to-one correspondence between x and z-values in the resulting reflector.

2. For each possible x-value,x_j, select, among the labels containing x_j, the label with maximum probability. Then for each x-value, there will be one and only one z-value.

For each reflector two results are presented- one for each of the two algorithm variants discussed above. The reconstructed shape is superimposed on the actual shape to give an idea of accuracy. The actual reflector shape is represented

by the lighter points and the reconstructed shape is represented by the dark pixels.

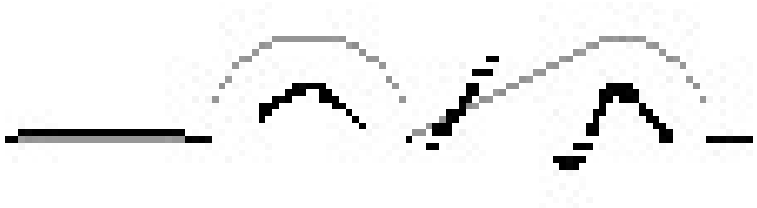


Figure 6.11 Reconstruction of reflector ‘S1’ using Algorithm 2, Variant 1

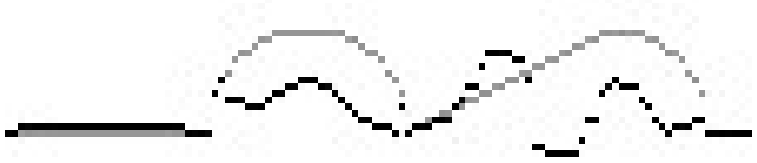


Figure 6.12 Reconstruction of reflector ‘S1’ using Algorithm 2, Variant 2

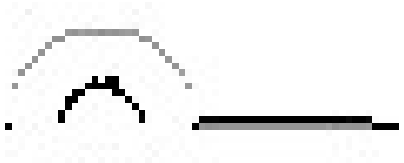


Figure 6.13 Reconstruction of reflector ‘S2’ using Algorithm 2, variant 1

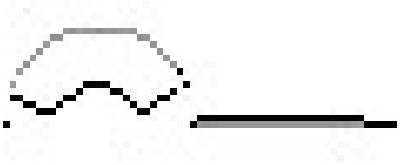


Figure 6.14 Reconstruction of reflector ‘S2’ using Algorithm 2, variant 2

Figure 6.11 through Figure 6.14 show that while the horizontal line segments are reconstructed fairly accurately, the rest of the features are not reproduced to a satisfactory degree. However, as seen from Figure 6.11 and Figure 6.13, the general shape of the semi-circular segments are reproduced; while the size is not.

6.4 Summary of Shape Reconstruction Results

Here, an accuracy measure is presented, using which the performance of the two algorithms can be assessed. The measure quantifies the results presented in sections 6.2 and 6.3. The accuracy measure was calculated as follows:

1. for the results of Algorithm 1 and Algorithm 2, Variant 2:
These include Figure 6.8, Figure 6.9, Figure 6.12 and Figure 6.14. In these reconstructions, for every x there exists a value of z. For each figure, the sum of absolute differences between the z-values of the original reflector and the reconstructed reflector was calculated. Then, the maximum possible inaccuracy, or, the maximum possible sum of absolute differences was calculated. The Percent Accuracy Measure, then, is the percentage of the actual sum of absolute difference w.r.t. the maximum possible sum of absolute differences. The sum of differences and % accuracy measure are entered in Table 2.

2. for the results of Algorithm 2, Variant 1:
These include Figure 6.11 and Figure 6.13. In these, for every x value, there need not necessarily exist a z value (as explained in section 6.3). Instead, for every (s_i,θ_i), there exists an (x,z). So for each result, the sum of differences was

calculated not over x, but over the tokens (s,θ). For each figure, the sum of differences among the actual (x,z) and the reconstructed (x,z) was calculated. The % accuracy measure was calculated as above and both these are presented in Table 2.

Table 2Summary of Algorithm 1 and 2 Results

Algorithm	Reflector	Sum of Differences	% Accuracy Measure
Algorithm 1	S1	88	94.6 %
	S2	43	95.27 %
Algorithm 2, Variant 1	S1	614	62.31 %
	S2	271	70.22 %
Algorithm 2, Variant 2	S1	566	65.25 %
	S2	232	74.5 %

6.5 Conclusion

In this thesis, two algorithms were presented for shape recovery for 2-dimensional specular reflectors from multiple brightness images. Both these algorithms employed a Corner Extraction Algorithm to obtain seed points used as an input for the algorithm. The results for these algorithms are presented in sections 6.1, 6.2 and 6.3. The analysis of these results is presented here:

• The corner extraction algorithm results for 2 simulation data sets and 2 experimental data sets are presented in Table 1. As seen from the table, the algorithm gives the correct solution to within 2 pixels of accuracy.

• Algorithm 1, employing optimization, was tested on 2 simulation data sets. As discussed in section 6.2, it was not possible to test it on experimental data. A visual perusal of Figure 6.8 and Figure 6.9 indicates that the shape reconstruction is accurate. This is quantified in Table 2 as the accuracy measure. For Algorithm 1, the accuracy measure was found to be around 95%.

• Algorithm 2, employing relaxation labeling, was also tested on 2 simulation data sets, the results of which are presented in Figure 6.11 through Figure 6.14. From Table 2, the accuracy measure for the reconstruction varies around 70%.

A visual inspection of the results confirms the low accuracy. A reason for this could be that the quantization error, introduced by converting from the continuous (distance, length) domain to the discrete (pixels) domain, causes an error in the consistency calculations. This error may then be propagated and worsened through the iterations. The order of complexity of this algorithm is $O(2a^2N^2+aN)$, where N is the number of input (s,θ) and a is the average number of labelings for every (s,θ). This is further detailed in section 5.3.1.

6.6 Further Work

Suggestions for further work follow:

• The work presented here can be extended for 3-D surfaces. Briefly, this can be done by relating corners in 2-D reflectors to edges and corners in 3-D, and generalizing the corner extraction algorithm to extract edges (lines) and corners (points) for the 3-D reflector. Similarly, the lines and curves need to be related to planes and planar curves.

• Further attempts may be made to improve the performance of Algorithm 2.

Bibliography

[1] P. Beckmann and A. Spizzichino, “The Scattering of Electromagnetic Waves from Rough Surfaces” *Pergamon Press, New York*, 1963.

[2] A. Blake and G. Brelstaff, “Geometry from Specularities” *IEEE International Conference on Computer Vision*, pp. 394-403, 1988.

[3] G. Brelstaff and A. Blake, “Detecting Specular Reflection Using Lambertian Constraints” *IEEE International Conference on Computer Vision* pp. 297-302, 1988.

[4] R.O. Duda and P.E. Hart, “Use of the Hough Transform to Detect Lines and Cuves in Pictures” *Comm. of Association Computing Machinery,* 15,1972.

[5] J. Franke and W. Snyder, “Determination of Part Pose with Unconstrained Moving Lighting”, 1984.

[6] R. Gershorn, A.D. Jepson and J.K. Tsotsos, “Highlight Identification Using Chromatic Information” *Proc. 1st International Conference on Computer Vision*, 1987.

[7] J. J. Gibson, “The Perception of the Visual World” *Boston: Houghton Mifflin*, 1950.

[8] G. Healey and T.O. Binford, “Local Shape from Specularity” *Computer Vision, Graphics, and Image Processing*, vol. 42, pp. 62-86, 1988.

[9] B.P.K. Horn, “Shape from Shading: A Method for Obtaining The Shape of a Smooth Opaque Object from one view” *PhD Thesis, Department of Electrical Engineering, MIT,* 1970.

[10] P.V.C.Hough “Method and Means for Recognizing Complex Patterns” *U.S. Patent 3069654*, 1962.

[11] K. Ikeuchi, “Determining Surface Orientations of Specular Surfaces by using the Photometric Stereo Method”*IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 3, pp.661-669, 1981.

[12] T. Kanade,”Recovery of the Three-Dimensional Shape of an Object from a Single View”,*Artificial Intelligence Journal*, vol.17, pp.409-460,1981.

[13] J.H. Lambert, “Photometria sive de mensura de gratibus luminis, colorum et umbrae” *Augsberg, Germany: Eberhard Klett*, 1760.

[14] S.K. Nayar and K. Ikeuchi, “Photometric Sampling: A Method for determining Shape and Reflectance of Surfaces” *Machine Vision for Inspection and Measurement, Academic Press*, 1989.

[15] S.K. Nayar, K. Ikeuchi, and T. Kanade, “Extracting Shape and Reflectance of Hybrid Surfaces by Photometric Sampling” *Proc. Image Understanding Workshop, Palo Alto, CA*, 1989.

[16] S.K. Nayar, K. Ikeuchi, and T. Kanade, “Surface Reflection: Physical and Geometrical Perspectives” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, p. 611-634, July 1991.

[17] M. Oren and S.K.Nayar, “A Theory of Specular Surface Geometry” *Proc. 5th International Conference on Computer Vision*, 1995.

[18] J.S. Park and J.T. Tou, “Hightlight Separation and Surface Orientation for 3-D Specular Objects” *Proc. 10th International Conference on Pattern Recognition*, 1990.

[19] B.T. Phong, “Illumination for Computer Generated Pictures” *Communications on ACM*, 1975.

[20] H. Ragheb and E. R. Hancock, “Separating Lambertian and Specular Reflectance Components using Iterated Conditional Modes”, *British Machine Vision Conference*, 2001

[21] A. Rosenfeld, R. Hummel and S. Zucker, “Scene labeling by Relaxation Operations”, *IEEE Trans. on Systems, Man and Cybernetics*, 1976.

[22] H. Schultz, “Retrieving Shape Information from Multiple Images of a Specular Surface” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, February 1994.

[23] K.E. Torrance and E.M. Sparrow, “Theory for Off-Specular Reflection from Roughened Surfaces” *J. Optical Soc. Am.*, vol. 57, p. 1105-1114, 1967.

[24] R. Zhang, P. Tsai, J. Cryer and M. Shah, “Shape from Shading: A Survey”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, August 1999.