Shape From Shading for Non-Lambertian Surfaces

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Department of Computational Science

University of Saskatchewan

Saskatoon, Saskatchewan

by

Sanjay M. Bakshi

July, 1994

PERMISSION TO USE

Head of the Department of Computational Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 0W0

ACKNOWLEDGEMENTS

I wish to thank Professor Yang for his instruction, support, and friendship. The skills I have gained working with him will be useful to me throughout my professional career.

I wish to acknowledge the contribution of members of the Vision Lab. I have benefited from useful discussions with many students– Russell Muzzollini, Gamal Abdel-Hamid, Nan Weng, Jackie Wu, Jennipher Liu. I also appreciate the friendship of several other graduate students in our department. They have made my time here more enjoyable.

Thanks to Valerie Larmour whose support and proofreading skills have made my research work productive.

And finally, I wish to acknowledge the contribution of my parents, Mohan and Subhadra Bakshi. Without their support and encouragement, this thesis could never have been written.

Shape From Shading for Non-Lambertian Surfaces

Candidate: Sanjay M. Bakshi

Supervisor: Herb Yang

Master of Science Thesis

Presented to the College of Graduate Studies and Research

University of Saskatchewan

July, 1994

ABSTRACT

The human visual system has the capacity to infer the shape of a three-dimensional object using only two-dimensional information. One of the cues the human visual system uses to perform this inference is shading information.

An object's shading is its variation in brightness over a given domain. Computer algorithms have been developed that recover the three-dimensional shape of objects depicted in images. These algorithms are called shape-from-shading algorithms.

Most shape-from-shading algorithms rely on idealistic conditions. They usually assume orthographic projection, a distant point light source, and Lambertian reflectance.

It is known that most real surfaces are neither perfectly diffuse (Lambertian) nor ideally specular (mirror-like); however, most shape-from-shading algorithms assume Lambertian reflectance. It is necessary to develop new techniques to recover the shape of objects whose surfaces are not necessarily Lambertian.

In this thesis two algorithms are proposed that can recover the shape of non-Lambertian surfaces given reflectance information. These algorithms are incorporated into the LIRAS (LIght, Reflectance, And Shape) Recovery System. The LIRAS Recovery System is a practical application of computer vision. Given a single image with multiple objects (each having different reflectance properties), the LIRAS Recovery system can reconstruct the three-dimensional shape of each object in the scene. Both real and synthesized images are used to evaluate the proposed algorithms.

Contents

List of Figures

Chapter 1

Introduction

An image is a two-dimensional representation of a three-dimensional world. From images, the human visual system has the capacity to recover three-dimensional information. One of the cues the human visual system uses is shading.

If an object has constant surface reflectance properties, the variations in brightness are due to the fact that some parts of the surface are oriented to reflect more of the incident light toward the viewer than are others. The human visual system uses these variations in brightness to recover the surface orientations of the object, in this way recovering the three-dimensional shape of the object.

Computer algorithms have been developed that recover the three-dimensional shape of objects depicted in images using variations in brightness. These algorithms are called shape-from-shading algorithms.

Shape recovery from shading information is not a trivial problem. Image brightness is dependent on surface orientation and surface reflectance properties, as well as the distribution of light sources. Also, shape has two degrees of freedom, as orientation can be expressed as a unit vector in the direction of the gradient. However, a brightness measure at a single point provides only one constraint. To recover surface orientation, either additional information is needed or additional assumptions are required.

The problem can be sufficiently constrained if additional images of the same object illuminated under different lighting conditions are available. However, the human visual system can recover the three-dimensional shape of objects given just one image. This suggests that either enough information is present or implicit assumptions are used.

Surface brightness is dependent on three factors: surface orientation, the distribution of the light sources, and surface reflectance properties. The shape-from-shading problem is concerned with reconstructing the shape (surface orientation) of the object. To perform this recovery, researchers have made further assumptions so that brightness is solely dependent on surface orientation.

Most shape-from-shading algorithms assume that the light source is far enough away that it can be treated as a point source. This facilitates the formulation of the problem, as brightness now depends only on surface orientation and reflectance properties. It also means that orthographic projection can be employed, which further simplifies the problem. Early shape-from-shading algorithms assume that the direction of the light source is available to the algorithm [20, 23]; recently algorithms have been developed to recover the direction of the light source [6, 29, 38, 43].

The human visual system assumes that the reflecting properties are homogeneous over the object's surface. If this is not the case, i.e., the reflectance properties vary on the object's surface, the brain can be fooled and a shape that is different from the real one may result. This phenomenon is exploited by the use of cosmetics to change the reflectance properties of a person's face. Features can be made to look more defined by carefully applying the cosmetics. Shape-from-shading algorithms assume homogeneous surface reflectance, but also make additional assumptions about surface reflectance to make the problem more tractable. For example, reflectance properties are assumed to be known. In addition, most algorithms are formulated assuming that the object's surface obeys Lambertian reflectance.

Matte surfaces observe Lambertian reflectance. These surfaces are so rough that incident light is reflected equally in all directions. Consequently, the direction of the viewer is irrelevant to the formulation of the problem. Researchers often assume Lambertian reflectance because of this simplification. However, it is known that most real surfaces are neither perfectly diffuse (Lambertian) nor ideally specular (mirror-like). It is necessary to develop new techniques to solve the shape-from-shading problem. These techniques must be able to recover the shape of objects whose surfaces are not necessarily Lambertian.

The problems incurred when attempting to recover the three-dimensional shape of non-Lambertian surfaces are the motivation of this research project. When dealing with Lambertian surfaces, the reflectance map is completely defined by the illuminant direction. A non-Lambertian reflectance function can model both diffuse as well as specular surfaces; the degree of specularity is controlled by additional parameters. This generalization introduces several complexities to the shape-from-shading problem, some of which are addressed in this thesis. The LIRAS (LIght, Reflectance And Shape) Recovery System, proposed in this thesis, incorporates algorithms to estimate the reflectance map prior to the shape-from-shading process. The LIRAS Recovery System is a practical application of computer vision. Given a single image containing several objects with various reflectance properties, the shape of each object can be reconstructed. This system is evaluated using both real and synthesized images.

There are currently very few shape-from-shading algorithms for non-Lambertian[1] surfaces. The results of the survey portion of this thesis make it apparent that more work in this area is needed.

1.1 Thesis Organization

A discussion of the shape-from-shading process is given in Chapter 2. A survey of the algorithms used to perform the shape recovery process is presented, and new techniques are proposed to perform the height integration process.

The different methods used to model light reflection from a surface are discussed in Chapter 3. The Torrance-Sparrow [37] model is presented and its incorporation into shape recovery algorithms is proposed. Various other shape recovery algorithms are surveyed in this chapter.

In Chapter 4 two new algorithms are proposed to recover shape from non-Lambertian

[1]The term non-Lambertian refers to surfaces that are not necessarily Lambertian but could be.

surfaces. These algorithms are formulated to incorporate the Torrance-Sparrow model of light reflection.

The LIRAS Recovery System is discussed in Chapter 5. New algorithms are proposed to recover the reflectance parameters given the illuminant direction and locate the light source given knowledge about the surface reflectance parameters. A new algorithm is presented to recover the reflectance map given a calibration object.

Chapter 6 includes a discussion of the experimental evaluation methods. The techniques discussed in this chapter are made use of in Chapter 7, which also includes the evaluation of the algorithms proposed in this thesis. Both real and synthesized images are used.

Finally, in Chapter 8 the expected contributions are discussed and some final conclusions are presented.

Chapter 2

Shape From Shading

Synthesizing realistic images depicting three-dimensional objects is a computer graphics problem. These images can be synthesized using the shape of the object (usually represented as surface normals), the reflectance properties of the object's surface, and the distribution of the light sources. Using this information, one can employ techniques such as ray tracing [8, 15] to create realistic-looking images.

If we assume a single point light source and Lambertian reflectance, the relationship between image intensity and object shape is [19]:

$$E(x, y) = \mathbf{n}(x, y) \cdot \mathbf{s}. \tag{2.1}$$

This is interpreted to mean that the image intensity, $E(x, y)$, is equal to the angle between the surface normal, $\mathbf{n}(x, y)$, and the light source direction vector, $\mathbf{s}$. Horn [18] expresses surface orientation using the partial derivatives p and q:

$$n(x, y) = (-p, -q, 1) \tag{2.2}$$

where $p = \frac{\partial z}{\partial x}$ and $q = \frac{\partial z}{\partial y}$ [18, 19].

The shape-from-shading problem is to unravel the image formation process, i.e., given an image of an object, how can the shape of the object be recovered?

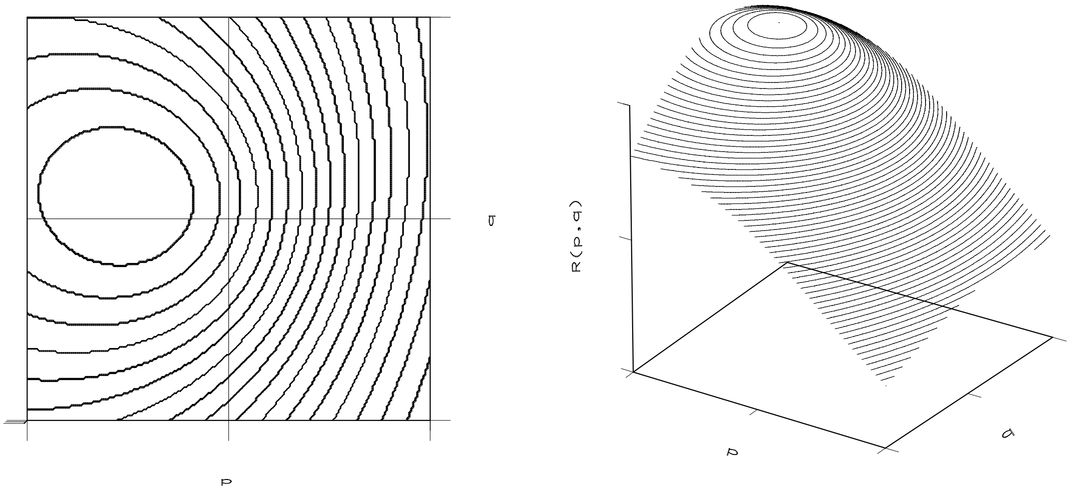


Figure 2.1: Reflectance maps for Lambertian surfaces. Left: 2-D, Right: 3-D.

2.1 Reflectance Maps

The reflectance map is a useful tool that has been used in both photometric-stereo algorithms as well as shape-from-shading algorithms [18, 19, 21]. It defines the relationship between image brightness and surface orientation, and encodes information about the surface reflectance properties and the light source distribution.

To construct the reflectance map, the surface orientations are mapped onto gradient-space. This is accomplished by mapping the surface normal, $(-p, -q, 1)$, onto the point (p, q) in gradient-space [18]. This is equivalent to translating the surface normal to the origin and finding its intersection with a plane at a unit distance from the origin.

For a given type of surface and light source, there is a fixed value of reflectance for every surface orientation. Hence, for every point in the gradient-space there is a unique image intensity. For a Lambertian surface, the reflectance map, $R(p, q)$, is:

$$R(p, q) = (-p, -q, 1) \cdot (-p_s, -q_s, -1), \tag{2.3}$$

where $(-p_s, -q_s, 1)$ is the orientation of the light source. The reflectance function can be interpreted as a contour map. The contours are formed by the points in gradient-space with equal image intensity values. The reflectance maps given in Figure 2.1 reveal that image brightness is dependent on surface orientation. Figure 2.1 also shows that several surface orientations will give the same image intensity.

2.2 Occluding Contour

The closed curve on an object's surface where the surface normals are parallel to the image plane is called the occluding contour. The occluding contour provides vital information because the gradient is known on this curve. This information is used by many shape-from-shading algorithms as a means to initialize an iterative solution [6, 17, 20, 23, 36]. However, depending on the direction of the light source, the occluding contour can be obscured from view by self-shadowing. Points lie on the self-shadowed boundary if the surface normals at these points form a 90-degree angle with the light source. The problem at these points is that the surface normals are not on the image plane as are the points on the occluding contour. It has been noted [23, 26] that the surface normals on the self-shadowed boundary are perpendicular to the light source direction. This is an important observation that has not been utilized by the shape-from-shading algorithms presented in the literature.

Most shape-from-shading algorithms assume that the occluding contour is available. However, depending on the orientation of the light source, occluding contour determination is not trivial. This is a problem that has not been addressed in the literature but will be studied in this research work.

2.3 Shape-From-Shading Algorithms

Horn [17] was the first to attempt to solve the shape-from-shading problem. His solution, which employs standard calculus, is called the characteristic strip method. More recently, a new algorithm similar in nature to the characteristic strip method was proposed by Bruckstein [7]. These methods suffer from a high degree of error if noise is present in the image. Thus, other approaches were developed.

The next significant advancement is a tool to understand the relationship between surface orientation and image brightness. This tool, called the reflectance map, was proposed by Horn [18, 19, 21].

Ikeuchi and Horn [23] proposed the first shape-from-shading algorithm using the calculus of variations. This was the first attempt to develop an algorithm that incorporates the reflectance map. This method prompted a flood of research that continues today. While approaches based on the calculus of variations are interesting, they have several fundamental flaws to be discussed later in this thesis.

Pentland [30] has devised approaches that assume local spherical properties. Other researchers [19, 25, 31] make assumptions about the reflectance map rather than the surface shape. These works assume that the reflectance map is locally linear. Recently, Vega and Yang [39] proposed a heuristic-based shape-from-shading algorithm. These approaches are described in Subsection 2.3.3.

2.3.1 Characteristic Strip Methods

The characteristic strip method was the first approach proposed to solve the shape-from-shading problem [17, 19]. More recently, Bruckstein [7] presented the equal height contour method which is similar in flavour to the characteristic strip approach.

The method presented by Horn [17, 19] assumes that initial values are available, namely the three-dimensional position (x_o, y_0, z_0) and the gradient p_0, q_0. These sets of values are called a characteristic as they characterize the surface at a point.

With respect to (x_0, y_0) at (x, y):

$$\begin{aligned} x &= x_0 + \Delta x \\ y &= y_0 + \Delta y \\ \Delta z &= z_x \Delta x + z_y \Delta y \end{aligned} \tag{2.4}$$

where $z_x = \frac{\partial z}{\partial x} = p$ and $z_y = q$.

The change in gradient $(\Delta p, \Delta q)$ can be obtained as follows:

$$\begin{aligned} \Delta p &= p_x \Delta x + p_y \Delta y \\ \Delta q &= q_x \Delta x + q_y \Delta y. \end{aligned} \tag{2.5}$$

These five differential equations define the characteristic at a new point. To solve this set of equations, the image intensity values must be utilized. Recall that $E(x, y) = R(p, q)$. Using the chain rule for differentiation the change in intensity is:

$$\begin{aligned} E_x &= R_p p_x + R_q q_x \\ E_y &= R_p p_y + R_q q_y. \end{aligned} \tag{2.6}$$

However, the direction to proceed is not constrained. If a direction is chosen such that,

$$\begin{aligned} \Delta x &= R_p \\ \Delta y &= R_q, \end{aligned} \tag{2.7}$$

then Equations (2.4) and (2.5) can be expressed using (2.7):

$$\begin{aligned} \Delta z &= pR_p + qR_q \\ \Delta p &= p_x R_p + p_y R_q = E_x \\ \Delta q &= q_x R_p + q_y R_q = E_y. \end{aligned} \tag{2.8}$$

The solution is then propagated along the characteristic strip recursively. Also note that the direction is in the path of steepest ascent and the strips are initiated from the occluding contour.

Bruckstein's approach [7] is similar in nature, but uses equal height contours as opposed to characteristic strips. These two approaches were compared experimentally, the results of which are in a paper by Tandri and Yang [36], who show that Horn's characteristic strip method [17, 19] performs better than the method proposed by Bruckstein [7].

The characteristic strip method suffers from a high degree of error even in the absence of image noise. This error is propagated and accumulated along the characteristic strip. Furthermore, the method is unstable computationally. A more robust technique is required that can produce good results when there is unavoidable noise in the image.

2.3.2 Variational Calculus-Based Approaches

Another approach to the shape-from-shading problem is to seek a surface that "fits" the image data. The problem is expressed as the minimization of a cost function that measures the departure from a perfect surface.

The cost function is:

$$|E(x,y) - R(p,q)|. \tag{2.9}$$

The desired surface gradient, (p,q), should generate an image intensity equal to the observed image intensity. Both $E(x,y)$ and $R(p,q)$ are three-dimensional surfaces; the difference in the volume between these surfaces (the total error) can be expressed as:

$$\int\int (E(x,y) - R(p,q))^2 dxdy. \tag{2.10}$$

The gradients that minimize this expression must be found. Equation (2.10) is commonly called a functional, which is a function of functions. To find extrema of functionals the calculus of variations is required. It is well-known that the extrema of a functional satisfies an associated set of Euler equations each of which is determined from the functional [41]. Thus, solving the set of Euler equations is analogous to equating a differential of a function to zero.

Equation (2.10) is ill-posed because there are numerous solutions to minimize the problem. The problem must be further constrained so that a solution can be computed.

Ikuechi and Horn [23] were the first to apply the calculus of variations to the shape-from-shading problem. They assume that the surface is smooth and hence, neighbouring patches should have similar orientations. Employing such a constraint is called *regularization*. A regularization term can transform an ill-posed problem to a well-posed problem.

The smoothness constraint is expressed as:

$$\int\int (p_x^2 + p_y^2 + q_x^2 + q_y^2)dxdy. \tag{2.11}$$

Ikeuchi and Horn [23] use Equation (2.11) as a penalty term and incorporate it into Equation (2.10) as follows:

$$\int\int \left[(E(x,y) - R(p,q))^2 + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2)\right] dxdy. \tag{2.12}$$

The functional with the form:

$$\int\int F(x, y, p, q, p_x, p_y, q_x, q_y)dxdy \tag{2.13}$$

has the Euler equations [11, 20]:

$$\begin{aligned} F_p - \tfrac{\partial}{\partial x}F_{p_x} - \tfrac{\partial}{\partial y}F_{p_y} &= 0 \\ F_q - \tfrac{\partial}{\partial x}F_{q_x} - \tfrac{\partial}{\partial y}F_{q_y} &= 0. \end{aligned} \tag{2.14}$$

It is easy to show that:

$$\begin{array}{cc} F_p = 2[E(x,y) - R(p,q)][-R_p] & F_q = 2[E(x,y) - R(p,q)][-R_q] \\ F_{p_x} = 2\lambda p_x & F_{q_x} = 2\lambda q_y \\ F_{p_y} = 2\lambda p_y & F_{q_y} = 2\lambda q_y. \end{array} \tag{2.15}$$

Substituting back into Equation (2.14) gives:

$$\begin{aligned} -R_p[E(x,y) - R(p,q)] - \tfrac{\partial}{\partial x}\lambda p_x - \tfrac{\partial}{\partial y}\lambda p_y &= 0 \\ -R_q[E(x,y) - R(p,q)] - \tfrac{\partial}{\partial x}\lambda q_x - \tfrac{\partial}{\partial y}\lambda q_y &= 0. \end{aligned} \tag{2.16}$$

Using the substitution $p_x = \frac{\partial}{\partial x}p$ and noting that the Laplacian operator, $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$:

$$\begin{aligned} (E - R)R_p + \lambda\nabla^2 p &= 0 \\ (E - R)R_q + \lambda\nabla^2 q &= 0. \end{aligned} \tag{2.17}$$

These equations must be satisfied for an extremum to exist. Introducing a discrete approximation to the Laplacian [19, 23]:

$$\nabla^2 p_{i,j} \approx \frac{4}{\epsilon^2}(\bar{\mathbf{n}}_{i,j} - n_{i,j}) \tag{2.18}$$

where ϵ is the grid distance, $\bar{p}_{i,j}$ is the local average of the normals and (i, j) denotes the position on the xy-plane. A similar equation can be derived for q. Substituting the discrete approximation of the Laplacian into (2.17) gives:

$$\begin{aligned} p_{i,j} &= \bar{p}_{i,j} + \tfrac{\epsilon^2}{4\lambda}\left(E_{i,j} - R(p_{i,j}, q_{i,j})\right) R_p(p_{i,j}, q_{i,j}) \\ q_{i,j} &= \bar{q}_{i,j} + \tfrac{\epsilon^2}{4\lambda}\left(E_{i,j} - R(p_{i,j}, q_{i,j})\right) R_p(p_{i,j}, q_{i,j}). \end{aligned} \tag{2.19}$$

An iterative scheme can be derived:

$$\begin{aligned} p_{i,j}^{k+1} &= \bar{p}_{i,j} + \tfrac{\epsilon^2}{4\lambda}\left(E_{i,j} - R(p_{i,j}^k, q_{i,j}^k)\right) R_p(p_{i,j}^k, q_{i,j}^k) \\ q_{i,j}^{k+1} &= \bar{q}_{i,j} + \tfrac{\epsilon^2}{4\lambda}\left(E_{i,j} - R(p_{i,j}^k, q_{i,j}^k)\right) R_p(p_{i,j}^k, q_{i,j}^k). \end{aligned} \tag{2.20}$$

This iterative algorithm uses the information on the occluding contour for initialization. This method is the first attempt to base a solution on the calculus of variations. Many other algorithms have been developed that are similar in nature but minimize different functionals.

Further work is necessary, as the algorithm of Ikeuchi and Horn [23] is flawed. The attempt to constrain the problem by means of a penalty smoothness function introduces errors. Incorporating the smoothness constraint means that the surface sought is not the same surface that satisfies (2.10). In fact, Horn and Brooks [20] report that given the correct gradients of a sphere as initial input, the algorithm "walks away" to a squashed sphere. The tradeoff between surface smoothness and accuracy is controlled by λ. Determining the value of λ is still an open problem.

Horn and Brooks [20] recognize the problem of the regularization term [23]. They suggest that using an integrability constraint could alleviate the problems. They define the integrability constraint as:

$$z_{xy} = z_{yx}. \tag{2.21}$$

This is a more general form of a smoothness constraint, as no discontinuous surface can satisfy the integrability constraint. Unfortunately, Horn and Brooks [20] fail to find a convergent algorithm that employs this integrability constraint.

Frankot and Chellappa [14] address the integrability issue in a radically different manner. They propose an algorithm that coexists with Brooks and Horn's [6] method. At every iteration, the new surface normals calculated by Brooks and Horn's [6] algorithm are modified. That is, the normal at point (i, j) is projected onto the nearest solution that is integrable. These modified normals are then fed back to Brooks and Horn's [6] algorithm for the next iteration. Although this is an innovative approach to enforce integrability, there are several problems. Bakshi [1] has reported that the algorithm performs poorly compared to Brooks and Horn's [6] algorithm. Also, Horn and Brooks [20] have published that given the solution, their variationally-based shape-from-shading algorithm still "walks away" from the solution. They cite regularization as the reason that their algorithm behaves in this manner. Frankot and Chellappa's [14] algorithm is only useful if it cooperates with an algorithm that stabilizes after finding the solution.

Brooks and Horn [6] use the surface normals themselves rather than the surface gradient. Using this notation, Equation (2.12) can be expressed as:

$$\int\int \Big((E - \mathbf{n} \cdot \mathbf{s})^{\mathbf{2}} + \lambda(\|\mathbf{n_x}\|^{\mathbf{2}} + \|\mathbf{n_y}\|^{\mathbf{2}}) + \mu(\|\mathbf{n}\|^{\mathbf{2}} - \mathbf{1})\Big)\, dxdy. \tag{2.22}$$

Although more elegant, this solution is the same as the method proposed by Ikeuchi and Horn [23]. It suffers from the same problems due to regularization.

More recently, Horn [19] has proposed using a different constraint, namely ensuring that the surface $z(x, y)$ has partial derivatives z_x, z_y that come closest to the computed

$p(x, y)$ and $q(x, y)$. This can be enforced by minimizing:

$$\int\int \left((z_x - p)^2 + (z_y - q)^2\right) dxdy. \tag{2.23}$$

Horn [19] incorporates this new constraint into Equation (2.12) derived by Ikeuchi and Horn [23]:

$$\int\int \left[(E(x, y) - R(p, q))^2 + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2) + \mu((z_x - p)^2 + (z_y - q)^2)\right] dxdy. \tag{2.24}$$

This is an attractive approach as the surface $z(x, y)$ is computed directly, unlike previous methods [18, 20, 23] that required an additional step to integrate the gradient to obtain the surface. Horn [19] addresses the problem of regularization by suggesting that varying λ dynamically as the iterative algorithm is computed will suppress the "walking away" problem. Regularization is necessary in the early stages of the algorithm when far away from the solution. When close to the solution, Horn [19] suggests λ should be reduced. However, a mechanism to reduce λ automatically is still an open issue.

Zheng and Chellappa [43] seek to derive a cost function that does not require a smoothness constraint. They replace the smoothness term with the constraint that the gradient of the observed image intensities, $(E_x(x, y), E_y(x, y))$, be equal to the gradients of the reconstructed image intensities $(R_x(p, q), R_y(p, q))$. They replace the smoothness term in Equation (2.24) with this new constraint. The resulting functional is:

$$\int\int F(p, q, z)dxdx \tag{2.25}$$

where

$$\begin{aligned} F = {} & [(E(x, y) - R(p, q))]^2 \\ & + [(R_p(p, q)p_x + R_q(p, q)q_x - E_x(x, y))]^2 \\ & + [(R_p(x, y)p_y + R_q(p, q)q_y - E_y(x, y))]^2 \\ & +\mu\left[(z_x - p)^2 + (z_y - q)^2\right]. \end{aligned} \tag{2.26}$$

This functional can be minimized if the associated Euler equations are satisfied. This formulation has the advantage of not using the smoothness constraint.

There are several problems with the calculus of variations approach to solve the shape-from-shading problem. The associated set of Euler equations can help to locate extremum of the functional, but there is no guarantee that the extremum is a minimum. Also, because the shape-from-shading problem is inherently ill-posed, additional constraints have to be incorporated into the minimization functional. The constraints that have been proposed all lead to solutions that are not the exact desired surfaces. Another problem with the variational formulations is that incorporating more realistic reflectance maps is difficult. This problem is discussed more thoroughly in Chapter 4.

2.3.3 Other Shape-From-Shading Approaches

Pentland [30] proposes a shape-from-shading algorithm that does not require as much information as the traditional algorithms. His method determines the surface albedo as well as the point source direction. Pentland's algorithm does not require information from the occluding contour; to overcome the inherent ambiguity in shading, the method assumes local spherical properties. This constraint, along with first and second partial derivatives of image intensity, is sufficient to recover the shape. To determine surface orientation, the derivatives of the intensities at a surface patch are matched to the derivatives of brightness on a sphere. Lee and Rosenfeld [24] also assume local spherical shape, but propose a method that does not require second-order partial derivatives. However, only spheres exhibit local spherical properties, these methods will invariably produce incorrect surface normals for differently shaped objects. Ferrie and Levine [13] extend local shading analysis, proving for which local surface properties can surface orientation be reconstructed.

Rather than making assumptions about surface shape, Pentland [31] proposes a local analysis method that assumes properties about the reflectance map. His algorithm approximates the reflectance map to be linear. This approximation allows for an efficient one-step algorithm. Horn [19] also assumes the reflectance map to be linear and proposes a variational formulation. More recently, a locally linear reflectance map approach was suggested by Lee and Kuo [25]. Their method is unique as it uses a finite element-based approach to model the surface. The shape of the surface is modified by assuming linearity in the reflectance map. However, these approaches that model the reflectance map as being locally linear are flawed. When dealing with reflectance maps that can model specular reflection, the equal intensity contours are especially steep. These steep contours cannot be estimated using a linear model.

Vega and Yang [38, 39] propose a heuristic-based shape-from-shading algorithm. The heuristics are derived from interpreting Brooks and Horn's [6] variational algorithm geometrically. This approach is easily modified to incorporate different reflectance maps (see Chapter 4).

2.4 Height Integration by Interpolation Polynomials

Most shape-from-shading algorithms represent the reconstructed shape in vector form. That is, the surface normals of the object are found at each image point. Another representation of surface shape is the actual height at each point.

The height, $z(x, y)$, can be recovered by integrating along a curve [19]:

$$z(x, y) = z(x_0, y_0) + \int_{(x_0,y_0)}^{(x,y)} (p dx + q dy). \tag{2.27}$$

The partial derivatives of z can be computed from the surface normal directly:

$$\mathbf{n} = (\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z) \tag{2.28}$$

$$\mathbf{n} = \frac{(p, q, -1)}{\sqrt{p^2 + q^2 + 1}}. \tag{2.29}$$

Thus:

$$p = -\frac{\mathbf{n}_x}{\mathbf{n}_z} \tag{2.30}$$

$$q = -\frac{\mathbf{n}_y}{\mathbf{n}_z}. \tag{2.31}$$

However, integrating along a curve provides an unreliable solution when the surface normal information is recovered by imperfect methods. Errors can propagate along the curve.

Vega [38] proposes a height reconstruction technique that estimates the height at point P using local information. For each of the eight neighbouring pixels of point P, a linear polynomial is constructed. This interpolation polynomial is used to estimate the new height at point P. The eight estimates of the height are then averaged.

For each neighbouring pixel the following equation is used to estimate the new height at point P (H_P):

$$H_P = H_i + p_i \Delta x + q_i \Delta y \tag{2.32}$$

where H_i is the height at neighbour i and p_i, q_i are the partial derivatives at neighbour i.

Several iterations are required for the algorithm to converge to a solution. Vega [38] suggests using a "grassfire approach" to speed up the convergence time. This method is described in Subsection 2.4.4.

Equation (2.32) is a linear polynomial estimation of the height at point P based on the slope and height at neighbour i. In the following section, a new height integration algorithm is described that uses higher-order polynomials to estimate the height profile.

2.4.1 Second-Order Polynomial Interpolation

Rather than using a simple linear polynomial approach to estimate the heights, it may be useful to use higher-order polynomials. Higher-order polynomial interpolation should provide more accurate estimates at the expense of increasing the convergence time.

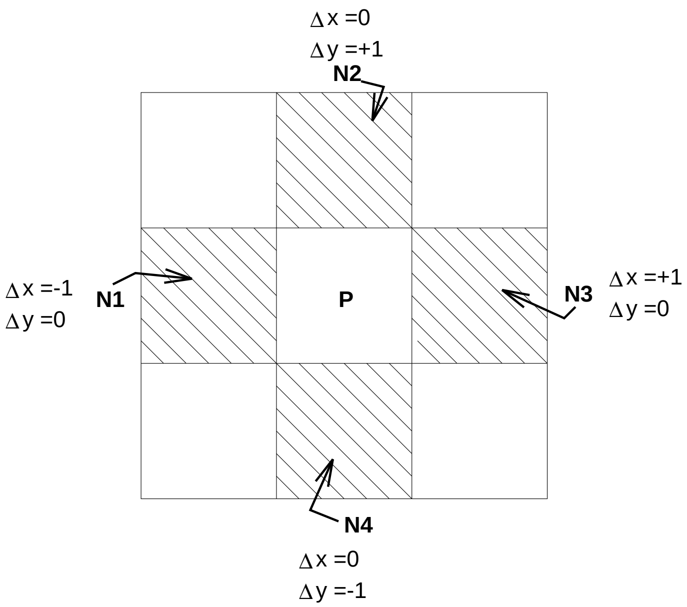


Figure 2.2: Using point P and a neighbour N_j ($j = 1, 2, 3, 4$), a second-order polynomial is constructed.

A general form of a second-order polynomial is:

$$y = a_2 x^2 + a_1 x + a_0. \tag{2.33}$$

The derivative of y is:

$$y\prime = 2a_2 x + a_1. \tag{2.34}$$

Using P and its four neighbours (see Figure 2.2) a second-order polynomial is constructed. Each image point has a two-dimensional index. Neighbour i is mapped from a two-dimensional index (x_i, y_i) to a single number using: $i = x_i \Delta x + y_i\ \Delta y$. Point P (x_P, y_P) is similarly mapped using: $P = x_P \Delta x + y_P \Delta y$. The slope at neighbour i (m_i), the slope at point P (m_p), and the height at neighbour i (H_i) is used to determine the coefficients a_0, a_1, and a_2. The slope at neighbour i is $m_i = p_i \Delta x + q_i \Delta y$. The slope at point P is: $p_P \Delta x + q_P \Delta y$.

Knowing the height at neighbour i and using Equation (2.33):

$$H_i = a_2 i^2 + a_1 i + a_0. \tag{2.35}$$

The slope at neighbour i and Equation (2.34) gives:

$$m_i = 2a_2 i + a_1 \tag{2.36}$$

The slope at point P is also known:

$$m_P = 2a_2 P + a_1 \tag{2.37}$$

There are now three equations and three unknowns. Solving for the coefficients a_0, a_1, and a_2 gives:

$$\begin{aligned} a_0 &= \tfrac{1}{2}\tfrac{-2PH_i+2Pim_i-i^2m_P+2iH_i-i^2m_i}{-P+i} \\ a_1 &= \tfrac{-m_iP+im_P}{-P+i} \\ a_2 &= \tfrac{1}{2}\tfrac{-m_P+m_i}{-P+i}. \end{aligned} \tag{2.38}$$

Thus, the height at point P can be estimated using the coefficients above and:

$$H_P = a_2P^2 + a_1P + a_0. \tag{2.39}$$

2.4.2 Third-Order Polynomial Interpolation

The general form of the degree-three polynomial is:

$$y = a_3x^3 + a_2x^2 + a_1x + a_0. \tag{2.40}$$

The derivative of y is:

$$y\prime = 3a_3x^2 + 2a_2x + a_1. \tag{2.41}$$

To determine the coefficients a_0, a_1, a_2, a_3, four equations are needed. The slopes at points i, P, and j are known. The height at points i and j are also known (see Figure 2.3). Five equations can be expressed for four unknowns. A least-sum-of-squares approach is described in Subsection 2.4.3 that can handle this overdetermined system. However, in this section, the height at point j will be discarded and four equations will be solved for the four coefficients.

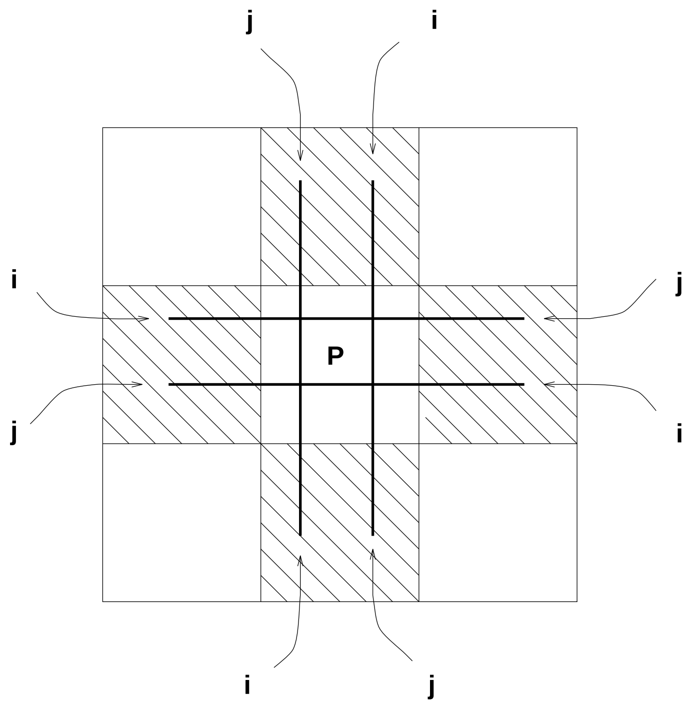


Figure 2.3: Figure shows the four pairs of points used. Pairs of i and j are used along with P to construct a third-order polynomial.

At each point P, four third-order polynomials are constructed. Each of the four polynomials estimates a height at point P. These four estimates are then averaged.

Given the slopes m_i, m_P, and m_j and the height at point i (H_i) the coefficients of Equation (2.40) can be computed.

Knowing the height at neighbour i and using Equation (2.40):

$$H_i = a_3 i^3 + a_2 i^2 + a_1 i + a_0. \tag{2.42}$$

The slope at neighbour i and Equation (2.41) gives:

$$m_i = 3a_3 i^2 + 2a_2 i + a_1. \tag{2.43}$$

The slope at point P is also known:

$$m_P = 3a_3 P^2 + 2a_2 P + a_1. \tag{2.44}$$

Finally, the slope at point j is also available. Thus:

$$m_j = 3a_3j^2 + 2a_2j + a_1. \tag{2.45}$$

Solving the above four equations for the four unknowns yields:

$$a_0 = \frac{1}{6}\frac{(6ij^2 - 6Pj^2 + 6Pi^2 - 6ji^2 + 6jP^2 - 6iP^2)H_i}{(P-j)(-j+i)(i-P)} \tag{2.46}$$

$$+\frac{1}{6}\frac{(i^4 - 4i^3P + 3i^2P^2)m_j}{(P-j)(-j+i)(i-P)} + \frac{1}{6}\frac{(-i^4 + 4i^3j - 3i^2j^2)m_P}{(P-j)(-j+i)(i-P)}$$

$$+\frac{1}{6}\frac{(2i^3j - 2i^3P - 3i^2j^2 + 3i^2P^2 + 6iPj^2 - 6ijP^2)m_i}{(P-j)(-j+i)(i-P)}$$

$$a_1 = -\frac{(-jm_i + im_j)P^2 + (m_ij^2 - i^2m_j)P - m_Pij^2 + i^2jm_P}{(j-i)P^2 + (i^2-j^2)P + ij^2 - ji^2}$$

$$a_2 = \frac{1}{2}\frac{(-m_j + m_P)i^2 + (m_i - m_P)j^2 + (m_j - m_i)P^2}{(j-i)P^2 + (i^2-j^2)P + ij^2 - ji^2}$$

$$a_3 = -\frac{1}{3}\frac{(m_j - m_i)P + (-m_j + m_P)i + (m_i - m_P)j}{(j-i)P^2 + (i^2-j^2)P + ij^2 - ji^2}$$

Using the above coefficients and the following equation, the height at point P can be estimated:

$$H_P = a_3P^3 + a_2P^2 + a_1P + a_0. \tag{2.47}$$

2.4.3 Least-Sum-of-Squares Third-Order Polynomial Interpolation

In the previous section, the height at point j was discarded so that four equations could be solved for the four coefficients. In this section, a third-order polynomial is constructed using all of the information available. The approach used is the least-sum-of-squares method.

To determine the coefficients a_0, a_1, a_2, a_3 it is necessary to use the three points: i, j, and P (note that i and j are on either side of P either horizontally or vertically, see Figure 2.4). The heights are known at points i and j. The slopes are known at

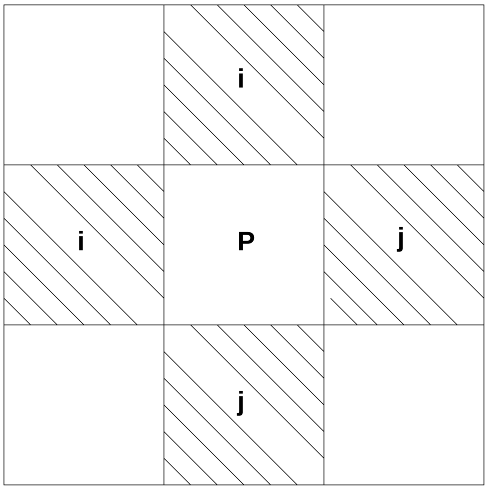


Figure 2.4: Using point P and neighbours i and j, a third-order polynomial is constructed.

points i, j, and P. There are four unknowns (coefficients a_0, a_1, a_2, and a_3) and five equations. A least-sum-of-squares polynomial is constructed to interpolate the height at point P.

The basic approach of constructing a least-sum-of-squares polynomial is to construct a function $y(x)$ in such a way as to minimize the sum of the squares of the errors.

The height at point i is known. Thus the square of the error between the real height and the height interpolated using $y(x)$ is:

$$(H_i - y(i))^2. \tag{2.48}$$

Remembering the general form of the degree-three polynomial (given in Equation (2.40)), Equation (2.48) can be written:

$$(H_i - a_3 i^3 - a^2 i^2 - a_1 i - a_0)^2. \tag{2.49}$$

The height is also known at point j. The square of the error using this information is:

$$(H_j - a_3 j^3 - a^2 j^2 - a_1 j - a_0)^2. \tag{2.50}$$

The slopes at points i, j and P are also known. The square of the error in estimating the slope at i using Equation (2.41) is:

$$(m_i - 3a_3i^2 - 2a_2i - a_1)^2. \tag{2.51}$$

Similarly, using the slopes at points P and j gives:

$$(m_P - 3a_3P^2 - 2a_2P - a_1)^2 \tag{2.52}$$

$$(m_j - 3a_3j^2 - 2a_2j - a_1)^2. \tag{2.53}$$

The sum of the squares of the errors is:

$$\begin{aligned} S = \; & (H_i - a_3i^3 - a^2i^2 - a_1i - a_0)^2 + (H_j - a_3j^3 - a^2j^2 - a_1j - a_0)^2 + \\ & (m_i - 3a_3i^2 - 2a_2i - a_1)^2 + (m_P - 3a_3P^2 - 2a_2P - a_1)^2 + \\ & (m_j - 3a_3j^2 - 2a_2j - a_1)^2. \end{aligned} \tag{2.54}$$

S represents the sum of the squared errors. The minimum of S is obtained by differentiating S with respect to the coefficients:

$$\begin{aligned} \tfrac{\partial S}{\partial a_0} = \; & 4a_0 + (2i + 2j)a_1 + (2i^2 + 2j^2)a_2 + (2i^3 + 2j^3)a_3 - 2(H_i - H_j) \\ \tfrac{\partial S}{\partial a_1} = \; & (2i + 2j)a_0 + (2i^2 + 2j^2 + 6)a_1 + (2i^3 + 2j^3 + 4i + 4P + 4j)a_2 + \\ & (2i^4 + 2j^4 + 6i^2 + 6P^2 + 6j^2)a_3 - 2(m_i + 2m_P + 2m_j + H_ii + H_jj) \\ \tfrac{\partial S}{\partial a_2} = \; & (2i^2 + 2j^2)a_0 + (2i^3 + 2j^3 + 4i + 4P + 4j)a_1 + \\ & (2i^4 + 2j^4 + 8i^2 + 8P^2 + 8j^2)a_2 + (2i^5 + 2j^5 + 12i^3 + 12P^3 + 12j^3)a_3 - \\ & 2(H_ii^2 + H_jj^2) - 4(im_i + Pm_P + jm_j) \\ \tfrac{\partial S}{\partial a_3} = \; & (2i^3 + 2j^3)a_0 + (2i^4 + 2j^4 + 6i^2 + 6P^2 + 6j^2)a_1 + \\ & (2i^5 + 2j^5 + 12i^3 + 12P^3 + 12j^3)a_2 + (2i^6 + 2j^6 + 18i^4 + 18P^4 + 18j^4)a_3 - \\ & 2(H_ii^3 + H_PP^3) - 6(i^2m_i + P^2m_P + j^2m_j). \end{aligned} \tag{2.55}$$

The equations above (2.55) are equated to zero. The system of equations is solved for the coefficients a_0, a_1, a_2 and a_3. The solution is given in Appendix A.

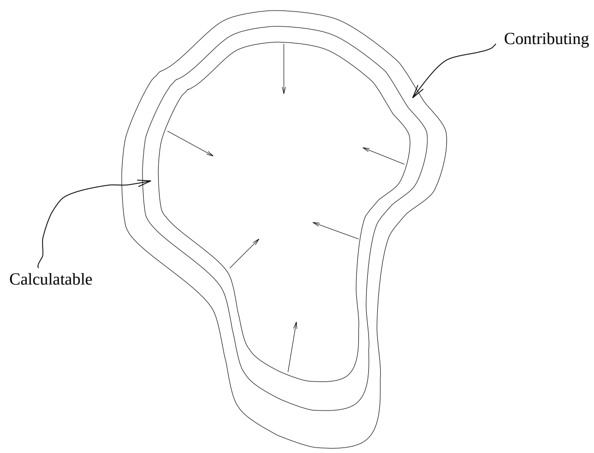


Figure 2.5: During the first wave, only the contour points can contribute. The adjacent interior points are set to be calculatable. All other interior points are in a wait state. The algorithm proceeds inwards; each iteration is called a wave.

Using the above coefficients and the following equation, the height at point P can be estimated:

$$H_P = a_3P^3 + a_2P^2 + a_1P + a_0. \tag{2.56}$$

2.4.4 Implementation Details

It was suggested in the previous section that second and third-order polynomials can be used to integrate height from surface normals. To perform this integration, four interpolation polynomials are constructed at each image point. These four interpolation polynomials estimate a new height for point P. The four estimates of the height at point P are averaged to determine a single new estimate.

Initially, the heights are assumed to be zero. If the above algorithm is iterated to determine the heights, the solution converges slowly. It was suggested by Vega [38] that the grassfire technique can be used to speed up the convergence.

Initially, the image points are assigned a height of zero. Each point is assigned a state. Points on the occluding contour are initialized to be in a *contributing* state. Points on the object immediately adjacent to the contour are set to be in a *calculatable* state. The remaining points on the object are set to be in a *wait* state.

The grassfire algorithm proceeds in waves (see Figure 2.5). During the first wave, points that are in a *calculatable* state are considered. Points immediately adjacent to the *calculatable* ones that are in a *contributing* state are used to interpolate the height. When all of the points in a *calculatable* state have been considered, the states are modified as follows: (1) points that have just been calculated are now set to be in a *contributing* state, (2) points adjacent to the just *calculated* states that are on the object and not yet calculated, are assigned a state of calculatable. This process continues until all points have been calculated. After all waves are complete, more iterations can occur with all neighbours at each point contributing. This algorithm is described in more detail in Figure 2.6.

```
{initialization }
for all points (x, y)
    if (x, y) is on object then State(x, y) = on object
       Height(x, y) = 0
       Buffer(x, y) = 0
for all points (x, y) on the occluding contour
    State(x, y) = contribute
    for all points (nx, ny) adjacent to (x, y)
       if State(nx, ny) = on object
            State(nx, ny) = calculatable
{main algorithm}
while more waves to be done do
    for all points (x, y)
       if State(x, y) = calculatable
            for all points (nx, ny) adjacent to (x, y)
                  if State(nx, ny) = contributable
                         Hi = Do polynomial height estimation
                  Buffer(x, y) = ∑(Hi) / numberofpoints
                  State(x, y) = next wave can contribute
                  for all points (nx, ny) adjacent to (x, y)
                         if State(nx, ny) = on object
                         State(x, y) =next wave calculatable
{ Update states for next wave }
    for all points (x, y)
       if State(x, y) = next wave can contribute
            State(x, y) = contribute
       else if State(x, y) = Next wave calculatable
            State(x, y) = calculatable
       Heights(x, y) = Buffer(x, y)
       Buffer(x, y) = 0
```

Figure 2.6: Grassfire approach that uses higher-order polynomial estimation to integrate height from gradient.

Chapter 3

Light Reflection

Most surfaces are neither ideally specular (mirror-like) nor perfectly diffuse (Lambertian). Early work in shape from shading assumed Lambertian reflectance properties, as this simplified the problem. However, shape-from-shading algorithms have to be developed to accommodate non-Lambertian surfaces.

A Lambertian surface reflects light equally in all directions. This is due to the nature of the object's surface. When the surface is very rough, incident light is scattered internally, and multiple reflections occur (see Figure 3.1). A perfectly smooth

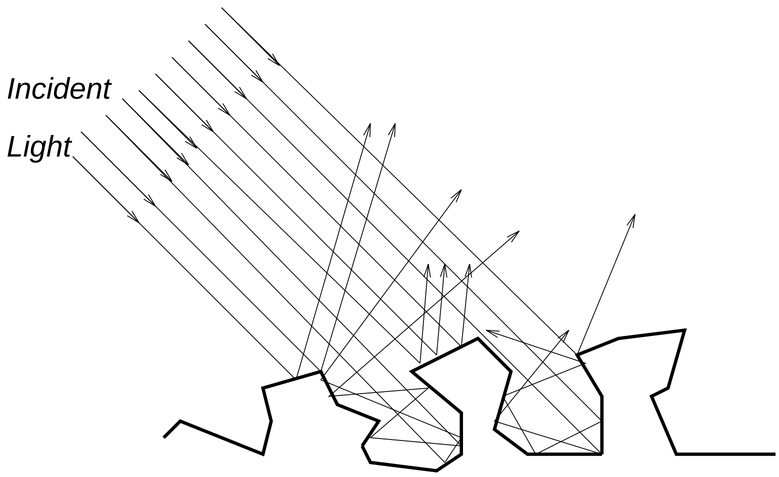


Figure 3.1: Rough surface scatters light in all directions

surface behaves like a mirror. Only those points whose surface normal is oriented in such a way to reflect light to the viewer are illuminated (see Figure 3.2).

However, most real surfaces are neither perfectly diffuse nor ideally specular, but are somewhere in between; reflecting light specularly as well as diffusely [42] (see Figure 3.2).

A physical model of light reflection is required to develop shape-from-shading algorithms that can reconstruct the shape of objects with realistic surfaces. Only by understanding image formation can effective shape-from-shading algorithms be developed.

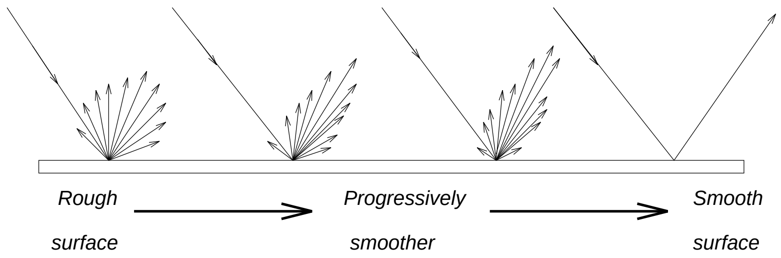


Figure 3.2: Diffuse and specular components exist in partly rough surfaces. The length of each arrow indicates the relative intensity of the light reflected in that direction

3.1 Models of Light Reflection

Physicists observe how light is reflected from rough surfaces experimentally and then try to develop physical models to account for their observations. Torrance and Sparrow [37] developed a technique that models surfaces as a collection of microfacets oriented in random directions. This method became very popular in computer graphics decades later. The Torrance-Sparrow [37] model is discussed in more detail in the next section.

Scientists in computer graphics tried to synthesize objects that had specular highlights. In 1969, Romney [33] suggested that the angle between the light source, $\mathbf{s}$, and the surface normal, $\mathbf{n}$, raised to a power c will give the impression of specular highlights:

$$\frac{(\mathbf{n} \cdot \mathbf{s})^c}{d^4} \tag{3.1}$$

where d is the distance from the viewer to the surface. This technique has no physical meaning and is thus of limited value.

Phong [32] formalizes a general model that incorporates ambient[1], diffuse, and specular reflection:

$$E(x, y) = K_a + K_d(\mathbf{n} \cdot \mathbf{s}) + K_s(\mathbf{w} \cdot \mathbf{s})^c \tag{3.2}$$

where K_a is the amount of ambient light, K_d is the diffuse component, K_s is the specular component and c is a measure of surface roughness. Note that $\mathbf{w}$ is a vector that estimates the direction of reflection. Also note that $K_d + K_s = 1$.

Phong [32] successfully incorporates the different components of light, but his calculation of specular reflection is not based on the physics of light reflection.

Blinn [4] was the first to use the Torrance-Sparrow [37] model in computer graphics. Blinn formulates the microfacet model in a convenient way for image synthesis. The approach is similar to that of Phong in the respect that ambient, diffuse, and specular reflection are calculated as a linear sum. However, the specular component is computed using the Torrance-Sparrow physical model of specular reflection:

$$\frac{DGF}{\mathbf{n} \cdot \mathbf{e}} \tag{3.3}$$

where D represents the microfacet distribution, G is the geometric attenuation function, and F is the Fresnel reflectance. These components are discussed in detail in the next section.

Cook and Torrance [10] expand on the work of Blinn [4] by formulating a model that can account for multiple light sources and colour images.

Recently, a new model has been proposed for computer vision. Nayar *et al.* [28] compare the Torrance-Sparrow [37] model to one developed by Beckmann and Spizzochino [3]. The Beckmann-Spizzochino model is unique, as it models light as a wave. This is a more complete albeit more complicated model. Nayar *et al.* [28] report that the models are similar except they observe that the Torrance-Sparrow model cannot

[1] A constant illumination arising from the environment which is independent of the light source is accounted for by the ambient constant.

account for a specular spike. They suggest the reason for this is that a microfacet model is only valid when the microfacets are larger than the wave length of the incident light. For very smooth surfaces, the microfacets will be infinitely small and the microfacet model will fail in this case. They propose adding an additional term to the Torrance-Sparrow model that can account for the specular peak.

For the purpose of this research, the Torrance-Sparrow model will be used. This model is physically-based and has been experimentally validated [37]. It is also the most widely accepted model in the literature.

3.1.1 Torrance-Sparrow Model

Torrance and Sparrow [37] developed a physical model of light reflection. Their model accounts for all of the light that is reflected from a surface patch. The incident light is reflected diffusely and specularly. The intensity at point $E(x, y)$ is the sum of two reflection components:

$$E(x, y) = K_d E_d + K_s E_s \tag{3.4}$$

where E_d is the intensity of the light reflected diffusely and E_s is the intensity of the light reflected specularly. Recall K_d and K_s are the diffuse and specular components, respectively. Also recall $K_d + K_s = 1$.

The surface is assumed to be a collection of mirror-like microfacets which are oriented in various directions. All the incident light that falls on the surface patch must be accounted for. If the surface is rough, multiple reflections will occur between facets. This phenomenon is described in Figure 3.1. This light will be visible to viewers in all directions. The intensity of light reflected diffusely is given by:

$$E_d = \mathbf{n}(x, y) \cdot \mathbf{s}. \tag{3.5}$$

This is simply the Lambertian law.

Some of the microfacets will be oriented to reflect light towards the viewer. This is the light that is reflected specularly. A portion of the incident light will be absorbed into the surface itself. This absorbed light can be calculated using the Fresnel equation. Finally, some of the incident light will be masked by the microfacets. This amount of light is estimated using the geometrical attenuation function. The Torrance-Sparrow [37] model calculates the specular reflection using the following function:

$$E_s = D\left(\frac{G}{\mathbf{n}\cdot\mathbf{e}}\right)F. \tag{3.6}$$

D (the distribution of the microfacets) indicates what proportion of the light is reflected specularly. G (the geometrical attenuation function) reveals how much of the incident light is shadowed by the facets, and F (the Fresnel equation) indicates what proportion of the incident light is absorbed into the surface. Each of these factors is addressed in turn.

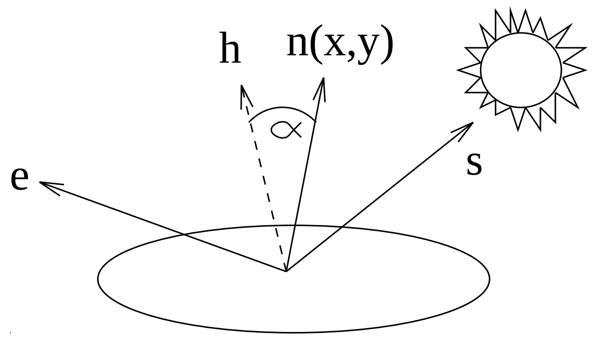


Figure 3.3: The scene geometry.

The microfacets oriented towards the viewer make up the specular component. If the surface is a perfect mirror, light would reach the eye only if the surface normal bisects the angle between the light source and the direction vector toward the viewer ($\mathbf{e}$). This reflection geometry is described in Figure 3.3. The bisector, $\mathbf{h}$, can be calculated using the following computation:

$$\mathbf{h} = \frac{\mathbf{e}+\mathbf{s}}{\|\mathbf{e}+\mathbf{s}\|}. \tag{3.7}$$

If the surface normal is oriented in the same direction as $\mathbf{h}$ there is a maximum specular component. For rough surfaces, the specular component falls off slowly as the angle between $\mathbf{n}(x, y)$ and $\mathbf{h}$ (call this angle α) increases.

The microfacets are modeled statistically. Only those microfacets oriented towards the maximum specular direction, $\mathbf{h}$, can contribute to the specular reflection. Although several functions have been used to model the microfacets statistically, the Gaussian is used in this work. The Gaussian can represent the percentage of the microfacets oriented at an angle α:

$$D = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\alpha^2}{2\sigma^2}}. \tag{3.8}$$

The factor σ is the standard deviation which can be interpreted as a measure of the roughness of the surface. A graph of this distribution is given in Figure 3.4. When the surface is smooth (e.g. $\sigma = 0.2$), the microfacets do not have much variation in their orientations. If the surface normal is in the same direction as $\mathbf{h}$, most of the light will be reflected towards the viewer. If the surface is rough (e.g. $\sigma = 0.6$), light is reflected from the microfacets towards the viewer independent of the surface orientation. This is due to the large variation in the directions of the microfacets.

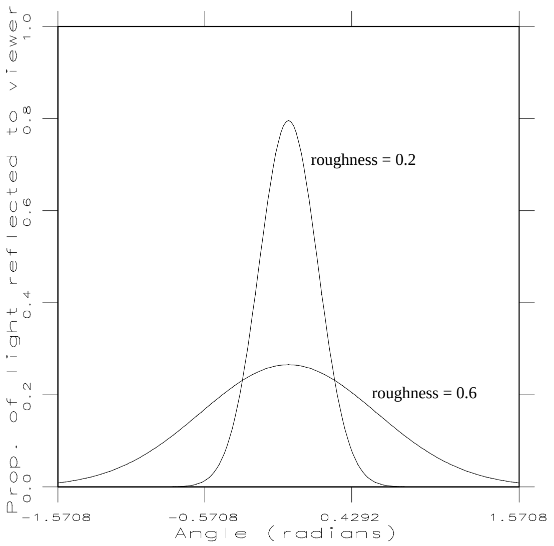


Figure 3.4: Plot of the Gaussian distribution.

The geometric attenuation function, G, accounts for the fact that the microfacets can shade or mask other parts of the surface. G is a value from 0 to 1 representing the proportion of the incident light that remains after masking. Torrance and Sparrow [37] model adjacent microfacets as forming V-groove cavities (see Figure 3.5). The amount of light masked is dependent on the shape of the V-grooved cavity as well as the angle to the viewer and the angle of the light source. Using geometry, Torrance and Sparrow [37] are able to derive a formula that calculates the proportion of the incident light that is masked on the object's surface. Blinn simplifies this expression to be:

$$G = min\left[1, \frac{2(\mathbf{n}\cdot\mathbf{h})(\mathbf{n}\cdot\mathbf{e})}{(\mathbf{h}\cdot\mathbf{e})}, \frac{2(\mathbf{n}\cdot\mathbf{h})(\mathbf{n}\cdot\mathbf{s})}{(\mathbf{h}\cdot\mathbf{e})}\right]. \tag{3.9}$$

For a complete discussion of the derivation of G see [4, 37].

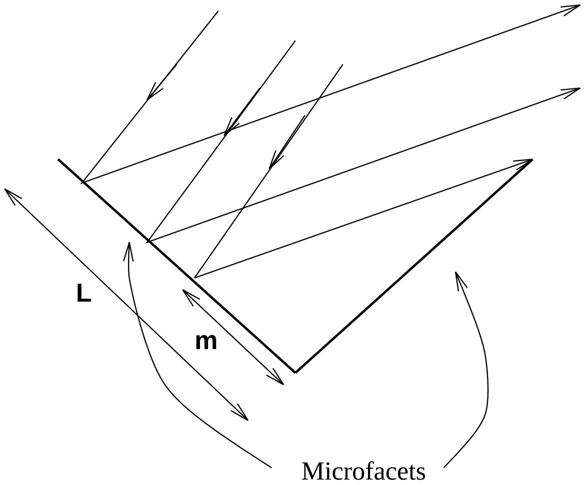


Figure 3.5: Masking can occur in the V-groove cavities. The proportion $m : l$ is used to estimate the amount of light that is masked.

Finally, F represents the proportion of incident light reflected as opposed to absorbed into the surface. This value can be obtained from the Fresnel equation:

$$F = \frac{1}{2}\left[\frac{sin^2(\phi-\theta)}{sin^2(\phi+\theta)} + \frac{tan^2(\phi-\theta)}{tan^2(\phi+\theta)}\right] \tag{3.10}$$

where ϕ is the angle of incidence, n is the index of refraction and $sin\theta = \frac{sin\phi}{n}$.

It has been shown experimentally that for most surfaces, the amount of light that is absorbed into the surface is minute. It has also been shown that the amount of light that is masked by the facets shadowing each other is also negligible. For the purposes of this research these factors (G, F) will not be used. The light reflection model then becomes:

$$E(x,y) = [K_d \mathbf{n}(x,y) \cdot \mathbf{s}] + \left[K_s \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\alpha^2}{2\sigma^2}} \right]. \tag{3.11}$$

3.2 Shape Recovery for Non-Lambertian Surfaces

Shape from shading and photometric-stereo are both shape recovery techniques. Most shape recovery methods require prior knowledge of the object's surface reflectance properties. Some algorithms assume that a calibration object of known reflectance is present in the scene. As this is an impractical assumption, other algorithms assume that the object's surface is Lambertian. However, most surfaces are not perfectly diffuse, thus it is essential that new shape recovery techniques are formulated for non-Lambertian surfaces.

Several photometric-stereo algorithms have been proposed for recovering the shape of non-Lambertian objects [9, 22, 27, 35]. The photometric-stereo method requires control of the lighting. Multiple images of the same scene have to be provided in which the light source is at different orientations. Shape recovery from a single image (shape from shading) is a more challenging problem but also more practical, as having control of the lighting is sometimes impossible.

Very few shape-from-shading algorithms have been presented that can recover the shape of non-Lambertian surfaces. Brooks and Horn [6] propose a shape-from-shading algorithm for a general reflectance map. Their formulation is based on the calculus of variations. The Torrance-Sparrow reflectance function is incorporated into this formulation in Chapter 4.

The variational formulations have inherent problems which were discussed in Subsection 2.3.2. These problems are present in this algorithm. However, this formulation requires the availability of $R_{\mathbf{n}}(\mathbf{n}_{(i,j)})$ which is the partial derivative of R with respect to $\mathbf{n}$. For complicated reflectance maps (physically-based) this derivative will not be easy to determine.

Healey and Binford [16] propose a new shape-from-shading algorithm that recovers shape from specular surfaces. Their approach uses the specular highlight to determine curvature properties of the surface. The point of maximum intensity in the image indicates the surface orientation at that point uniquely. At this point the angle of incidence will equal the angle of reflection, thus $\mathbf{n} = \mathbf{h}$. The reflection model they use is also based on the Torrance-Sparrow [37] model but the distribution function is slightly different. The specular component for their model is:

$$E_s = K_s e^{-\frac{\alpha}{\sigma}^2}. \tag{3.12}$$

They identify the specular highlight as the region surrounding the point of maximum intensity. This region, called the specular irradiance image, is easily identified as the specular highlight has a high contrast. The specular irradiance image is then transformed into the α angle image using the following equation:

$$|\alpha| = \sigma\sqrt{-ln\frac{E_s}{K_s}}. \tag{3.13}$$

Local curvature properties are computed from the α angle image by taking a finite number of straight line samples intersecting the point of maximum intensity. The curvature is then extrapolated for the part of the object outside of the specular highlight.

This approach assumes that the specular highlight is easily detected. In some cases the object will be diffuse so that there is no specular highlight, but yet smooth enough that it cannot be modeled using Lambertian reflectance properties. In these cases the behavior of the Healey-Binford [16] is hard to predict.

It is clear that new algorithms must be developed to recover the shape of non-Lambertian objects from a single image, which is the focus of the proposed work.

Chapter 4

Shape From Shading For Non-Lambertian Surfaces

Research conducted in shape from shading has focused on recovering the shape of objects whose surfaces obey Lambertian reflectance properties. The physical models behind Lambertian reflectance are well understood and easy to express mathematically. However, most surfaces are non-Lambertian. New algorithms must be developed to reconstruct the shape of specular surfaces. This problem is much more difficult, as the underlying physical properties of reflectance from rough surfaces is not as well understood. The mathematical models used to explain non-Lambertian surfaces [37] are relatively more complex and more difficult to manipulate. The algorithms derived in the next section reflect this fact. Because the models for specular surface are complex, the algorithms used to reconstruct the shape of non-Lambertian surfaces are slightly less elegant than their Lambertian counterparts.

4.1 Variational Algorithm for Non-Lambertian Surfaces

Brooks and Horn [6] propose a shape-from-shading algorithm for a general reflectance map. They argue that the desired surface should minimize the following integral:

$$\int\int \left((E - R(\mathbf{n}))^2 + \lambda(\|\mathbf{n}_x\|^2 + \|\mathbf{n}_y\|^2) + \mu(x,y)(\|\mathbf{n}\|^2 - 1)\right) dxdy. \tag{4.1}$$

In minimizing the above functional the following Euler equation is obtained:

$$(E - R)R_{\mathbf{n}} + \lambda\nabla^2\mathbf{n} - \mu\mathbf{n} = 0. \tag{4.2}$$

This equation must be satisfied for an extremum to occur. After discretization of the Laplacian, the following iterative scheme is derived:

$$\begin{aligned} \mathbf{m}_{i,j}^{(k+1)} &= \bar{\mathbf{n}}_{i,j}^{(k)} + \tfrac{\epsilon^2}{4\lambda}\left(E_{i,j} - R(\mathbf{n}_{i,j})\right) R_{\mathbf{n}}(\mathbf{n}_{i,j}) \\ \mathbf{n}_{i,j}^{(k+1)} &= \frac{\mathbf{m}_{i,j}^{(k+1)}}{\left\|\mathbf{m}_{i,j}^{(k+1)}\right\|} \end{aligned} \tag{4.3}$$

where $\bar{\mathbf{n}}_{i,j}^{k}$ is the average normal centered around pixel i, j at iteration k. This formulation requires the availability of $R_{\mathbf{n}}$ which is the derivative of the reflectance map with respect to $\mathbf{n}$. However, for complicated reflectance maps, $R_{\mathbf{n}}$ may be difficult to determine.

Using the Torrance-Sparrow model of reflectance [37], R($\mathbf{n}$) is:

$$R(\mathbf{n}) = [K_d\mathbf{n}(x,y)\cdot\mathbf{s}] + \left[\left(K_s\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\alpha^2}{2\sigma^2}}\right)\right]. \tag{4.4}$$

Taking the derivative of $R(\mathbf{n})$ with respect to $\mathbf{n}$ yields:

$$R_{\mathbf{n}} = K_d\mathbf{s} + \frac{K_s}{\sqrt{2\pi}\sigma}e^{-\left(\frac{\arccos(\mathbf{n}\cdot\mathbf{h})^2}{2\sigma^2}\right)}\frac{arccos(\mathbf{n}\cdot\mathbf{hh})}{\sigma^2\sqrt{1-(\mathbf{n}\cdot\mathbf{h})^2}}. \tag{4.5}$$

$R_{\mathbf{n}}$ is substituted into the above algorithm (4.3). This is the first time the Torrance-

Sparrow reflectance model has been incorporated into a variational-based shape-from-shading algorithm. The performance of the Variational Algorithm for Non-Lambertian Surfaces (hereafter referred to as V.A.N.L.) is described in a later chapter.

4.2 General Shading Logic Algorithm

Many variational-based shape-from-shading algorithms have been proposed that recover the shape of Lambertian objects. These algorithms are very difficult to modify to incorporate non-Lambertian surfaces. The only way to extend these approaches is to return to the basic variational formulation and reformulate the problem. Depending on the complexity of the reflectance model, the resulting equation may or may not be solvable.

The Shading Logic algorithm developed by Vega and Yang [39] is the only known heuristic-based approach to recover the shape of an object from its shading. This method is generalized to cope with any reflectance map. The heuristic-based method proves more easily modified than variational-based approaches.

The heuristics for the shading logic algorithm are derived by interpreting Brooks and Horn's algorithm [6] geometrically. In Brooks and Horn's algorithm the functional:

$$\int\int \Big((E - \mathbf{n}\cdot\mathbf{s})^2 + \lambda(\|\mathbf{n}_x\|^2 + \|\mathbf{n}_y\|^2) + \mu(x,y)(\|\mathbf{n}\|^2 - 1)\Big)\, dxdy \tag{4.6}$$

is minimized. The following iterative equation results from the Euler equation that satisfies the functional (4.6):

$$\mathbf{n}_{i,j}^{(k+1)} = \bar{\mathbf{n}}_{i,j}^{(k)} + \frac{\epsilon^2}{4\lambda}(E_{i,j} - \mathbf{n}_{i,j}^{(k)}\cdot\mathbf{s})\mathbf{s}. \tag{4.7}$$

This can be expressed in a simplified form if it is assumed that the solution is already smooth, that is, $\mathbf{n}_{i,j} = \bar{\mathbf{n}}_{i,j}$, then Equation (4.7) becomes

$$\mathbf{n}_{i,j}^{(k+1)} = \mathbf{n}_{i,j}^{(k)} + \delta \xi \mathbf{s}, \tag{4.8}$$

where the error between pixel intensity and calculated intensity is $\xi(\; E_{i,j} - \mathbf{n}_{i,j}^{(k)} \cdot \mathbf{s})$ and $\delta = \frac{\epsilon^2}{4\lambda}$ is a constant. From Equation (4.8) it is clear that the surface normal calculated on iteration $(k+1)$ is a vector sum between the surface normal on iteration (k) and the light source direction vector. This means that the calculated normal must lie on the plane formed by these two vectors. Similar to the Shading Logic algorithm, the General Shading Logic algorithm (hereafter referred to as G.S.L.) relaxes this constraint.

To simplify the discussion, it is assumed that the normal $\mathbf{n}(x, y)$ at the point P is required to be updated. The inputs to the algorithm are: (1) the surface normals on the occluding contour, (2) the shaded image, (3) the light source direction vector, (4) the viewer direction vector, and (5) the roughness parameters. The three basic operations of the G.S.L. algorithm are now discussed. These three operations each suggest a new normal. This new normal (calculated at iteration $(k + 1)$) is evaluated to determine if it is better than the previous normal (calculated at iteration (k)). The criterion for evaluation: *is the brightness that the new normal would generate more similar to the given image intensity than the previous normal?* The reflectance map is used to determine the intensity that the normal would generate. Stated mathematically:

$$\left| E(x, y) - R(\mathbf{n}^{(k+1)}) \right| \leq \left| E(x, y) - R(\mathbf{n}^{(k)}) \right|. \tag{4.9}$$

Three operations suggest a new normal and use Equation (4.9) to determine the quality of the proposed normal.

1. **Weighted Neighbour Operation (W)**

   The G.S.L. algorithm ranks the neighbours of P based on their credibility.

Credibility is defined as:

$$w_i = \frac{1}{|E(x, y) - R(\mathbf{n})| + 1}. \tag{4.10}$$

If the intensity calculated using the ith neighbour of P produces an intensity closer to that at P than the shading estimator using the jth neighbour of P, then the ith neighbour is given a higher credibility than the jth neighbour.

The calculated normal at P will be a weighted average of the neighbours of P. Brooks and Horn's algorithm [6] gives equal weight to each neighbour. It is important to note that this simple weighting scheme does not require the normals to be on the solution plane and can achieve smoothness in an implicit manner.

2. **Minor Slant Adjustment (MS)**

   The weighted neighbour method described above can get "stuck" if the neighbour's surface normals do not vary in direction. It is necessary to propose a new normal that varies considerably and determine if this new normal will lead to a better solution. The original Shading Logic algorithm changed the slant of the normal by ten degrees. In particular, two normals are constructed one ten degrees closer to the z-axis and one ten degrees farther away from the z-axis. As there is no real justification for choosing ten degrees as the angle to perturb the surface normal, the G.S.L. algorithm generates two normals at 5, 10, 15, 20, and 25 degrees. At each angle two different normals are constructed, one slightly closer to the z-axis and one farther away. These two surface normals are generated by elongating and shortening the projection of the the normal onto the xy-plane. The two generated normals are tested to determine if they can produce a smaller intensity error than the current normal using the method described in Equation (4.9).

3. **Normal Complementary Operation (NC)**

   A potential problem with variational approaches is that there exists the possibility that the solution may get caught at a local minimum or maximum. A simple solution to this problem is the normal complementary operation. The goal of the normal complementary operation is to change the convexity of the reconstructed surface.

The first step in the normal complementary operation is to project the surface normal onto the xy-plane, this projected normal is $\mathbf{n}_p$. The complementary normal, $\mathbf{f}_p$, will have magnitude

$$\|\mathbf{f}_p\| = 1 - \|\mathbf{n}_p\|. \tag{4.11}$$

The projected normal is computed as:

$$\mathbf{f}_p = \frac{\|\mathbf{f}_p\|}{\|\mathbf{n}_p\|}\mathbf{n}_p. \tag{4.12}$$

Because the complementary normal is unitary, the three-dimensional normal can be computed from its projection. At each step the complemented normal is compared with the current one using the method described in Equation (4.9). The normal which produces the least error is selected.

For the reflectance model discussed in Section 3.1.1, the reflectance map is:

$$R(\mathbf{n}) = [K_d \mathbf{n}(x,y) \cdot \mathbf{s}] + \left[K_s \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\alpha^2}{2\sigma^2}}\right]. \tag{4.13}$$

4.3 Summary of Shape-from-Shading Approaches for Non-Lambertian Surfaces

Two new algorithms have been proposed to recover the shape of non-Lambertian objects. The variational approach is an extension of Brooks and Horn's work, as the Torrance-Sparrow reflectance model has been incorporated into the variational formulation. The second algorithm presented, the G.S.L., is a generalization of the Shading Logic algorithm [39].

The heuristic-based algorithm (Shading Logic [39]) proved to be more easily modifiable. Brooks and Horn's algorithm, for a general reflectance map, requires the derivative with respect to the reflectance map ($R_{\mathbf{n}}$). When dealing with complicated reflectance functions, calculating $R_{\mathbf{n}}$ can be difficult. A heuristic-based shape-from-shading algorithm for a general reflectance map is presented (G.S.L. algorithm). Incorporating different reflectance maps into this formulation is trivial. The performance of the G.S.L. algorithm and V.A.N.L. are compared in a later chapter.

The two algorithms developed in this section require the reflectance coefficients and the illuminant direction to be available. However, it may be impractical to assume that these parameters are available. The next section deals with techniques that can be used to estimate the parameters prior to the shape recovery process.

Chapter 5

LIRAS Recovery System

Shape-from-shading algorithms presented in the literature assume the reflectance map is available. That is, these algorithms assume the surface is diffuse (Lambertian) and that the light source direction is known. More recently, researchers have proposed algorithms that estimate the light source direction [5, 6, 24, 29, 40, 43], however, these techniques still assume Lambertian reflectance.

The focus of this thesis is non-Lambertian surfaces. Because a more general class of surfaces is considered, complexities are introduced that are not present when considering Lambertian reflectance. The Torrance-Sparrow model of reflectance [37] introduces two additional parameters: (1) the coefficients ($K_d + K_s = 1$) and (2) the distribution of the microfacets (σ). The light source direction also has to be available for the reflectance map to be defined.

It is not practical to assume that these parameters are available to the shape-from-shading algorithms in all cases. The LIRAS (LIght, Reflectance, And Shape) Recovery System includes techniques that estimate the reflectance map and use this information to recover the shape.

In a practical application of shape-from-shading, a photograph may be taken with many objects in the image. If a calibration object is available in the scene, the illuminant direction and the reflectance of the surface can be estimated using the algorithm proposed in Section 5.3. If no calibration object is identifiable, the illuminant direction can be estimated if the reflectance parameters of one object are known. This algorithm, called the Contour-Based Illuminant Locator, is presented in Section 5.2. Once the light source direction has been found, the shape of each object in the scene has to be determined. It is not practical to assume that all of the objects in the scene are diffuse, thus an algorithm is proposed that can estimate the reflectance parameters of each object given the illuminant direction. This algorithm, called the Region-Based Reflectance Estimator, is proposed in Section 5.1. Once the reflectance parameters of each object are known, and the illuminant direction for the entire scene is available, the algorithms proposed in Chapter 4 can be used to reconstruct the shape of each object in the scene. The LIRAS system includes algorithms that can recover the shape of each non-Lambertian object in a scene, given only one piece of information (either a calibration object or the reflectance parameters of one object).

5.1 Region-Based Reflectance Estimator

An algorithm has yet to be proposed that can recover the shape of an object without a reliable estimate of the reflectance map. The experiments described in Subsection 7.2.1 demonstrate the importance of the reflectance map in estimating surface shape. In these experiments, traditional algorithms formulated for Lambertian surfaces are used to recover the shape of non-Lambertian surfaces. As expected, the algorithms designed for diffuse objects cannot effectively recover the shape of specular surfaces. As it is not practical to assume that reflectance information is available in all situations, it is necessary to develop new techniques that can estimate the reflectance map.

In an image of a single object, there are two identifiable regions that hold useful information about the reflectance map. Recently, methods have been developed that can remove specular highlights prior to the shape recovery process [12]. In this work, the specular highlight is used as an invaluable resource in estimating the reflectance parameters. In this region, the Torrance-Sparrow reflectance function [37] can be simplified. Also, along the occluding contour, the surface orientations are defined to be on the image plane. This additional information is exploited to estimate σ.

5.1.1 Estimating K_d on the Occluding Contour

As discussed in Section 2.2, the surface normals are known along the occluding contour. This additional information can be exploited in an algorithm to recover the reflectance coefficient, K_d. An algorithm similar in spirit to the one presented here is that of Zheng and Chellappa [43]. However, they use the occluding contour to estimate the light source direction. Also Zheng and Chellappa's algorithm is only useful for surfaces with Lambertian reflectance.

Initially, the points on the occluding contour are assumed to be diffuse. This is a reasonable assumption that is relaxed in subsequent iterations. The specular highlight is the region where the surface normals are oriented in such a way that they reflect light directly to the viewer. The surface normals on the closed occluding contour will vary considerably. It is unlikely that the majority of the normals on the occluding contour will be oriented in such a direction as to reflect light directly to the viewer. Thus, it is reasonable to initially assume that the occluding contour is diffuse.

In a diffuse region, the Torrance-Sparrow model of light reflection [37] can be reduced from:

$$E(x,y) = K_d(\mathbf{n}\cdot\mathbf{s}) + K_s\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{arccos(\mathbf{n}\cdot\mathbf{h})}{2\sigma^2}} \tag{5.1}$$

to

$$E(x,y) = K_d(\mathbf{n}\cdot\mathbf{s}). \tag{5.2}$$

Since $\mathbf{s}$ is given, and $\mathbf{n}$ is known along the occluding contour, K_d can be estimated by:

$$K_d = \frac{E(x,y)}{(\mathbf{n}\cdot\mathbf{s})}. \tag{5.3}$$

Clearly, the occluding contour consists of many points. Thus, the calculation of K_d is overdetermined. The technique employed to deal with this overdetermined system is robust statistics.

For each point along the occluding contour, there will be an estimate of K_d.

After sorting the estimates in ascending order, a least median of squares (LMS) fit is employed [34]. It is assumed that there are an even number of estimates. Let $K_{d_1} - K_{d_m}$ represent the first set of estimates (the half with the lesser-valued estimates) and $K_{d_{m+1}} - K_{d_n}$ represent the second set of estimates (the half with the greater-valued estimates):

$$\underbrace{K_{d_1}, K_{d_2}, ..., K_{d_{m-1}}, K_{d_m}}_{lesser}, \underbrace{K_{d_{m+1}}, K_{d_{m+2}}, ..., K_{d_{n-1}}, K_{d_n}}_{greater}. \tag{5.4}$$

m differences are computed in the following fashion:

$$\begin{aligned} d_1 &= K_{d_n} - K_{d_m} \\ d_2 &= K_{d_{n-1}} - K_{d_{m-1}} \\ d_3 &= K_{d_{n-2}} - K_{d_{m-2}} \\ &\vdots \\ d_m &= K_{d_{m+1}} - K_{d_1}. \end{aligned} \tag{5.5}$$

The minimum difference is found:

$$d_{min} = min\{d_1, d_2, ..., d_m\}. \tag{5.6}$$

The final estimate of K_d is the average of the estimates of K_d that comprise the minimum difference. For example, if d_2 is the minimum difference then the estimate of K_d would be:

$$K_{d_{est}} = \frac{K_{d_{n-1}} + K_{d_{m-1}}}{2}. \tag{5.7}$$

This technique is more robust than a simple averaging formulation, as up to one-half of the estimates can be outliers, yet the mean will be reliable [34]. Thus, the points on the occluding contour that reflect light in the specular direction will be considered outliers and not have an impact on the solution. However, subsequent iterations will not assume that the occluding contour region is Lambertian.

A method has been described that can give an initial estimate of K_d, however, it has been assumed that the occluding contour is Lambertian. When σ is large, the microfacets at a point are distributed in many directions. Even if the points on the occluding contour are not in the specular highlight, it does not mean they cannot reflect light specularly when σ is large. Thus, it is necessary to refine the calculation of K_d to handle the full Torrance-Sparrow reflectance function [37].

Assume that once K_d is found, there exists a method of estimating σ (such an algorithm is described in Subsection 5.1.2). How can the estimate of σ be used to refine the estimation of K_d?

The complete Torrance-Sparrow reflectance model is used (Equation 5.1). Noting that $K_s = 1 - K_d$:

$$K_d = \frac{E(x,y) - \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\alpha^2}{2\sigma^2}}}{(\mathbf{n}\cdot\mathbf{s}) - \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\alpha^2}{2\sigma^2}}} \tag{5.8}$$

The incorporation of σ is used to obtain a more accurate estimate of K_d. The LMS approach is again employed to determine a single estimate of K_d using all of the estimates on the occluding contour.

This iterative process of using K_d to estimate σ and then using σ to estimate K_d can continue for several steps. Initially assuming the occluding contour is Lambertian gives a *ballpark* estimate of K_d which bootstraps the whole process.

5.1.2 Estimating σ in the Specular Highlight

The specular highlight is the region where maximum light is reflected directly to the viewer. In this region, the surface normals ($\mathbf{n}$) are oriented along $\mathbf{h}$ so that the surface behaves like a mirror reflecting maximum light to the viewer (see Figure 3.3). In this region the Torrance-Sparrow reflectance function can be simplified.

In the specular highlight it is known that $\mathbf{n} = \mathbf{h}$. This is a very useful observation as $arccos(\mathbf{n}\cdot\mathbf{h}) = 0$ in this region. The reflectance function given in Equation (5.1) is reduced to:

$$E(x,y) = K_d(\mathbf{n}\cdot\mathbf{s}) + (1 - K_d)\frac{1}{\sqrt{2\pi}\sigma} \tag{5.9}$$

in the specular highlight. The coefficient K_d is available from the occluding contour algorithm described in Subsection 5.1.1. Isolating σ, Equation (5.9) becomes:

$$\sigma = \frac{(K_d - 1)(\mathbf{n} \cdot \mathbf{s})}{E(x,y)\sqrt{2\pi}}. \tag{5.10}$$

The specular highlight can contain many points making the problem of estimating σ overdetermined. The LMS strategy described in Subsection 5.1.1 is again employed.

5.1.3 Region-Based Reflectance Estimator Summary

A new approach at estimating reflectance parameters given the light source direction has been derived. The approach uses the occluding contour to estimate the reflectance coefficient K_d and the specular highlight to estimate σ. These two regions are selected because they represent an area on the object where the Torrance-Sparrow reflectance model can be simplified. Initially, the occluding contour is assumed to be diffuse. With this simplification, an estimate of K_d can be calculated. Call this initial estimate $K_d^{(0)}$. Using $K_d^{(0)}$, the specular highlight is used to gain an estimate of σ (call the initial estimate $\sigma^{(0)}$). The estimate $\sigma^{(0)}$ can be used to refine the estimate of K_d using the occluding contour. This new estimate of K_d is $K_d^{(1)}$. It seems logical to use $K_d^{(1)}$ to improve the estimate of σ. This iterative process can continue until the solution converges to stable estimates of K_d and σ. A stable estimate of the reflectance parameters occurs when the values of K_d and σ remain constant from one iteration to the next.

5.2 Contour-Based Illuminant Locator

A complete reflectance map includes both reflectance parameters and light source orientation. In this section, the problem of determining the light source direction given the reflectance parameters is considered. Several researchers [5, 6, 24, 29, 40, 43] have addressed the similar problem of determining light source orientation assuming diffuse reflectance, however, dealing with non-Lambertian surfaces is more difficult.

Zheng and Chellappa [43] use the occluding contour to estimate the azimuth and slant of the light source. They are able to isolate the light source direction and use the known surface orientations along the contour to constrain the problem. A similar approach is difficult to formulate for the non-Lambertian case. The Zheng and Chellappa algorithm is an elegant search strategy that can locate an estimate of the light source direction. Because of the complexity of the Torrance-Sparrow model, the search strategy used here is less elegant but still effective.

As stated in Section 2.2, the surface orientations are known along the occluding contour. The reflectance parameters (K_d, σ) are also available. Thus, the only unknown in the Torrance-Sparrow reflectance function is the light source direction. At any point on the image, numerous light source orientations can satisfy the reflectance function. That is, given the reflectance parameters and surface normal, several light source orientations can equate the Torrance-Sparrow reflectance function to the given pixel intensity. However, as more and more points are considered, the number of light source orientations that are candidates decreases. Although there is no guarantee of uniqueness, it is possible to use a sufficient number of points to obtain a good estimate of the light source orientation.

This is the motivation for the algorithm. At each point on the occluding contour, the light source azimuth and slant are varied incrementally to determine which orientations can satisfy the reflectance function. Each light source orientation that satisfies the reflectance function obtains a vote. When all light source orientations have been considered, the next point on the occluding contour is considered. After all of the points on the occluding contour have been used, the light source direction with the most votes is the estimation of the light source direction. In the event that several light source orientations have the same number of votes, a coarse-to-fine search strategy can be used to limit the number of orientations. A pseudo-code algorithm is presented in Figure 5.1.

5.3 Estimating Reflectance Map Using a Calibration Object

A calibration object is an object with a known shape that is placed in the scene and is used to determine the reflectance map. However, in a scene with several objects whose surfaces have different reflectance properties, a calibration object is of limited value. The reflectance map obtained from the calibration object is not necessarily applicable to all objects in the scene (the reflectance parameters can be different). Of more value is a technique that can estimate the light source direction using a calibration object. The light source will be the same for all objects in the scene. Thus, for each of the other objects in the scene, the algorithm described in Section 5.1 can be employed to recover the reflectance parameters of an object given the light source direction. In this manner a reflectance map for each object in the scene can be obtained.

The algorithm presented in this section can recover the reflectance map given a calibration object. Even though the reflectance parameters of the calibration object may not be useful for the other objects in the scene, they must be calculated in order to calculate the light source direction.

Again, because of the complexity of the reflectance function, a coarse-to-fine search strategy (similar to the one used to determine light source direction given reflectance) is employed. There are two important differences between the algorithm presented here and the approach proposed in Section 5.2: (1) the shape is known everywhere on a calibration object, not just on the occluding contour, (2) the reflectance parameters must be varied as well as the light source orientation. Other than that, this algorithm is similar to the one presented in Figure 5.1.

```
{Vary the light source at each point to determine which
 illuminant directions are candidates}

 for all points (x,y) on the occluding contour do
     for slnt = 0 to 2π step slantinc
         for tlt = 0 to π/2 step tiltinc
             compute light source (s) with slant= slnt and tilt= tlt
             E_est = Calculate intensity using Torrance-Sparrow reflectance
                   function and n(x,y), s, K_d, σ
             if E(x,y) = E_est
             then Votes(slnt,tlt)++

{Determine which illuminant direction has the greatest
 number of votes}

 find maximum of Votes(slnt,tlt)
 return(slnt,tlt)
```

Figure 5.1: Pseudo-code for algorithm to estimate light source direction given reflectance parameters.

Chapter 6

Experimental Evaluation

A complete experimental evaluation was conducted to understand the effectiveness of the algorithms proposed in this thesis. Both synthesized and real images were used.

Conducting experiments using synthesized images has several advantages:

- Experimental parameters can be easily controlled in a synthetic environment, thus the effects of varying one parameter can be studied while the other variables are held constant. The value of conducting experiments in a synthetic world is apparent when studying the effects of reflectance on the shape recovery process. Surface reflectance properties are varied systematically and the behaviour of the algorithms is studied. In a real setting, varying the reflectance of the surfaces is very difficult.

- It is difficult to justify why a specific set of real images is used over another.

- Information required for evaluating synthetic images is readily available. In the case of the shape-from-shading experiments, an effective way of evaluating the algorithms is to compare the normals and heights of the real object to those obtained using the shape-from-shading algorithms. When dealing with real objects, the height and surface orientation data are difficult to obtain.

In this thesis, superquadric surfaces are synthesized for the purpose of experimentation. These superquadrics model surfaces with properties that are representative of commonly-seen objects. Vega and Yang [39] propose using superquadrics to evaluate shape-from-shading algorithms because the shape of the superquadrics can be varied by changing some controlled parameter. Thus, the behavior of the shape-from-shading algorithms can be studied as a function of the shape of an object. However, superquadrics are symmetrical in three dimensions. It is suggested in this thesis that performing experiments on less symmetrical objects is important in understanding the complete behaviour of the shape-from-shading process. Surfaces that are formed by rotating a curve around an axis are used, as they have one less degree of symmetry. These *surfaces of revolution* are also representative of commonly-seen objects (cylinders, cones).

It is also important to verify that the algorithms proposed in this thesis are effective when using real images. Because of the difficulties outlined above, a complete experimental procedure for real images is not included in this work. The purpose of conducting experiments using real images is to demonstrate that the algorithms are effective in real world situations.

6.1 Superquadratic Surfaces

A set of functions called superquadrics are employed because of their ability to change shape dramatically by "tweaking" parameters.

- **Superellipsoids**

  The surface of the superellipsoid can be generated using the following equation [2]:

$$f(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{e_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{e_2}} \right)^{\frac{e_2}{e_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{e_1}}. \qquad (6.1)$$

  In the above equation a_1, a_2, a_3 control the length, width, and height of the object, respectively. The parameters e_1 and e_2 control the sharpness of the object. When $e_1 = e_2 = 1$, Equation (6.1) reduces to the standard equation of the ellipse.

- **Supertoroids**

  The following is a function for the supertoroid [2]:

$$f(x,y,z) = \left(\left(\left(\frac{x}{a_1}\right)^{\frac{2}{e_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{e_2}} \right)^{\frac{e_2}{2}} - a_4 \right)^{\frac{2}{e_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{e_1}}. \tag{6.2}$$

  In the above equation a_1, a_2, a_3 control the length, width, and height of the supertoroid respectively. a_4 is defined as follows:

$$a_4 = \frac{\hat{a}}{\sqrt{a_1^2 + a_2^2}}, \tag{6.3}$$

  where $\hat{a}$ is the radius of the supertorus. As in the equation for the superellipsoid, the parameters e_1 and e_2 control the sharpness of the object. When $e_1 = e_2 = 1$, Equation (6.2) reduces to the standard equation of a toroid.

The surface normals of the superellipsoids and supertoroids can be computed using the partial derivatives of Equations (6.1) and (6.2) respectively. The vector $(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, -1)$ represents the surface normal at any point on the xy-plane.

The superquadric equations describe the solids at standard positions and orientations. If it is assumed that the viewing direction is $(0,0,1)$, the superquadrics can be transformed by rotating the image plane as well as the viewing direction vector. The points on the rotated image plane are then projected along the new direction vector (using orthographic projection) and the superquadrics functions are queried to determine where an intersection occurs. This type of raytracing is described in [8, 15]. A sample of these rotated superquadrics is given in Figure 6.1

6.2 Surfaces of Revolution

The superquadric equations can model surfaces with properties that are representative of commonly-seen objects. However, the standard superquadrics are symmetrical in three dimensions. A wide variety of commonly-shaped objects are less symmetrical (cylinders, cones). In this thesis, it is proposed that performing experiments on

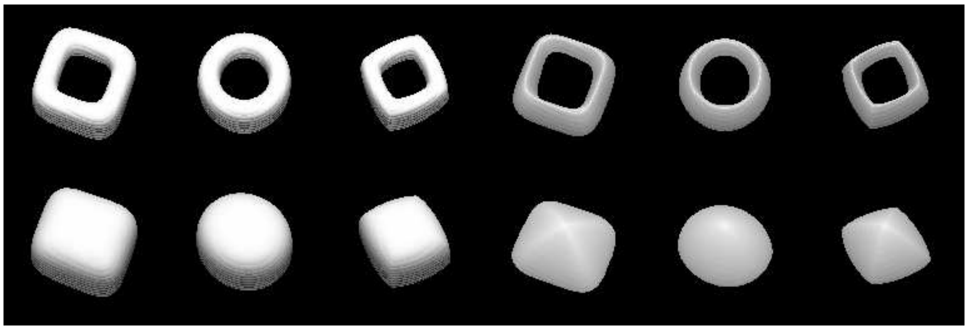


Figure 6.1: Sample of superellipsoids and supertoroids. Parameters e_1 and e_2 range from 0.5 to 1.5 in steps of 0.5. The a_1, a_2, a_3 values for the supertoroids (top row) are 10, 10 and 30 respectively. The radius is 40. The a_1, a_2, a_3 values for the superellipsoids are 40, 35 and 50 respectively. All objects are rotated -20 degrees around the x-axis and z-axis.

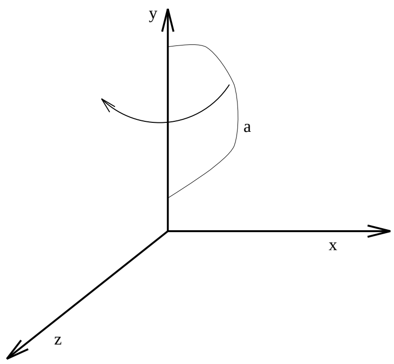


Figure 6.2: A surface is formed by rotating curve a around the $y-$axis.

surfaces of revolution is valuable, as the effectiveness of the algorithms can be studied on less symmetrical objects. The shapes of the surfaces of revolution are significantly different than the superquadrics. When dealing with superquadric equations, all of the normals on the occluding contour lie on the image plane. However, the occluding contour normals for the surfaces of revolution are not necessarily on the image plane. Several of the algorithms presented in this thesis use the occluding contour region as an important clue for the shape recovery process. It is important to understand the behaviour of these algorithms in all scenarios.

A surface of revolution is formed by rotating a curve around an axis (see Figure 6.2). The heights and slopes are known on the original curve $x = f(y)$. The original curve is rotated around the y-axis 360^o. The surface normals can be determined on the surface of revolution by rotating the surface normal on the original curve.

It is important to be able to understand the behaviour of the shape-from-shading algorithms as the shape of the object is varied systematically. In Section 6.1 it was suggested that the shape of the superquadric surface can be controlled by varying e. A good choice for a curve to be rotated around the $y-$axis is a superquadric curve:

$$f(x,y) = \left(\left(\frac{y}{a_1}\right)^{\frac{2}{e_2}} + \left(\frac{x}{a_2}\right)^{\frac{2}{e_2}} \right)^{\frac{e_2}{e_1}}. \tag{6.4}$$

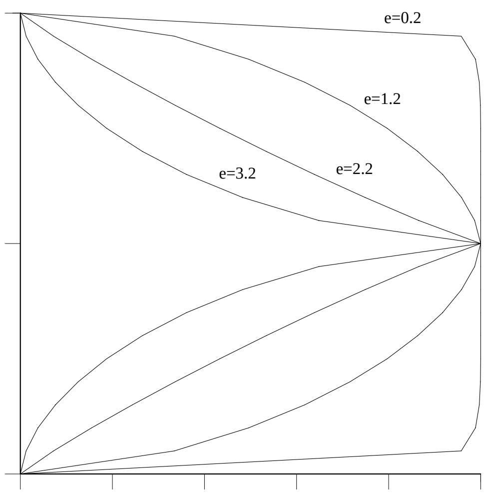


Figure 6.3: The superquadric curves change shape in a consistent manner by varying the parameter e.

The shape of this curve can be changed by varying the parameter e (where $e = e_1 = e_2$. A plethora of two-dimensional superquadric curves are shown in Figure 6.3. The corresponding surfaces of revolution are given in Figure 6.4. It is easy to see that the corresponding surfaces have one less degree of symmetry. The various shape reconstruction algorithms are tested using these less symmetrical surfaces as well as the more symmetrical superquadratic surfaces.

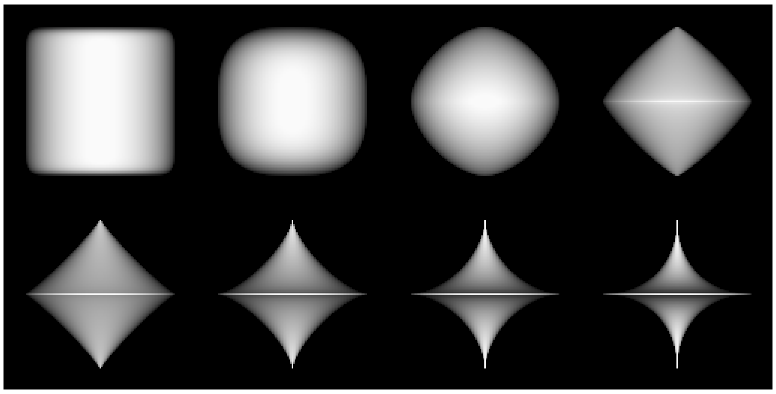


Figure 6.4: Various superquadric surfaces of revolution. Top row: $e = 0.2, 0.7, 1.2, 1.7$, Bottom row: $e = 2.2, 2.7, 3.2, 3.7$.

6.3 Evaluation Measures

Several error measures are used to evaluate the algorithms proposed in this thesis. When dealing with synthesized images the evaluation is much easier, as height and surface orientation are known on the original object.

The goal of the shape-from-shading algorithms is to reconstruct the three-dimensional shape of an object depicted in an image. The output of the shape-from-shading algorithms are the surface normals at each point. These surface normals can be integrated using the algorithms described in Section 2.4 to reconstruct the height profile. When dealing with synthesized objects the surface normals and heights are known precisely at each point. The average angular error indicates how close the reconstructed normals are to the original surface normals. The angle between the real normal and reconstructed normal is computed for each point on the object. The following formula is used to determine the average angular error ($A.A.E.$):

$$A.A.E. = \frac{1}{n} \sum_{i=0}^{n} \left(arccos(\mathbf{n}_o^i \cdot \mathbf{n}_r^i) \right) \tag{6.5}$$

where $\mathbf{n}_o$ is the normal at point i on the original image. $\mathbf{n}_r^i$ is the normal at point i on the reconstructed object.

For the shape-from-shading experiments, the real surface normals are integrated using a height-from-gradient algorithm. The average height error is found by comparing these recovered heights to the height profile obtained by integrating the reconstructed object normals. The average height error ($A.H.E.$) is computed as:

$$A.H.E. = \frac{1}{n}\sum_{i=0}^{n}\left(|H_o^i - H_r^i|\right) \tag{6.6}$$

where n is the total number of points on the object, H_o^i is the height obtained by integrating the real object's surface normal at point i and H_r^i is the reconstructed height at point i.

The $A.A.E.$ is used when evaluating the results of the shape-from-shading process using synthetic images. The $A.H.E.$ is used when evaluating the shape-from-shading algorithms as well as the height reconstruction algorithms. When considering real images the height map and the surface orientations are not known for the original object. The pixel intensities are the only information provided. Once the surface orientations are determined using the shape-from-shading algorithms, an image is synthesized and compared with the input image. The average intensity error ($A.I.E.$) is determined using:

$$A.I.E. = \frac{1}{n}\sum_{i=0}^{n}\left(|I_o^i - I_r^i|\right). \tag{6.7}$$

where I_o^i is the intensity of the original image at point i and I_r^i is the reconstructed intensity at point i.

A new algorithm has been proposed to determine the reflectance parameters when given the light source direction. To evaluate this algorithm the K_{d_e} and σ_e that are estimated are compared to the original K_{d_o} and σ_o. The error is simply the absolute difference between the original reflectance parameters and the estimated reflectance parameters. When dealing with real images; the original reflectance parameters are unknown. There is no direct way of estimating the effectiveness of the algorithm using real images only qualitative evaluation is employed.

It is assumed that a single infinite point source illuminates the object. A new algorithm has been proposed to estimate the direction of the light source when given the reflectance parameters. To evaluate this algorithm, the angle between the estimated illuminant direction $\mathbf{l}_e$ and the real illuminant direction $\mathbf{l}_r$ is computed. Again, there is no direct way of estimating the effectiveness of the light source determination algorithm using real images.

Chapter 7

The Experiments

Several new algorithms have been presented to reconstruct the three-dimensional shape of non-Lambertian objects. In this chapter, these new algorithms are evaluated to determine their effectiveness in recovering the shape-from-shading information.

The first three experiments were conducted to determine values for some of the parameters of the algorithms. The optimal values for the following variables must be determined before the experiments studying the new algorithms can proceed:

- The number of iterations required for the height reconstruction algorithms to converge,
- The value of λ guaranteeing convergence for Brooks and Horn's [23] algorithm,
- The number of iterations required for the shape-from-shading algorithms to converge.

In determining the optimal value of the above variables, several different objects are used to ensure that the selected parameter values are appropriate for a wide variety of shapes. Also, different reflectance parameters are used to study whether the trends observed are consistent for surfaces that reflect light differently.

Experiments were conducted to evaluate the two new shape-from-shading algorithms (G.S.L. and V.A.N.L.). The reflectance parameters, illuminant direction, and object shape are all varied to determine the effectiveness of the two approaches.

Several experiments are done to evaluate the Region-Based Reflectance Estimator. The light source direction, original reflectance parameters, and object shape are all varied to determine the effectiveness of the algorithm.

7.1 Determining Appropriate Parameter Values for Subsequent Experiments

The goal of this research work is to study and develop shape recovery approaches that can determine the three-dimensional shape of non-Lambertian surfaces. Several algorithms have been proposed that have input parameters. The optimal values for these input parameters must be determined before the algorithms can be evaluated fairly.

The first set of experiments is conducted to evaluate the height reconstruction algorithms proposed in Section 2.4. The average height error is computed as the number of iterations is varied. A wide variety of synthetic objects are used. From this experiment, a height reconstruction algorithm will be selected for use in subsequent experiments. The number of iterations that are necessary for convergence is also determined for use in the following experiments.

The optimal value of λ is determined. λ is a regularization parameter used in the variational shape-from-shading algorithms [6, 20, 23]. The average angular error and average height error are computed as λ is varied. The value of λ is chosen that best allows the variational shape-from-shading algorithm to converge; this λ is then used in the subsequent experiments.

By nature, the shape-from-shading algorithms presented in this thesis are iterative. The number of iterations required for convergence is determined in this set of experiments. The average angular error and average height error are computed as the number of iterations is varied.

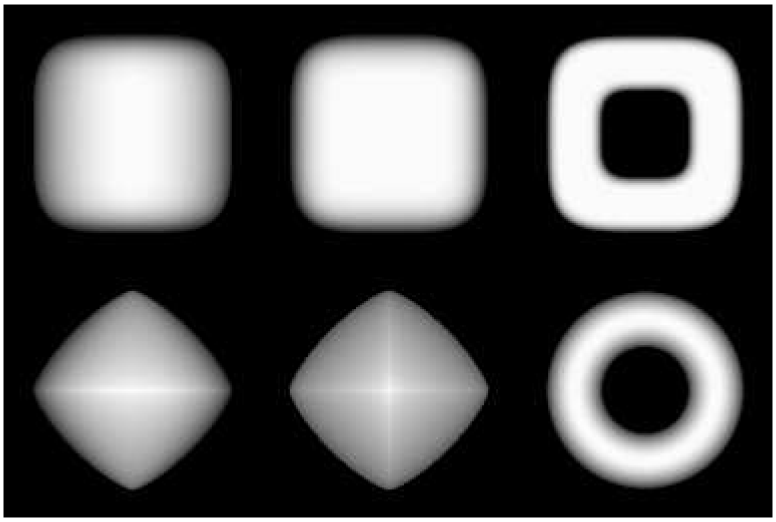


Figure 7.1: Input images for the height reconstruction experiment. Left: Surface of revolution, Middle: Superellipsoid, Right: Supertoroid. Top: $e = 0.5$, Bottom: $e = 1.5$.

7.1.1 Effect of the Number of Iterations on the Height Reconstruction Process

Experiments were conducted to understand the performance of the various height reconstruction approaches. In this set of experiments, the real surface normals are integrated by the height reconstruction algorithms discussed in Section 2.4. The resulting height map is compared with the original height map at each pixel point. The average height error is computed as described in Section 6.3.

The input objects are described in two figures. Figure 7.1 is a rendered version of the input objects. Figure 7.2 is a three-dimensional mesh diagram of the input objects. The linear, degree-two, degree-three and least-sum-of-squares height reconstruction algorithms are all studied. The algorithms are run from 200 to 1400 iterations in increments of 200. The results of this experiment are given in Figures 7.3 to 7.8. Note, the curve labeled *linear* is the linear algorithm, the curve labeled *deg 2* is the result of the degree-two polynomial approach, the plot labeled *deg 3* is the degree-three polynomial method described in Subsection 2.4.2, and finally, the curve labeled *deg 3 ssm* is the results of the degree-three least-sum-of-squares minimization method discussed in Subsection 2.4.3.

The linear height reconstruction algorithm has average height errors between 16.9 and 41.7 units. Until 500 iterations, the error increases and finally stabilizes at a solution after 700 iterations.

The degree-two polynomial interpolation approach has much better results. The average error is between 1.7 and 18.6 units. The error decreases until it converges at 1000 iterations.

The degree-three polynomial interpolation approach has similar results to the degree-two approach. In fact, it is impossible to distinguish between the plots of the degree-two and degree-three methods. In the degree-three algorithms, the four coefficients are determined using four pieces of information. The only extra piece of information is the slope at the third point. The results of this experiment show that this has little effect on the polynomial constructed.

A different third-order polynomial is constructed using the least-sum-of-squares approach. The error in the slopes and heights are minimized using the standard calculus. Five pieces of information are used to determine the four coefficients. The results of this experiment show that this technique is not very effective. The error in slope and error in heights are minimized simultaneously. An inherent property of the least-sum-of-squares approach is that the terms to be minimized are assumed to be at the same scale. However, the error in slopes and error in heights are different measures. It is possible that the technique is minimizing the error in slopes at the expense of the error in heights. To circumvent this problem, it is possible to introduce a parameter to control the contributions of the two error measures. However, it is argued that such a constant is a *fudging factor* that should be avoided, if possible. The optimal value of such a *fudging factor* is difficult to determine. The degree-two approach produces good results and is relatively faster than the computationally expensive least-sum-of-squares approach. The average height error using the sum-of-squares approach is between 5.4 and 20.7 units.

The shape of the input object dramatically affects the results of the experiment. Using the degree-two approach for 500 iterations on the supertoroid with $e = 1.5$

results in an average height error of 4 units. A surface of revolution with $e = 0.5$ results in an average height error of 12 units after the same number of iterations. More rigorous experiments are included in Section 7.2.3 concerning the effect of object shape on the shape recovery process.

Based on the above discussion, the degree-two approach is used in the subsequent experiments. The technique produces good results in a reasonable amount of time. For subsequent experiments, the number of iterations is set at 1000.

A three-dimensional height profile of the reconstructed objects is given in Figure 7.9. It is evident from this figure that the shape is reconstructed best by the degree-two approach. Results of the degree-three algorithm are not included, as they are similar to the degree-two approach.

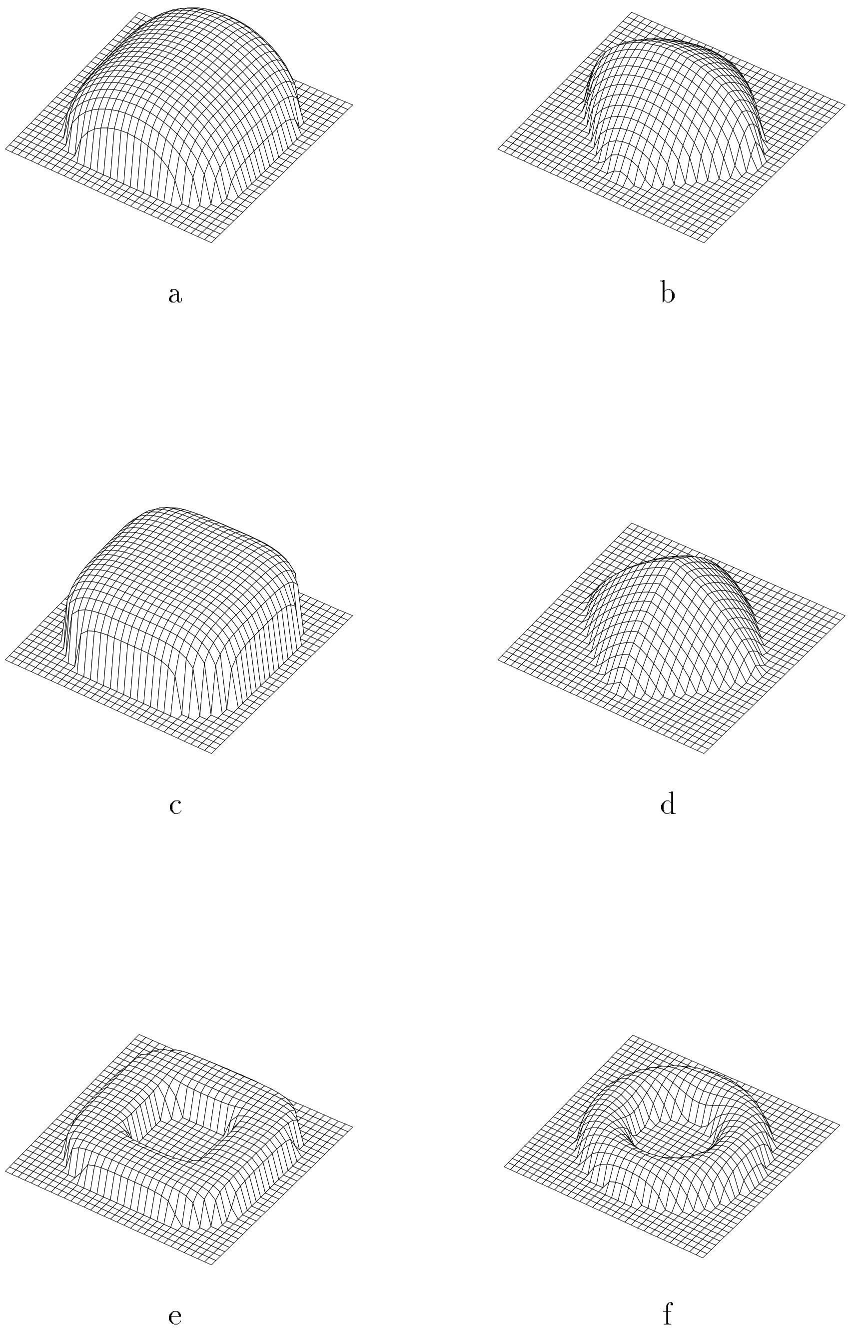


Figure 7.2: Three-dimensional mesh plots of input objects. (a) Surface of revolution with $e = 0.5$, (b) Surface of revolution with $e = 1.5$, (c) Superellipsoid with $e = 0.5$, (d) Superellipsoid with $e = 1.5$, (e) Supertoroid with $e = 0.5$, (f) Supertoroid with $e = 1.5$.

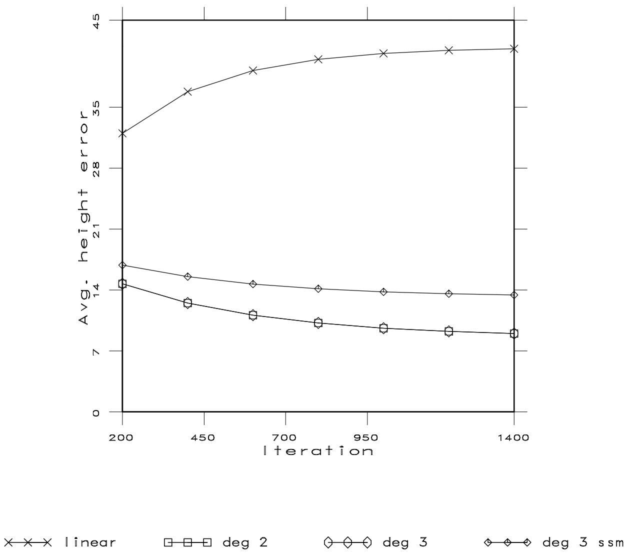


Figure 7.3: Results of the height reconstruction experiment when the input object is a surface of revolution with $e = 0.5$.

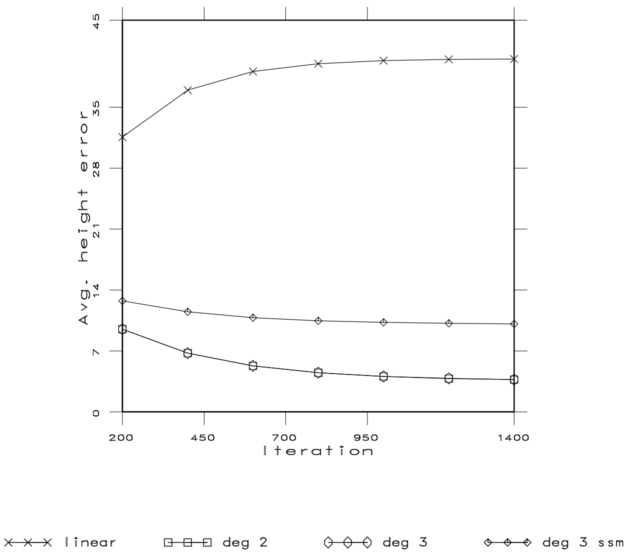


Figure 7.4: Results of the height reconstruction experiment when the input object is a surface of revolution with $e = 1.5$.

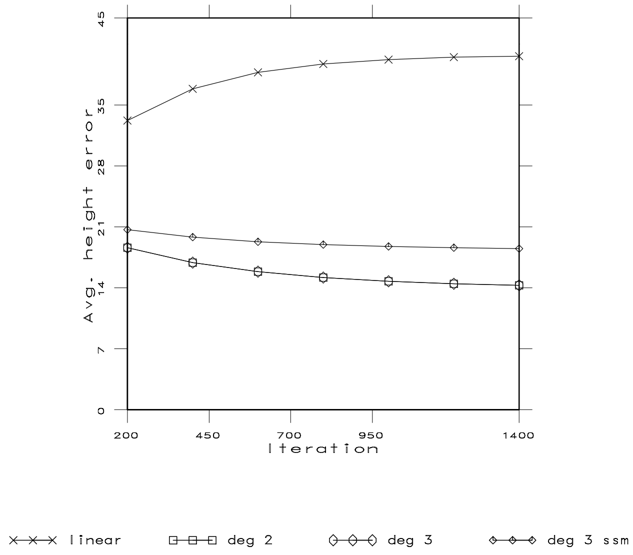


Figure 7.5: Results of the height reconstruction experiment when the input object is a superellipsoid with $e = 0.5$.

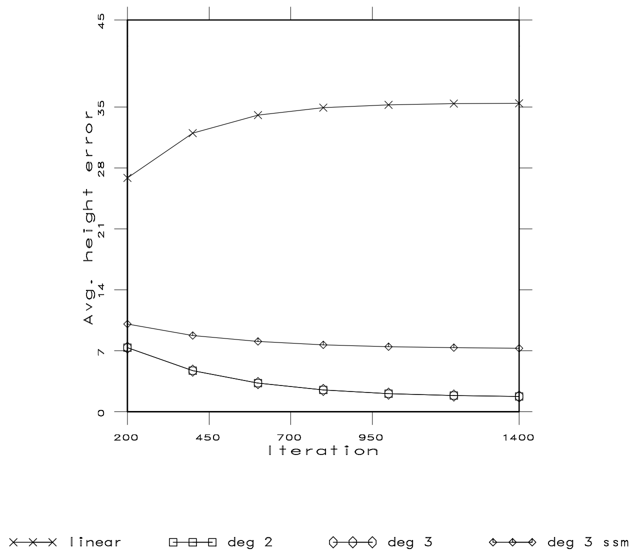


Figure 7.6: Results of the height reconstruction experiment when the input object is a superellipsoid with $e = 1.5$.

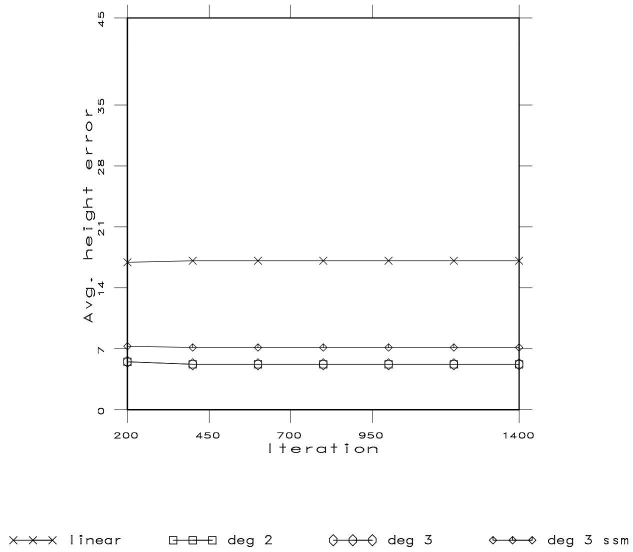


Figure 7.7: Results of the height reconstruction experiment when the input object is a supertoroid with $e = 0.5$.

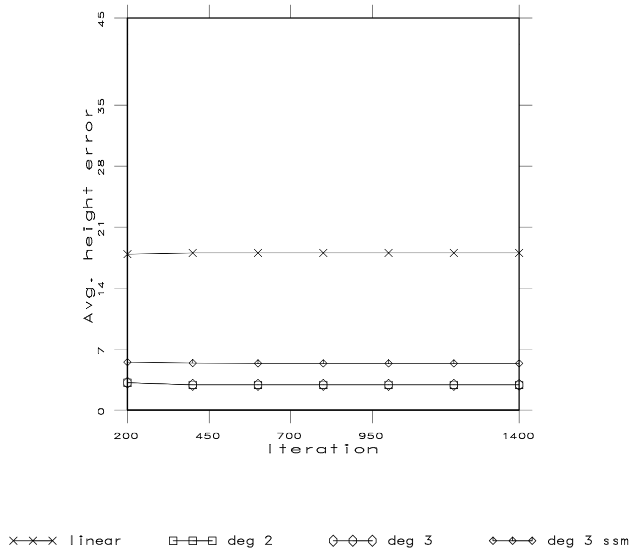
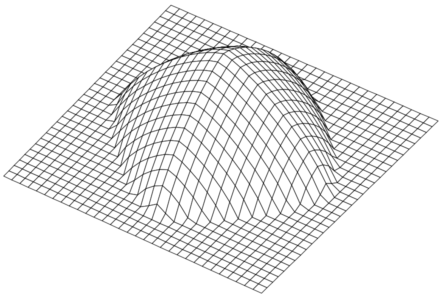
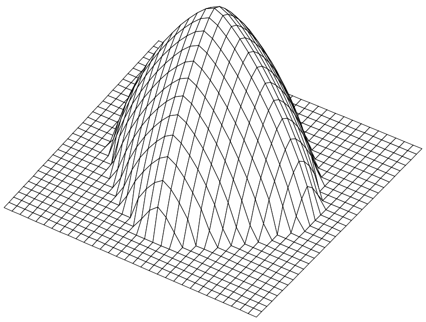


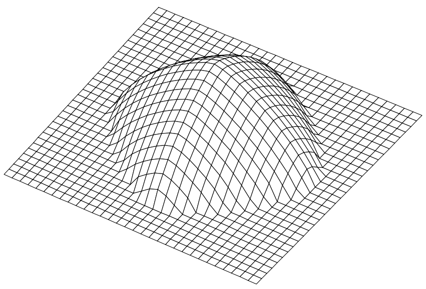
Figure 7.8: Results of the height reconstruction experiment when the input object is a supertoroid with $e = 1.5$.



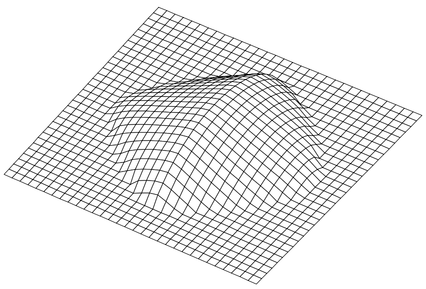
a



b



c



d

Figure 7.9: Three-dimensional mesh plots comparing original height map to reconstructed height profiles. The algorithms were run for 1000 iterations. (a) the input object (superellipsoid with $e = 1.5$) (b) results of the linear algorithm, (b) results of the degree-two algorithm, (c) results of degree-three, least-sum-of-squares approach.

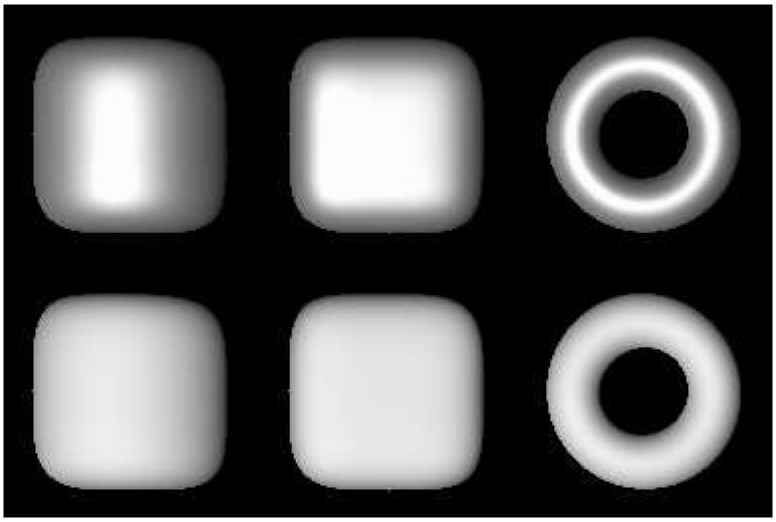


Figure 7.10: Input images for the λ experiment. Left: Surface of revolution $e = 0.5$, Middle: Superellipsoid $e = 0.5$, Right: Supertoroid $e = 1.0$. Top: $(K_d = 0.3, \sigma = 0.3)$, Bottom: $(K_d = 0.7,\ \sigma = 0.7)$.

7.1.2 Effect of Lambda (λ)

The variational shape-from-shading algorithms make use of a regularization constant called lambda (λ). λ establishes a trade-off between departure from smoothness and the brightness error. The optimal choice of λ is an open problem.

If λ is too small, the resulting solution may not be smooth; if λ is too large the solution may not be accurate. In general, λ has to be in a limited range. A natural question is: what is the maximum range of λ such that uniqueness and convergence are still guaranteed. This remains an open problem that is addressed experimentally.

In this experiment, λ was varied from 0.05 to 2.65 in increments of 0.2. The light source was fixed at (3,2,-9) and the V.A.N.L. was run for 500 iterations. The input objects are given in Figure 7.10.

Two error measures are used. The average error in height that was described in Section 6.3 is again used; as well, the average angular error is computed. The results of the λ experiment are given in Figures 7.11 to 7.16. From the figures it is observed that the results of the V.A.N.L. are greatly affected by the choice of λ. If λ is too small or too large, the algorithm breaks down and is unable to reconstruct the shape.

The results also vary greatly depending on the reflectance parameters. If it is a highly specular object ($K_d = 0.3$, $\sigma = 0.3$), the optimal choice of λ is 1.3, however if the object is diffuse ($K_d = 0.7$, $\sigma = 0.7$) the minimum error occurs when λ is 0.85. Also, for specular surfaces the choice of λ has less impact than for diffuse surfaces. As λ is varied on specular surfaces, the average angular error only varies from 6 to 11 degrees; however for diffuse surfaces the average angular error varies from 6 to 43 degrees.

Another trend observed is that the shape of the object also plays a role as to which λ is optimal. For a diffuse supertoroid the optimal λ occurs at 1.65, however, for a diffuse surface of revolution the optimal λ occurs at 1.25.

The average angular error ranges from 6 to 43 degrees while the average height error varies from 1 to 14 units. The dependence of the variational algorithm on a good choice of λ is reflected by this diverse range of errors. Also, the optimal choice of λ is highly dependent on the reflectance properties of the surface.

In subsequent experiments, λ will be set at 1.0. It is impractical to adjust λ depending on the shape of the object and the reflectance parameters, thus λ is set at 1.0, which seems to be the best compromise.

To illustrate the dramatic effect λ has on the results of the V.A.N.L., an output mesh diagram is included in Figure 7.17. The V.A.N.L. is run with λ equal to 0.05, 0.70, and 1.70 on a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$. The errors obtained using this object are plotted in Figure 7.12.

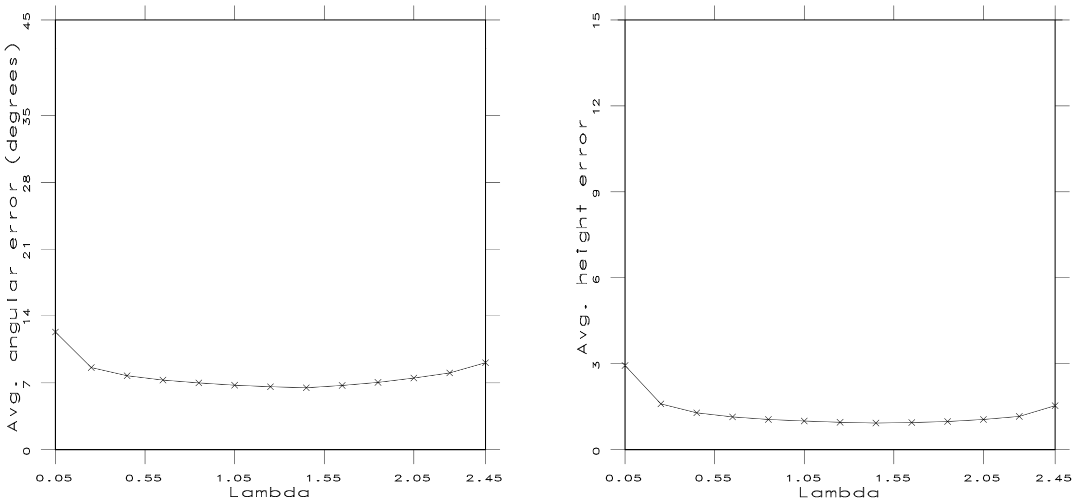


Figure 7.11: Results of the λ experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.3$, $\sigma = 0.3$. Left: the average angular error, Right: the average height error.

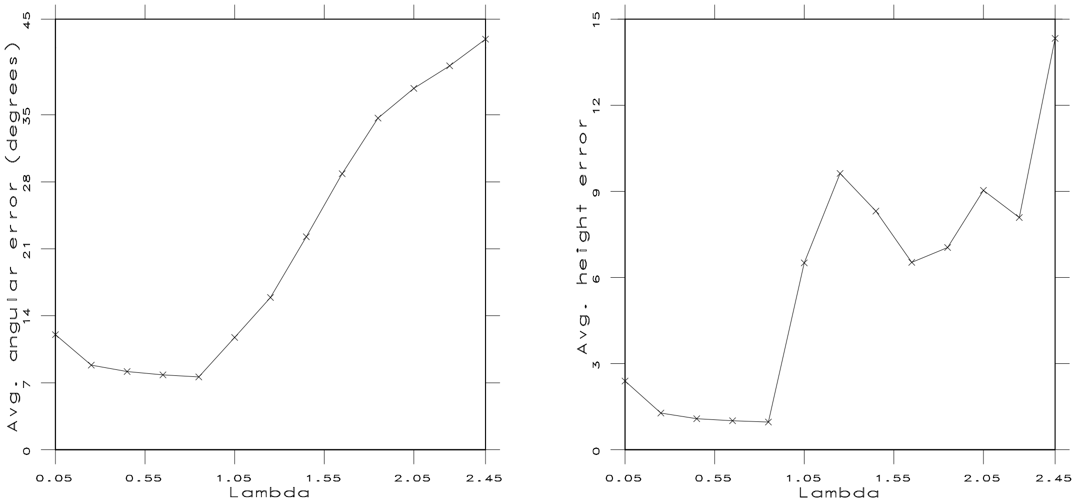


Figure 7.12: Results of the λ experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7$, $\sigma = 0.7$. Left: the average angular error, Right: the average height error.

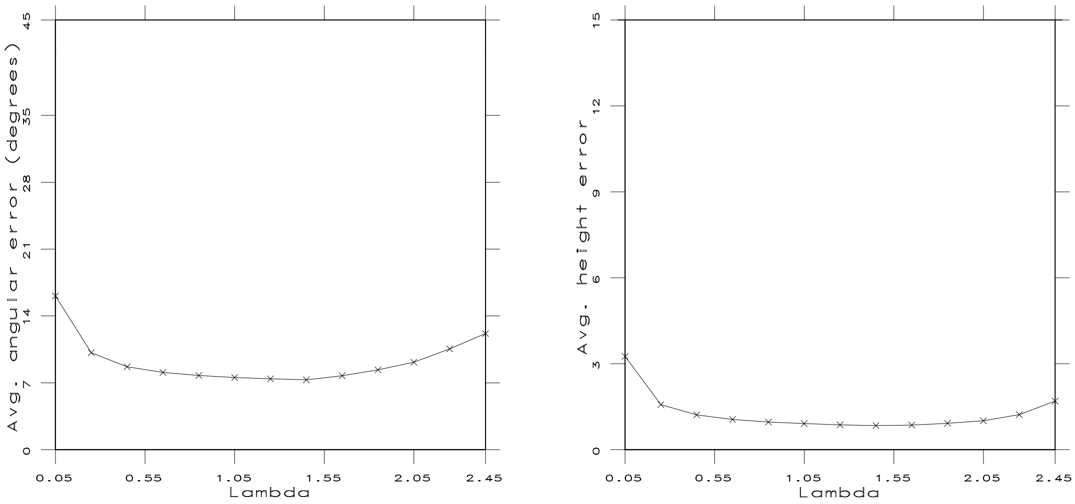


Figure 7.13: Results of the λ experiment when the input object is a superellipsoid with $e = 0.5$ and $K_d = 0.3$, $\sigma = 0.3$. Left: the average angular error, Right: the average height error.

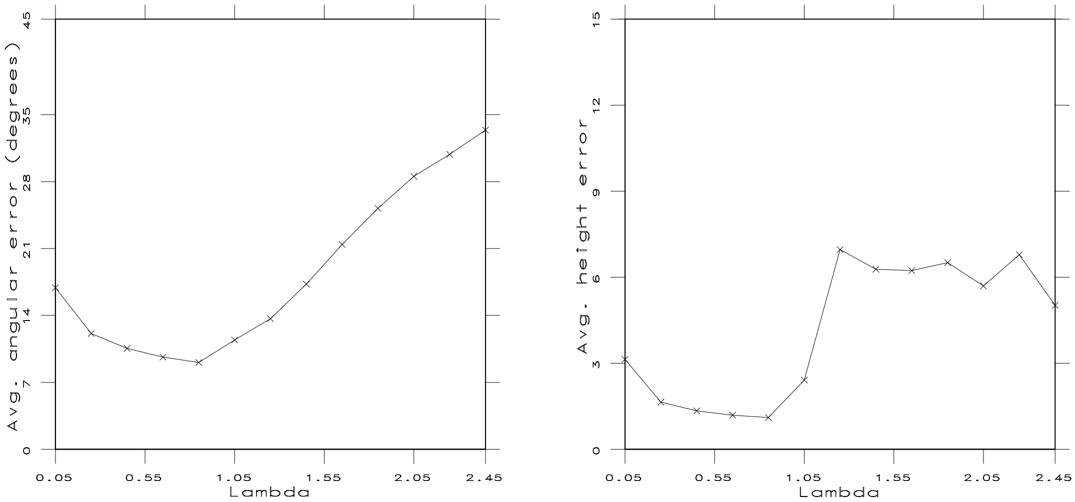


Figure 7.14: Results of the λ experiment when the input object is a superellipsoid with $e = 0.5$ and $K_d = 0.7$, $\sigma = 0.7$. Left: the average angular error, Right: the average height error.

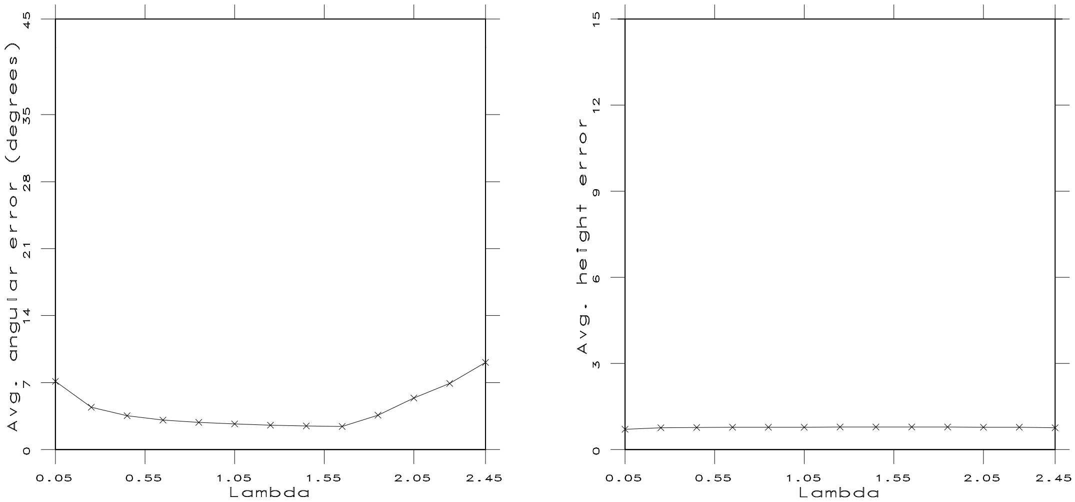


Figure 7.15: Results of the λ experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.3$, $\sigma = 0.3$. Left: the average angular error, Right: the average height error.

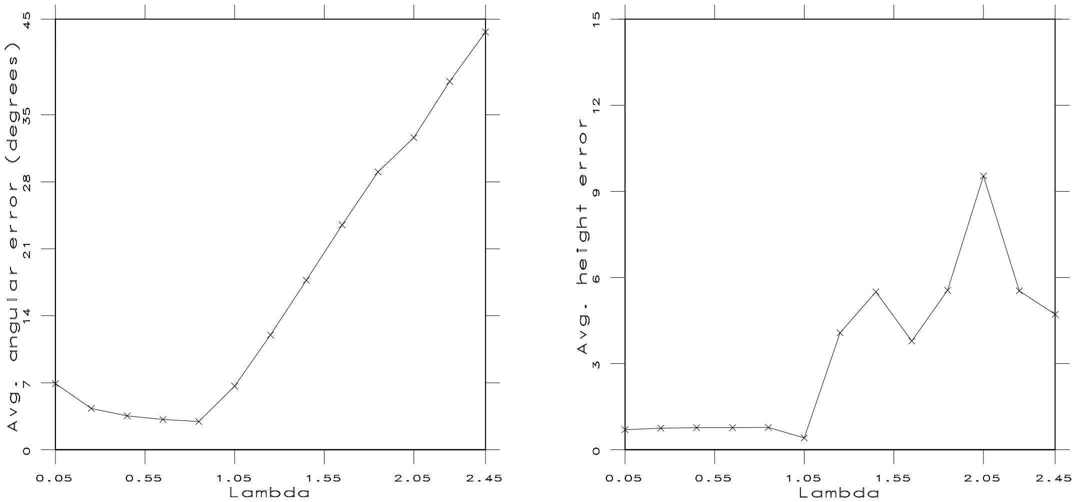


Figure 7.16: Results of the λ experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.7$, $\sigma = 0.7$. Left: the average angular error, Right: the average height error.

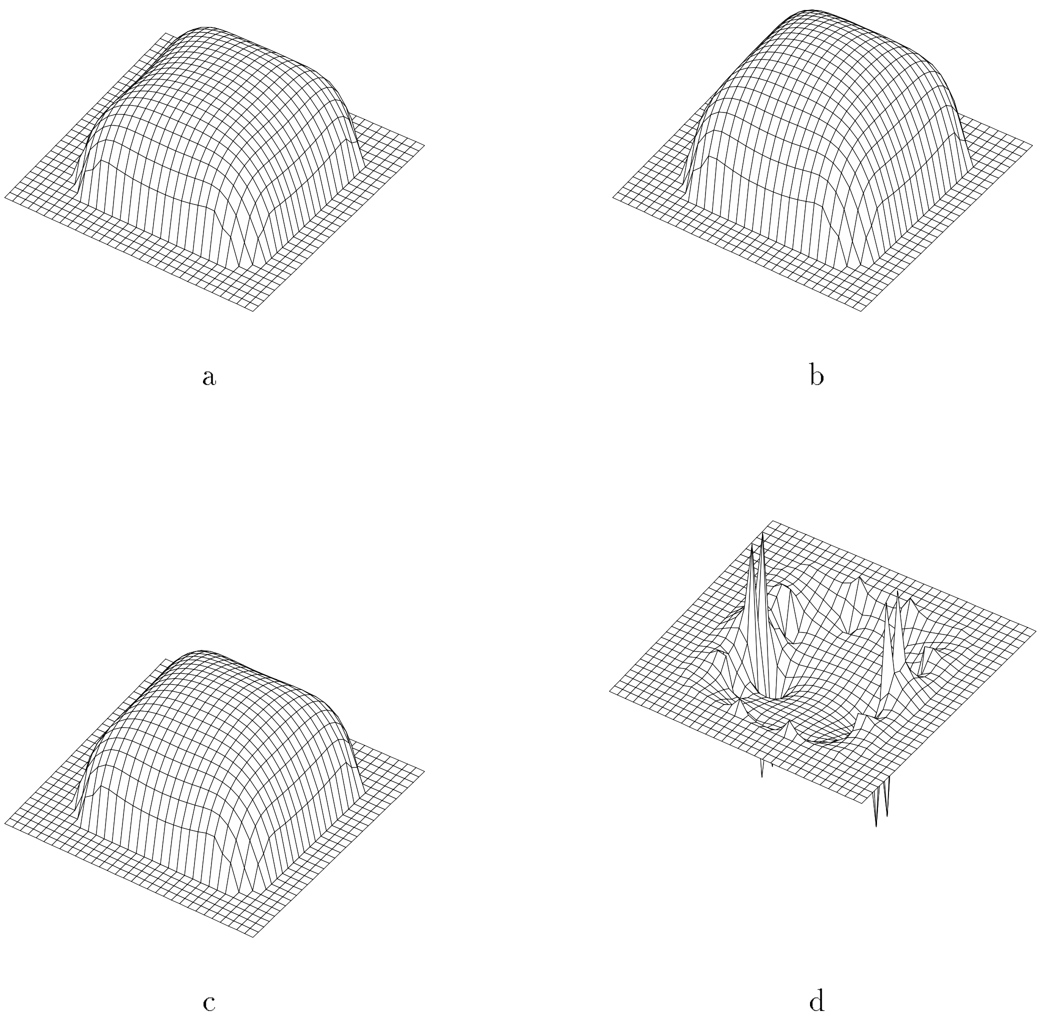


Figure 7.17: Three-dimensional mesh plots comparing results with different λ values. The V.A.N.L. was run for 1000 iterations on a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$ (a) the original height profile after integrating the real normals, (b) $\lambda = 0.05$, (c) $\lambda = 0.70$, (d) $\lambda = 1.70$.

7.1.3 Effect of the Number of Iterations on the Shape-From-Shading Algorithms

The shape-from-shading algorithms studied in this thesis are all iterative. An experiment is conducted to study the behaviour of the algorithms as the number of iterations is varied. This should reveal the relative speeds of the algorithms as well as indicate whether the algorithms do in fact converge.

Variational algorithms that employ regularization have been accused of "walking-away" from a solution as the number of iterations increases [1, 20]. It has been proposed that this "walking-away" property is due to the trade-off between departure from smoothness and the brightness error that λ imposes. The G.S.L. algorithm imposes no such trade-off. Therefore it is expected that the G.S.L. algorithm will converge to a stable solution.

In the following set of experiments, the number of iterations is varied and the average height error and average angular error is computed. The number of iterations is varied from 50 to 210 in increments of 40. The input objects are again given in Figures 7.10 and 7.2. The results of the experiment are given in Figures 7.18 to 7.23. Note, the curve labelled *V.A.N.L.* represents the results of the Variational Algorithm for Non-Lambertian Surfaces and the *G.S.L.* curve represents the results obtained from running the General Shading Logic algorithm.

The V.A.N.L. algorithm has average angular errors between 3 and 11 degrees. In some cases the solution does indeed "walk-away" from the solution. Considering Figure 7.19, the average angular error is only 7.6 degrees after 50 iterations. However, after 130 iterations the error has increased to 9.57 degrees. This "walking-away" phenomenon is also observed in the average height errors. It is interesting to note that the reflectance of the surface does have an impact on the ability of the V.A.N.L. to recover shape. From the results of using diffuse ($K_d = 0.7$, $\sigma = 0.7$) objects (see Figures 7.19 and 7.21) as input images, it is observed that the algorithm "walks-away" from the solution. From the experiments using more specular surfaces ($K_d = 0.3$, $\sigma = 0.3$) (see Figures 7.18 and 7.20), this "walking-away" phenomenon does not occur. The average height error using the V.A.N.L. is between 1 and 16 units.

The G.S.L. algorithm performs better in all cases. The average angular error is between 1 and 4 degrees. The average height error is between 1 and 2 units. The algorithm is much more stable than the V.A.N.L. The error decreases or remains the same as the number of iterations increases. The reflectance of the surface has no impact on the number of iterations required for convergence.

From the results of this experiment it is observed that the V.A.N.L. surfaces exhibits the same "walking-away" property observed in other variational shape-from-shading algorithms employing a regularization term. It is also observed that the reflectance properties of the surface play a role as to the number of iterations needed for the V.A.N.L. to converge to a solution.

The G.S.L. is less affected by varying the number of iterations. Regardless of shape or reflectance properties, the G.S.L. algorithm converges to a solution in very few iterations. It is also noted that the G.S.L. algorithm can recover the shape of non-Lambertian surfaces better than the V.A.N.L. The two algorithms are compared more rigorously in a subsequent experiment. The number of iterations is set to 150 for subsequent experiments.

A mesh diagram is included in Figure 7.24 to reveal the effectiveness of the algorithms. The input object is a superellipsoid with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$. Because the G.S.L. algorithm's error is stable as the number of iterations is varied, only one mesh diagram is presented (after 70 iterations). The mesh diagram for the V.A.N.L. is presented after 70 iterations and after 200 iterations. The errors obtained using this object are plotted in Figure 7.21.

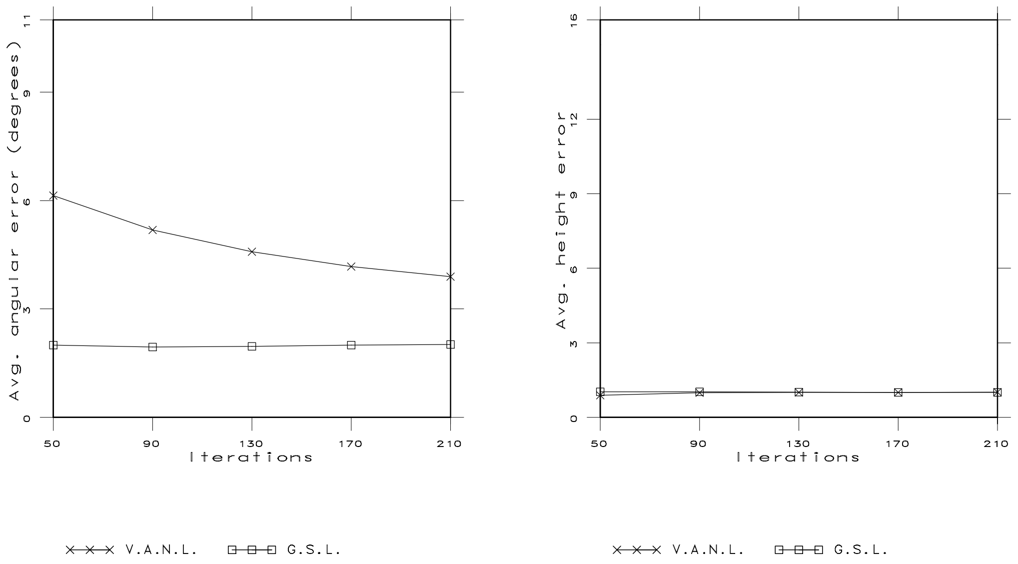


Figure 7.18: Results of the iteration experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.3$, $\sigma = 0.3$. Left: average angular error, Right: average height error.

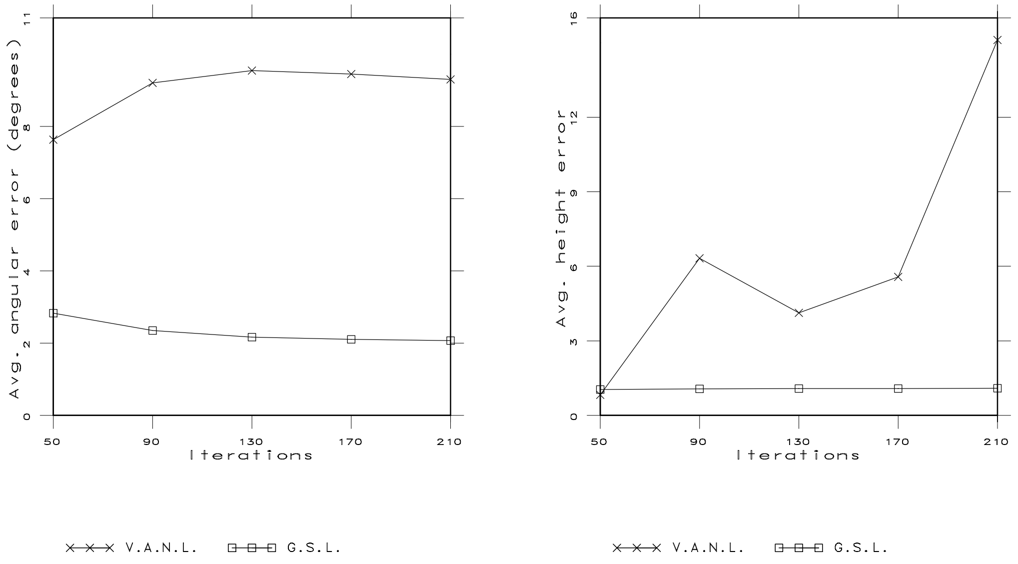


Figure 7.19: Results of the iteration experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7$, $\sigma = 0.7$. Left: average angular error, Right: average height error.

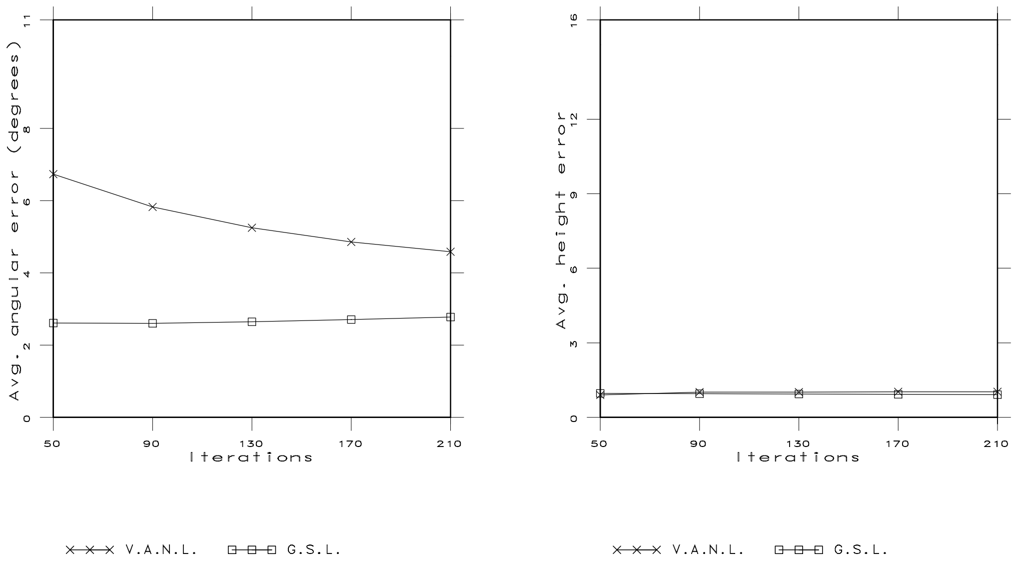


Figure 7.20: Results of the iteration experiment when the input object is a superellipsoid with $e = 0.5$ and $K_d = 0.3$, $\sigma = 0.3$. Left: average angular error, Right: average height error.

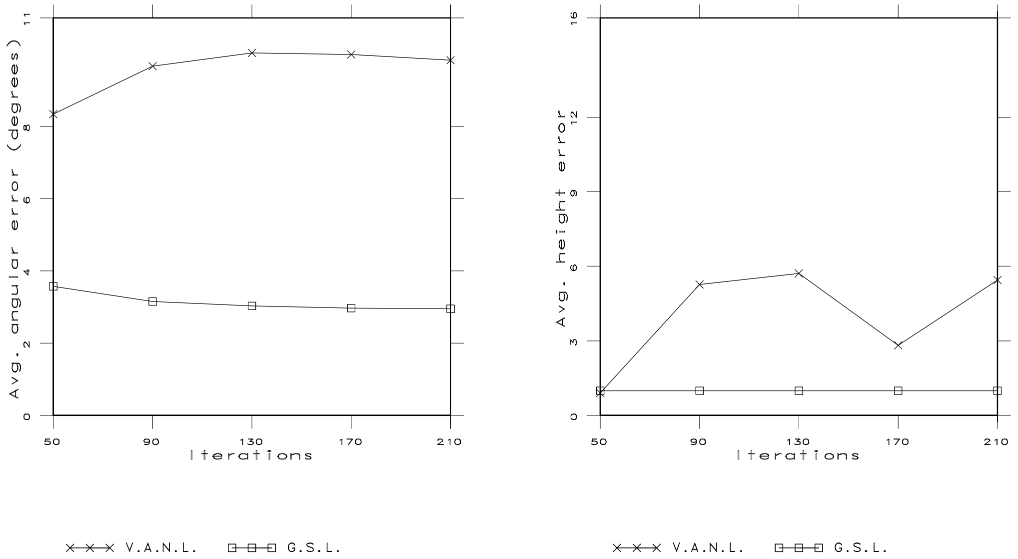


Figure 7.21: Results of the iteration experiment when the input object is a superellipsoid with $e = 0.5$ and $K_d = 0.7$, $\sigma = 0.7$. Left: average angular error, Right: average height error.

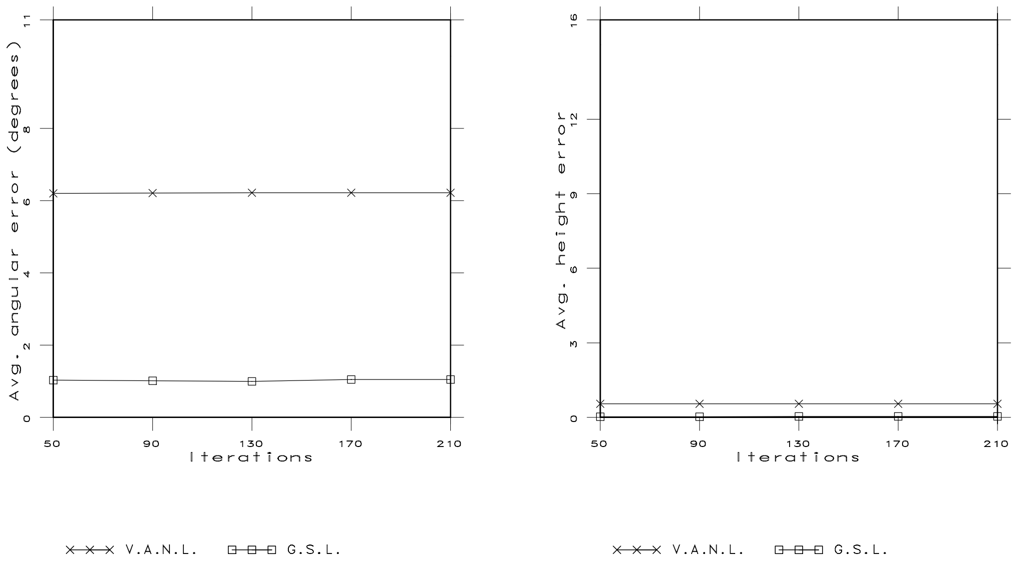


Figure 7.22: Results of the iteration experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.3$, $\sigma = 0.3$. Left: average angular error, Right: average height error.

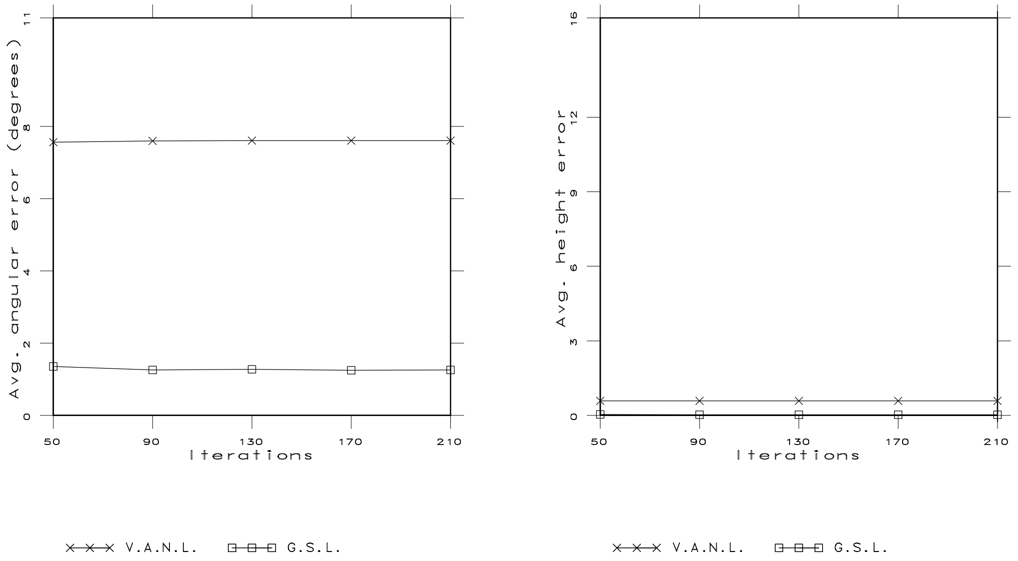
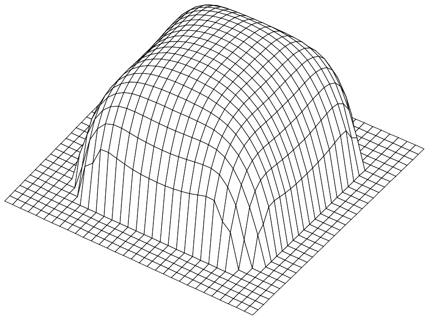
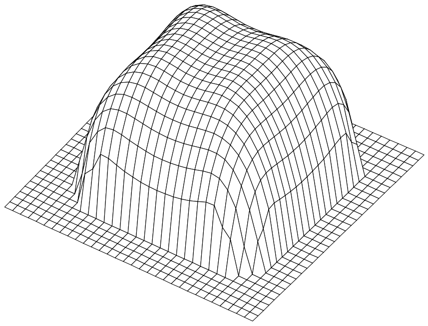


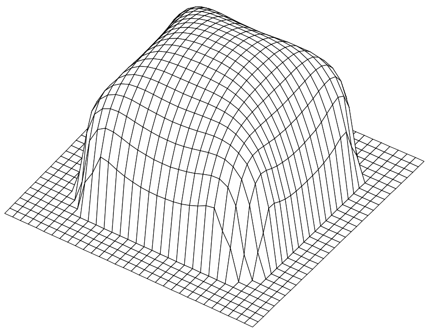
Figure 7.23: Results of the iteration experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.7$, $\sigma = 0.7$. Left: average angular error, Right: average height error.



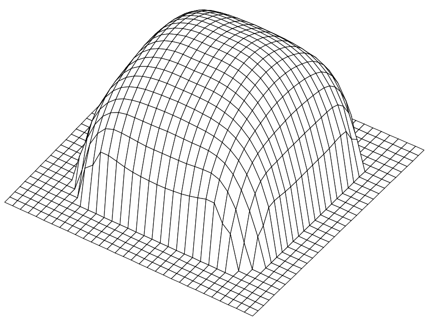
a



b



c



d

Figure 7.24: Three-dimensional mesh plots comparing results as the number of iterations is varied. Input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$ (a) the original height profile after integrating the real normals, (b) V.A.N.L. after 50 iterations, (c) V.A.N.L. after 210 iterations, (d) G.S.L. algorithm after 210 iterations.

7.1.4 Summary of Algorithm Parameter Experiments

In Section 2.4 several new height reconstruction algorithms have been proposed. It was suggested that using higher-order polynomials to integrate height-from-gradient information could produce better results than simply using linear interpolation. These algorithms have been evaluated experimentally and it was found that using a degree-two or degree-three polynomial interpolation method produces much better results than using linear interpolation.

It was also found that using the degree-three approach proposed in Subsection 2.4.2 gave the same results as the degree-two approach. That is, using one extra piece of information (the slope at an adjacent point) did not have much impact in the polynomial constructed. Another experimental finding was that using a least-sum-of-squares approach as suggested in Subsection 2.4.3 did not give better results than the degree-two approach. It was hypothesized that this is due to the trade-off between minimizing gradient and height information. In subsequent experiments the second-order height reconstruction algorithm will be run for 1000 iterations.

The effect of the value for the regularization factor λ was studied experimentally. It was found that the value of λ greatly affects the ability of the V.A.N.L. to recover the shape of non-Lambertian surfaces. If λ is too large or too small the algorithm breaks down and is unable to produce a good solution. The optimal choice of λ varies, depending on the surface shape and reflectance properties of the surface. For diffuse surfaces, λ is smaller than for highly specular surfaces. However, it is concluded that a λ of 1.0 should be used in subsequent experiments, as this value seems to be the best compromise for all surface types.

The V.A.N.L. has the same "walking-away from the solution" property exhibited by other variational formulations using a regularization term. The number of iterations that the V.A.N.L. needs for convergence is also dependent on the reflectance properties of the surface. The G.S.L. algorithm converges to a solution after very few iterations. The reflectance properties of the surface do not affect the number of iterations the G.S.L. algorithm requires for convergence. In subsequent experiments the algorithms are run for 150 iterations.

7.2 Shape-from-Shading Experiments

Two new algorithms have been presented that can recover the three-dimensional shape of objects whose surfaces obey the Torrance-Sparrow reflectance model. The first algorithm, (V.A.N.L) was originally derived by Brooks and Horn [23]. Their approach has been modified to include the Torrance-Sparrow reflectance model. The second algorithm presented (G.S.L.), is a heuristic-based algorithm. The G.S.L. algorithm is a generalization of an algorithm first proposed by Vega and Yang [39]. Vega and Yang's formulation has been modified to cope with any reflectance model.

Experiments are performed using the V.A.N.L. and the G.S.L. algorithm to determine their effectiveness in reconstructing the shape of non-Lambertian surfaces. In the first set of experiments, the reflectance parameters are changed systematically. This experiment gives some indication whether the shape-from-shading algorithms can recover the shape of objects with varying specularity. In this experiment, several objects are used to ensure that the results of the algorithms are consistent over many shape types.

In the second set of experiments, the light source is moved systematically. The location of the specular highlight is dependent on the proximity of the light source. When performing experiments on non-Lambertian surfaces, it is especially important to study the behaviour of the shape-recovery algorithms as the light source is moved. Unlike Lambertian surfaces, the light source direction has a dramatic impact on the appearance of non-Lambertian objects. Various objects are used in this experiment as well.

In the third and final set of experiments, the shape-from-shading algorithms are studied with respect to the shape of the object. It has been found by Vega and Yang [39] that the shape of the object has a dramatic effect on the shape-from-shading algorithms' ability to recover the three-dimensional shape. The two new algorithms are tested to determine the behaviour of the algorithms as the shape is changed.

Objects are rendered using many different reflectance parameters to study whether the trends observed are consistent over different reflectance maps.

7.2.1 Effect of Reflectance on Recovering Shape

The G.S.L. algorithm and the V.A.N.L. have been derived to recover the shape of objects whose surfaces can be modeled using the Torrance-Sparrow [37] reflectance function. Traditional shape-from-shading algorithms were designed to handle diffuse surfaces. In this set of experiments, the surface reflectance properties are varied systematically. The behaviour of the shape-from-shading algorithms are studied as the reflectance is changed.

The original Brooks and Horn [23] approach as well as Vega and Yang's Shading Logic [39] algorithm are included in the first experiment for the sake of demonstration. A sphere with varying reflectance is reconstructed using four algorithms: Brooks and Horn's (B.H.), Shading Logic (S.L.), Variational Algorithm for Non-Lambertian Surfaces (V.A.N.L.), General Shading Logic Approach (G.S.L). The input images are given in Figure 7.25. The spheres are illuminated from (3,2,-9). The coefficients K_d and σ are varied from 0.1 to 0.895 in increments of 0.325. The light source direction, reflectance parameters, and image are available to the shape-from-shading algorithms.

It is clear from the results given in Figures 7.26 to 7.29 that the algorithms designed for Lambertian surfaces cannot recover the shape of non-Lambertian surfaces. Brooks and Horn's formulation for non-Lambertian surfaces has an average angular error between 6 and 40 degrees. The average height error varies from 1 to 36 units. Over all diffuse coefficients (K_d), the average A.A.E. is 15.99 degrees. The Shading Logic algorithm has an average angular error between 6 and 41 degrees and an average height error between 1 and 36 units. The average A.A.E. over all reflectance parameters is 15.47 degrees. Contrast this with the results obtained from using the new algorithms. The V.A.N.L. algorithm had an average A.A.E. of 4.43 degrees and the range is between 3 and 7 degrees. The G.S.L. algorithm had an average A.A.E. of 1.98 degrees with the range being between 1 and 3 degrees.

The algorithms designed for Lambertian surfaces have a very large range of errors when applied to non-Lambertian surfaces. The high end of this diverse range occurs when the diffuse coefficient (K_d) is small. That is, when the surface has a lot of specular reflection, the ability of the Lambertian shape-from-shading algorithms to recover the shape decreases. This is a logical result. When the surface is more diffuse ($K_d = 0.85$), all of the algorithms perform reasonably well. The algorithms designed for non-Lambertian surfaces (G.S.L and V.A.N.L.) consistently produce good results. The small range of error reveals how effective these algorithms are at recovering the shape of all surface types, diffuse or specular. These algorithms are evaluated more rigorously in the following experiments.

To further illustrate the nature of the algorithms as the reflectance is varied, a mesh plot is included in Figure 7.30. Mesh plots are presented for a sphere with the reflectance parameters: $K_d = 0.10, \sigma = 0.365$. The errors obtained from using this object are given in Figure 7.27. Note that the reconstructed height map for the V.A.N.L. is not included as the G.S.L. result is very similar.

In the subsequent experiments the results of the two algorithms designed for Lambertian surfaces (B.H. and S.L.) are not included. They are only effective when dealing with highly diffuse surfaces. Instead, the results obtained from running the non-Lambertian versions of these algorithms are presented. Experiments using a wide variety of objects is given to exhaustively study the behaviour of these algorithms when reflectance is varied.

Figures 7.32 to 7.35 are the results obtained from varying the reflectance of a surface of revolution with $e = 0.5$. Figures 7.37 to 7.40 are the results using a surface of revolution with $e = 1.5$ as the input object. Figures 7.43 to 7.46 are the results obtained using a superellipsoid with $e = 1.5$ as an input image. Finally, Figures 7.48 to 7.51 are the results from using a supertoroid with $e = 1.0$ as an input object.

The average angular error for the V.A.N.L. varies from 1 to 12 degrees. The average height error is between 1 and 4 units. The G.S.L. algorithm performs better in all situations. The average A.A.E. over all objects for the G.S.L. algorithm is 4.396 degrees. The average A.A.E. for the V.A.N.L. is 6.665 degrees. Also the range of errors for the G.S.L. algorithm is significantly smaller. The average angular error varies from 1 to 6 degrees. The average height error is between 1 and 2 units.

The G.S.L. algorithm is not significantly affected by the reflectance properties of an object's surface. From the graphs in Figures 7.26 to 7.51 it is observed that the error holds steady as σ is varied, as well as when K_d is varied. However, the V.A.N.L. seems to perform better as K_d increases. Consider Figures 7.32 to 7.35 which display the results of the experiments on a surface of revolution with $e = 0.5$. The average angular error when $K_d = 0.100$ is 11.086 degrees over all values of σ. When K_d increases to 0.365, the average angular error decreases to 8.749. The average angular error over all σ when $K_d = 0.895$ reduces to 8.173. This trend is observed in the experiments with different input objects. The G.S.L. algorithm exhibits no such behaviour.

A mesh plot is included in Figure 7.41 to illustrate the behaviour of the G.S.L. algorithm as the reflectance is varied. The input object is a surface of revolution with $e = 1.5$. The errors obtained using this object are presented in Figures 7.37 to 7.40. The resulting solution using the G.S.L. algorithm is presented for $(K_d = 0.10, \sigma = 0.365)$, $(K_d = 0.365, \sigma = 0.365)$, and $(K_d = 0.895, \sigma = 0.365)$.

Both algorithms can recover the shape of non-Lambertian surfaces. The V.A.N.L. performs slightly better when the surface is more diffuse. The G.S.L. can recover the shape equally well, regardless of reflectance parameters.

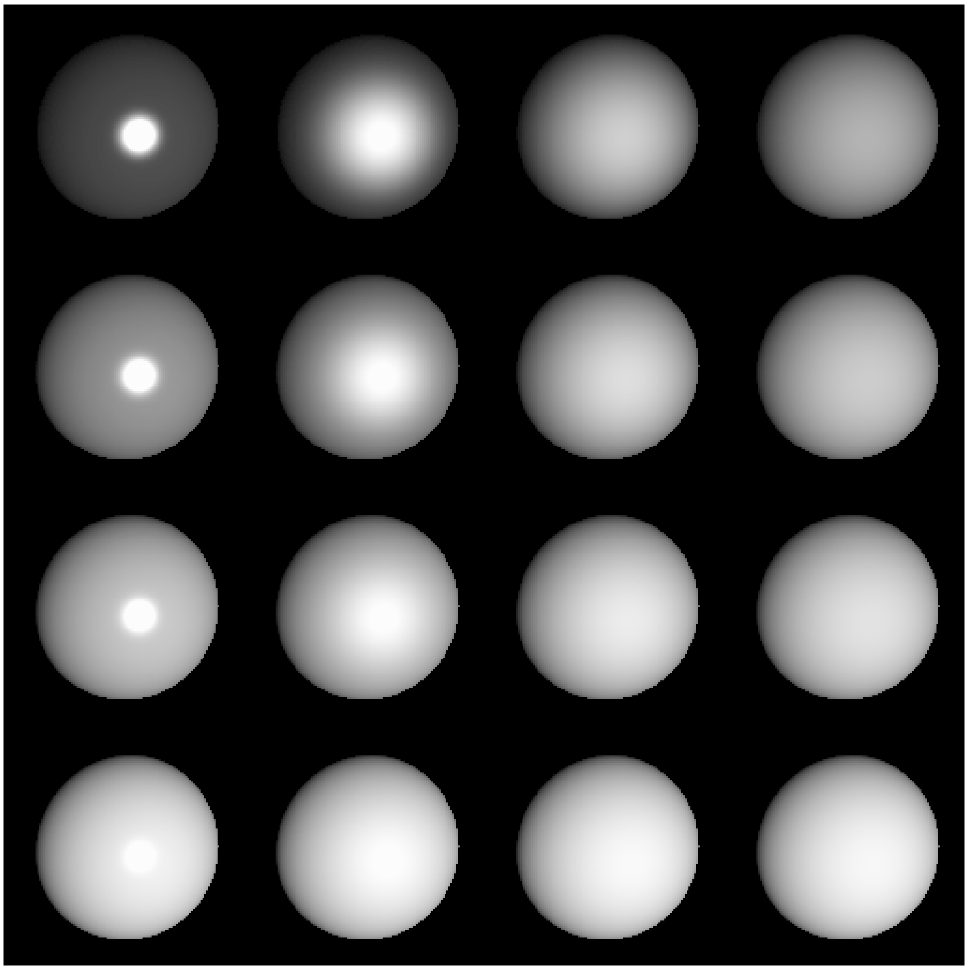


Figure 7.25: Input images used in the reflectance experiment. The object is a sphere with radius 50.0 units. K_d varies from 0.100 to 0.895 in increments of 0.265 from top to bottom. σ varies from 0.100 to 0.895 in increments of 0.265 from left to right.

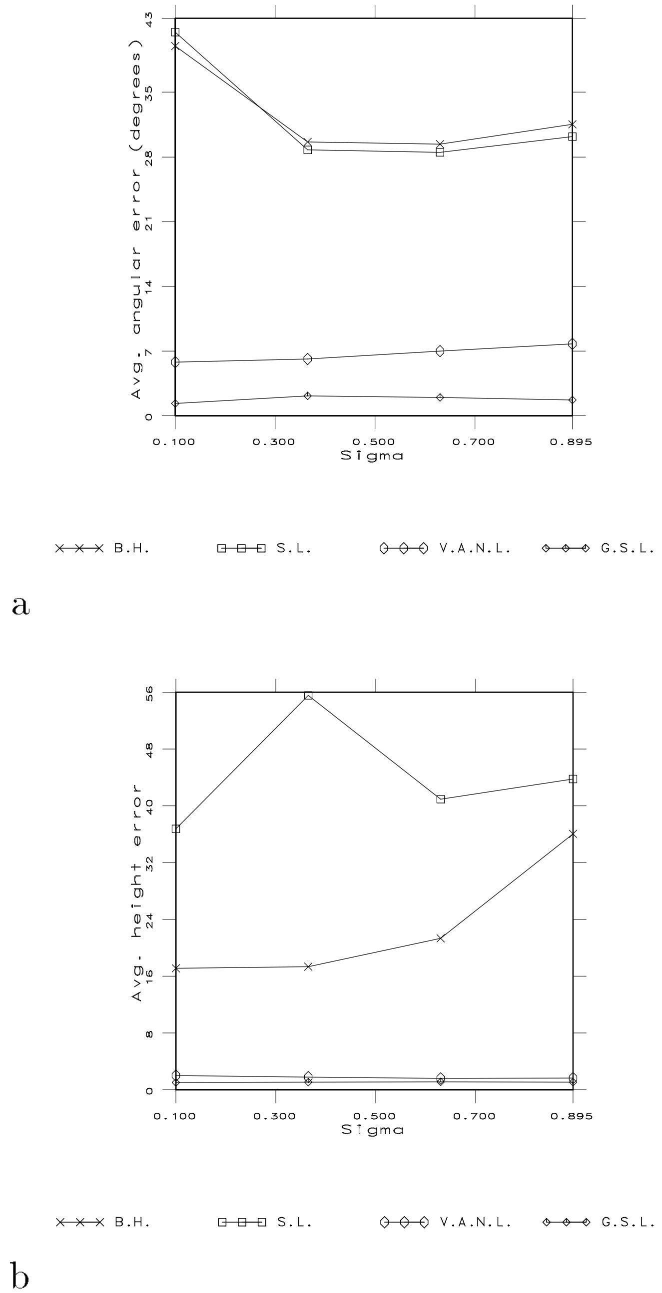
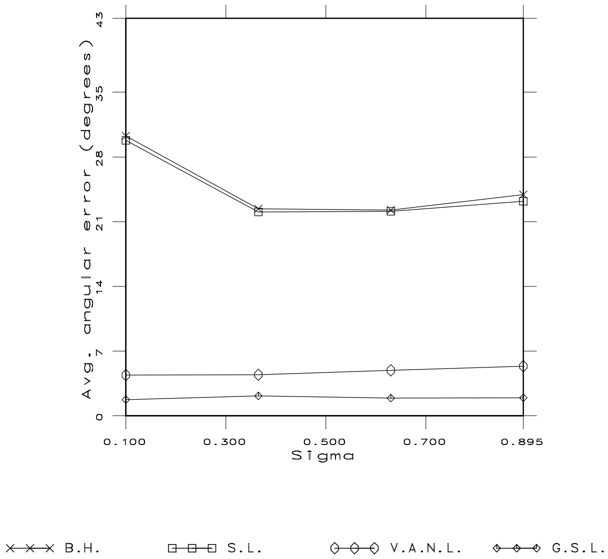
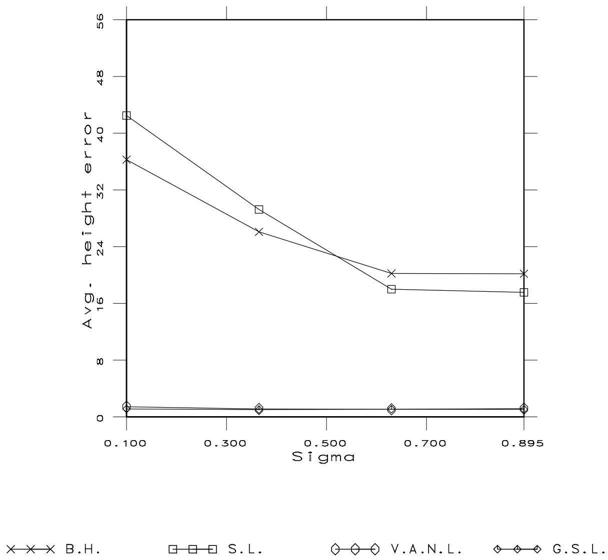


Figure 7.26: Results of the reflectance experiment when the input object is a sphere and $K_d = 0.10$. Plot (a) is the average angular error, Plot (b) is the average height error.

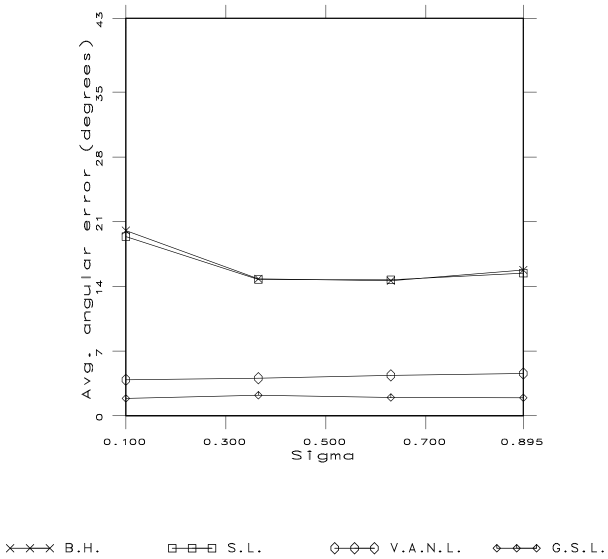


a

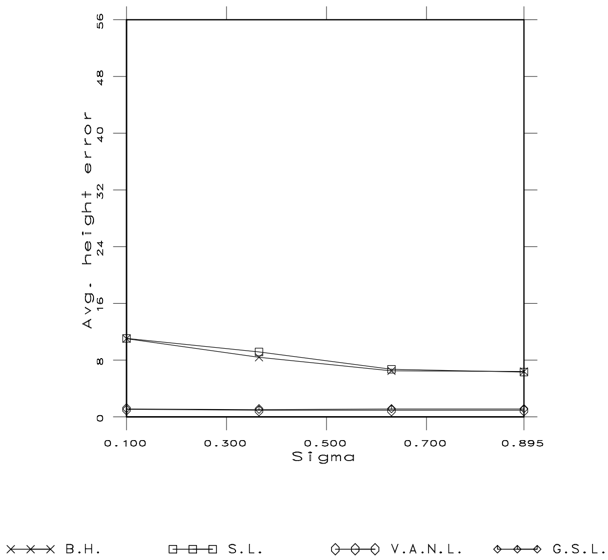


b

Figure 7.27: Results of the reflectance experiment when the input object is a sphere and $K_d = 0.365$. Plot (a) is the average angular error, Plot (b) is the average height error.

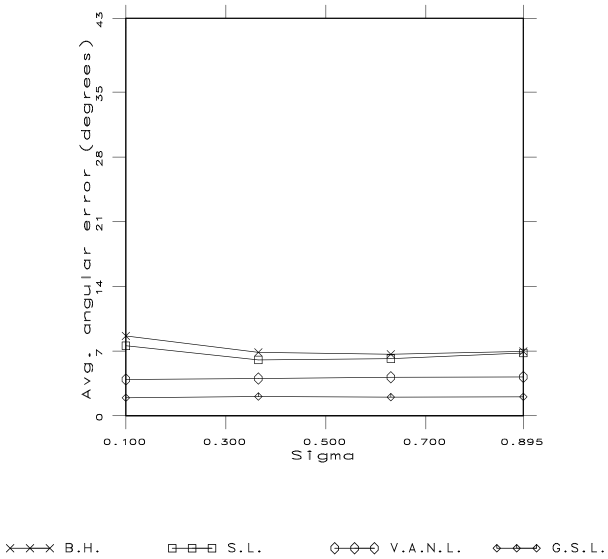


a

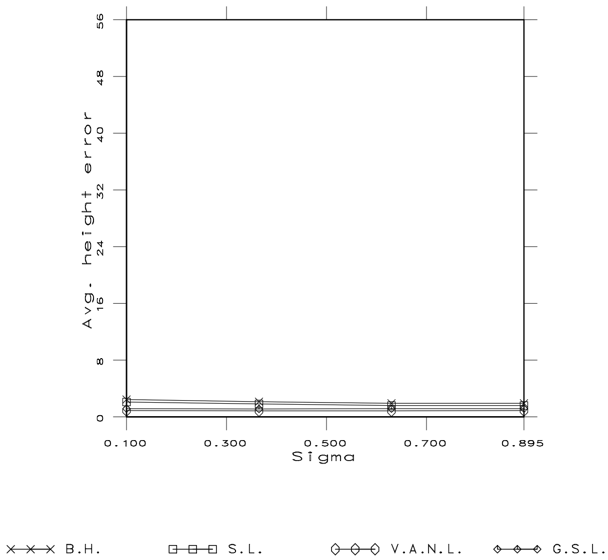


b

Figure 7.28: Results of the reflectance experiment when the input object is a sphere and $K_d = 0.63$. Plot (a) is the average angular error, Plot (b) is the average height error.



a



b

Figure 7.29: Results of the reflectance experiment when the input object is a sphere and $K_d = 0.895$. Plot (a) is the average angular error, Plot (b) is the average height error.

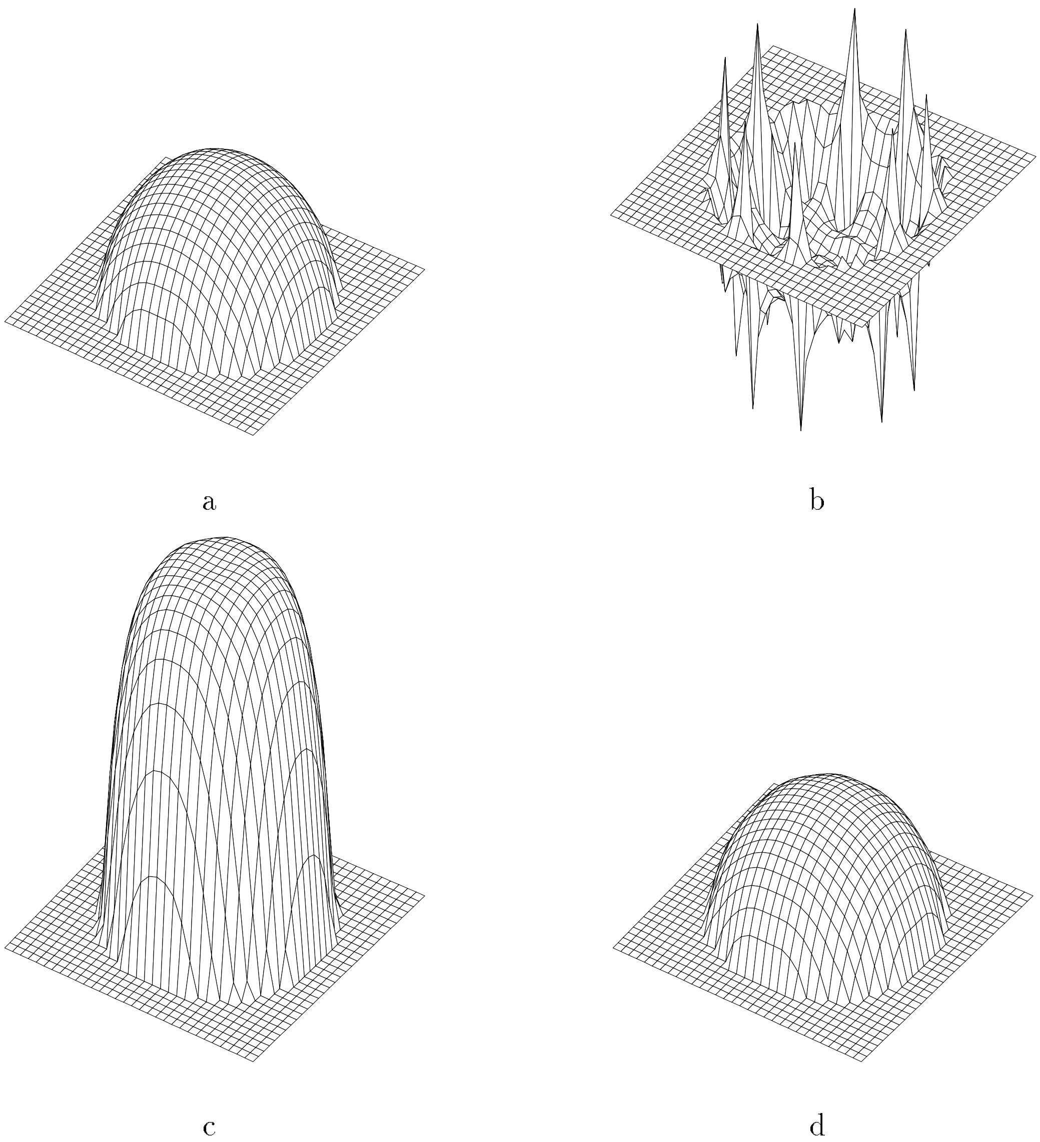


Figure 7.30: Three-dimensional mesh plots comparing results of the shape-from-shading algorithms for a sphere with $K_d = 0.10, \sigma = 0.365$ (a) the original height profile after integrating the real normals, (b) B.H. algorithm, (c) S.L. algorithm, (d) G.S.L. algorithm.

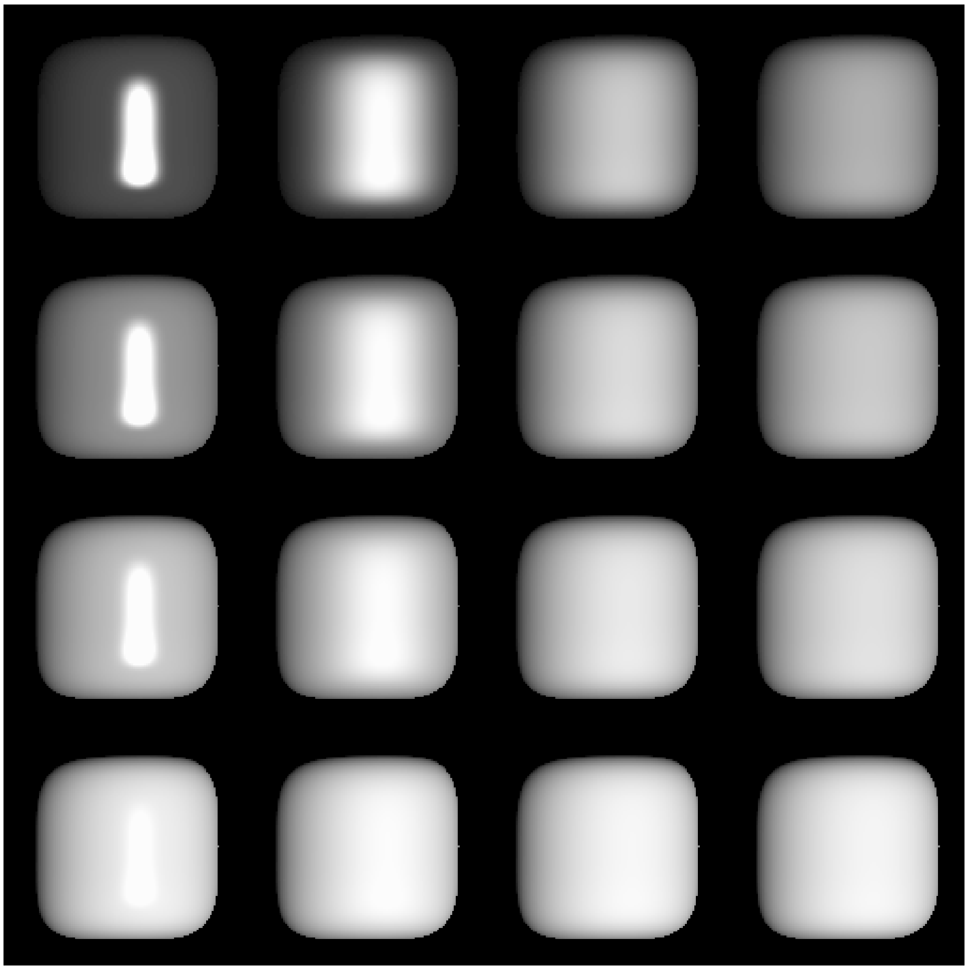


Figure 7.31: Input images used in the reflectance experiment. The object is a surface of revolution with $e = 0.5$. K_d varies from 0.10 to 0.895 in increments of 0.265 from top to bottom. σ varies from 0.10 to 0.895 in increments of 0.265 from left to right.

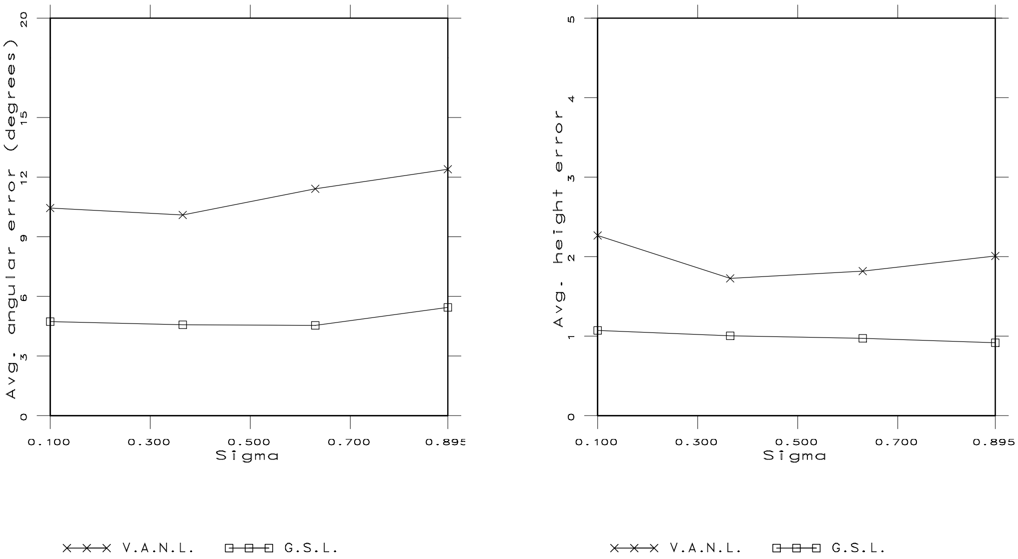


Figure 7.32: Results of the reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.10$. Left: average angular error, Right: average height error.

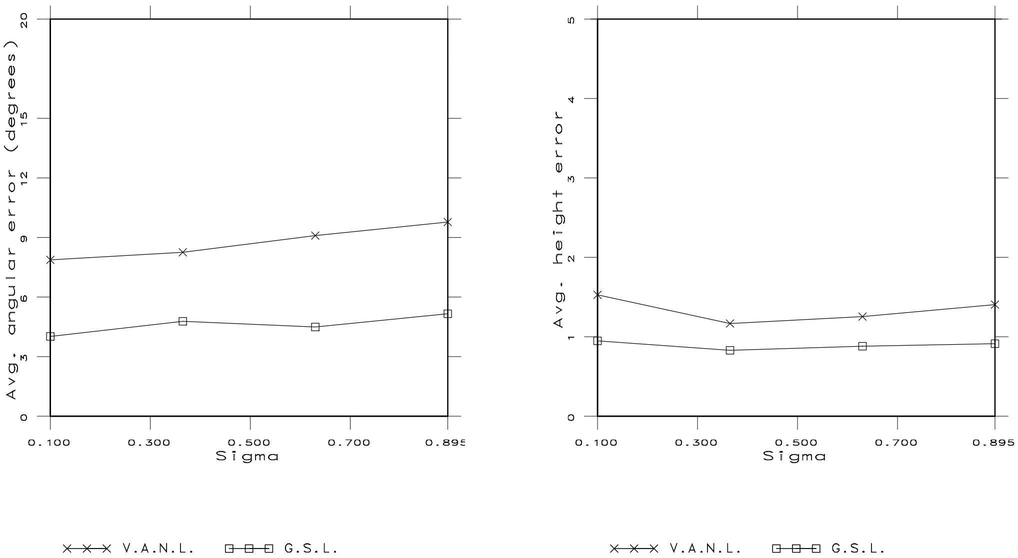


Figure 7.33: Results of the reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.365$. Left: average angular error, Right: average height error.

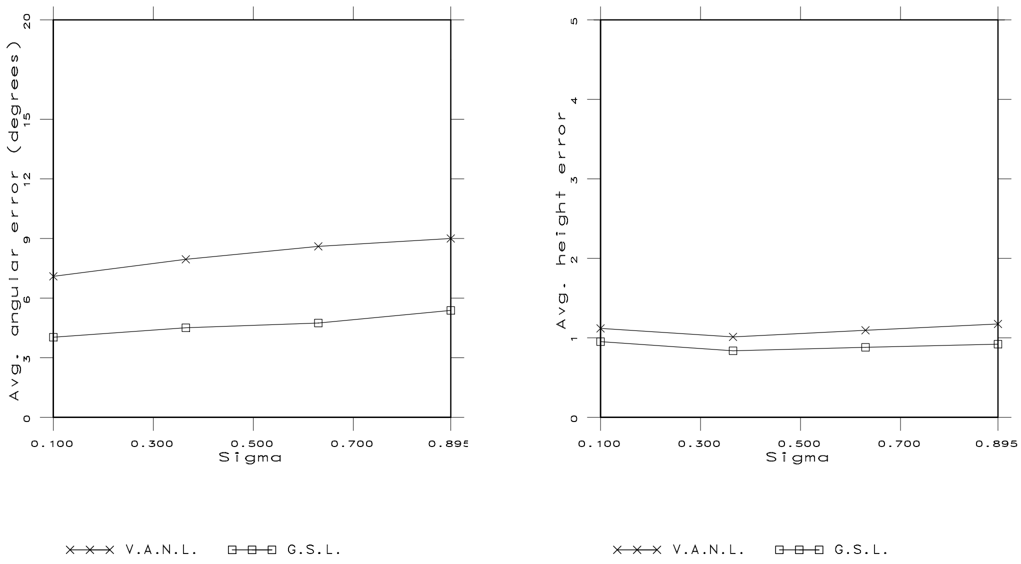


Figure 7.34: Results of the reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.63$. Left: average angular error, Right: average height error.

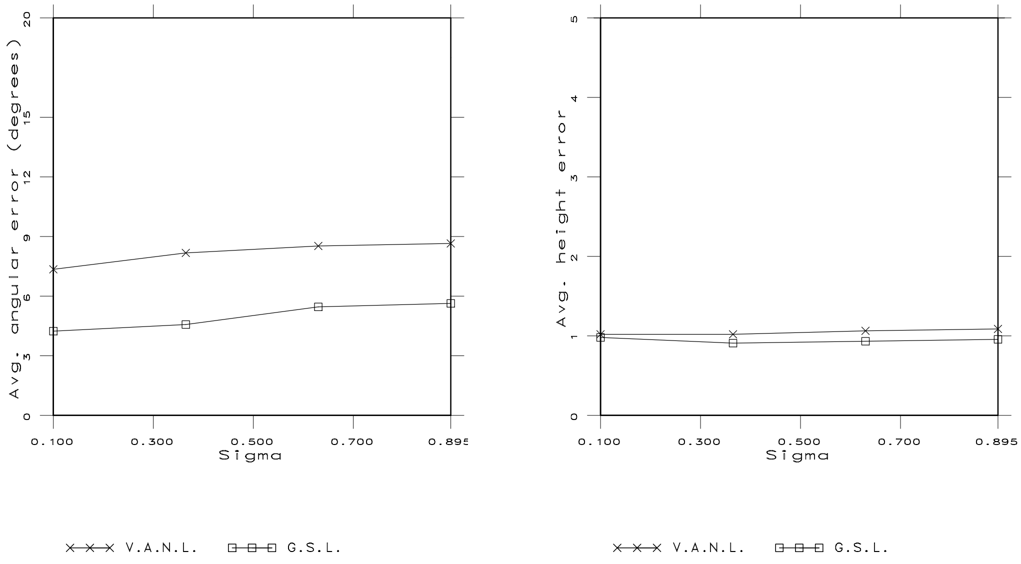


Figure 7.35: Results of the reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.895$. Left: average angular error, Right: average height error.

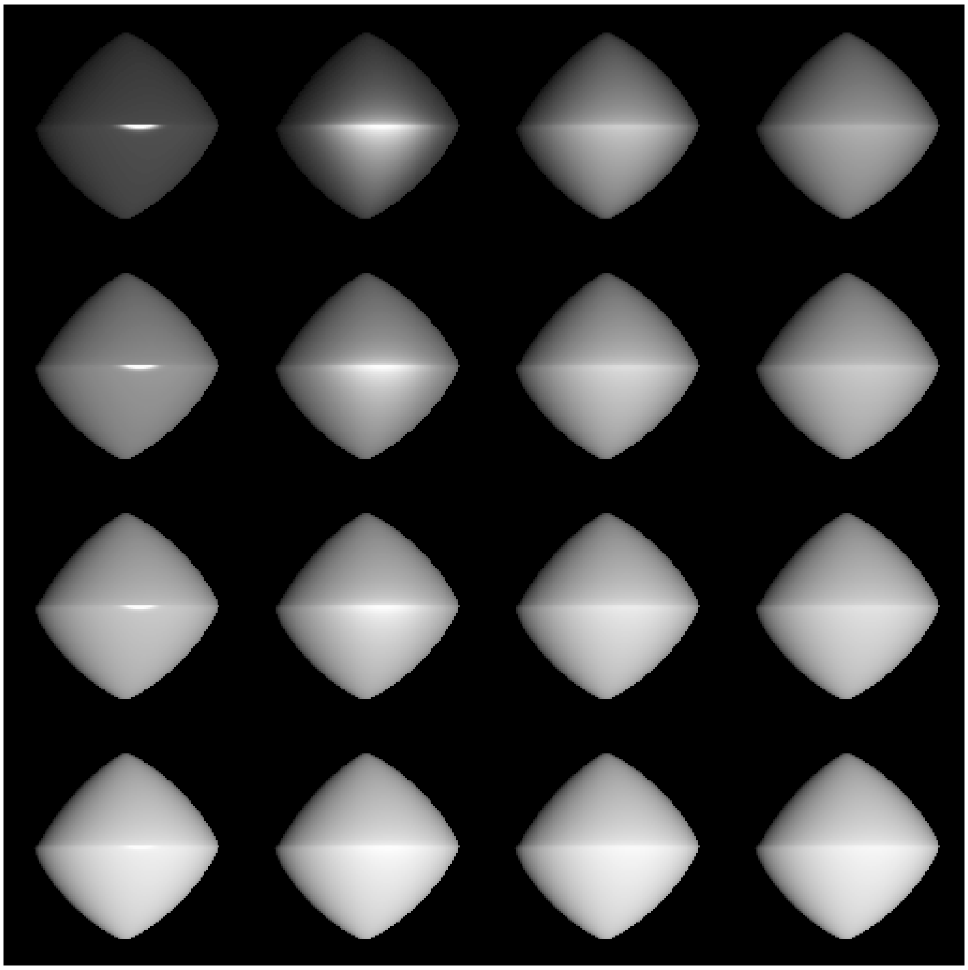


Figure 7.36: Input images used in the reflectance experiment. The object is a surface of revolution with $e = 1.5$. K_d varies from 0.10 to 0.895 in increments of 0.265 from top to bottom. σ varies from 0.10 to 0.895 in increments of 0.265 from left to right.

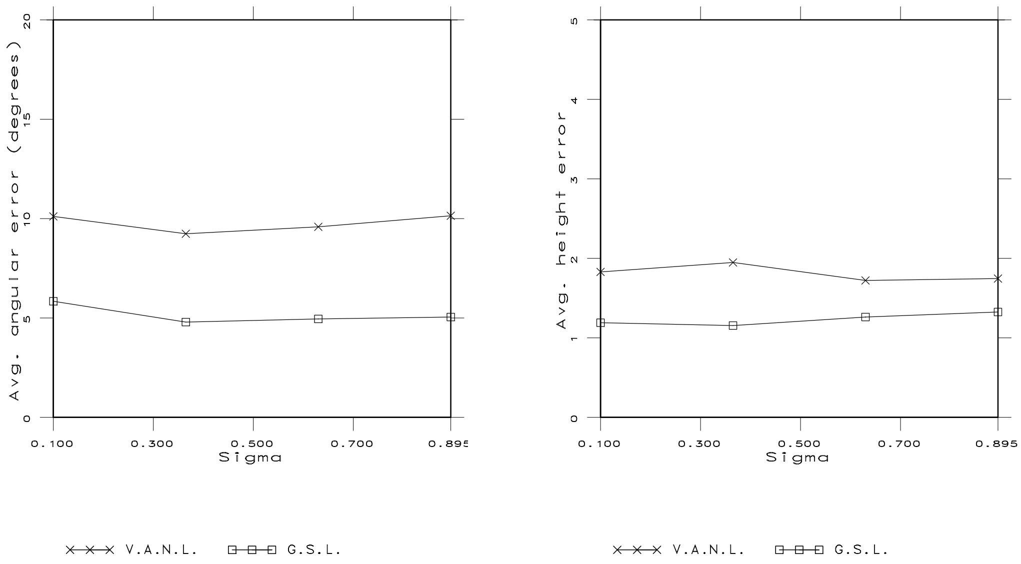


Figure 7.37: Results of the reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.10$. Left: average angular error, Right: average height error.

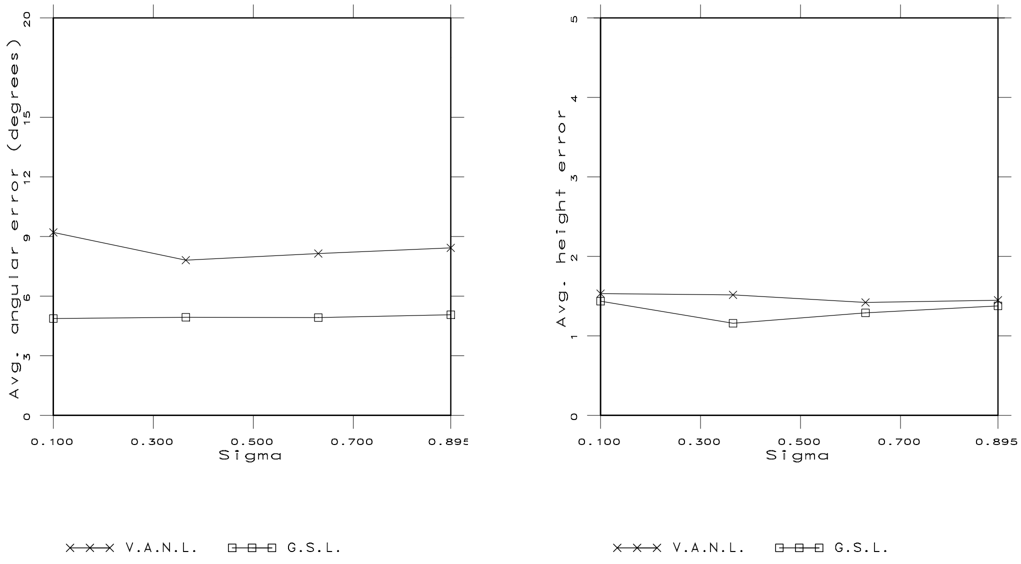


Figure 7.38: Results of the reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.365$. Left: average angular error, Right: average height error.

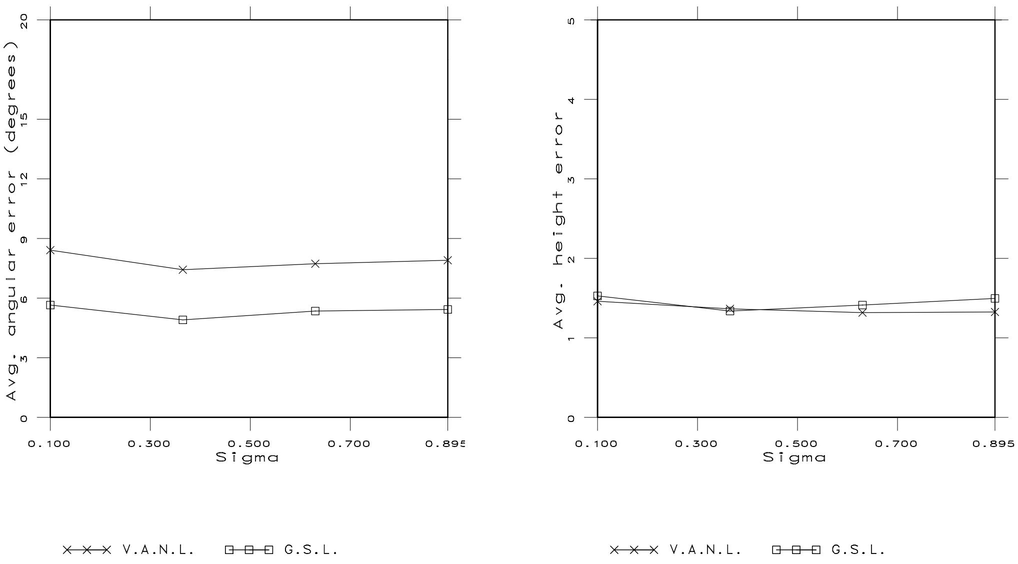


Figure 7.39: Results of the reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.63$. Left: average angular error, Right: average height error.

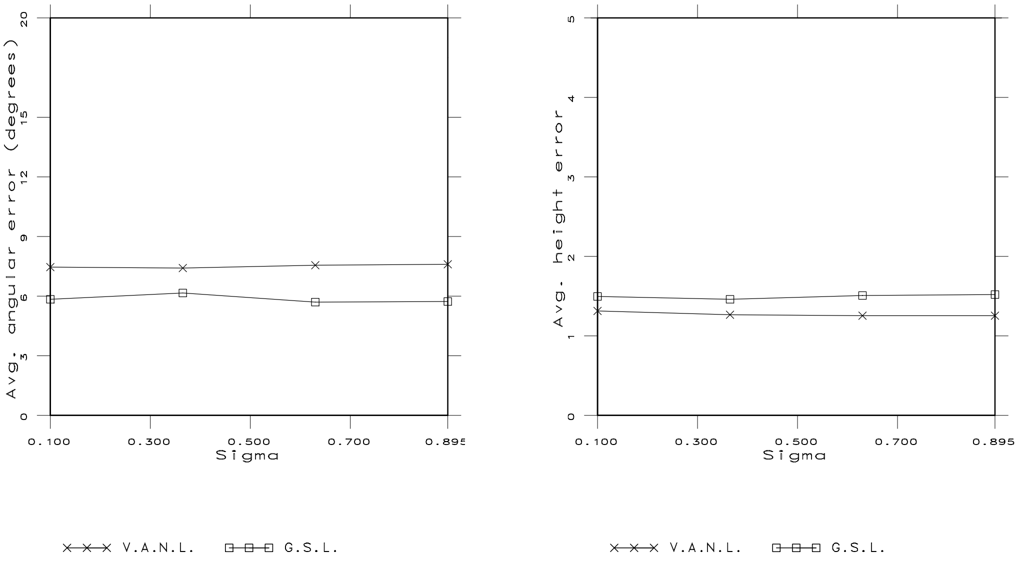


Figure 7.40: Results of the reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.895$. Left: average angular error, Right: average height error.

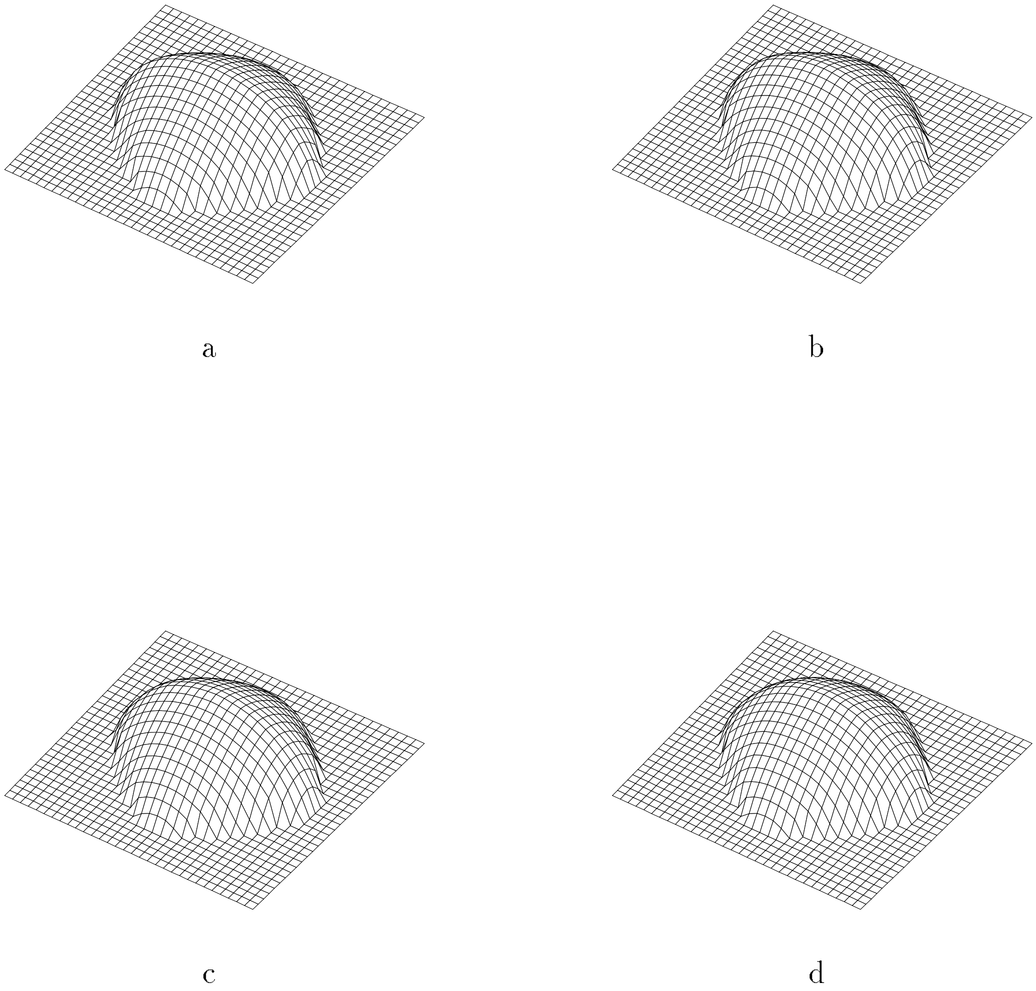


Figure 7.41: Three-dimensional mesh plots comparing results of the shape-from-shading algorithms for a surface of revolution with $e = 1.5$ as the reflectance is varied. (a) the original height profile after integrating the real normals, (b) $K_d = 0.10, \sigma = 0.365$, (c) $K_d = 0.365, \sigma = 0.365$, (d) $K_d = 0.895, \sigma = 0.365$.

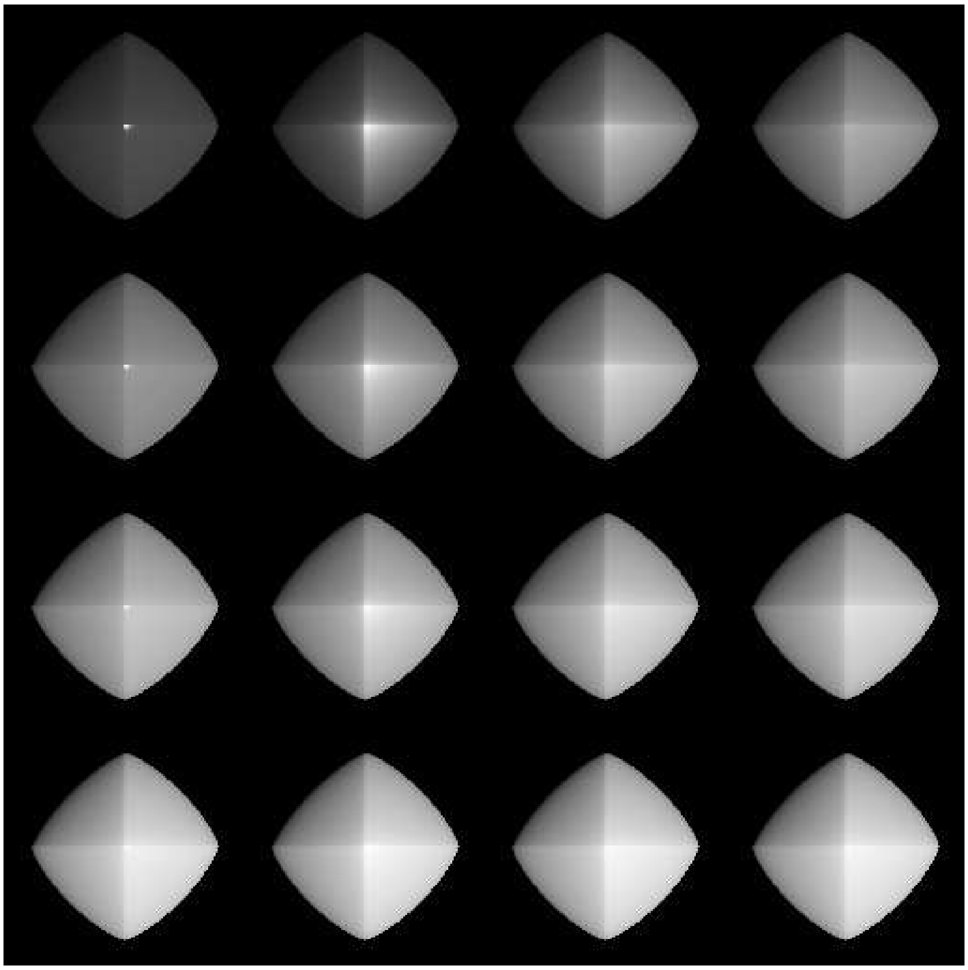


Figure 7.42: Input images used in the reflectance experiment. The object is a superellipsoid with $e = 1.5$. K_d varies from 0.10 to 0.895 in increments of 0.265 from top to bottom. σ varies from 0.10 to 0.895 in increments of 0.265 from left to right.

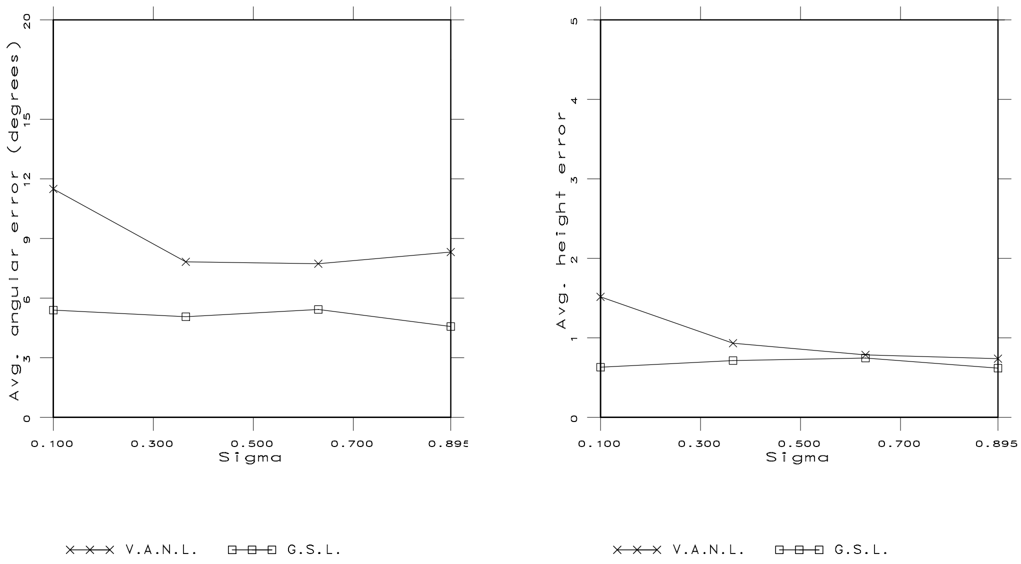


Figure 7.43: Results of the reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.10$. Left: average angular error, Right: average height error.

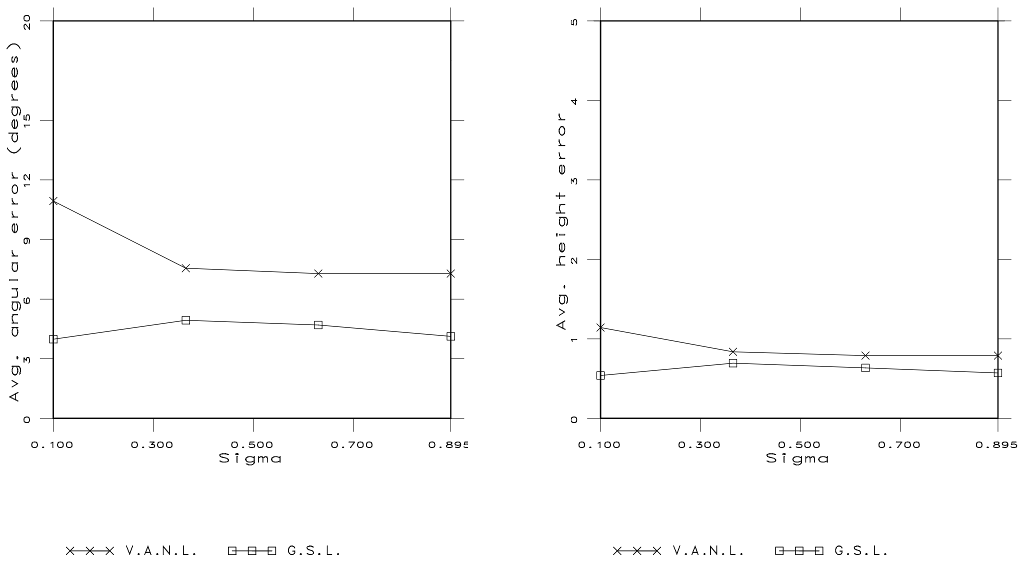


Figure 7.44: Results of the reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.365$. Left: average angular error, Right: average height error.

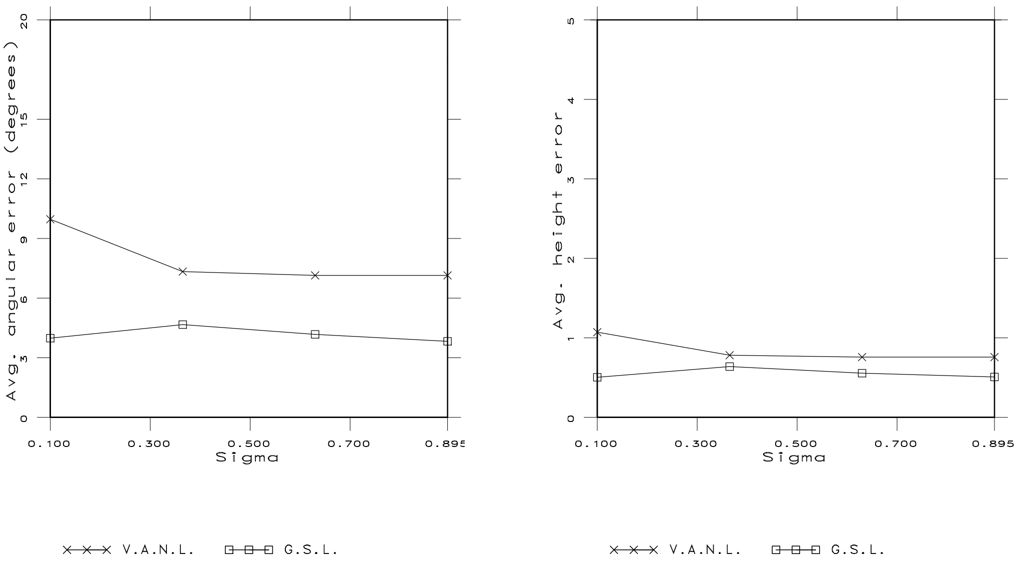


Figure 7.45: Results of the reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.63$. Left: average angular error, Right: average height error.

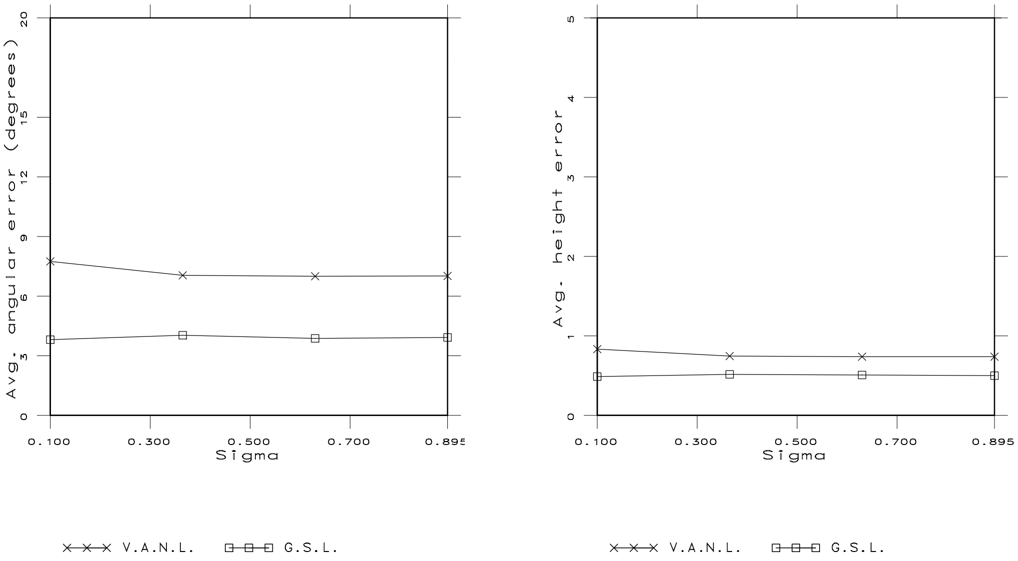


Figure 7.46: Results of the reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.895$. Left: average angular error, Right: average height error.

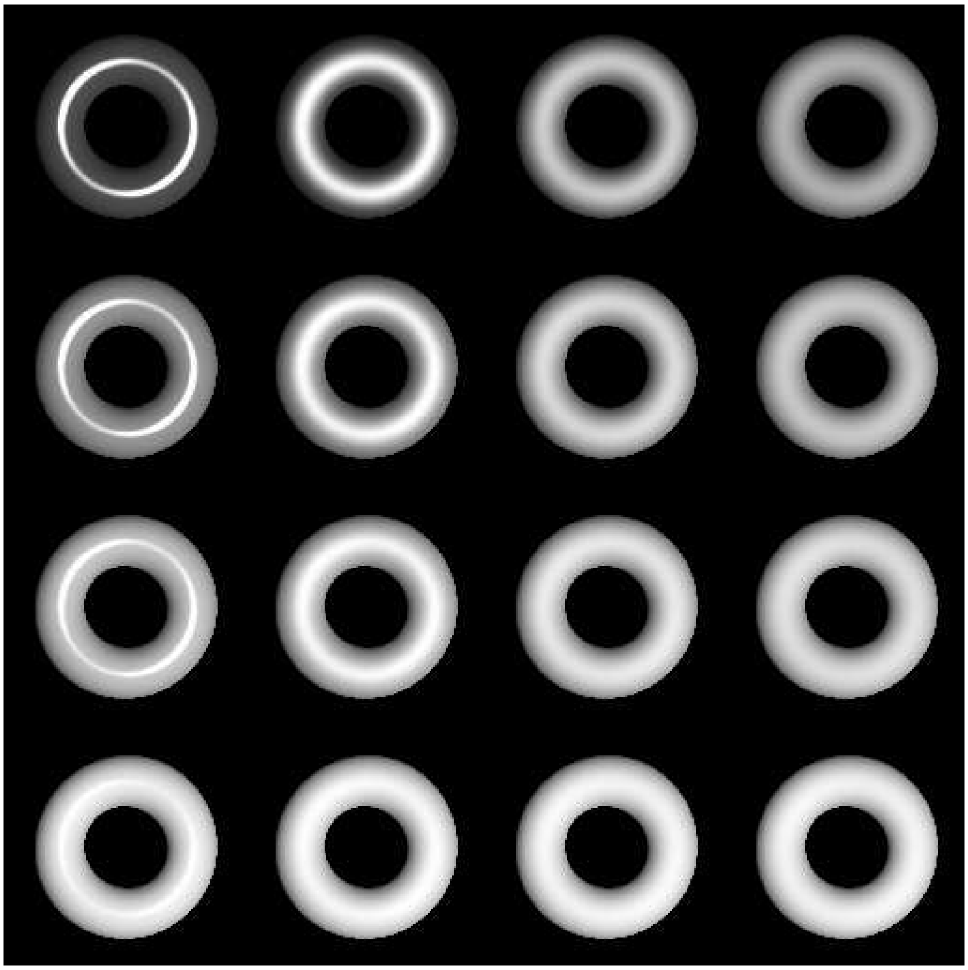


Figure 7.47: Input images used in the reflectance experiment. The object is a supertoroid with $e = 1.0$. K_d varies from 0.10 to 0.895 in increments of 0.265 from top to bottom. σ varies from 0.10 to 0.895 in increments of 0.265 from left to right.

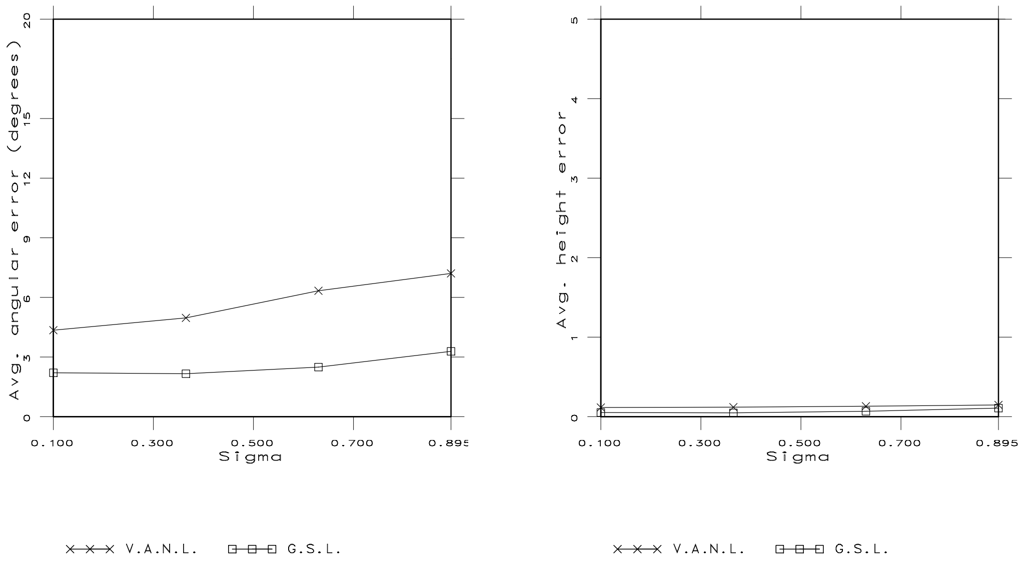


Figure 7.48: Results of the reflectance experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.10$. Left: average angular error, Right: average height error.

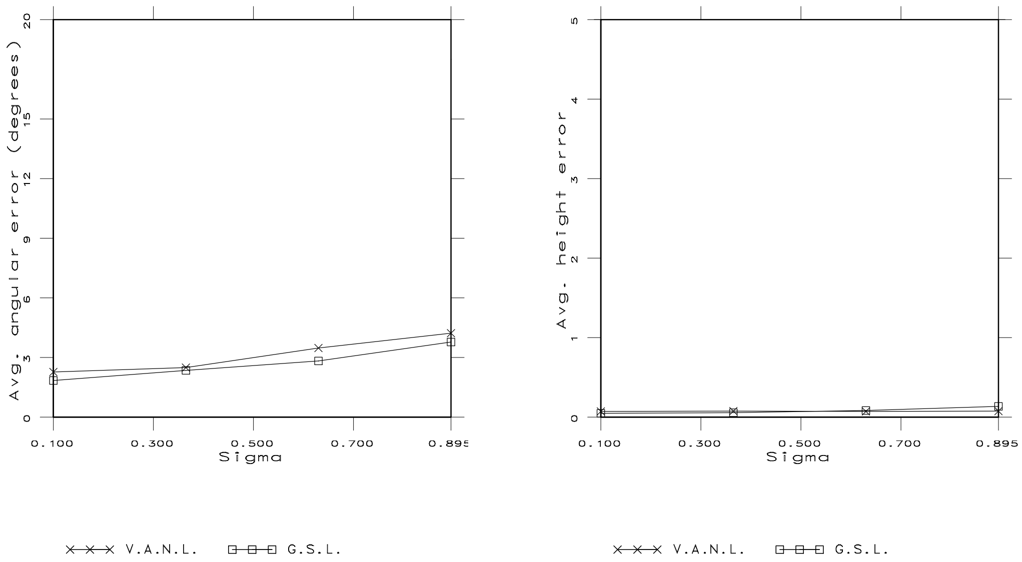


Figure 7.49: Results of the reflectance experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.365$. Left: average angular error, Right: average height error.

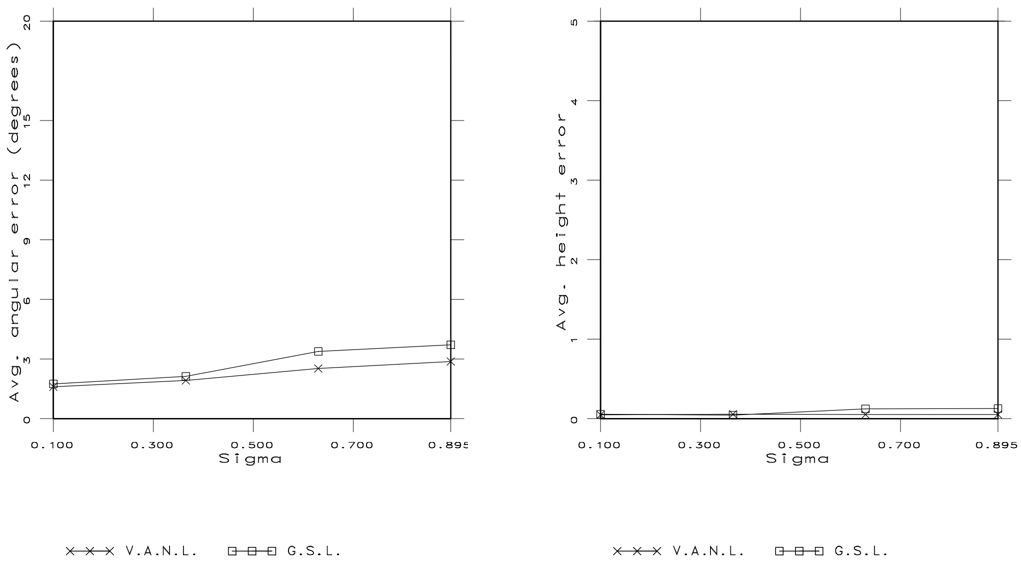


Figure 7.50: Results of the reflectance experiment when the input object is a super-toroid with $e = 1.0$ and $K_d = 0.63$. Left: average angular error, Right: average height error.

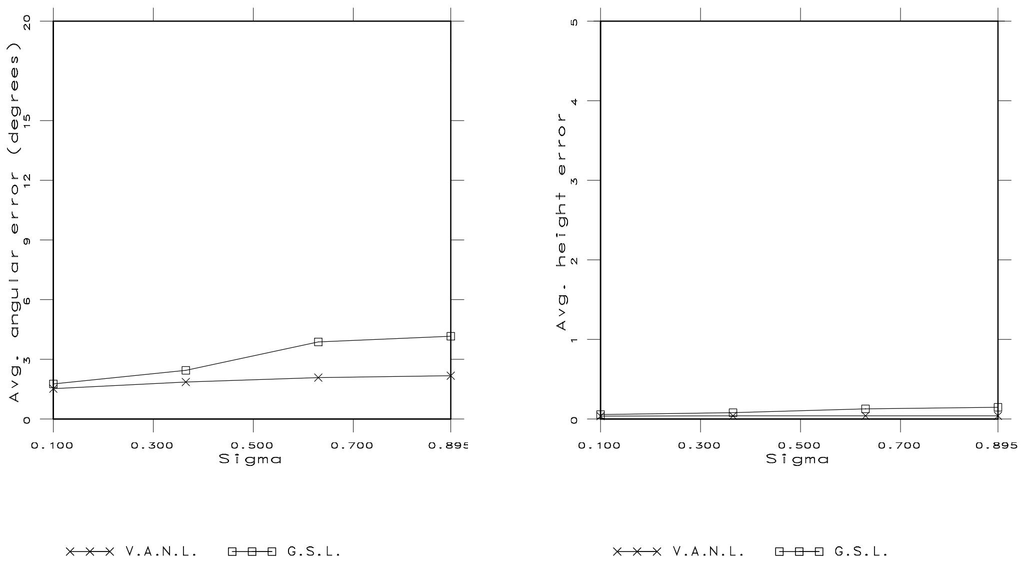


Figure 7.51: Results of the reflectance experiment when the input object is a super-toroid with $e = 1.0$ and $K_d = 0.895$. Left: average angular error, Right: average height error.

7.2.2 Effect of Illuminant Direction on Recovering Shape

In this series of experiments the V.A.N.L. is compared with the G.S.L. algorithm. The behaviour of the two algorithms is studied as the light source orientation is varied. Several objects were used to ensure that the behaviour is consistent over differently-shaped objects as well as different reflectance properties.

The input images are given in Figures 7.52, 7.54, 7.58, and 7.60. The slant of the light source is varied from 0 to 45 degrees in 15 degree increments. That is, the light source is fixed at (0.0, 0.0, -1.0) and rotated around the y-axis. The A.A.E. and A.H.E. are plotted versus the slant of the light source. Two objects are used: a surface of revolution with $e = 0.5$ and a superellipsoid with $e = 1.5$. Two sets of reflectance parameters are used for each object. The first set is a specular surface with $K_d = 0.3$ and $\sigma = 0.3$. The second set is a diffuse surface with $K_d = 0.7$ and $\sigma = 0.7$. The results of the experiment are given in Figures 7.53 to 7.61.

The V.A.N.L. has an A.A.E. that varies from 5 to 16 degrees. The A.H.E. varies from 1 to 16 units. In most cases, it is observed that as the slant of the light source increases, the ability of the V.A.N.L. to recover the shape decreases. Consider the experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$. When the slant is 0 degrees the A.A.E. is only 8.534 degrees. When the slant increases to 30 degrees the A.A.E. has increased to 12.208 degrees. Finally, when the slant is 45 degrees, the error has increased to 15.207 degrees. This is true regardless of shape or reflectance.

The G.S.L. algorithm performs much better than the V.A.N.L. The A.A.E. varies from 3 to 8 degrees and the A.H.E. varies from 1 to 2 units. The slant of the light source has little effect on the performance of the G.S.L. algorithm with regard to the A.H.E. The A.A.E. is slightly more sensitive to the illuminant direction; however, the G.S.L. algorithm is far less affected by the light source direction than the V.A.N.L.

To illustrate the behaviour of the shape-from-shading algorithms as the illuminant direction is varied, mesh diagrams are given in Figures 7.56 and 7.57. The resulting surface of the respective algorithms are presented for the surface of revolution with

$e = 0.5$ and $K_d = 0.7, \sigma = 0.7$. Figure 7.56 represents the results of the V.A.N.L. at slants 0, 15 and 45 degrees. Figure 7.57 represents the results of the G.S.L. algorithm at slants 0, 15, and 45 degrees.

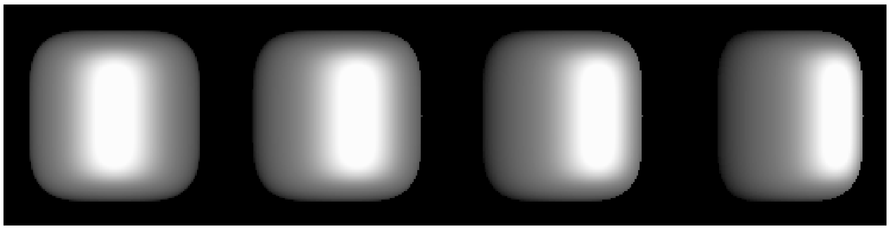


Figure 7.52: Input images used in the illuminant direction experiment. The object is a surface of revolution with $e = 0.5$ and $K_d = 0.3, \sigma = 0.3$. The slant of the light source increases from 0 to 45 degrees in increments of 15 degrees from left to right.

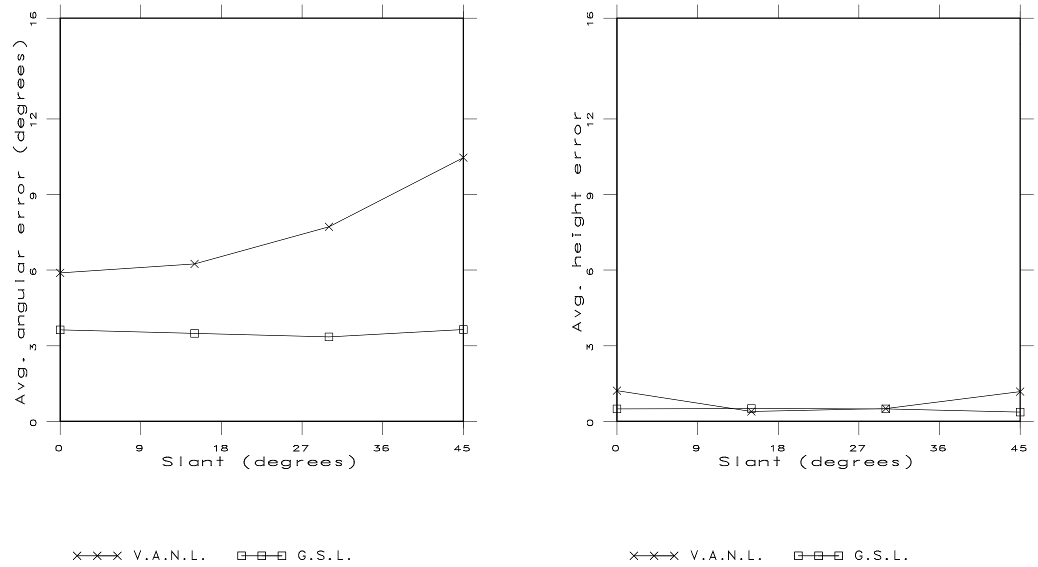


Figure 7.53: Results of the illuminant direction experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.3, \sigma = 0.3$. Left: average angular error, Right: average height error.

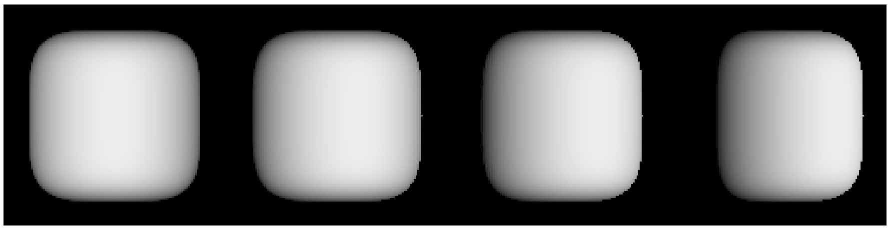


Figure 7.54: Input images used in the illuminant direction experiment. The object is a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$. The slant of the light source increases from 0 to 45 degrees in increments of 15 degrees from left to right.

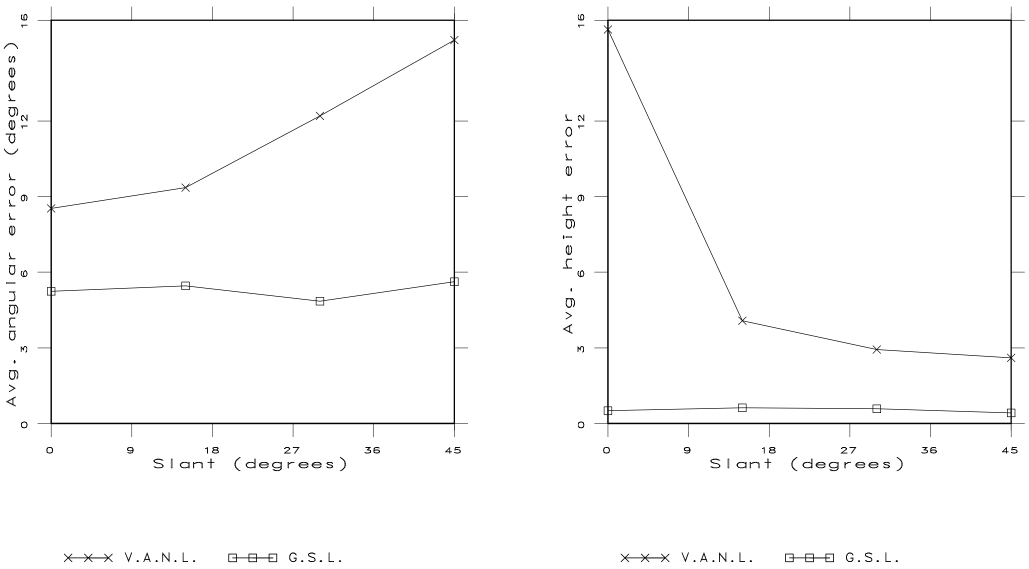


Figure 7.55: Results of the illuminant direction experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$. Left: average angular error, Right: average height error.

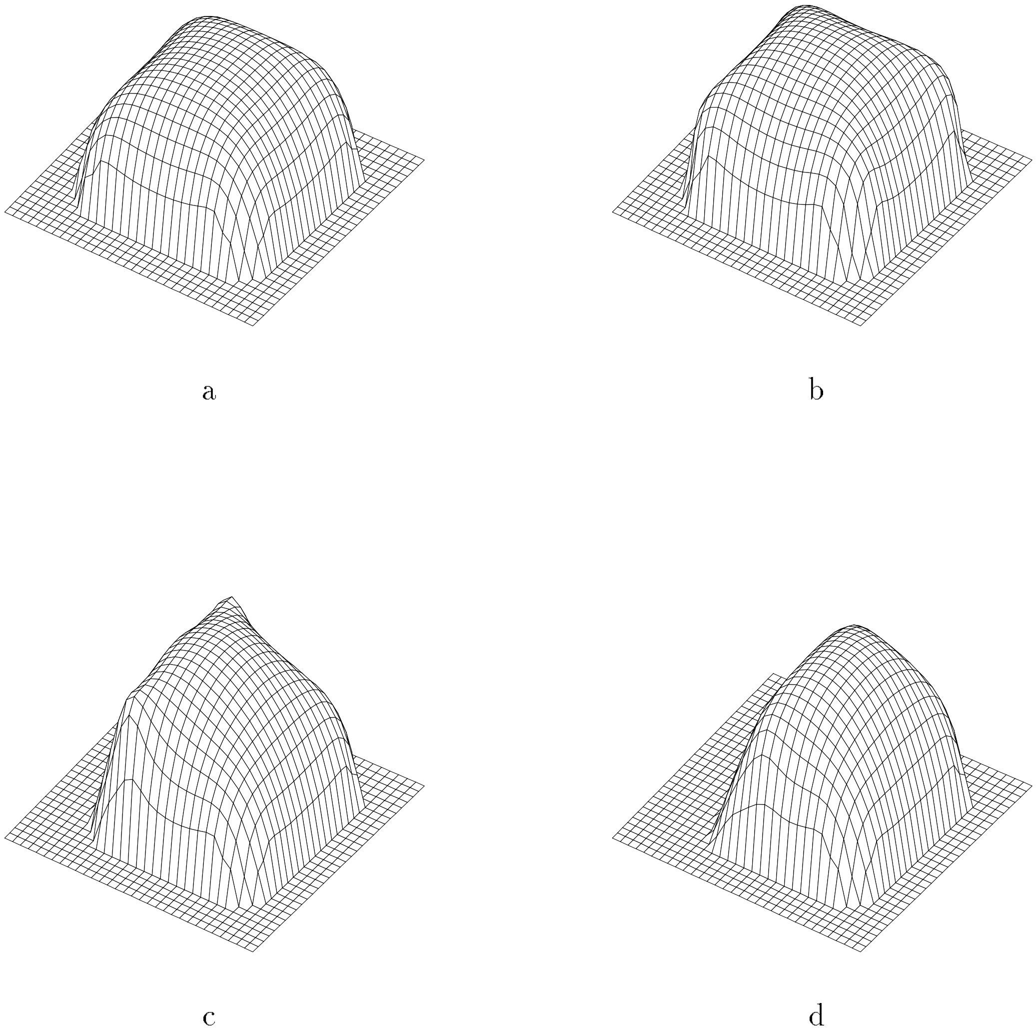
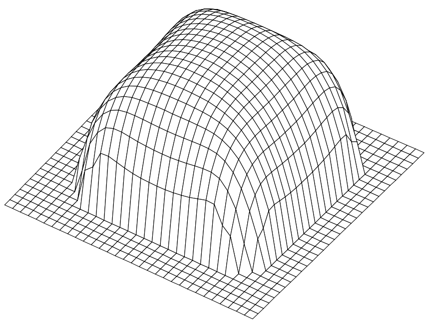
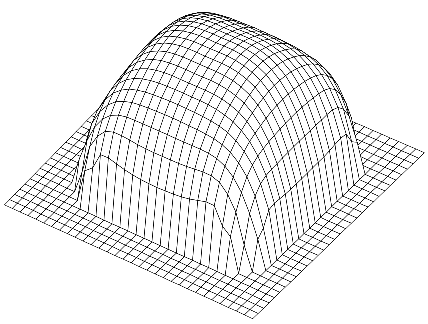


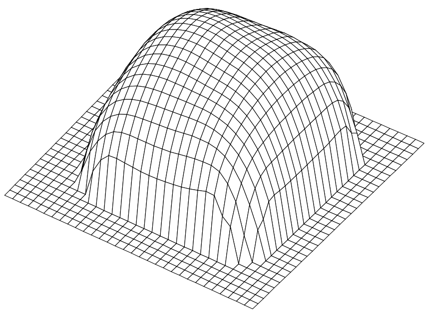
Figure 7.56: Three-dimensional mesh plots comparing results of the V.A.N.L. as the illuminant direction is varied. The input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$ (a) the original height profile after integrating the real normals, (b) slant is 0 degrees, (c) slant is 15 degrees, (d) slant is 45 degrees.



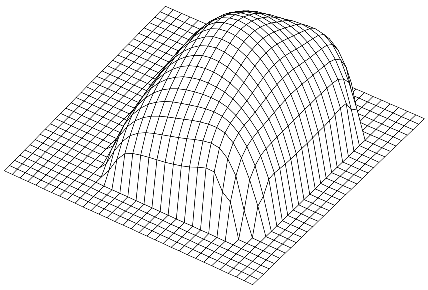
a



b



c



d

Figure 7.57: Three-dimensional mesh plots comparing results of the G.S.L. algorithm as the illuminant direction is varied. Input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7, \sigma = 0.7$ (a) the original height profile after integrating the real normals, (b) slant is 0 degrees, (c) slant is 15 degrees, (d) slant is 45 degrees.

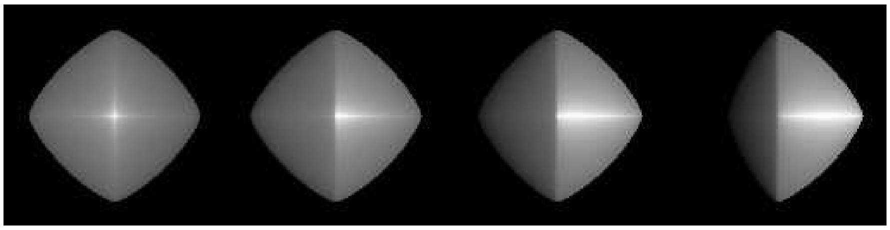
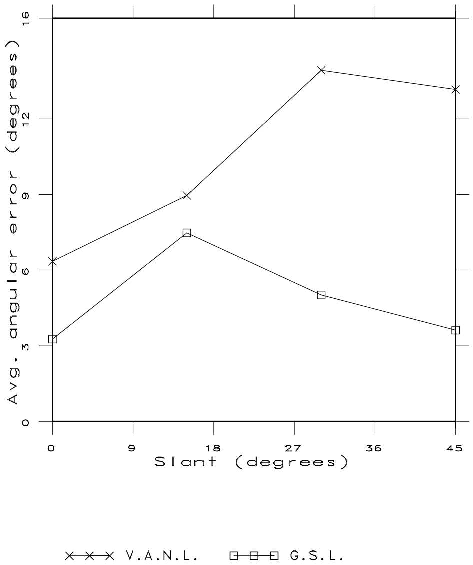


Figure 7.58: Input images used in the illuminant direction experiment. The object is a superellipsoid with $e = 1.5$ and $K_d = 0.3, \sigma = 0.3$. The slant of the light source increases from 0 to 45 degrees in increments of 15 degrees from left to right.



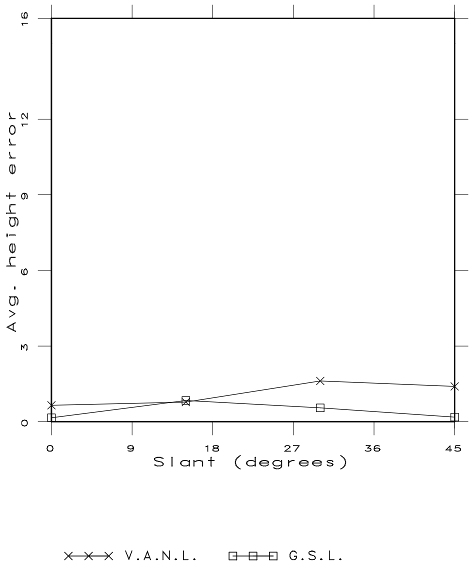


Figure 7.59: Results of the illuminant direction experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.3, \sigma = 0.3$. Left: average angular error, Right: average height error.

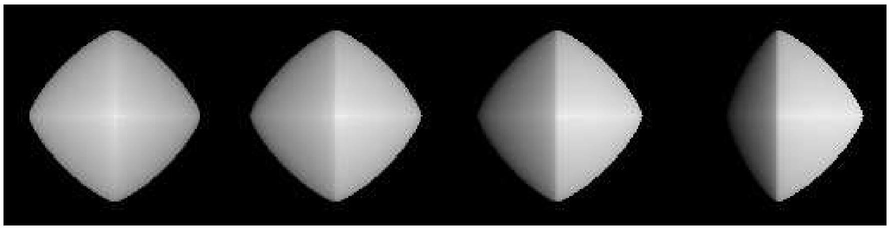


Figure 7.60: Input images used in the illuminant direction experiment. The object is a superellipsoid with $e = 1.5$ and $K_d = 0.7, \sigma = 0.7$. The slant of the light source increases from 0 to 45 degrees in increments of 15 degrees from left to right.

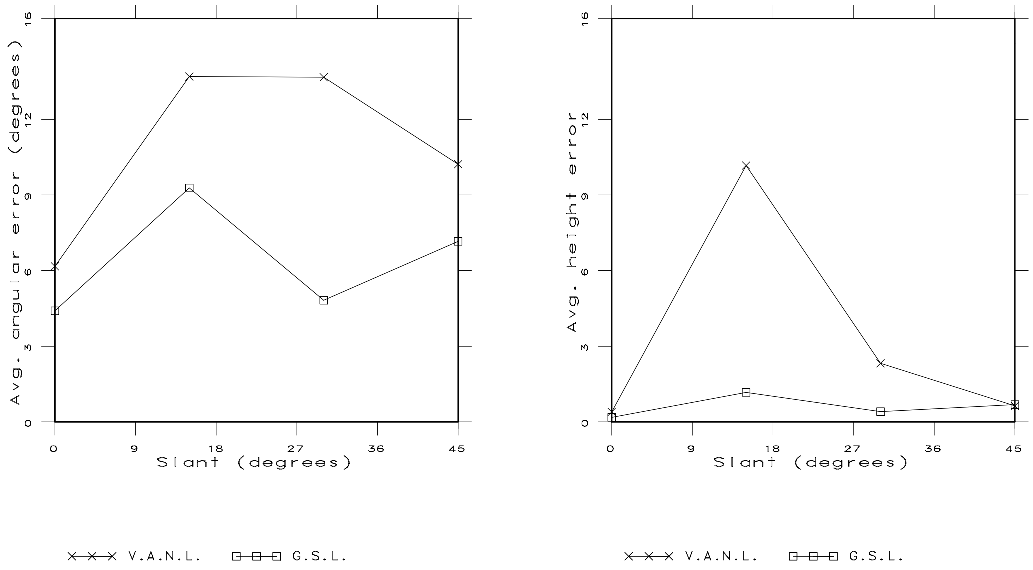


Figure 7.61: Results of the illuminant direction experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.7, \sigma = 0.7$. Left: average angular error, Right: average height error.

7.2.3 Effect of Object Shape on Recovering Shape

In this set of experiments, the shape parameters of the supertoroid, superellipsoid and surface of revolution are varied to determine the behaviour of the shape-from-shading algorithms as the object shape changes. Two parameters, e_1 and e_2, control the shape of the superquadric equations defined in Section 6.1. To reduce the number of parameters to one, it will be assumed that $e_1 = e_2 = e$. By varying e, a wide variety of shapes can be synthesized. It can be seen in Figure 6.1 that the shape of the object varies from a square planar surface to a sharply pointed diamond shaped object as e is increased. Note that $e = 1.0$ represents a standard toroid or ellipsoid.

Objects formed by rotating a curve around an axis is the subject of Section 6.2. By using a two-dimensional superquadric equation as the curve of rotation, a wide variety of surfaces can be created. Unlike the superquadric surfaces, the objects formed by rotating a curve around an axis are not symmetrical in three dimensions. Such objects are useful in shape-from-shading experiments as commonly-seen objects are not necessarily perfectly symmetrical. The two-dimensional superquadric curve is chosen because the shape of the object can be varied in a predictable fashion. In Figure 6.3 it is shown that as e increases, the shape of the two-dimensional superquadric curve changes from a square concave shape to a convex sharply pointed curve. Figure 6.4 displays the diverse range of three-dimensional objects that can be synthesized by rotating this curve around the $y-$axis.

All three object types are used (surface of revolution, superellipsoids, and supertoroids). The shape parameter, e, is varied from 0.2 to 1.8 in increments of 0.4. All of the different objects are rendered as both specular ($K_d = 0.3, \sigma = 0.3$) and diffuse ($K_d = 0.7, \sigma = 0.7$), to study whether the trends observed as the shape is varied is consistent over different reflectance types. The input objects are given in Figures 7.62, 7.64, 7.66, 7.68, 7.70, and 7.72. The G.S.L. algorithm and V.A.N.L. are tested and the A.H.E. and A.A.E. is plotted as e is varied. The results of the experiments are given in Figures 7.63 to 7.73.

The shape of the object has a dramatic effect on the ability of the V.A.N.L.

to reconstruct surface shape. The A.A.E. varies from 1 to 40 degrees, while the A.H.E. varies from 1 to 13 units. In all cases, the V.A.N.L. performs better when the sharpness is close to 1.0. That is, when the shape is standard (either a sphere, ellipsoid, or toroid), the V.A.N.L. can recover the shape effectively. For example, consider the surface of revolution with surface parameters $K_d = 0.7$ and $\sigma = 0.7$. When the shape is cylindrical ($e = 0.2$) the A.A.E. is 8.87 degrees. When the shape is a sphere ($e = 1.0$) the A.A.E. using the V.A.N.L. is 0.47 degrees. However, when e increases to 1.8, the A.A.E. also increases to 15.2 degrees.

The V.A.N.L. performs better on diffuse surfaces than specular surfaces. This observation was also noted in Subsection 7.2.1. The V.A.N.L. has a difficult time recovering the shape of supertoroids with high degrees of sharpness.

The G.S.L. algorithm outperforms the V.A.N.L. The A.A.E. varies between 1 and 24 degrees, while the A.H.E. range of error is between 1 and 2 units. The G.S.L. algorithm performs slightly better for superellipsoid and surfaces of revolution with standard shapes (when e is close to 1.0) but it is far less affected by surface shape than the V.A.N.L. surfaces. This phenomenon is not observed when the input object is a supertoroid. Small values of e give good results; large values of e produce high errors.

Both algorithms perform slightly less effectively on non-symmetrical objects. The superellipsoid and surface of revolution have the same parameters. That is, the width, length, height, and sharpness are the same; the only difference is that the surface of revolution is less symmetrical than the superellipsoid. With input objects being surfaces of revolution, the A.A.E. over all objects using the G.S.L. algorithm is 3.543 degrees. However, for the superellipsoids, the A.A.E. is 5.034 degrees.

A sample of the reconstructed mesh diagrams are given in Figure 7.74. The input object is a supertoroid with $K_d = 0.7, \sigma = 0.7$, the error plots using this object are presented in Figure 7.73.

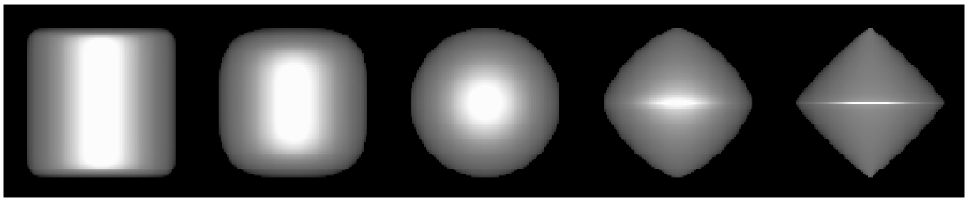


Figure 7.62: Input images used in the shape experiment. The object is a surface of revolution with $K_d = 0.3, \sigma = 0.3$. The shape parameter, e, varies from 0.2 to 1.8 in increments of 0.4.

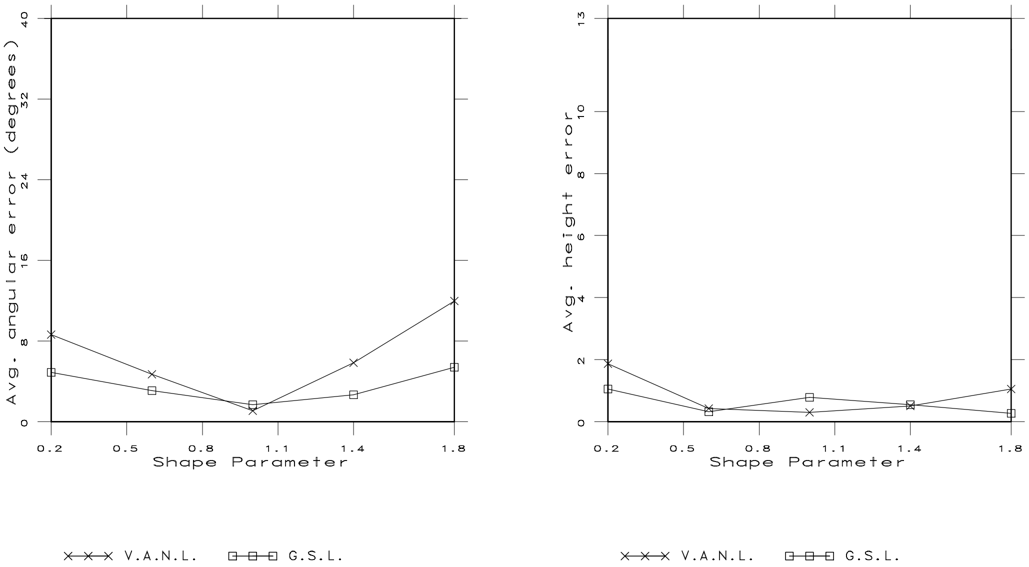


Figure 7.63: Results of the shape experiment when the input object is a surface of revolution with $K_d = 0.3, \sigma = 0.3$. Left: average angular error, Right: average height error.

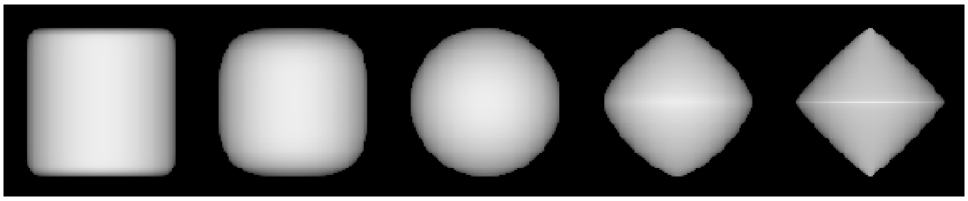


Figure 7.64: Input images used in the shape experiment. The object is a surface of revolution with $K_d = 0.7$, $\sigma = 0.7$. The shape parameter, e, varies from 0.2 to 1.8 in increments of 0.4.

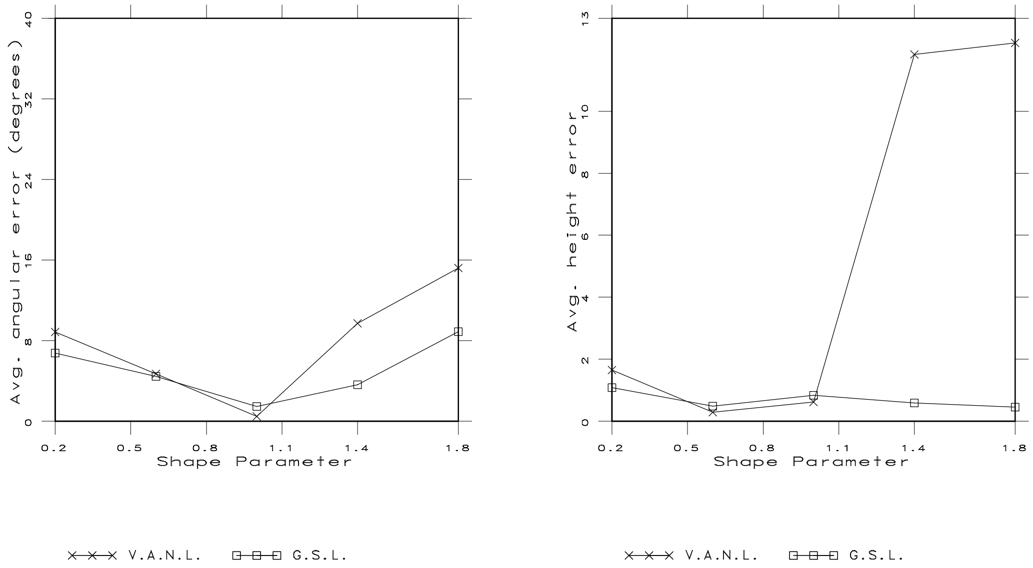


Figure 7.65: Results of the shape experiment when the input object is a surface of revolution with $K_d = 0.7$ and $\sigma = 0.7$. Left: average angular error, Right: average height error.

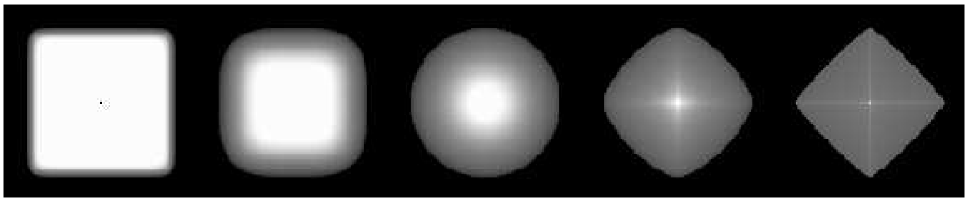
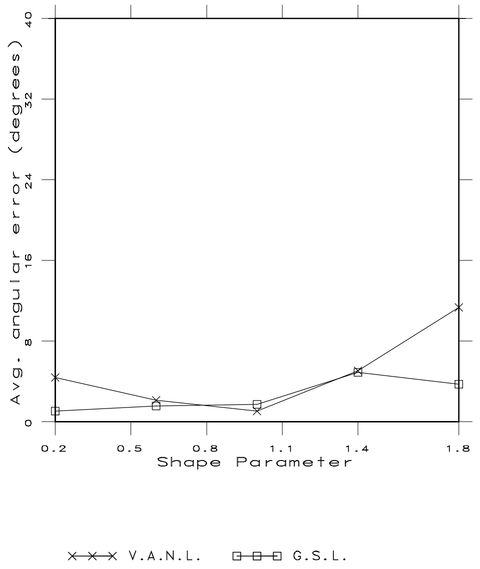


Figure 7.66: Input images used in the shape experiment. The object is a superellipsoid with $K_d = 0.3$ and $\sigma = 0.3$. The shape parameter, e, varies from 0.2 to 1.8 in increments of 0.4.



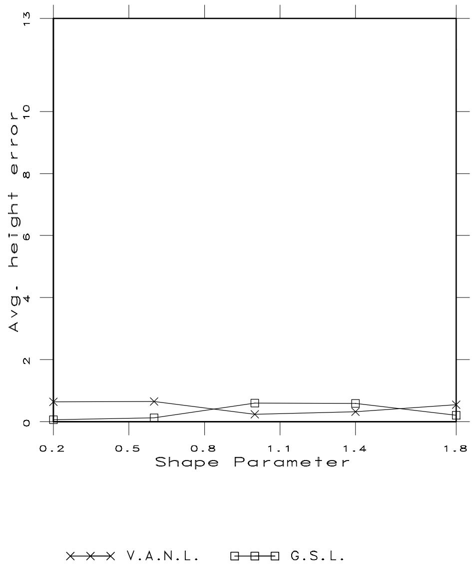


Figure 7.67: Results of the shape experiment when the input object is a superellipsoid with $K_d = 0.3$ and $\sigma = 0.3$. Left: average angular error, Right: average height error.

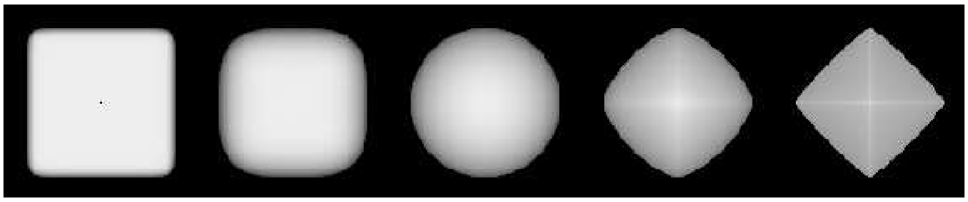


Figure 7.68: Input images used in the shape experiment. The object is a superellipsoid with $K_d = 0.7, \sigma = 0.7$. The shape parameter, e, varies from 0.2 to 1.8 in increments of 0.4.

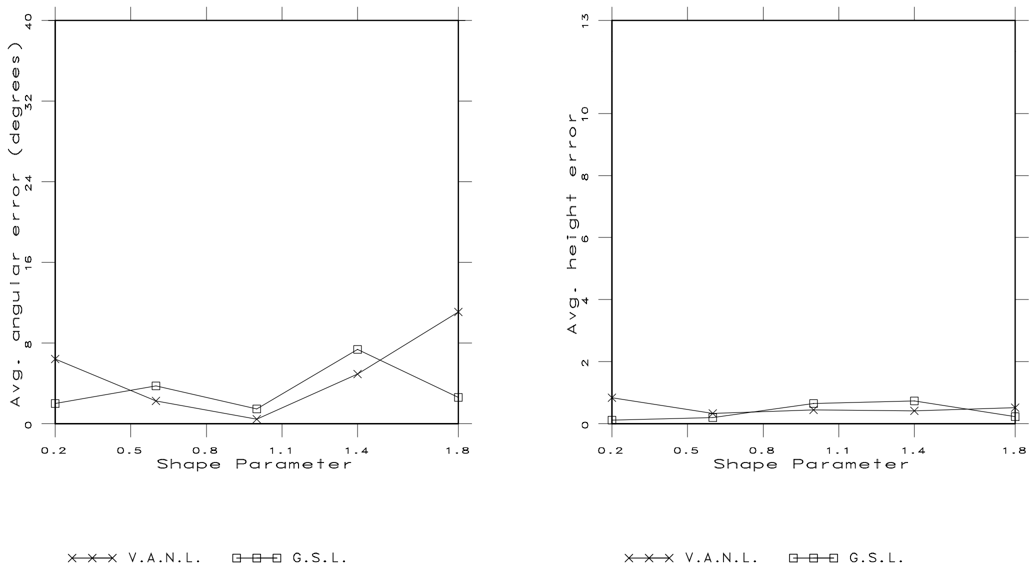


Figure 7.69: Results of the shape experiment when the input object is a superellipsoid with $K_d = 0.7$ and $\sigma = 0.7$. Left: average angular error, Right: average height error.

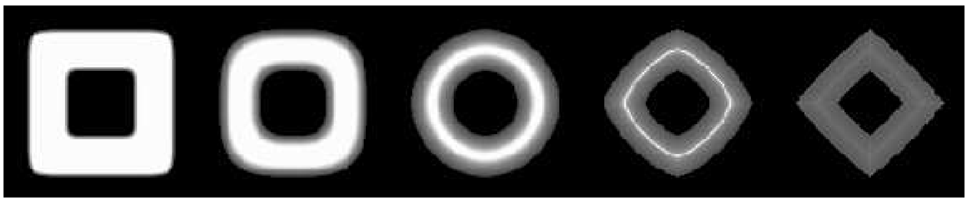


Figure 7.70: Input images used in the shape experiment. The object is a supertoroid with $K_d = 0.3$ and $\sigma = 0.3$. The shape parameter, e, varies from 0.2 to 1.8 in increments of 0.4.

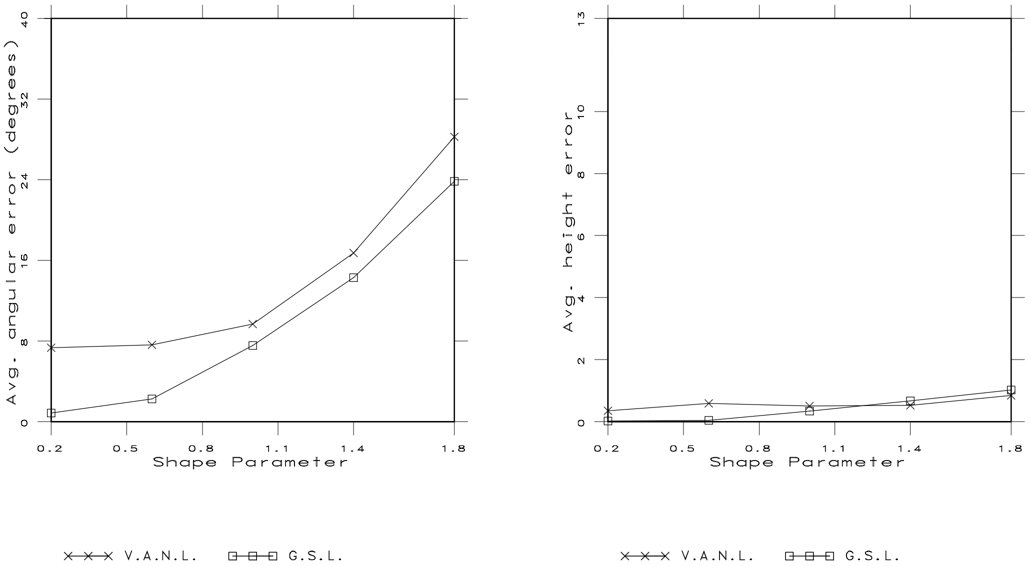


Figure 7.71: Results of the shape experiment when the input object is a supertoroid with $K_d = 0.3$ and $\sigma = 0.3$. Left: average angular error, Right: average height error.

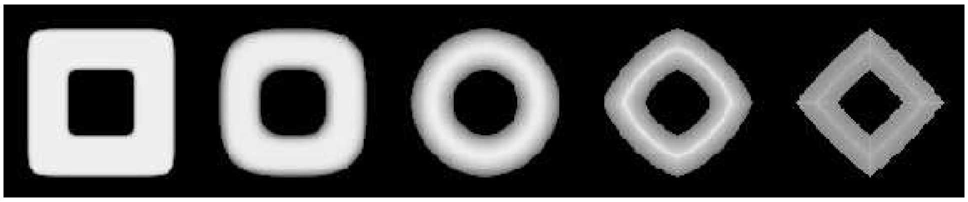


Figure 7.72: Input images used in the shape experiment. The object is a supertoroid with $K_d = 0.7, \sigma = 0.7$. The shape parameter, e, varies from 0.2 to 1.8 in increments of 0.4.

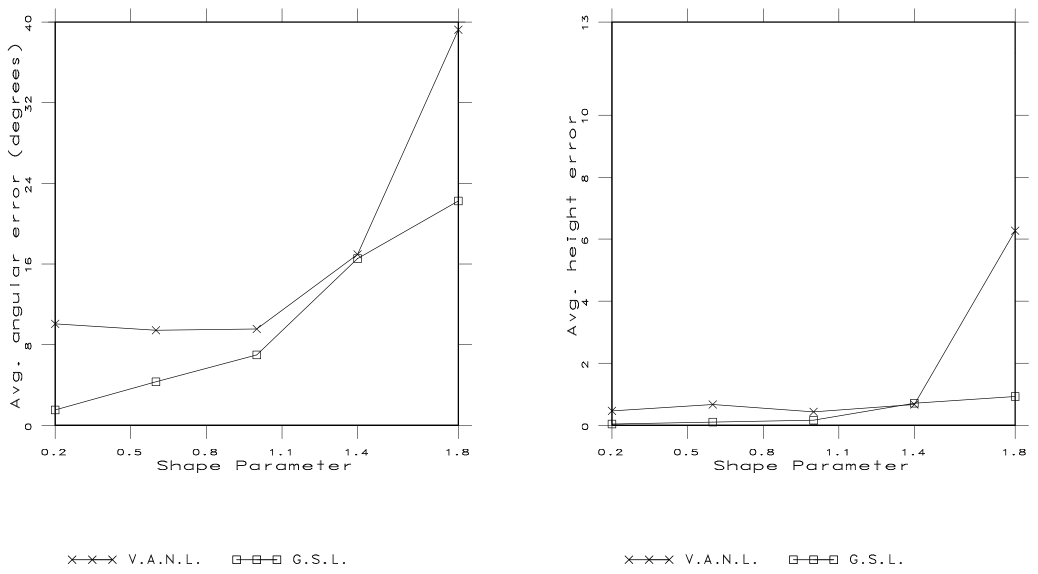


Figure 7.73: Results of the shape experiment when the input object is a supertoroid with $K_d = 0.7$ and $\sigma = 0.7$. Left: average angular error, Right: average height error.

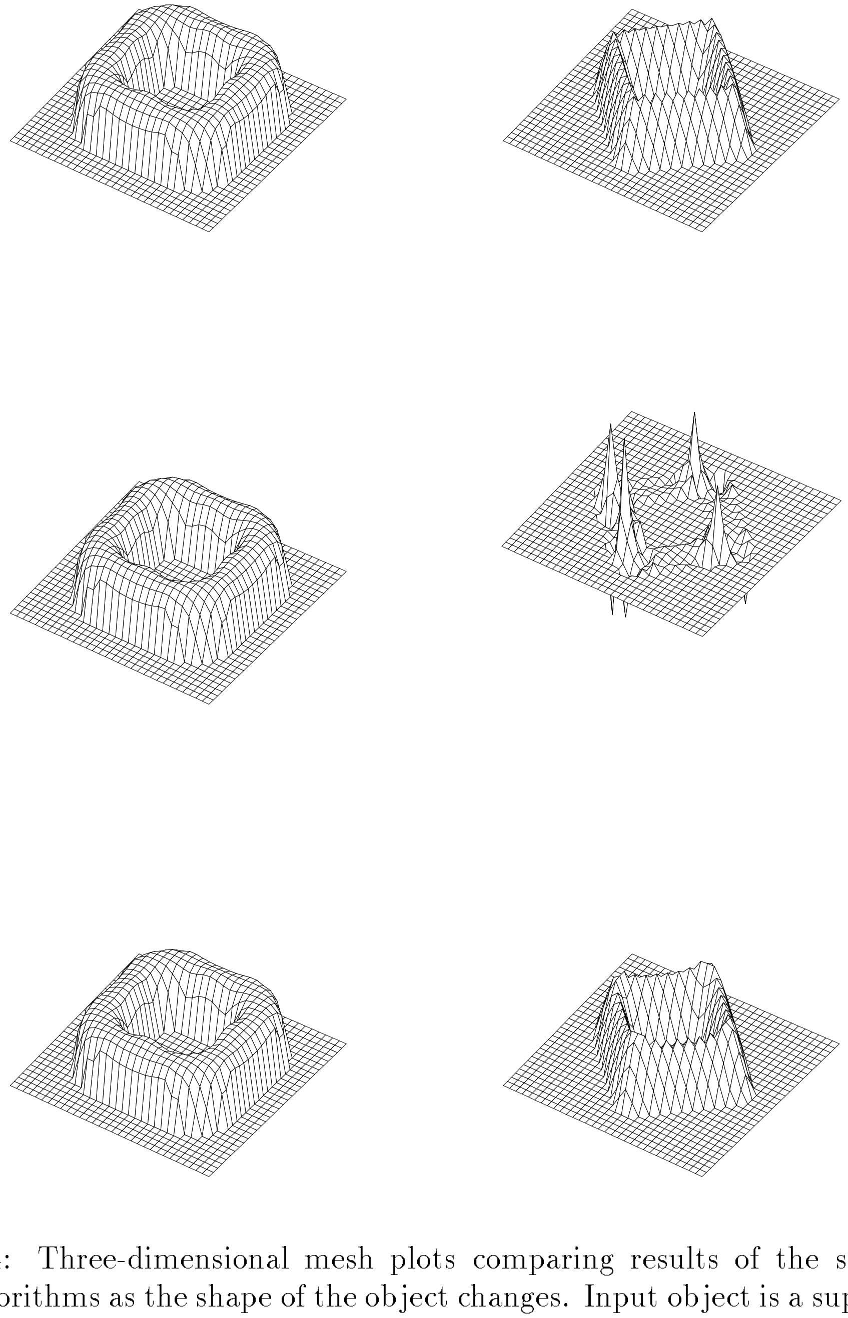


Figure 7.74: Three-dimensional mesh plots comparing results of the shape-from-shading algorithms as the shape of the object changes. Input object is a superellipsoid with $K_d = 0.7, \sigma = 0.7$. Left: $e = 0.6$, Right: $e = 1.8$, Top: the original height profile after integrating the real normals, Middle: results using V.A.N.L., Bottom: results using the G.S.L. algorithm.

7.2.4 Summary of Shape-From-Shading Experiments

In Subsection 7.2.1, experiments were performed to study the behaviour of several shape-from-shading algorithms as the reflectance was varied. It was found that the shape-from-shading algorithms for Lambertian surfaces could not effectively recover the shape of objects with non-Lambertian surfaces. However, the two new algorithms (V.A.N.L. and G.S.L. algorithm) can consistently recover the shape of non-Lambertian surfaces.

The G.S.L. algorithm was not affected by the reflectance properties of the object's surface. The G.S.L. algorithm performed well on all objects with all reflectance parameters. However, the V.A.N.L. performs slightly better on surfaces that are more diffuse than on specular surfaces. In all cases, the G.S.L. algorithm outperformed the V.A.N.L.

In Subsection 7.2.2 the performance of the G.S.L. algorithm and V.A.N.L. was studied as the illuminant direction varied. It was found that as the slant of the light source increases, the ability of the V.A.N.L to recover shape decreases. The G.S.L. algorithm outperformed the V.A.N.L. and was also less sensitive to the slant of the light source.

In Subsection 7.2.3 the behaviour of the shape-from-shading algorithms was studied as the shape of an object was varied. It was observed that the shape of the object had a dramatic effect on the ability of the algorithms to reconstruct the shape. When e is close to 1.0, the algorithms perform significantly better. That is, when the shape is smooth the algorithms are able to reconstruct the shape more effectively. It was again observed that the G.S.L. algorithm could recover shape much better than the V.A.N.L. also, the G.S.L. algorithm was less affected by surface shape than the V.A.N.L.

The shape-from-shading experiments were run on a SPARCstation 10. The surface normals and height map can be recovered in approximately five minutes using the G.S.L. algorithm and the degree-two height integration algorithm.

7.3 Region-Based Reflectance Estimation Experiments

The algorithm proposed in Section 5.1 is studied to determine its effectiveness in estimating the reflectance parameters for the Torrance-Sparrow reflectance model.

In the first set of experiments, an object's reflectance parameters are changed systematically. This experiment will give an indication whether the algorithm can estimate reflectance of diffuse as well as specular surfaces. Several shapes are used to study whether trends observed are consistent over different objects.

In the second experiment, the light source is moved systematically. The Region-Based Reflectance Estimation algorithm requires the illuminant direction as input information. It is interesting to study how the orientation of the light source effects the Region Based Reflectance Estimator. Various objects are used with different reflectance parameters to ensure that the results are consistent over different shapes and reflectance parameters.

7.3.1 Effect of Surface Reflectance on Region-Based Reflectance Estimator

The Region-Based Reflectance Estimation algorithm has been formulated to recover the reflectance properties of a surface. As the name suggests, the algorithm uses two regions of the surface to perform the estimation. The occluding contour is used to estimate K_d and the specular highlight is used to estimate σ.

The input images are the same ones used in the experiment to determine the behaviour of the shape-from-shading algorithms as the reflectance is varied. The input images can be found in Figures 7.31, 7.36, 7.42 and 7.47. The coefficients K_d and σ are varied from 0.100 to 0.895 in increments of 0.265. The light source direction (3,2,-9) is available to the algorithm, as well as the normals along the occluding contour. The specular highlight is assumed to be the region around the point of maximum intensity.

The results of the experiment are given in Figures 7.75 to 7.90. It is clear that the Region-Based Reflectance Estimation algorithm can recover the reflectance parameters that best fit the Torrance-Sparrow reflectance function. The absolute error in K_d varies between 0.004 to 0.02. The absolute error in σ varies between 0.0003 and 0.08. The average absolute error of K_d over all objects is 0.0086. The average absolute error of σ over all objects is 0.0167.

It is observed that when K_d is large, the error in estimating σ is large. That is, when the surface is very diffuse, it is more difficult to estimate the amount of specularity using the highlight. This trend is especially evident when considering the superellipsoid. When $K_d = 0.895$ the absolute error in the estimation of K_d is 0.0645. This is significantly larger than the average absolute error of 0.0167 in estimating σ. The difficulty of estimating σ when the surface is diffuse is probably due to the fact that the region used to perform the estimation is the specular highlight. When K_d is very large, there is no real specular highlight.

Another trend observed from the experiment is that as K_d increases, the error in estimating the reflectance parameters increases. This is again accounted for by noting that the specular highlight diminishes when the surface becomes increasingly diffuse.

It should be pointed out that the results of the Region-Based Reflectance Estimator are extremely consistent. The iterative nature of the formulation enables the algorithm to converge to a solution even with the initial assumption that the occluding contour region is diffuse. The average error in both K_d and σ over all objects is remarkably small (absolute error of $K_d = 0.0086$, absolute error of $\sigma = 0.0167$).

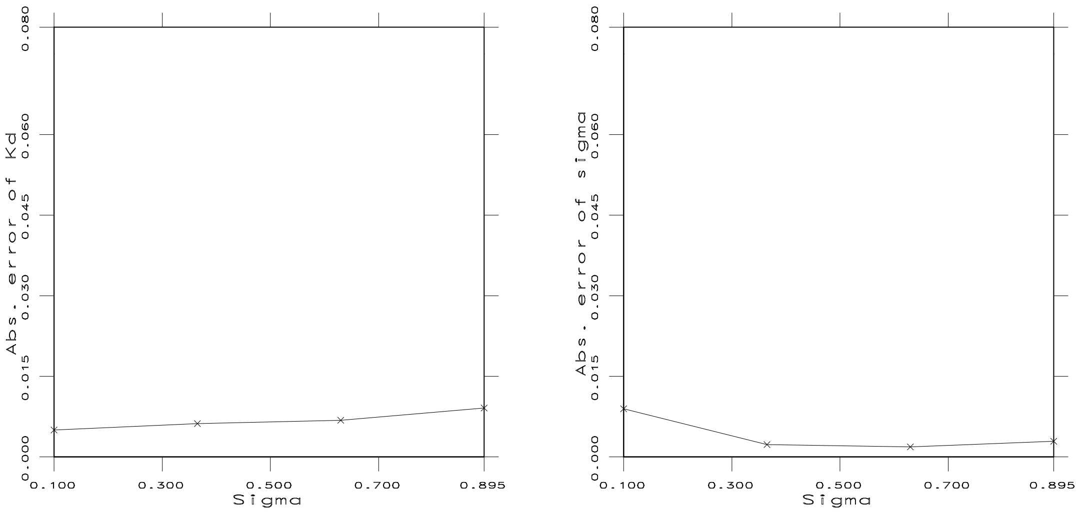


Figure 7.75: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.100$. Left: the absolute error in K_d, Right: the absolute error in σ.

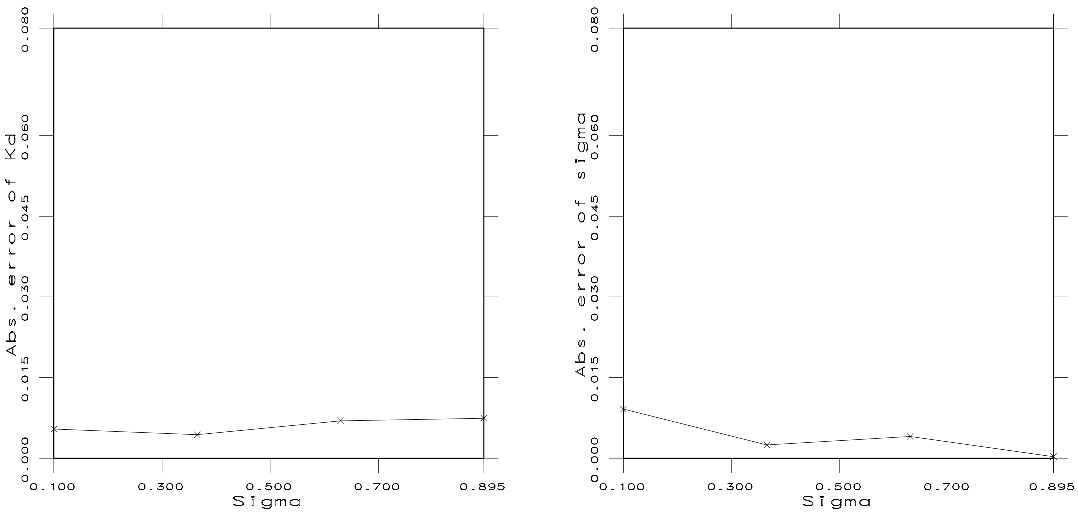


Figure 7.76: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.365$. Left: the absolute error in K_d, Right: the absolute error in σ.

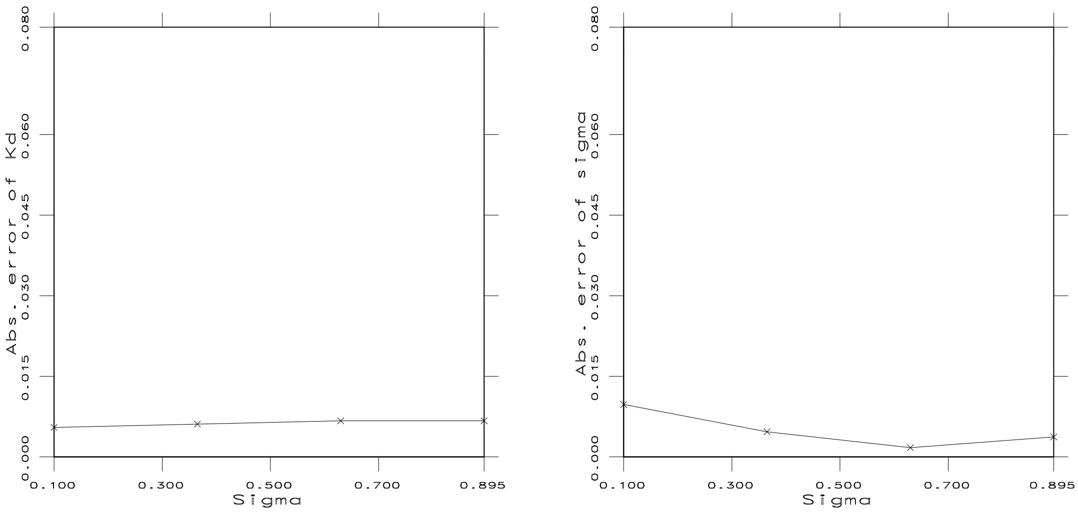


Figure 7.77: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.630$. Left: the absolute error in K_d, Right: the absolute error in σ.

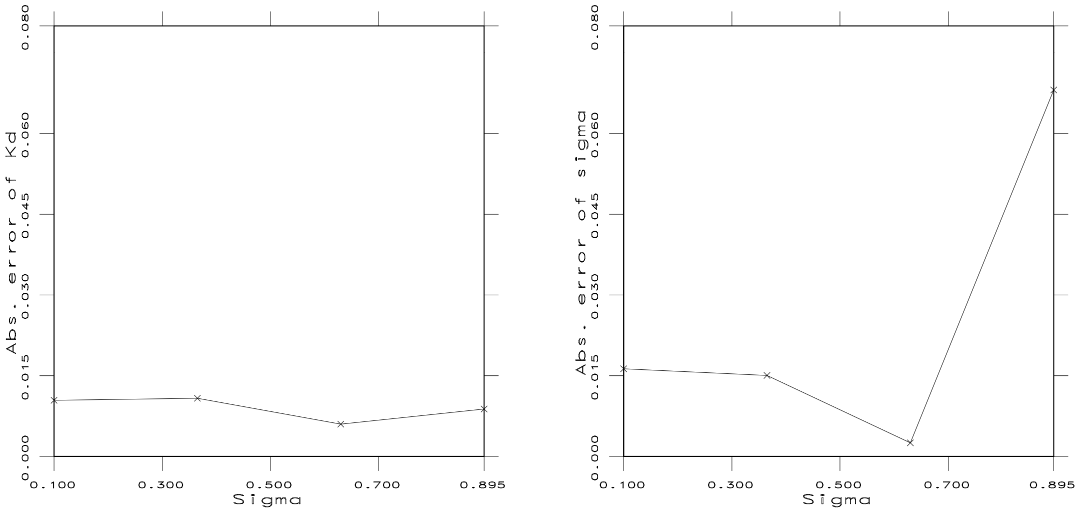


Figure 7.78: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.895$. Left: the absolute error in K_d, Right: the absolute error in σ.

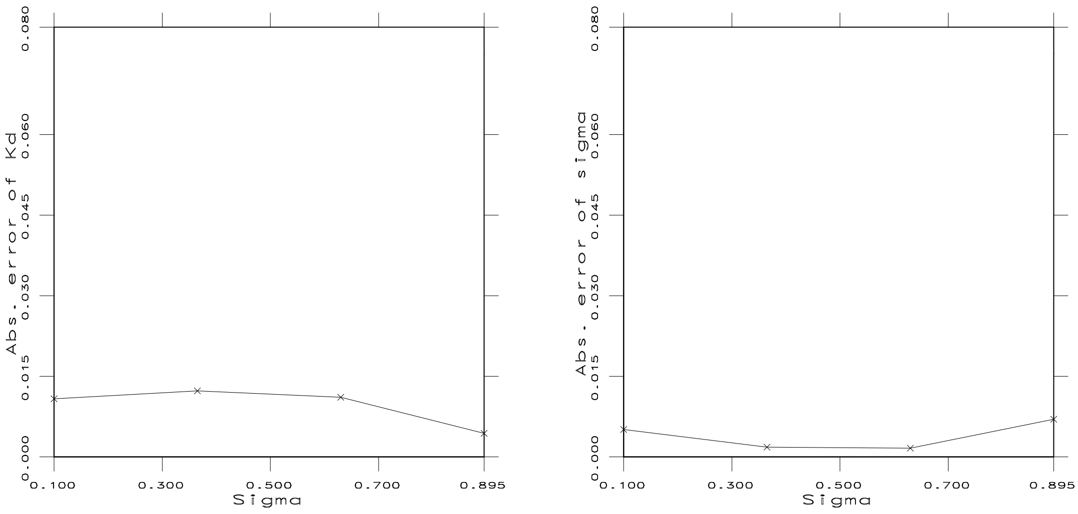


Figure 7.79: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.100$. Left: the absolute error in K_d, Right: the absolute error in σ.

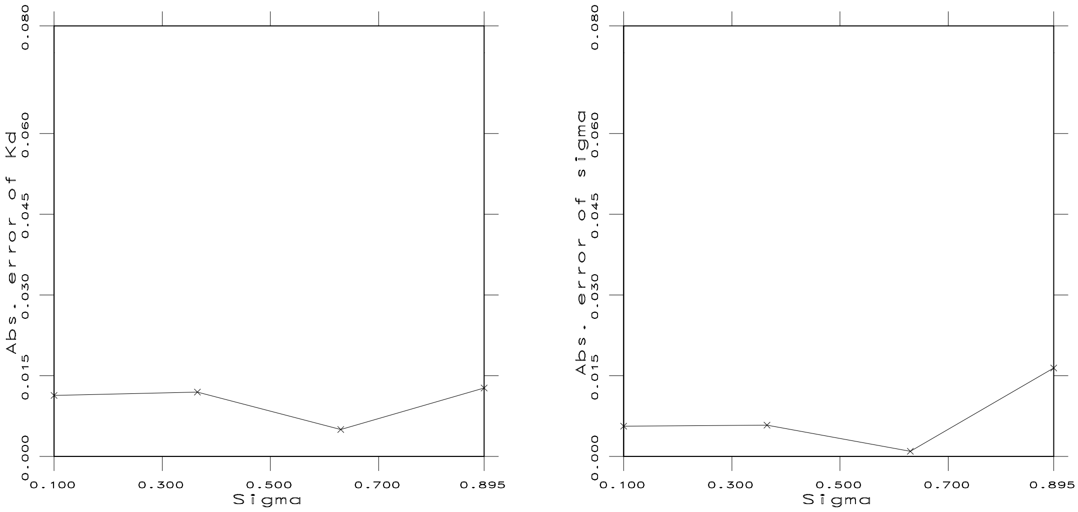


Figure 7.80: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.365$. Left: the absolute error in K_d, Right: the absolute error in σ.

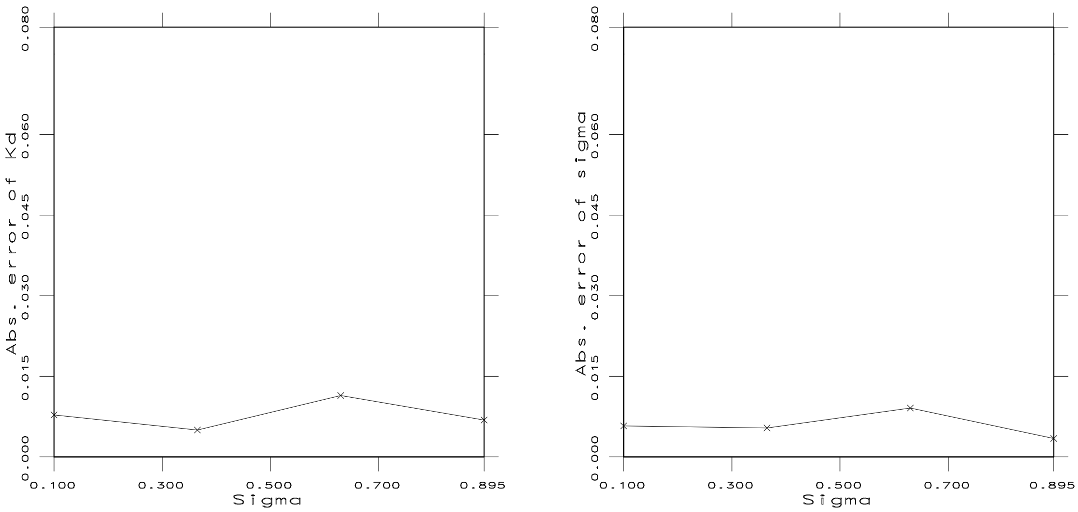


Figure 7.81: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.630$. Left: the absolute error in K_d, Right: the absolute error in σ.

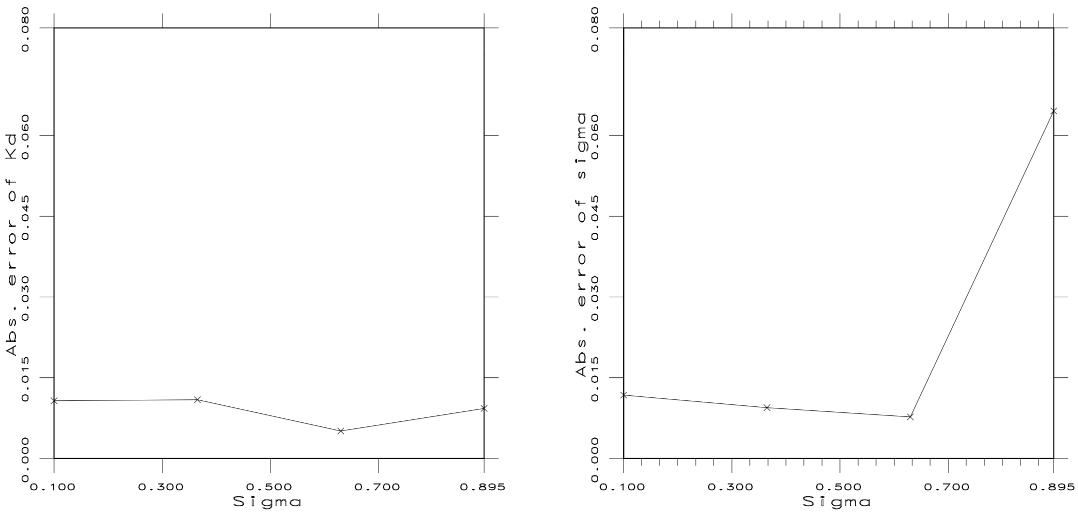


Figure 7.82: Results of estimating reflectance experiment when the input object is a surface of revolution with $e = 1.5$ and $K_d = 0.895$. Left: the absolute error in K_d, Right: the absolute error in σ.

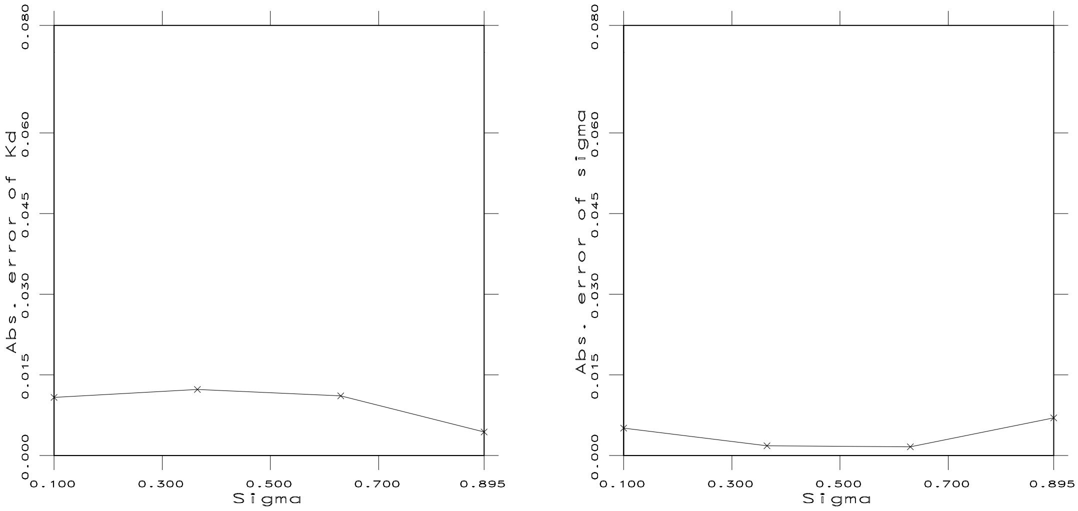


Figure 7.83: Results of estimating reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.100$. Left: the absolute error in K_d, Right: the absolute error in σ.

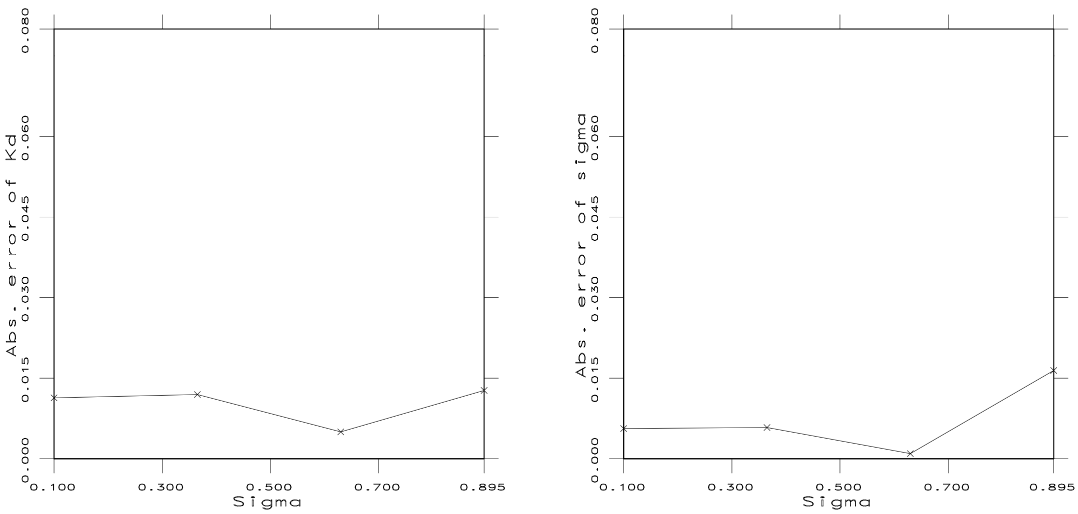


Figure 7.84: Results of estimating reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.365$. Left: the absolute error in K_d, Right: the absolute error in σ.

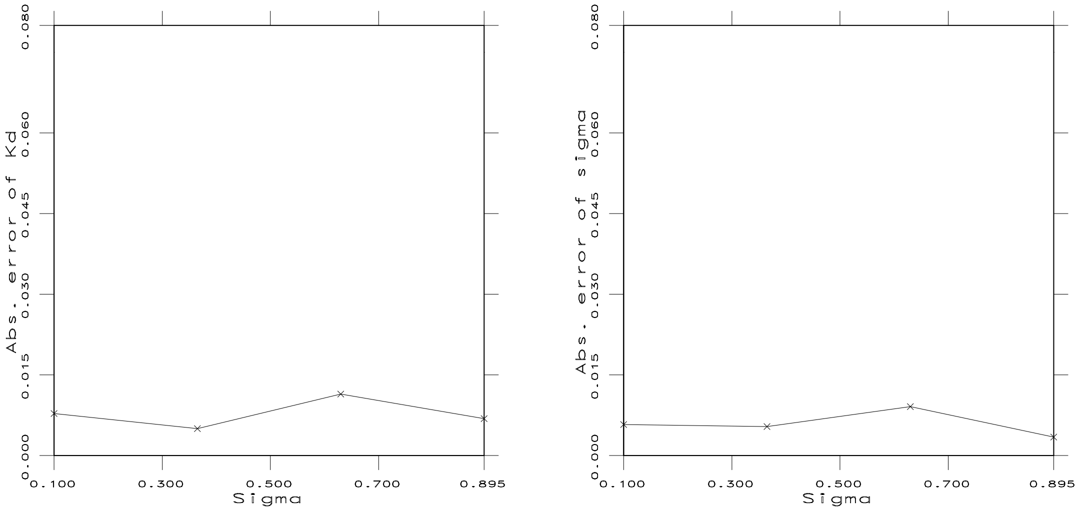


Figure 7.85: Results of estimating reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.630$. Left: the absolute error in K_d, Right: the absolute error in σ.

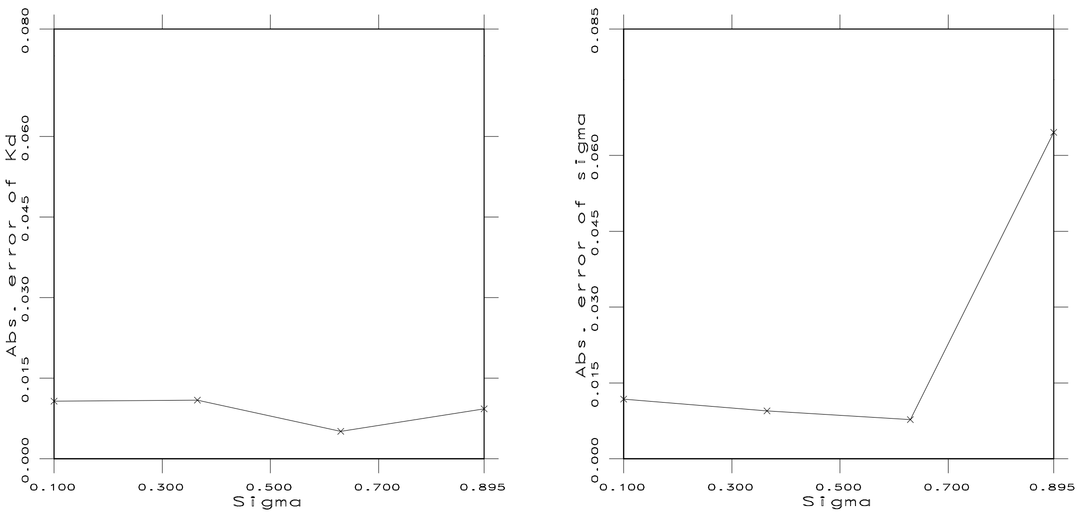


Figure 7.86: Results of estimating reflectance experiment when the input object is a superellipsoid with $e = 1.5$ and $K_d = 0.895$. Left: the absolute error in K_d, Right: the absolute error in σ.

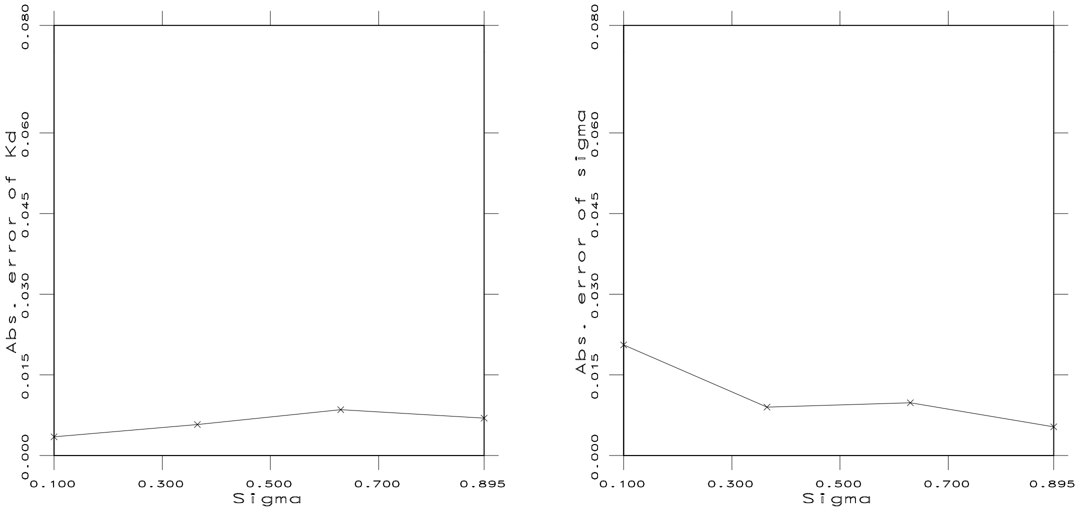


Figure 7.87: Results of estimating reflectance experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.100$. Left: the absolute error in K_d, Right: the absolute error in σ.

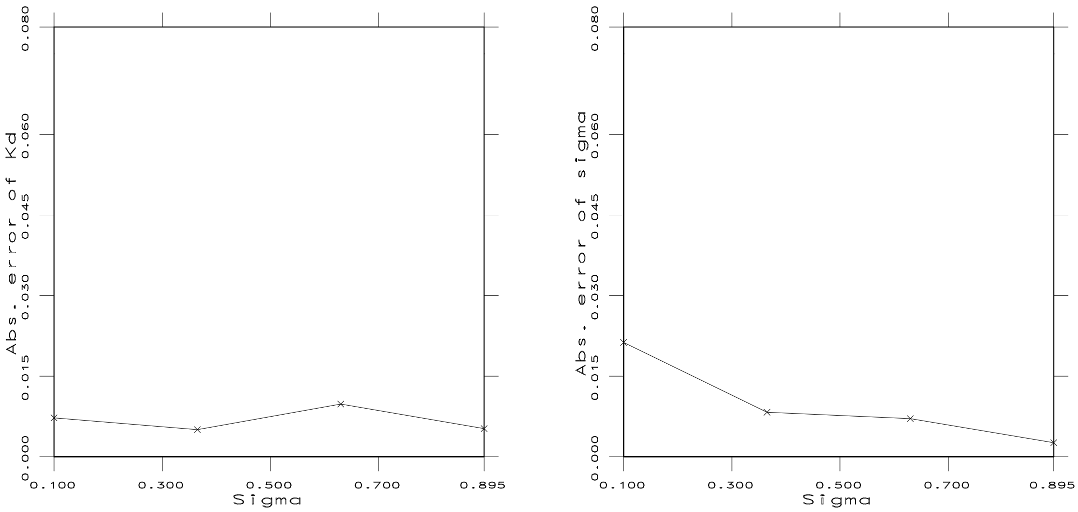


Figure 7.88: Results of estimating reflectance experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.365$. Left: the absolute error in K_d, Right: the absolute error in σ.

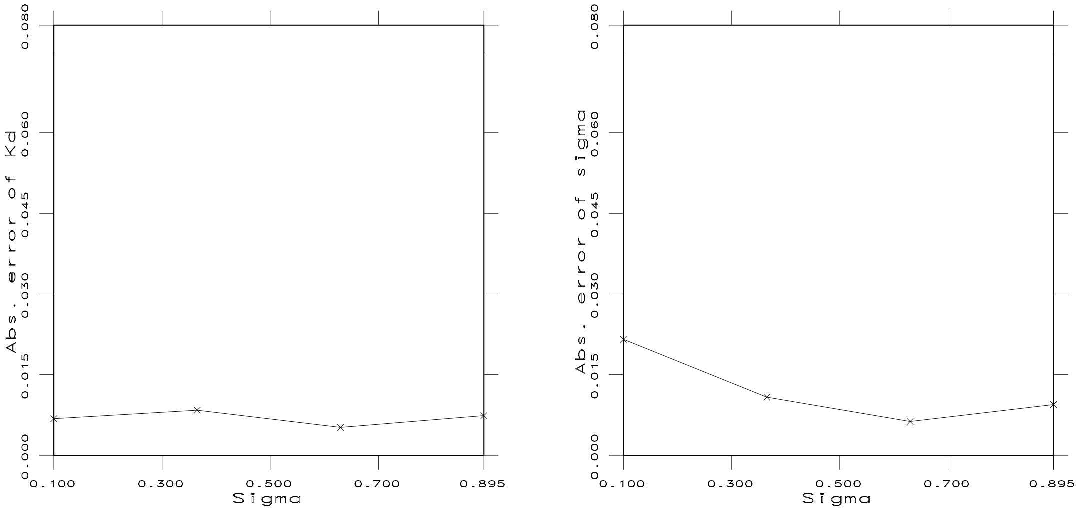


Figure 7.89: Results of estimating reflectance experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.630$. Left: the absolute error in K_d, Right: the absolute error in σ.

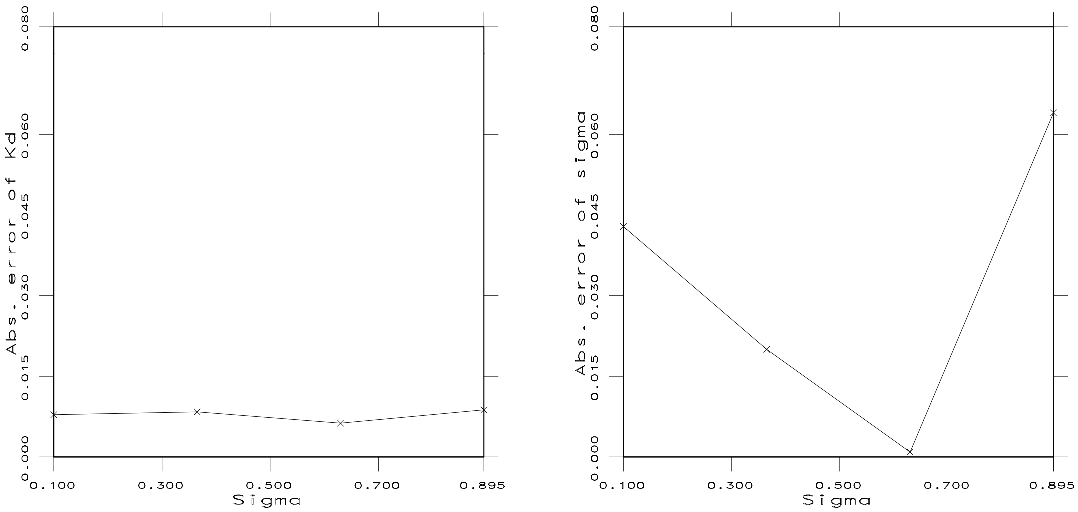


Figure 7.90: Results of estimating reflectance experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.895$. Left: the absolute error in K_d, Right: the absolute error in σ.

7.3.2 Effect of Illuminant Direction on Region-Based Reflectance Estimator

The specular highlight is located in the region that reflects light directly from the point source to the viewer. The specular highlight is very much dependent on the illuminant direction. One of the areas that the Region-Based Reflectance Estimator uses is the specular highlight. The other area used is the occluding contour. It is initially assumed that the region along the occluding contour is diffuse. This assumption is true depending on the illuminant direction. Because these regions (specular highlight and occluding contour) are so influenced by the light source direction, it is important to understand the behaviour of the Region-Based Reflectance Estimator as the light source is moved.

In this set of experiments the light source is varied from 0 to 45 degrees in 15 degree increments. The absolute error of σ and the absolute error of K_d is plotted versus the slant. The input images are given in Figures 7.52, 7.54, 7.58, and 7.60. The results of the experiment are given in Figures 7.91, 7.92, 7.93, and 7.94.

The light source orientation has little effect on the Region-Based Reflectance Estimator, except when the object is diffuse ($K_d = 0.7$ and $\sigma = 0.7$). When the object's surface is diffuse, a specular highlight may not be present. When the light source has a large slant, the estimate of σ produces an error of 0.008 when considering the surface of revolution with $e = 0.5$. However, the range of absolute error for the estimate of K_d is between 0.0008 and 0.0042. The absolute error in estimating σ is between 0.00005 and 0.013. The average absolute error in estimating K_d is only 0.00295 while the average absolute error in estimating σ over all objects is 0.00453.

It is also observed that object shape has little effect on the performance of the Region-Based Reflectance Estimator. The average absolute error in estimating K_d for the surface of revolution is 0.00293. The average absolute error in estimating K_d using the superellipsoid as the input object is 0.00289.

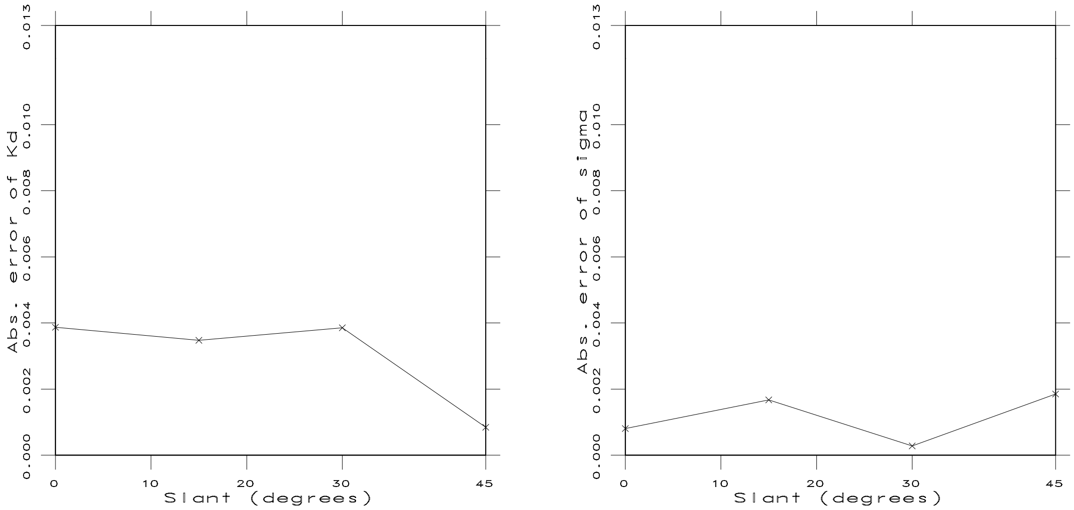


Figure 7.91: Results of estimating reflectance experiment when the illuminant direction is varied. The input object is a surface of revolution with $e = 0.5$ and $K_d = 0.3$, $\sigma = 0.3$. Left: absolute error in K_d, Right: absolute error in σ.

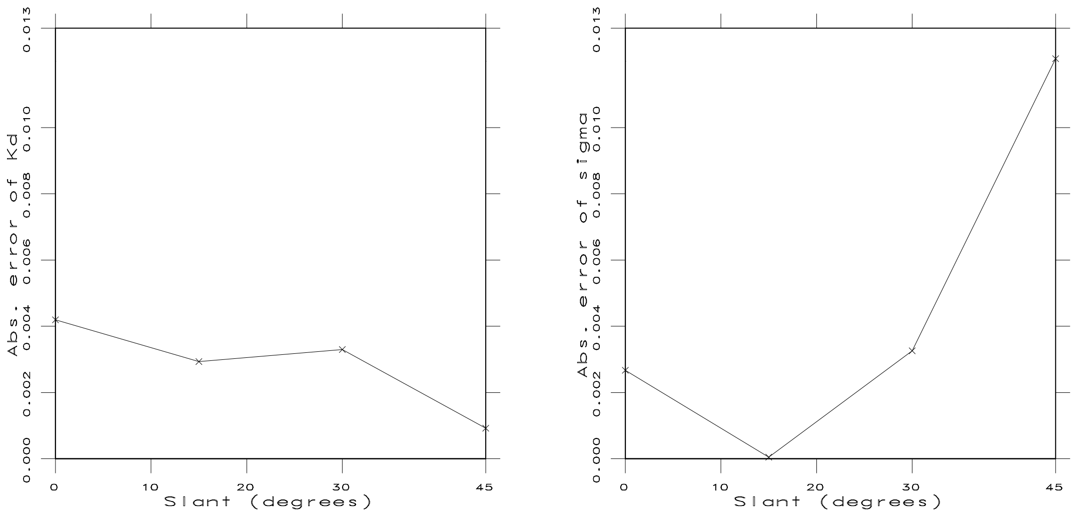


Figure 7.92: Results of estimating reflectance experiment when the illuminant direction is varied. The input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7$, $\sigma = 0.7$. Left: absolute error in K_d, Right: absolute error in σ.

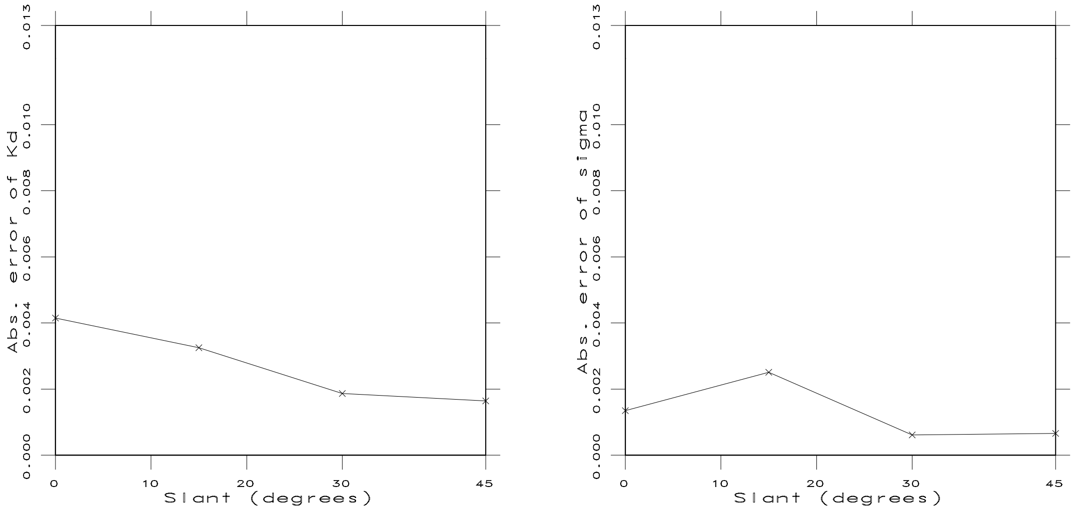


Figure 7.93: Results of estimating reflectance experiment when the illuminant direction is varied. The input object is a superellipsoid with $e = 1.5$ and $K_d = 0.3$, $\sigma = 0.3$. Left: absolute error in K_d, Right: absolute error in σ.

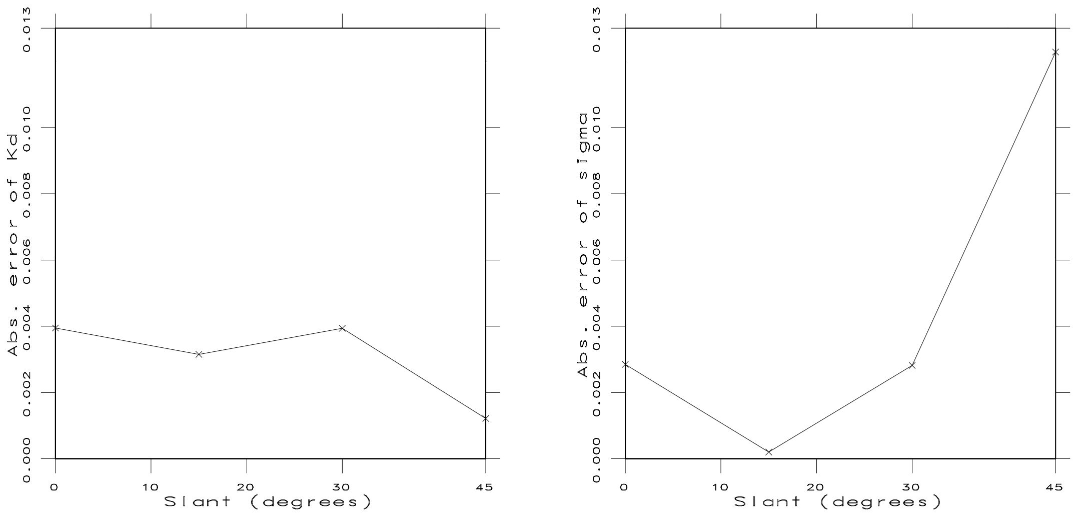


Figure 7.94: Results of estimating reflectance experiment when the illuminant direction is varied. The input object is a superellipsoid with $e = 1.5$ and $K_d = 0.7$, $\sigma = 0.7$. Left: absolute error in K_d, Right: absolute error in σ.

7.3.3 Summary of Region-Based Reflectance Estimator Experiments

In Subsection 7.3.1 the surface reflectance properties are varied to study the Region-Based Reflectance Estimator's performance. It is observed that when the surface is diffuse, the error in estimating σ increases. This is because the specular highlight becomes less and less discernible as the surface becomes more and more diffuse. However, the algorithm is able to achieve very small errors for almost all combinations of reflectance parameters.

In Subsection 7.3.2 the illuminant direction is varied to study the behaviour of the Region-Based Reflectance Estimator's performance. It was found that the orientation of the light source had very little effect on the algorithm's ability to estimate reflectance. It was also observed that the Region-Based Reflectance Estimator could recover reflectance parameters independent of surface shape.

7.4 Experiments using Contour-Based Illuminant Locator and a Calibration Object

The Contour-Based Illuminator estimates the light source direction given the reflectance parameters. The occluding contour is used because the surface normals are known in this region. The light source slant and tilt are varied incrementally to determine which reflectance map can generate brightness values equal to the intensities along the occluding contour given the normals on the contour. The algorithm proposed to estimate the total reflectance map given a calibration object is very similar to the Contour-Based Illuminant Locator. Both the surface normals and orientations are known at each point on the object, but the reflectance map is to be determined. The light source and reflectance parameters are all varied at each point to determine which set of parameters "fit" the intensity data. At each point, the reflectance maps (consisting of both reflectance parameters and illuminant direction) that can generate a brightness equal to the given pixel intensity are given a vote. After all the points are considered, the reflectance map with the most votes is the estimated reflectance map.

Experiments were conducted to determine the effectiveness of these algorithms. It was found that the algorithms performed very well if the choice in increments was sufficiently small. That is, if the light source and reflectance parameters were incremented in sufficiently small amounts, the algorithms could find the reflectance map. It was observed that the algorithms could locate the reflectance map independent of surface shape, surface reflectance, and illuminant direction if the increments were small enough. Thus, only the experiments to study the effect of the algorithms as the increment of the light source is varied are presented.

In this set of experiments, the increment of the light source tilt and slant is varied. The input images are given in Figure 7.10. For the Contour-Based Illuminant Locator, the normals along the occluding contour as well as the reflectance parameters are available to the algorithm. For the algorithm to estimate the reflectance map given a calibration object, the surface normals at each point are known and the light source and reflectance parameters are estimated. The pixel intensities at each point are available to both algorithms.

The results of the experiments are given in Figures 7.95 to 7.100. The absolute error in K_d and σ is plotted for the algorithm to estimate the reflectance map with a calibration object. The Contour-Based Illuminant Locator is used when the reflectance parameters are known and the illuminant direction is to be determined. The angular error between the estimated light source direction and the actual illuminant direction is plotted for both the Contour-Based Illuminant Locator and the algorithm to estimate the reflectance map given a calibration object. The Contour-Based Illuminant Locator is labeled *C.B.I.L.* while the algorithm that uses a calibration object is simply labeled *Calib.*. The increment in the slant and tilt of the light source is varied from 4 degrees to 40 degrees in steps of 4 degrees. The light source increment (labeled L.S. Increment) is along the $x-$axis of the graphs, along the $y-$axis is the different error measures. For the calibration object algorithm, the increment of the reflectance parameters is set at 0.02.

It is observed that the algorithms perform very well when the increment is small (4 or 8 degrees). When the light source increment angle is large, the error of the algorithms becomes erratic. The Region-Based Illuminant Locator is able to recover the illuminant direction regardless of object shape and surface reflectance if the increment of the light source direction is small. In fact, the angular error between the actual light source and the estimated light source is always less than 2 degrees for light source increments of 4 and 8 degrees. The algorithm to estimate the reflectance map given a calibration object can recover the reflectance parameters and illuminant direction if the increment of the light source is sufficiently small. Again, if the light source increment is 4 or 8 degrees, the error in estimating both the illuminant direction and the reflectance parameters is close to zero.

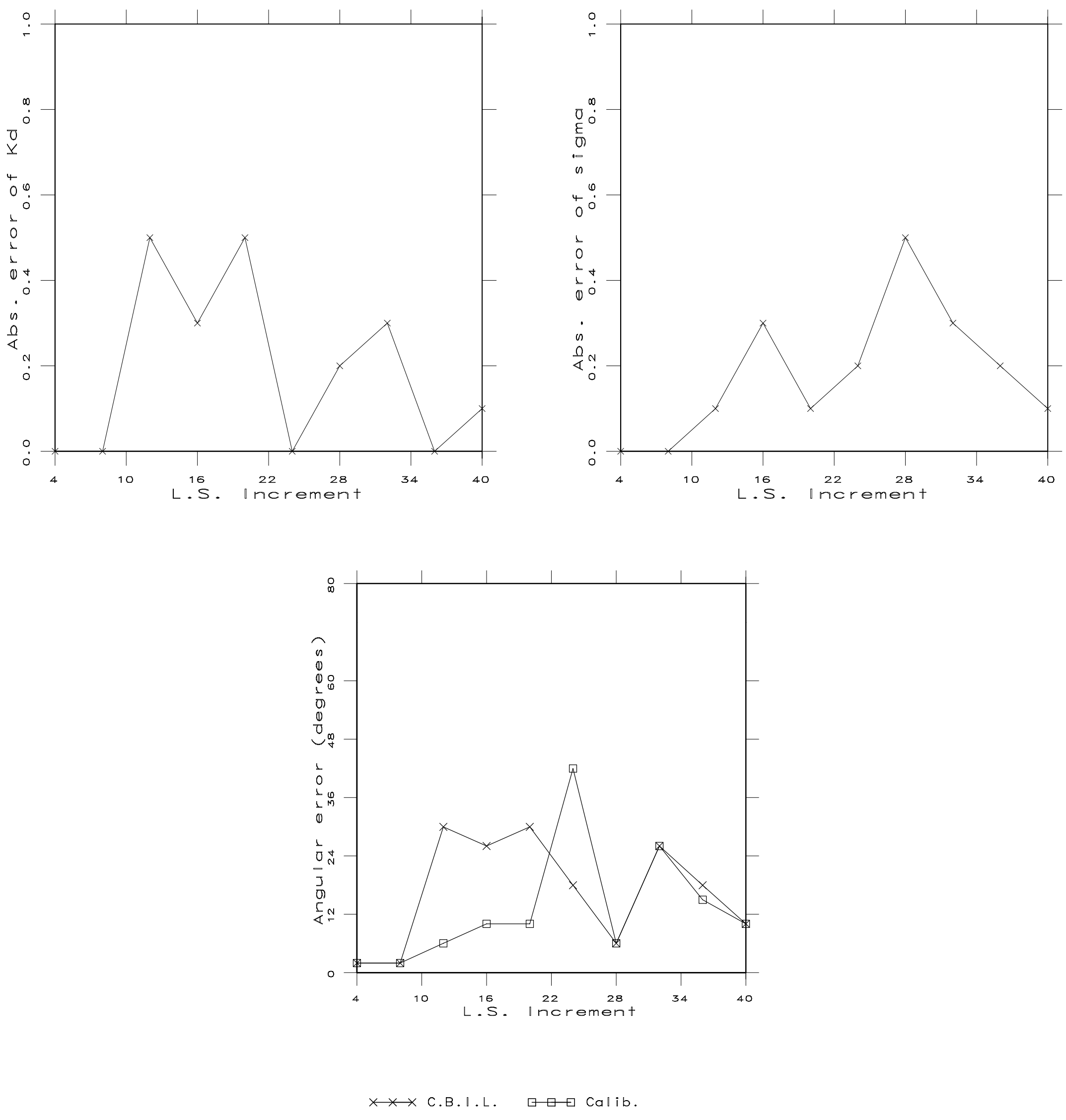


Figure 7.95: Results of the Contour-Based Illuminant Locator and calibration object experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.3$, $\sigma = 0.3$. Top left: the absolute error in K_d, Top right: the absolute error is σ, Bottom: the angular error between estimated illuminant and real illuminant direction.

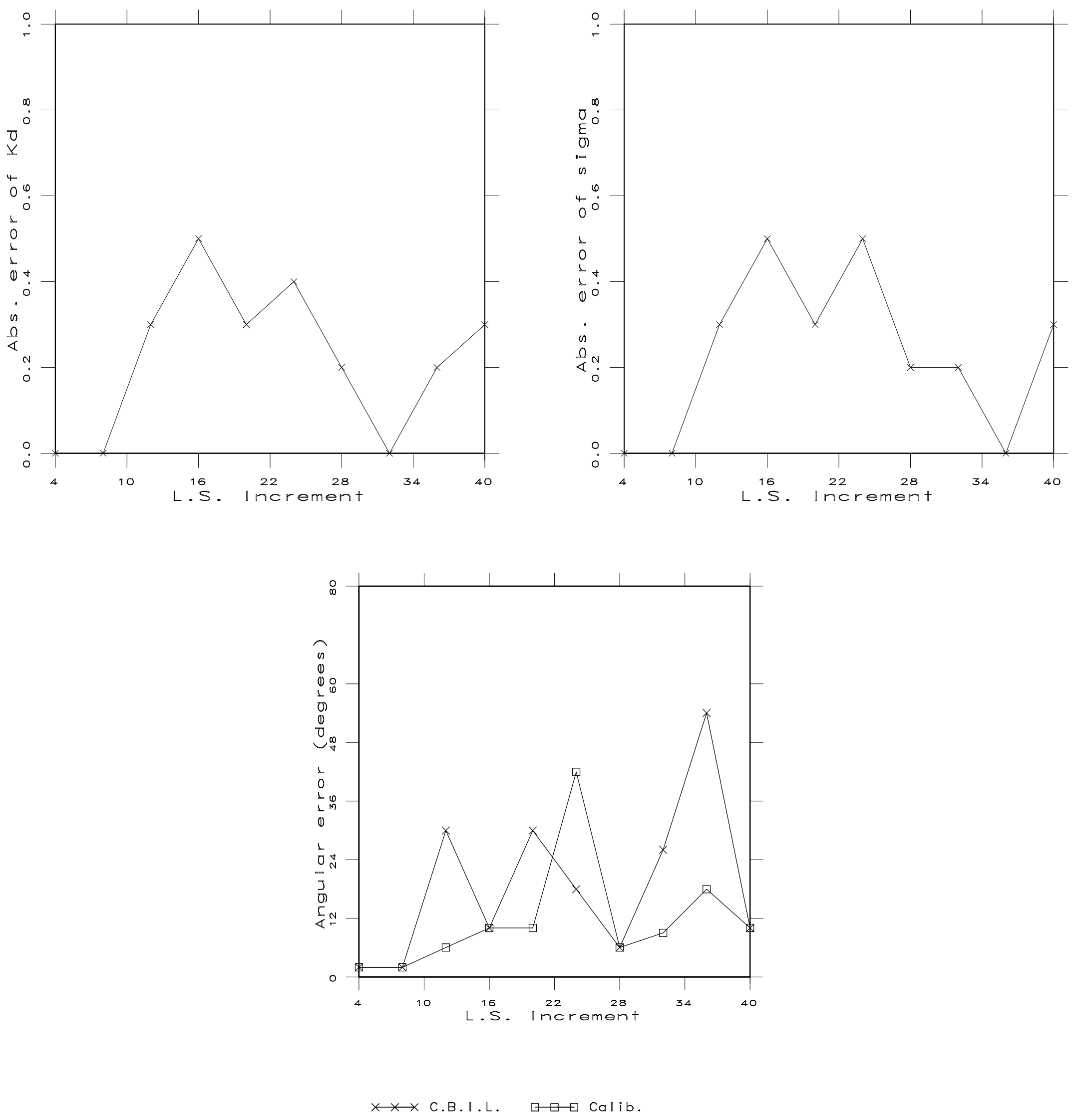


Figure 7.96: Results of the Contour-Based Illuminant Locator and calibration object experiment when the input object is a surface of revolution with $e = 0.5$ and $K_d = 0.7$, $\sigma = 0.7$. Top left: the absolute error in K_d, Top right: the absolute error is σ, Bottom: the angular error between estimated illuminant and real illuminant direction.

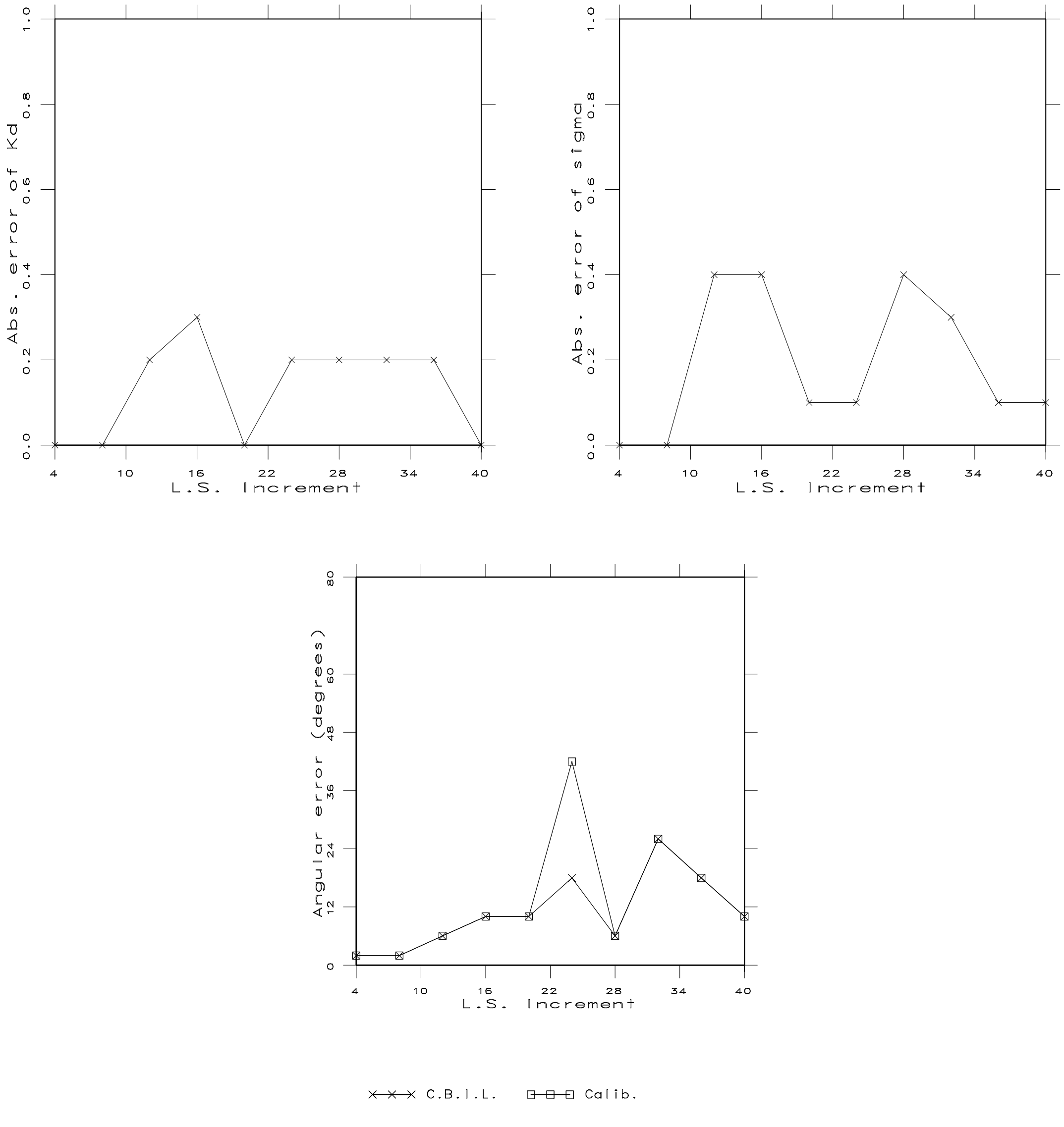


Figure 7.97: Results of the Contour-Based Illuminant Locator and calibration object experiment when the input object is a superellipsoid with $e = 0.5$ and $K_d = 0.3$, $\sigma = 0.3$. Top left: the absolute error in K_d, Top right: the absolute error is σ, Bottom: the angular error between estimated illuminant and real illuminant direction.

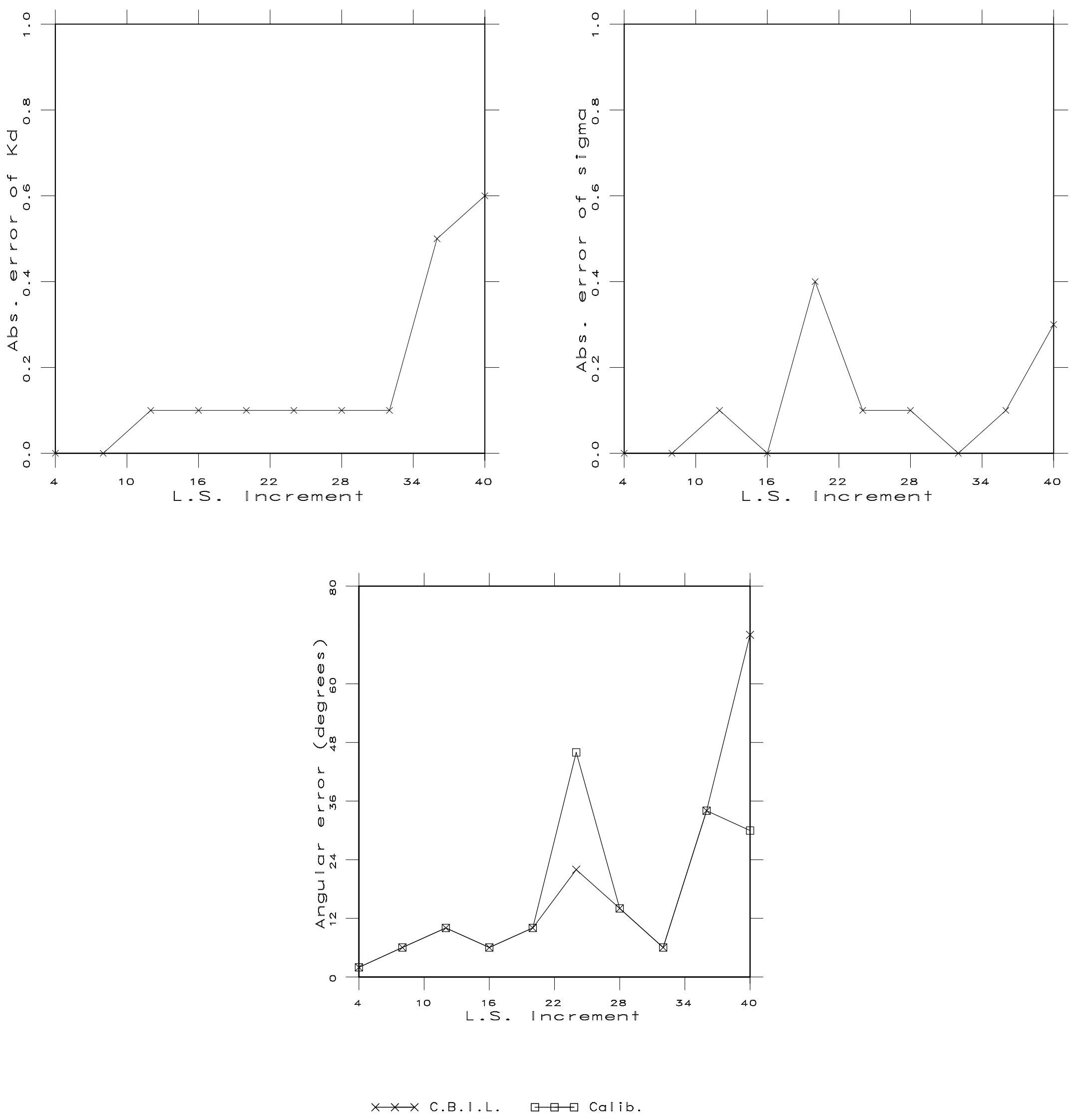


Figure 7.98: Results of the Contour-Based Illuminant Locator and calibration object experiment when the input object is a superellipsoid with $e = 0.5$ and $K_d = 0.7$, $\sigma = 0.7$. Top left: the absolute error in K_d, Top right: the absolute error is σ, Bottom: the angular error between estimated illuminant and real illuminant direction.

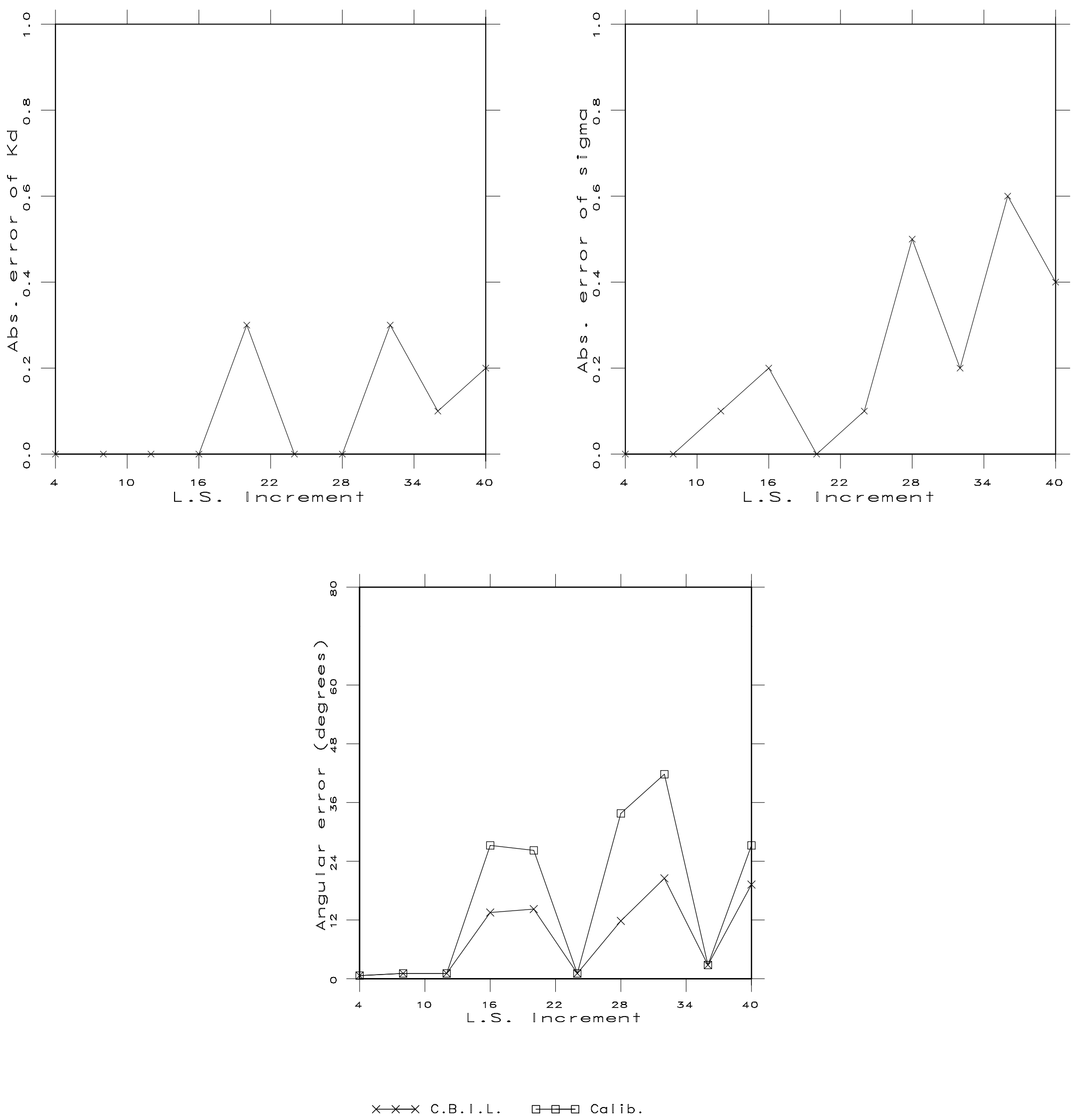


Figure 7.99: Results of the Contour-Based Illuminant Locator and calibration object experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.3$, $\sigma = 0.3$. Top left: the absolute error in K_d, Top right: the absolute error is σ, Bottom: the angular error between estimated illuminant and real illuminant direction.

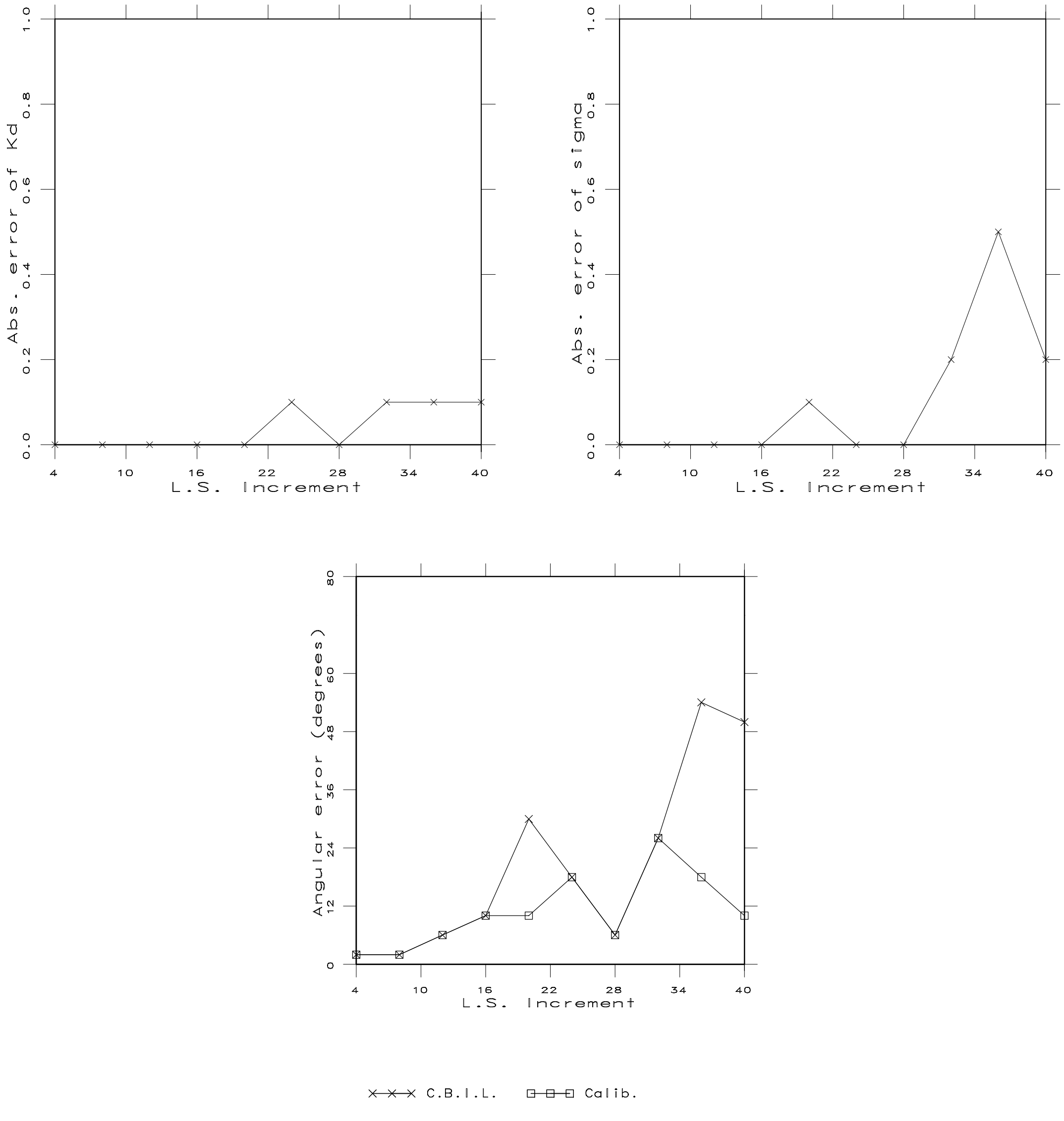


Figure 7.100: Results of the Contour-Based Illuminant Locator and calibration object experiment when the input object is a supertoroid with $e = 1.0$ and $K_d = 0.7$, $\sigma = 0.7$. Top left: the absolute error in K_d, Top right: the absolute error is σ, Bottom: the angular error between estimated illuminant and real illuminant direction.

7.5 Experiments Using Real Images

In this section, experiments are performed on real images. It is beyond the scope of this thesis to study the behaviour of the shape-from-shading algorithms as the reflectance properties of a real object's surface are varied. Rather, the purpose of these experiments is to demonstrate that the algorithms presented in this thesis are effective in coping with not only synthetic images but also real images.

The experimental set-up is described in Figure 7.101.

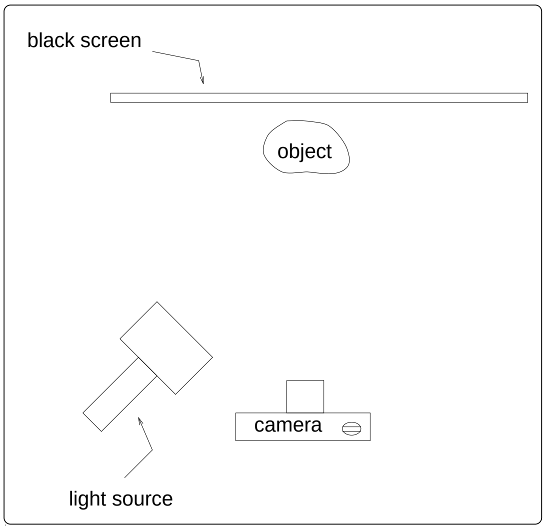


Figure 7.101: Experimental set-up used to capture the real images.

Several objects were used with a wide variety of shapes and reflectance properties. The real images used in this experiment are given in Figures 7.102 and 7.103. It can be visually verified that the shapes of the input images vary considerably and that the reflectance parameters of the input images are also quite different. A video camera connected to a frame buffer was used to capture the images. All input images were 512×512 but reduced to 256×256 by cropping the original image and bilinear interpolation. All of the images were taken with the same illuminant direction and viewing orientation. The input images were thresholded to differentiate between the object and the background region. An edge detector was used to determine the occluding contour and the surface normals along the contour were determined using the slope of the occluding curve.

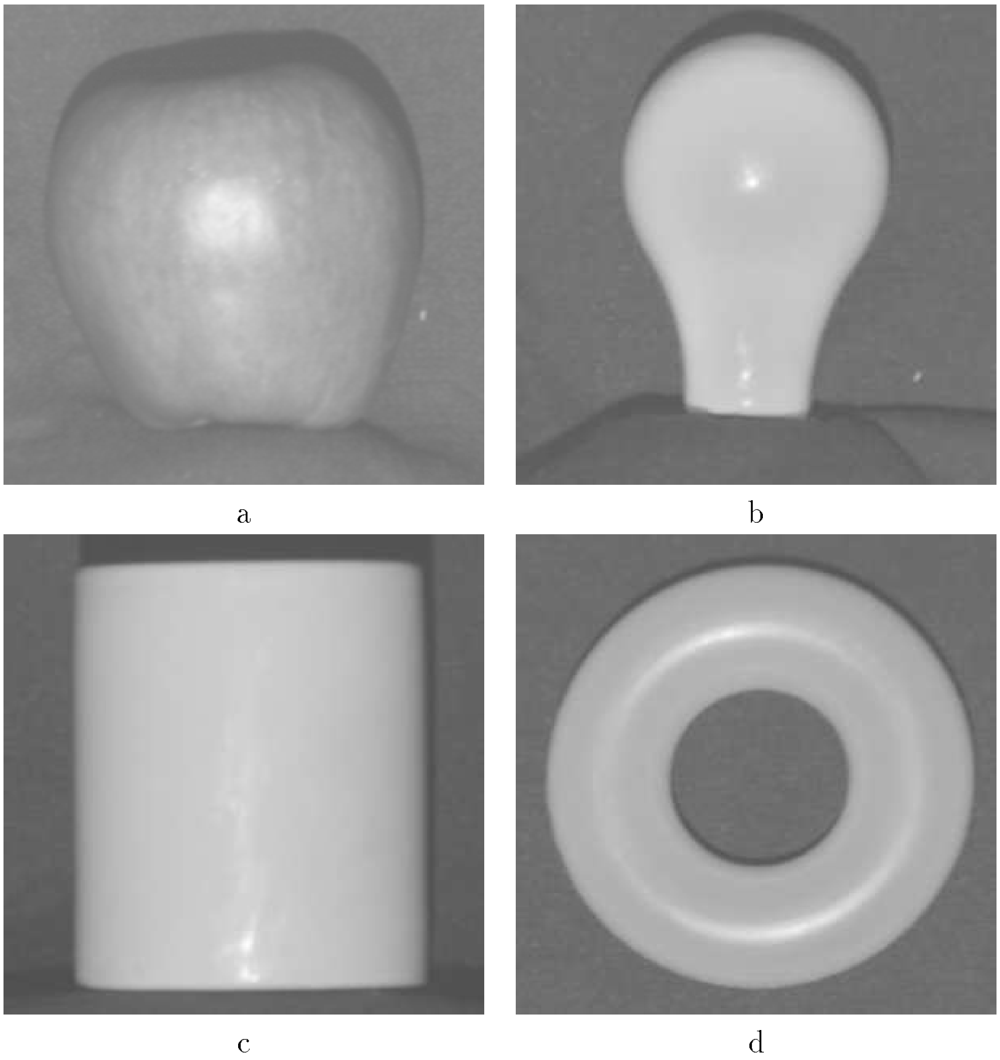


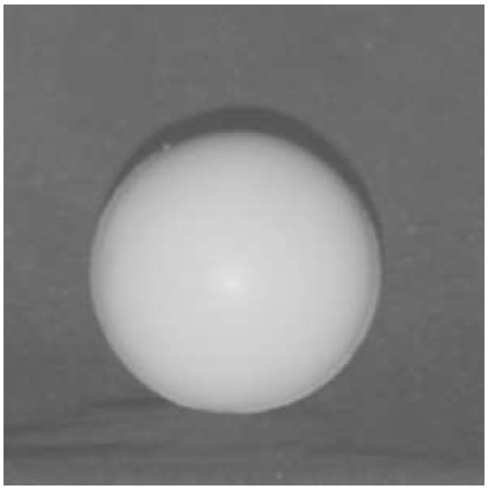
Figure 7.102: Real images used in experiment. (a) apple, (b) light bulb, (c) coffee mug, (d) plastic toroid

In real world situations the light source direction and object reflectance parameters are not known. This is also true in this experimental set-up. It is difficult to measure the exact location of the light source or the reflectance properties of the surface. Because there is no known algorithm that can recover this information given a single image, several new algorithms have been proposed in this thesis that can facilitate the recovery of shape from real images.

The LIRAS Recovery System proposed in this thesis allows for the shape recovery of each object in a scene given only one piece of information. These input images can be considered to be in the same scene, as the objects were all taken with the light source and camera view fixed. Using the LIRAS Recovery System, a calibration object in the scene can be used to recover the light source direction (as well as the reflectance properties of the object, although this information is not needed here). Once the illuminant direction has been found, the reflectance parameters of each object in the scene can be estimated using the Region-Based Reflectance Estimator (proposed in Section 5.1). When the reflectance map has been recovered (reflectance parameters and illuminant direction) the algorithms formulated in Chapter 4 can be used to recover the shape from the shading information. In other words, although no algorithm has been proposed to recover the reflectance map given a single image, if a calibration object is available, the light source direction can be estimated. Because the light source direction will not change for all objects in the same scene, the reflectance parameters of the various objects can be estimated. Finally, the shape-from-shading algorithms for non-Lambertian surfaces can be employed to recover shape. A calibration object *bootstraps* the whole process.

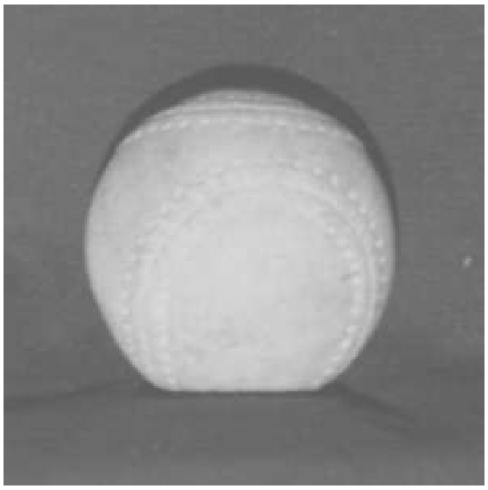
In this experiment, the specular sphere (Figure 7.102 e) is used as the calibration object. The sphere is a good choice for a calibration object as it is easily identifiable using the contour. The light source was estimated at (0.034899 0.000000 -0.999391) and the reflectance parameters were estimated as being ($K_d = 0.398949, \sigma = 0.630965$). Now that the illuminant direction has been determined using the calibration object, the reflectance parameters of the remaining objects can be determined using the Region-Based Reflectance Estimator. The reflectance parameters of the objects are summarized in Figure 7.104. The V.A.N.L. and the G.S.L. algorithm are both used to reconstruct the shape of the objects. Once the surface normals and reflectance parameters have been estimated, an image can be synthesized and compared to the input image. The average intensity error is calculated and used to evaluate the algorithms. These errors are also given in Figure 7.104. For each object, the thresholded input image, calculated occluding contour, resynthesized image and mesh diagram of the reconstructed object are given in Figures 7.105 to 7.111.

The A.I.E. for the G.S.L. algorithm is lower than the A.I.E. for the V.A.N.L. The A.I.E. over all objects for the G.S.L. is only 3.611957, while the A.I.E. for the V.A.N.L. is 5.718210. It can also be visually confirmed that the resynthesized images resemble the original image. From the three-dimensional mesh diagrams, it is observed that both algorithms can recover the shape of object using real images. The height maps are qualitatively confirmed to be accurate representations of the object shapes.



e

f

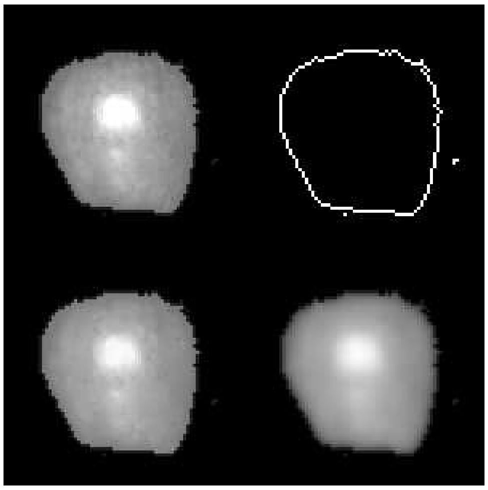


g

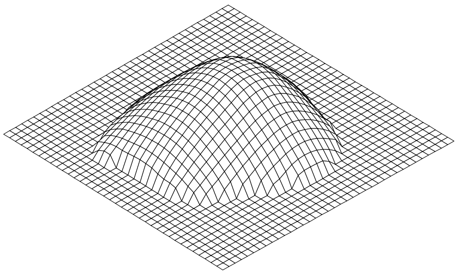
Figure 7.103: Real images used in experiment. (e) plastic ball, (f) new baseball, (g) old baseball

Image	*estimated* K_d	*estimated* σ	A.I.E. using V.A.N.L.	A.I.I. using G.S.L.
a	0.413489	0.582818	3.581689	2.094044
b	0.450786	0.492609	5.428901	3.500812
c	0.460959	0.482091	7.064533	4.947911
d	0.420101	0.556688	6.404081	2.499778
e	0.398949	0.630965	3.187378	1.267022
f	0.396200	0.643209	5.703644	2.357867
g	0.403841	0.783362	8.657245	8.616267

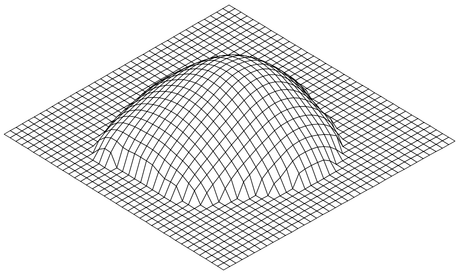
Figure 7.104: Errors calculated using real images.



a

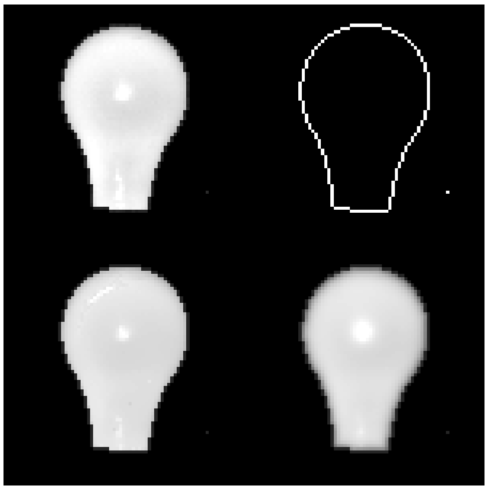


b

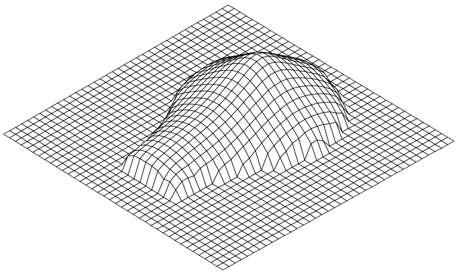


c

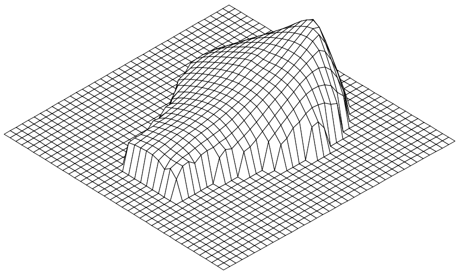
Figure 7.105: (a) Reconstructing the shape given the image of the apple. Top-Left: the original image after thresholding, Top-Right: the occluding contour located by edge detection, Bottom-Left: the resynthesized image using the G.S.L. algorithm, Bottom-Right: the resynthesized image using the V.A.N.L. (b) A mesh diagram showing the reconstructed heights using the G.S.L. algorithm (c) A mesh diagram showing the reconstructed heights using the V.A.N.L.



a

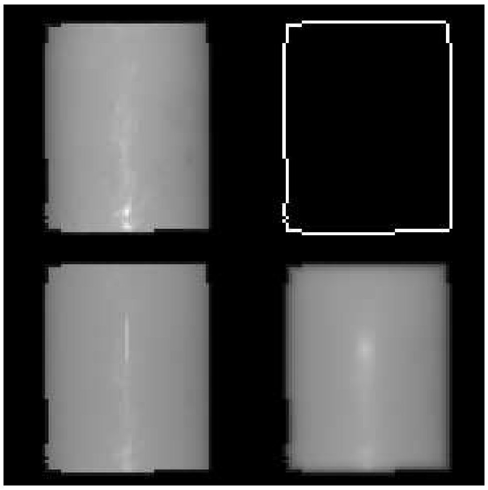


b

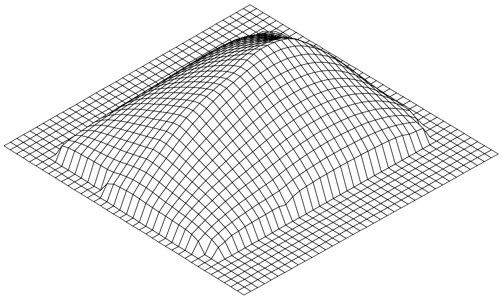


c

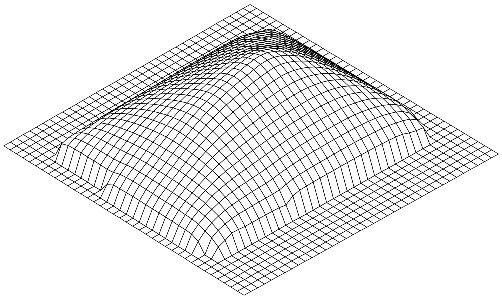
Figure 7.106: (a) Reconstructing the shape given the image of the light. Top-Left: the original image after thresholding, Top-Right: the occluding contour located by edge detection, Bottom-Left: resynthesized image using the G.S.L. algorithm, Bottom-Right: resynthesized image using the V.A.N.L. (b) A mesh diagram showing the reconstructed heights using the G.S.L. algorithm (c) A mesh diagram showing the reconstructed heights using the V.A.N.L.



a

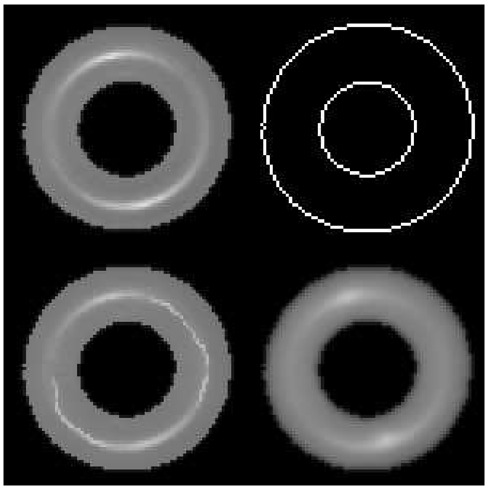


b

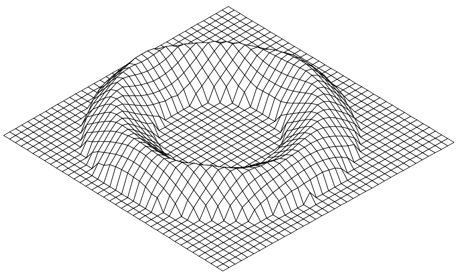


c

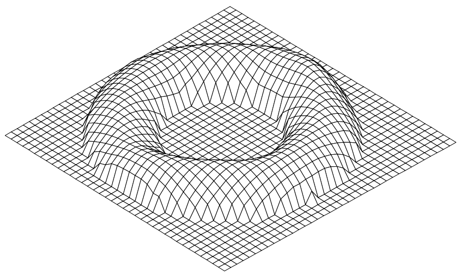
Figure 7.107: (a) Reconstructing the shape given the image of the mug. Top-Left: the original image after thresholding, Top-Right: the occluding contour located by edge detection, Bottom-Left: resynthesized image using the G.S.L. algorithm, Bottom-Right: resynthesized image using the V.A.N.L. (b) A mesh diagram showing the reconstructed heights using the G.S.L. algorithm (c) A mesh diagram showing the reconstructed heights using the V.A.N.L.



a

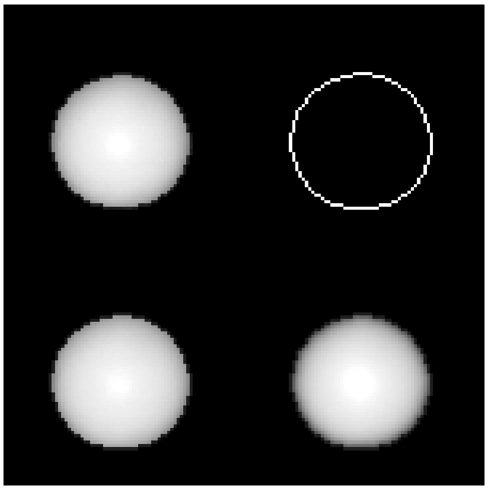


b

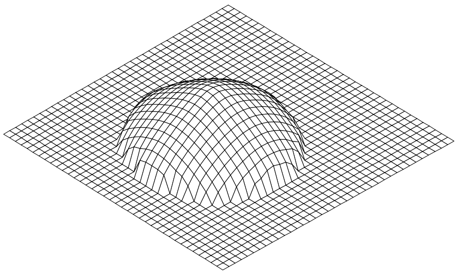


c

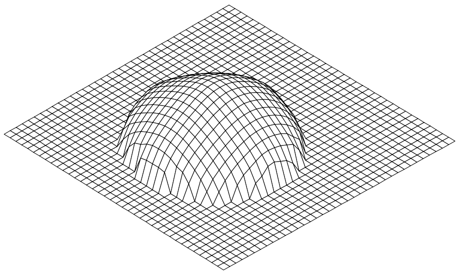
Figure 7.108: (a) Reconstructing the shape given the image of the plastic toroid. Top-Left: the original image after thresholding, Top-Right: the occluding contour located by edge detection, Bottom-Left: resynthesized image using the G.S.L. algorithm, Bottom-Right: resynthesized image using the V.A.N.L. (b) A mesh diagram showing the reconstructed heights using the G.S.L. algorithm (c) A mesh diagram showing the reconstructed heights using the V.A.N.L.



a



b



c

Figure 7.109: (a) Reconstructing the shape given the image of the plastic ball. Top-Left: the original image after thresholding, Top-Right: the occluding contour located by edge detection, Bottom-Left: resynthesized image using the G.S.L. algorithm, Bottom-Right: resynthesized image using the V.A.N.L. (b) A mesh diagram showing the reconstructed heights using the G.S.L. algorithm (c) A mesh diagram showing the reconstructed heights using the V.A.N.L.

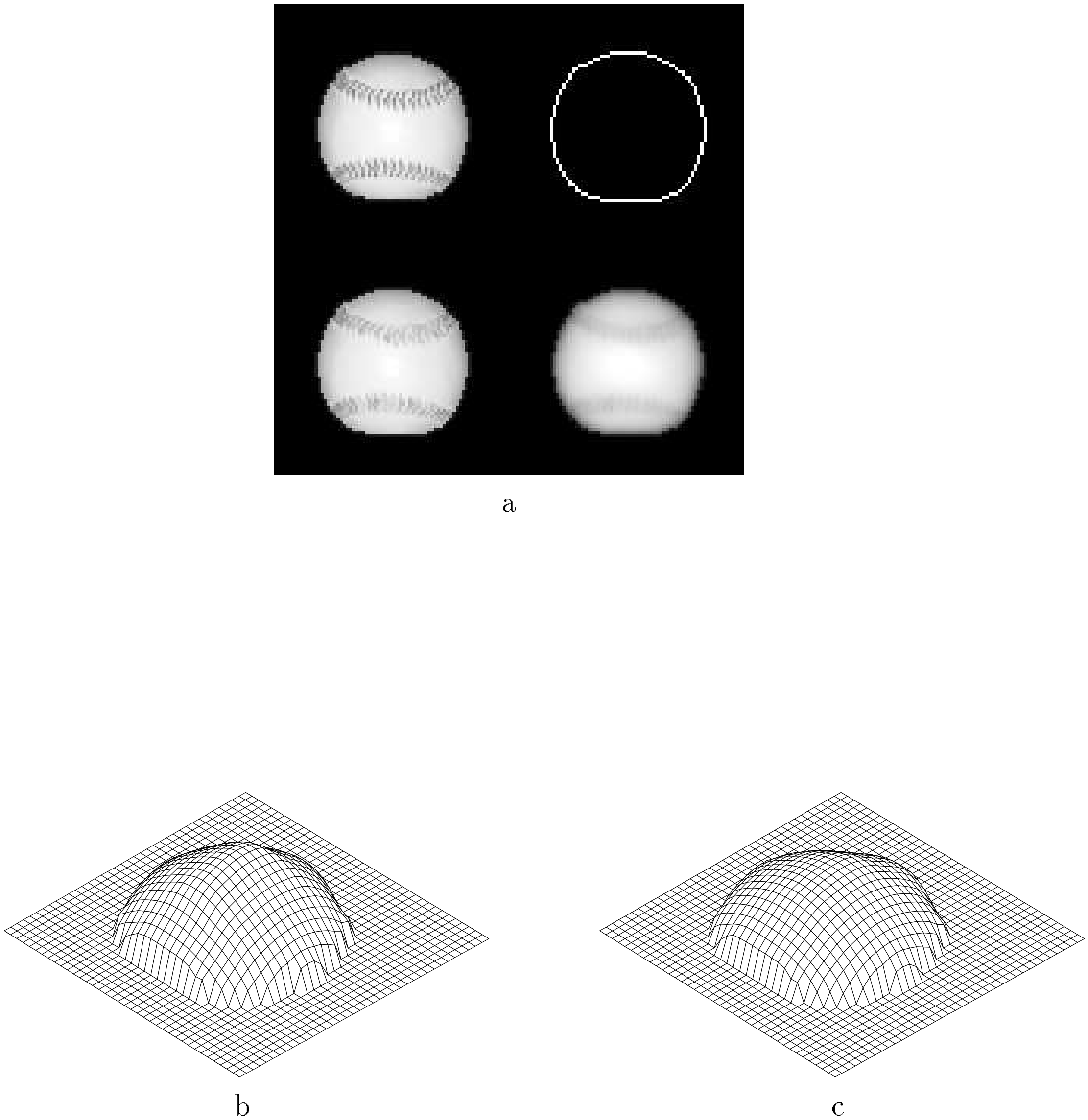
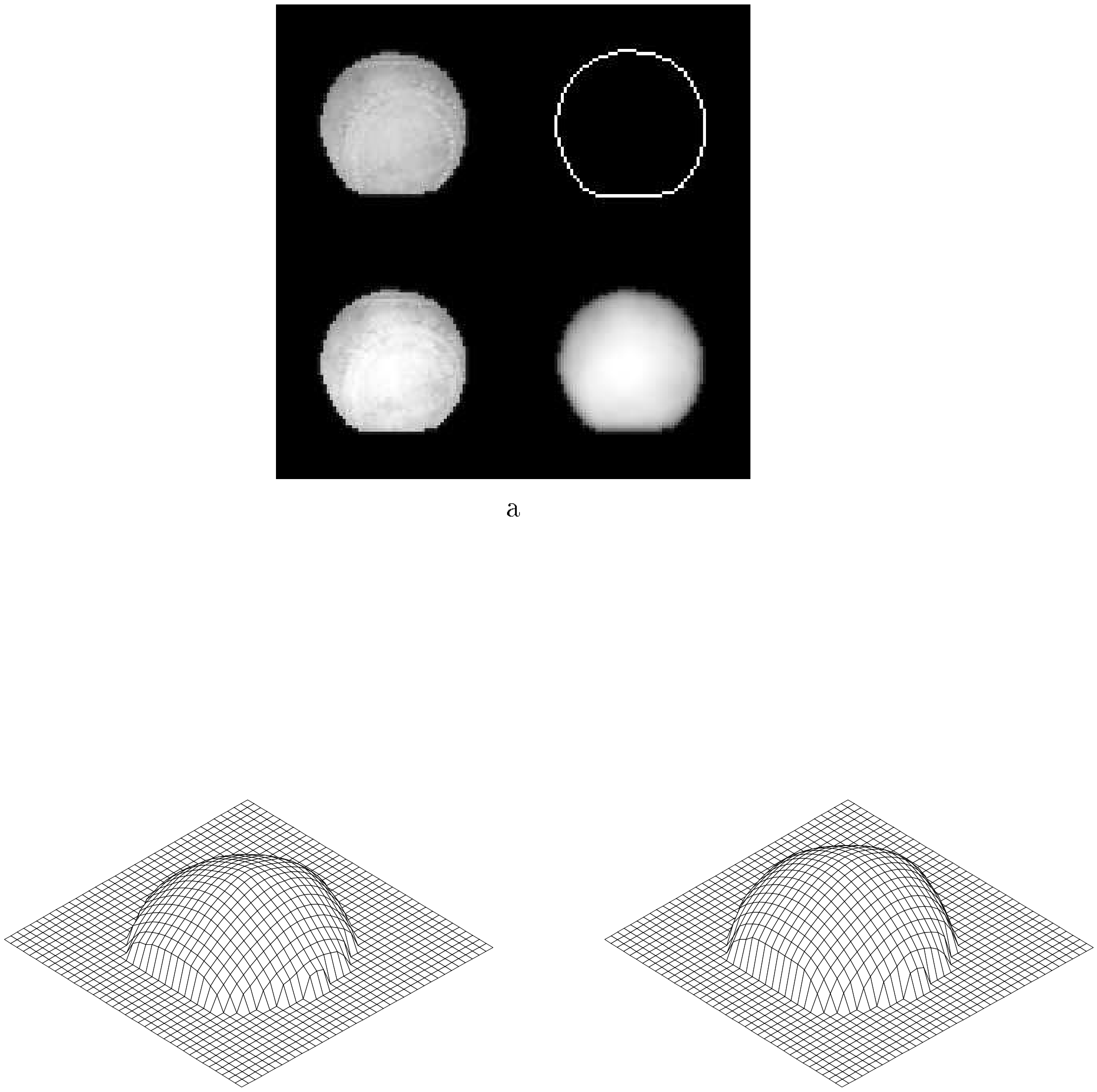


Figure 7.110: (a) Reconstructing the shape given the image of the new baseball. Top-Left: the original image after thresholding, Top-Right: the occluding contour located by edge detection, Bottom-Left: resynthesized image using the G.S.L. algorithm, Bottom-Right: resynthesized image using the V.A.N.L. (b) A mesh diagram showing the reconstructed heights using the G.S.L. algorithm (c) A mesh diagram showing the reconstructed heights using the V.A.N.L.



a

b c

Figure 7.111: (a) Reconstructing the shape given the image of the old baseball. Top-Left: the original image after thresholding, Top-Right: the occluding contour located by edge detection, Bottom-Left: resynthesized image using the G.S.L. algorithm, Bottom-Right: resynthesized image using the V.A.N.L. (b) A mesh diagram showing the reconstructed heights using the G.S.L. algorithm (c) A mesh diagram showing the reconstructed heights using the V.A.N.L.

Chapter 8

Conclusions and Future Work

The main contribution of this thesis, is the extension of shape-from-shading to handle non-Lambertian surfaces. It is commonly understood that most surfaces are not diffuse, yet most research in shape-from-shading has focused on Lambertian objects because of the inherent simplicity of such surfaces.

Two innovative algorithms have been formulated to handle non-Lambertian surfaces, the Variational Algorithm for Non-Lambertian Surfaces, and the General Shading Logic algorithm. The V.A.N.L. incorporates the Torrance-Sparrow reflectance function into Brooks and Horn's variational algorithm [6]. The General Shading Logic algorithm is an extension of Vega and Yang's [39] algorithm. In this thesis, Vega and Yang's algorithm was generalized to handle any reflectance model. It was suggested that incorporating different reflectance functions into heuristic algorithms (such as the G.S.L.) is easier than variational algorithms (like the V.A.N.L). Extensive experiments were conducted on both new algorithms to determine the effectiveness of the approaches as the reflectance, shape, illuminant direction, and number of iterations was varied. It was determined experimentally that both algorithms are effective in dealing with non-Lambertian surfaces, however, the V.A.N.L. exhibited the same unfavourable characteristics intrinsic to all previously formulated variational algorithms. In both real and synthesized experiments, the G.S.L. outperformed the V.A.N.L., however, both algorithms are effective at dealing with non-Lambertian surfaces, unlike the traditional algorithms.

Higher degree polynomial interpolation was presented as a novel approach to recover height from gradient information. It was found that using a degree-two polynomial interpolation was more effective than the previous linear approach. Experiments were conducted comparing the convergence time and errors in the reconstructed heights.

A practical shape-from-shading system was presented, called The LIRAS Recovery System. LIRAS includes algorithms to estimate the reflectance map prior to the shape-recovery process. When dealing with non-Lambertian surfaces, the reflectance properties vary from object to object, thus algorithms have to be formulated to estimate the surface type before the shape can be effectively recovered. The LIRAS Recovery System includes such algorithms and the effectiveness of the system was verified using real and synthesized data.

This thesis provides effective solutions to recover the shape of a more general class of surface. However, the real world is a complicated place and many other issues are left for future work. It is assumed that an object's reflectance is consistent, however, many objects have textures. A texture is a region on a surface in which the reflectance properties of the surface change. Generalizing shape-from-shading algorithms to handle objects with textures would be a significant contribution. Another assumption that needs to be relaxed for a more practical vision system is the assumption that the object is illuminated by a single infinite source. Algorithms have to be developed to handle multiple illuminant scenarios.

References

[1] S. Bakshi. A study of shape-from-shading algorithms. In *Proceedings of the Fifth Annual Graduate Symposium on Computational Science*, pages 214–225, Saskatoon, Sk., 1993. Department of Computational Science, University of Saskatchewan.

[2] A. Barr. Superquadrics and angle preserving transformations. *IEEE Transactions on Computer Graphics and Applications*, 1(1):11–23, January 1981.

[3] P. Beckman and A. Spizzochino. *The Scattering of Electromagnetic Waves from Rough Surfaces.* Pergamon, New York, 1963.

[4] J. Blinn. Models of lighht reflection for computer synthesized pictures. *ACM Computer Graphics (SIGGRAPH 77)*, 11(2):192–198, 1977.

[5] M. Brooks. Light source direction from a single image: A performance analysis. *The Australian Computer Journal*, 23(4):165–174, 1991.

[6] M. Brooks and B. Horn. Shape and source from shading. In *Proceedings of the 1985 International Joint Conference on Artificial Intelligence*, pages 932–936, Los Angeles, CA, 1985. Artificial Intelligence Conference Inc., Morgan Kaufmann Publishers.

[7] A. Bruckstein. On shape from shading. *Computer Vision, Graphics, and Image Processing*, 44(2):139–154, November 1988.

[8] P. Burger and D. Gillies. *Interactive Computer Graphics*, chapter 3.2. Addison-Wesley Publishing Company, Tokyo, 1989.

[9] E. Coleman and R. Jain. Obtaining 3D shape of textured and specular surfaces using four-source photometry. *Computer Vision, Graphics, and Image Processing*, 18(4):309–328, April 1982.

[10] R. Cook and K. Torrance. A reflectance model for computer graphics. *ACM Transactions on Computer Graphics*, 1(1):7–24, 1982.

[11] R. Courant and D. Hilbert. *Methods of Mathematical Physics, Volume I.* Wiley, 1953.

[12] M. Drew. Shape from color. Center for Systems Science CSS/LCCR TR 92-07, Simon Fraser University, Burnaby, B.C., 1992.

[13] F. Ferrie and M. Levine. Where and why local shading analysis works. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 11(2):198–206, 1989.

[14] R. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988.

[15] R. Hall. *Illumination and Color in Computer Generated Imagery*, chapter 4.2. Springer-Verlag, New York, 1989.

[16] G. Healey and T. Binford. Local shape from specularity. *Computer Vision, Graphics, and Image Processing*, 42:62–86, 1988.

[17] B. Horn. *Obtaining Shape From Shading Information*, chapter 4. The Psychology of Computer Vision. McGraw-Hill, 1975.

[18] B. Horn. Understanding image intensities. *Artificial Intelligence*, 8(2):201–231, April 1977.

[19] B. Horn. *Robot Vision.* The MIT Electrical Engineering and Computer Science Series. The MIT Press, McGraw-Hill Book Company, 1986.

[20] B. Horn and M. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics and Image Processing*, 33(2):174–208, 1986.

[21] B. Horn and R. Sjoberg. Calculating the reflectance map. *Applied Optics*, 18(11):1770–1779, 1979.

[22] K. Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 3(6):661–669, 1981.

[23] K. Ikeuchi and B. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1-3):141–184, 1981.

[24] C. Lee and A. Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence*, 26(2):125–143, 1985.

[25] K. Lee and C. Kuo. Shape from shading with a linear triangular element surface model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:815–822, August 1993.

[26] D. Marr. Analysis of occuding contour. In *Proceedings of the Royal Society of London B*, pages 441–475, 1977.

[27] S. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Transactions of Robotics and Automation*, 6(4):418–431, August 1990.

[28] S. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: Physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):611–634, July 1991.

[29] A. Pentland. Finding the illuminant direction. *Journal of the Optical Society of America*, 74(4):448–455, April 1982.

[30] A. Pentland. Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):170–187, 1984.

[31] A. Pentland. Linerar shape from shading. *International Journal of Computer Vision*, 4(2):153–162, 1990.

[32] B. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(8):311–317, 1975.

[33] G. Romney. *Computer Assisted Assembly and Rendering of Solids*. PhD Thesis, University of Utah, Department of Electrical Engineering, Salt Lake City, Utah, 1969.

[34] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, New York, 1987.

[35] H. Tagare and R. deFiguiredo. A theory of photometric stereo for a class of diffuse non-lambertian surfaces. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 13(2):133–152, February 1991.

[36] S. Tandri and Y.H. Yang. Comparison of two shape-from-shading algorithms. *Pattern Recognition Letters*, 11:637–642, September 1990.

[37] K. Torrance and E. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 57(9):1105–1114, 1967.

[38] O. Vega. Default shape theory: With applications to the recovery of shape and light source from shading. Master's thesis, University of Saskatchewan, Computational Science Department, October 1991.

[39] O. Vega and Y.H. Yang. Shading logic: A heuristic approach to recover shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):592–597, June 1993.

[40] O. Vega and Y.H. Yang. Default shape theory: With application to the recovery of light source direction. *Computer Vision, Graphics and Image Processing: Image Understanding*, to appear.

[41] R. Weinstock. *Calculus of Variations.* International Series in Pure and Applied Mathematics. McGraw-Hill Book Company Inc., 1952.

[42] S. Williamson and H. Cummins. *Light and Color.* John Wiley and Sons, New York, 1983.

[43] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):680–702, 1991.

Appendix A

Coefficients for Sum of Squares Approach

The general form of a degree three polynomial is:

$$y = a_3x^3 + a_2x^2 + a_1x + a_0. \tag{A.1}$$

Using a sum-of-squares approach as described in Section 2.4.3 the coefficients are solved to be:

$$
\begin{aligned}
&a_0 = \\
&\tfrac{1}{2}((-27i^3P^2j^3m_i - 13i^4P^2j^2m_j + 11i^5P^2jm_i - i^4P^2j^2m_i - 108j^2P^2iH_j - 108j^2P^2i \\
&H_i - 78P^2ji^3m_j - 30P^2i^4m_j + 30P^2j^4m_i + i^6j^2m_P + 26i^5P^2H_j - i^7jm_j - i^7jm_P + 6j^6iH_i \\
&-26j^5P^2H_i - 72i^2P^3H_j - 72i^2P^3H_i + 6i^5Pm_j - 6i^5Pm_P + 2i^7H_j - 2j^7H_i + 64i^2j^3P^2H_i - 8i \\
&j^4P^2H_i + 30ij^4P^2H_j - 64i^3j^2P^2H_j - 30i^4P^2jH_i + 8i^4P^2jH_j - 18i^2P^4jH_i - 36i^2P^4jH_j - 4i^5 \\
&P^2jm_j - 11j^5P^2im_j + 4j^5P^2im_i + j^4P^2i^2m_j + 13j^4P^2i^2m_i - 24i^2Pj^3m_P - 72i^3PjH_i - 72 \\
&i^3PjH_j - 4j^4i^3Pm_P - 12j^4i^3Pm_j + 2j^4i^3Pm_i - 2j^3i^4Pm_j - 6j^2i^5Pm_i + 12j^3i^4P \\
&m_i + 18j^2i^4PH_i + 14j^2i^4PH_j + 8Pi^2j^5m_P) + (6Pi^2j^5m_j - 6Pi^2j^5m_i + 4Pi^4j^3m_P + 6P \\
&i^5j^2m_j - 2Pi^6jm_i + 6Pi^5jH_i - 10Pi^5jH_j - 4Pj^6im_P + 2Pj^6im_j - 14Pj^4i^2H_i - 18P \\
&j^4i^2H_j + 18j^2P^4iH_j + 36j^2P^4iH_i - 36P^3i^4H_j + 42P^3i^3m_j + 30P^3i^3m_i + 36P^3j^4H_i - 30P^3 \\
&j^3m_j - 42P^3j^3m_i - 36ijP^4m_j + 36ijP^4m_i + 30ji^3P^3H_j + 42ji^3P^3H_i - 8j^2i^5Pm_P + 4ji^6 \\
&Pm_P + 90ji^2P^3m_j - 18ji^2P^3m_i - 30P^3ij^3H_i - 42P^3ij^3H_j + 18P^3ij^2m_j - 90P^3ij^2m_i - 6 \\
&Pj^5iH_j + 10Pj^5iH_i - 6i^3P^3j^2m_j - 18i^4P^3jm_i + 12i^4P^3jm_j + 24i^3P^3j^2m_i - 8i^6 \\
&PH_j + 8j^6PH_i + 18j^4P^3im_j - 24i^2P^3j^3m_j + 6i^2P^3j^3m_i - 12j^4P^3im_i + 9j^3m_iiP^4 + 9P^4 \\
&m_ii^3c) + (18i^3P^4H_j - 6j^5im_P - 18j^2m_ii^2P^4 - 36j^3P^2H_j - 18i^3j^3m_j - 36i^2j^3H_i + 6j^5P \\
&m_P - 9P^4m_ji^3c + 18j^2P^4m_i + 18j^2P^4m_j + 24i^2j^4m_P + 18i^3j^3m_i + 6i^4P^2m_P + 36j^2i^3H_i + 6i^5 \\
&jm_P + 36i^3P^2H_j - 18i^2P^4m_i + 36i^3P^2H_i + 72j^2P^3H_j - 18j^3P^4H_i + 6j^5im_i + 12j^4i^2m_j - 18i^2 \\
&P^4m_j - 12i^4j^2m_i - 6i^5jm_j - 6j^5Pm_i + 18i^2m_jj^2P^4 - 36i^2j^3H_j - 36j^3P^2H_i - 9P^4m_jij^3 + 72 \\
&j^2P^3H_i + 36j^2i^3H_j - 6j^4P^2m_P + 28i^3H_jPj^3 + 48i^2H_jj^2P^3 - 28j^3H_iPi^3 - 48j^2H_ii^2 \\
&P^3 + 72j^3PiH_j + 72j^3PiH_i + 54j^2Pi^3m_j + 24j^2Pi^3m_P + 24jPi^4m_i + 30jPi^4m_j + 18jP \\
&i^4m_P - 18j^4Pim_P) - (24j^4Pim_j - 30j^4Pim_i + 6j^3Pi^2m_j - 6j^2Pi^3m_i - 54j^3Pi^2 \\
&m_i + 24P^2ij^3m_P + 42P^2ij^3m_j + 78P^2ij^3m_i - 90P^2i^2j^2m_j - i^2j^6m_P - 5i^2j^6m_i - 5i^3j^5 \\
&m_j - 13i^4j^4m_i + 12i^3j^5m_i + 30i^3j^4H_i + 14i^3j^4H_j - i^3j^5m_P + 13i^4j^4m_j + 5i^5j^3m_i - 14i^4j^3 \\
&H_i - 30i^4j^3H_j + i^5j^3m_P - 12i^5j^3m_j + 5i^6j^2m_j - 6i^6jH_j + j^7im_P + j^7im_i - 20j^5i^2H_i + 20j^2i^5 \\
&H_j - 24j^2i^4m_P + 36P^4iH_j + 36P^4iH_i - 36P^4jH_i - 36P^4jH_j - 12P^2i^4m_i + 12P^2j^4m_j - 42P^2i^3j \\
&m_i + 90P^2j^2i^2m_i + 108P^2i^2jH_i + 108P^2i^2jH_j + 3ij^5P^2m_P + 6i^4j^2P^2m_P - 6i^2j^4P^2 \\
&m_P - 24ji^3P^2m_P - 3ji^5P^2m_P + 27i^3P^2j^3m_j))(1/(-2i^5jP + 27ij^2P^4 - 27i^2jP^4 - 11i^4j \\
&P^2 + 16i^4j^2P - 32i^3j^2P^2 - 9j^3P^4 - 36i^2j^3 + 36j^2i^3 + 36i^3P^3j - 72i^2P^3 + 72j^2P^3 + 36i^3P^2 - 36 \\
&j^3P^2 + 108ji^2P^2 - 108ij^2P^2 - 72Pji^3 + 72Pij^3 - 36jP^4 + i^7 - 16i^2Pj^4 + 32i^2j^3P^2 - 36ij^3P^3 \\
&+2ij^5P + 11ij^4P^2 + 18j^4P^3 + 4j^6P - 13P^2j^5 - j^7 - 4i^6P - 3i^6j + 10i^5j^2 + 13i^5P^2 - 18i^4 \\
&P^3 - 22i^4j^3 + 22i^3j^4 + 9i^3P^4 - 10i^2j^5 + 36iP^4 + 3ij^6))
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
a_1 = {} & \\
& \tfrac{1}{2}(-6ij^5H_j + 6ij^5H_i - 3ij^6m_i - 72iP^4m_j + 2ij^6m_j + 72i^3jPm_P + 144ij^2P^2m_i - 72ij^3Pm_P \\
& -6ij^3P^2H_j + 72jP^4m_i + 42P^3j^3H_i - 42P^3j^3H_j + 9j^3m_iP^4 + 72P^2j^3m_i + 3j^5P^2m_P \\
& -30j^4P^2H_i + 30j^4P^2H_j - 11j^5P^2m_j + 4j^5P^2m_i + j^7m_i + j^7m_P - 4Pj^6m_P + 2Pj^6 \\
& m_j - 144P^3j^2m_i - 6Pj^5H_j + 6Pj^5H_i + 18j^4P^3m_j - 12j^4P^3m_i - 10ji^4P^2m_j + 2j^4i^2Pm_i + 18j^2 \\
& i^2P^3m_i - 18j^2i^2P^3m_j + 16j^4i^2Pm_P - 3ji^4P^2m_P + j^2i^5m_P - j^5i^2m_P - 10i^4j^3m_j - 15i^4 \\
& j^3m_i - 2i^6jm_i - 3i^5P^2m_P - i^6jm_P + 30i^4P^2H_j + 9i^3P^4m_i - 30i^4P^2H_i + 6i^5jH_i - 6i^5jH_j - 24 \\
& i^2j^4H_i + 15j^4i^3m_j - 9i^3P^4m_j + 9i^5j^2m_i - 9j^5i^2m_j + 3i^6jm_j + 24i^2j^4H_j - 9j^4i^3m_P + 10 \\
& j^4i^3m_i - 72i^3P^2m_j - 72i^3j^2m_P + 72i^2j^3m_P + 144i^2P^3m_j + 11P^2i^5m_i + 18i^2P^4H_j - 18i^2P^4H_i \\
& +10iP^2j^4m_i - 12iP^3j^3m_j + 6i^4PjH_i - 6i^4PjH_j + 42i^2jP^3H_j - 42i^2jP^3H_i) - (12i^3j^2P \\
& H_i + 12i^3j^2PH_j - 48j^2P^2i^2H_j + 48j^2P^2i^2H_i + 12P^3i^4m_j - 18P^3i^4m_i + 6i^5PH_i - 6i^5PH_j + 9 \\
& j^3i^4m_P + 11j^2i^3P^2m_j - 13j^2i^3P^2m_i - 4j^4Pi^2m_j + 13P^2i^2j^3m_j + 2ji^5Pm_P + 12 \\
& ji^3P^3m_i + 2j^3i^3Pm_i + 4ji^5Pm_j - P^2i^4jm_i + P^2ij^4m_j + 24j^2i^3P^2m_P + 6j^4Pi \\
& H_i - 2j^3Pi^3m_j + 4j^2Pi^4m_i - 2ij^5Pm_P - 9ij^2P^4m_i + 9ij^2P^4m_j + 42ij^2P^3H_j - 4i \\
& j^5Pm_i + 2ij^5Pm_j - 12j^3Pi^2H_i - 2jPi^5m_i + 12j^3Pi^2H_j - 6j^4PiH_j) + (6ij^3P^2 \\
& H_i - 42ij^2P^3H_i + 3ij^4P^2m_P - 6ji^3P^2H_j - 9ji^2P^4m_i + 6ji^3P^2H_i + 9ji^2P^4m_j - 11j^3 \\
& i^2P^2m_i - 2j^2i^4Pm_j - 16j^2i^4Pm_P - i^7m_j - i^7m_P - 24j^3i^2P^2m_P + j^6im_P - 24i^4j^2H_i + 24i^4 \\
& j^2H_j - 4P^2i^5m_j - 2Pi^6m_i - 42i^3P^3H_j + 42i^3P^3H_i + 4i^6Pm_P + 72i^3jPm_j - 72i^2jP^2m_P - 144 \\
& i^2jP^2m_j - 72ij^3Pm_i - 72ijP^3m_i + 72ijP^3m_j + 72ij^2P^2m_P - 36i^3j^3H_j + 36i^3j^3H_i - 36ij \\
& P^4H_j + 36ijP^4H_i - 9P^4m_jj^3 + 18P^4H_jj^2 - 18P^4H_ij^2)1/(-2i^5jP + 27ij^2P^4 - 27i^2jP^4 - 11i^4 \\
& jP^2 + 16i^4j^2P - 32i^3j^2P^2 - 9j^3P^4 - 36i^2j^3 + 36j^2i^3 + 36i^3P^3j - 72i^2P^3 + 72j^2P^3 + 36i^3 \\
& P^2 - 36j^3P^2 + 108ji^2P^2 - 108ij^2P^2 - 72Pji^3 + 72Pij^3 - 36jP^4 + i^7 - 16i^2Pj^4 + 32i^2j^3P^2 - 36i \\
& j^3P^3 + 2ij^5P + 11ij^4P^2 + 18j^4P^3 + 4j^6P - 13P^2j^5 - j^7 - 4i^6P - 3i^6j + 10i^5j^2 + 13i^5P^2 - 18 \\
& i^4P^3 - 22i^4j^3 + 22i^3j^4 + 9i^3P^4 - 10i^2j^5 + 36iP^4 + 3ij^6)
\end{aligned}
\tag{A.3}
$$

$a_2 =$

$$\frac{1}{2}(-2j^5im_j + 11i^5jm_j + 2j^5Pm_i - 4j^5Pm_j - 36j^3Pm_P - 36i^2P^2m_P + 6j^3P^2H_i + 6ij^4H_j + 6j^2P^3H_i + -16i^4j^2m_j + j^5im_P + 36i^2P^2m_j + 36j^3Pm_i + 36ij^3m_P - 6j^3P^2H_j + 9i^3j^3m_j - 36ij^3m_i - 10j^5Pm_P + 9j^2P^4m_i - 9j^2P^4m_j - 8i^2j^4m_P - 9i^3j^3m_i + 2i^5jm_i - 9i^4P^2m_P - i^5jm_P - 6i^3P^2H_j + 9i^2P^4m_i + 6i^3P^2H_i - 6j^2P^3H_j - 6i^4jH_i - 36jP^3m_i + 6i^4jH_j - 6ij^4H_i - 11j^5im_i - 2j^4i^2m_j - 9i^2P^4m_j + 2i^4j^2m_i + 36jP^3m_j + 36j^2P^2m_P + 36ji^3m_j - 36ji^3m_P + 16j^4i^2m_i + 9j^4P^2m_P + 6i^5H_i - 6i^5H_j - 6j^5H_j + 6j^5H_i - 2i^6m_i + 2j^6m_j - 2i^6m_j - 2i^6m_P + 2j^6m_i - 6j^3PiH_j + 6j^3PiH_i + 4j^2Pi^3m_j - 8j^2Pi^3m_P - 2jPi^4m_i - 36jPi^2m_j + 36jPi^2m_P - 2jPi^4m_j - 14jPi^4m_P + 2j^6m_P - 36P^4m_j + 14j^4Pim_P + 2j^4Pim_j + 2j^4Pim_i + 2j^3Pi^2m_j - 2j^2Pi^3m_i - 4j^3Pi^2m_i - 18P^2ij^3m_P - 3P^2ij^3m_j + 9P^2ij^3m_i + 9P^2i^2j^2m_j + 8j^2i^4m_P - 36j^2P^2m_i + 3P^2i^3jm_i - 9P^2j^2i^2m_i - 6P^2i^2jH_i + 6P^2i^2jH_j + 18ji^3P^2m_P + 6j^2P^2iH_j - 6j^2P^2iH_i - 9P^2ji^3m_j + 3P^2i^4m_j - 3P^2j^4m_i + 4Pi^5m_i - 6i^2P^3H_j + 6i^2P^3H_i - 36i^3Pm_j + 36i^3Pm_P - 2i^5Pm_j + 10i^5Pm_P + 36iP^3m_j - 36iP^3m_i + 36P^4m_i + 8i^2Pj^3m_P + 6i^3PjH_i - 6i^3PjH_j + 36iPj^2m_i - 36iPj^2m_P + 6P^3i^3m_j - 12P^3i^3m_i + 12P^3j^3m_j - 6P^3j^3m_i - 12i^4PH_i - 36ijP^2m_j + 36ijP^2m_i + 18ijP^4m_j - 18ijP^4m_i + 18ji^2P^3m_i - 18P^3ij^2m_j + 12i^4PH_j - 12j^4PH_i + 12j^4PH_j + 12ijP^3H_j - 12ijP^3H_i + 12i^2j^2PH_i - 12i^2j^2PH_j 1/(-2i^5jP + 27ij^2P^4 - 27i^2jP^4 - 11i^4jP^2 + 16i^4j^2P - 32i^3j^2P^2 - 9j^3P^4 - 36i^2j^3 + 36j^2i^3 + 36i^3P^3j - 72i^2P^3 + 72j^2P^3 + 36i^3P^2 - 36j^3P^2 + 108ji^2P^2 - 108ij^2P^2 - 72Pji^3 + 72Pij^3 - 36jP^4 + i^7 - 16i^2Pj^4 + 32i^2j^3P^2 - 36ij^3P^3 + 2ij^5P + 11ij^4P^2 + 18j^4P^3 + 4j^6P - 13P^2j^5 - j^7 - 4i^6P - 3i^6j + 10i^5j^2 + 13i^5P^2 - 18i^4P^3 - 22i^4j^3 + 22i^3j^4 + 9i^3P^4 - 10i^2j^5 + 36iP^4 + 3ij^6)) \tag{A.4}$$

$$
\begin{aligned}
&a_3 = \\
&\tfrac{1}{2}(-6i^3jH_i + 6i^3jH_j - 24iP^2m_P + 6ij^3H_j + 10j^3PH_j - 11i^3j^2m_j + 24jP^2m_P - 6i \\
&j^3H_i - 10j^3PH_i - 4i^2j^2H_j + 24i^2Pm_P + 24i^2jm_j + 4i^2j^2H_i - 24j^2im_i - 24j^2Pm_P - 24ji^2 \\
&m_P + 4i^4H_i - 4i^4H_j - 4j^4H_j + 4j^4H_i + 10i^2PjH_i - 12i^3jPm_P + 3ij^2P^2m_i + 24ij^2m_P - 15iP^2 \\
&j^2m_j + 12ij^3Pm_P - 2i^3jPm_i + j^5m_P + j^5m_i + 2j^5m_j - 6j^4Pm_P + 6P^2j^3m_P + 11P^2j^3 \\
&m_j - 7P^2j^3m_i - 10j^2P^2H_j + 10j^2P^2H_i - 6j^4Pm_j + 2j^4Pm_i - 24P^3m_j + 24P^3m_i + 24P \\
&j^2m_i + 24jP^2m_j - 48jP^2m_i - 6P^3j^2m_j + 6P^3j^2m_i + 4i^4jm_i + 6i^4Pm_P + 6i^4Pm_i + 7i^4j \\
&m_j - 2i^4Pm_j - i^4jm_P - 7i^3j^2m_i - 10i^3PH_i - 6i^3P^2m_P + 7i^3P^2m_j + 10i^3PH_j + 8i^3j^2m_P - 11 \\
&i^3P^2m_i + 7i^2j^3m_j - 24i^2Pm_j + 11i^2j^3m_i - 8i^2j^3m_P - 10i^2P^2H_j + 10i^2P^2H_i - 6i^2P^3m_j + 6 \\
&i^2P^3m_i - 4ij^4m_j - 7ij^4m_i + ij^4m_P + 48iP^2m_j - 24iP^2m_i - 2i^5m_i - i^5m_j - i^5m_P - 6i^3jP \\
&m_j + 18i^2jP^2m_P - 10i^2PjH_j + 12i^2Pj^2m_j - 12i^2Pj^2m_i - 3i^2jP^2m_j + 15i^2jP^2m_i + 6i \\
&j^3Pm_i - 48ijPm_j + 2ij^3Pm_j - 12ijP^3m_i + 12ijP^3m_j - 18ij^2P^2m_P - 20iP^2jH_i + 20iP^2j \\
&H_j + 48ijPm_i + 10ij^2PH_i - 10ij^2PH_j 1/(-2i^5jP + 27ij^2P^4 - 27i^2jP^4 - 11i^4jP^2 + 16i^4 \\
&j^2P - 32i^3j^2P^2 - 9j^3P^4 - 36i^2j^3 + 36j^2i^3 + 36i^3P^3j - 72i^2P^3 + 72j^2P^3 + 36i^3P^2 - 36j^3P^2 + 108j \\
&i^2P^2 - 108ij^2P^2 - 72Pji^3 + 72Pij^3 - 36jP^4 + i^7 - 16i^2Pj^4 + 32i^2j^3P^2 - 36ij^3P^3 + 2ij^5P + 11i \\
&j^4P^2 + 18j^4P^3 + 4j^6P - 13P^2j^5 - j^7 - 4i^6P - 3i^6j + 10i^5j^2 + 13i^5P^2 - 18i^4P^3 - 22i^4j^3 + 22 \\
&i^3j^4 + 9i^3P^4 - 10i^2j^5 + 36iP^4 + 3ij^6))
\end{aligned}
\tag{A.5}
$$

Appendix B

List of Common Symbols

E	Image intensity, $E(x, y)$ is the intensity at point x, y
R	reflectance map
$\mathbf{n}$	surface normal, $\mathbf{n}(x, y)$ is the surface normal at point x, y
$\bar{\mathbf{n}}$	local average of the normals
p	partial derivative of z with respect to x
q	partial derivative of z with respect to y
$\mathbf{s}$	light source direction vector
$\mathbf{h}$	vector that bisects that angle formed by $\mathbf{e}$ and $\mathbf{s}$
$\mathbf{e}$	viewer direction vector
λ	regularization constant used in variational algorithms
ϵ	grid distance
∇	Laplacian operator
H	height map
m	slope
D	distribution of the microfacets
G	geometric attenuation function
F	Fresnel reflectance
q K_a	ambient coefficient
K_d	diffuse coefficient
K_s	specular coefficient
E_d	diffuse component
E_s	specular component
σ	roughness of the surface using the Torrance-Sparrow model
α	angle between the vectors $\mathbf{n}$ and $\mathbf{h}$