

USING HYBRID APPROACHES TO SOLVE THE CHALLENGES OF
SHAPE FROM SHADING

A Thesis

Presented to

The Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Ryan Alan Murphy

March 2007

AUTHORIZATION FOR REPRODUCTION OF MASTER'S THESIS

I reserve the reproduction rights of this thesis for a period of seven years from the date of submission. I waive reproduction rights after the time span has expired.

Signature

Date

APPROVAL PAGE

TITLE: Using Hybrid Approaches to Solve the Challenges of Shape from
Shading

AUTHOR: Ryan Alan Murphy

DATE SUBMITTED: March 2007

Dr. Zoë Wood
Advisor or Committee Chair

Signature

Dr. Clark S. Turner
Committee Member

Signature

Dr. Franz J. Kurfess
Committee Member

Signature

Abstract

Using Hybrid Approaches to Solve the Challenges of Shape from Shading

by

Ryan Alan Murphy

For over thirty years researchers have been trying to solve the Shape from Shading problem of determining 3D shape from a single image with a single light source. This paper shows that this problem can never be solved perfectly due to challenges of light orientation, camera type, ambiguity, multiple materials, and specular highlights.

However, we can get accurate results on small, simple shapes where we can make assumptions to remove these challenges. This paper shows how each of these challenges can be overcome through the use of other techniques such as pattern matching and stereopsis. The paper will then present a new hybrid method of Shape from Shading that can be used to autonomously capture 3D information from 2D images of single objects with multiple peaks and multiple materials with specular components.

Contents

List of Figures	vii
1 Introduction.....	1
1.1 Challenges with Shape from Shading.....	4
1.1.1 Challenge of Ambiguity.....	5
1.1.2 Challenge of Multiple Materials	7
1.1.3 Challenge of Specularity.....	8
2 Previous Work	10
2.1 Minimization.....	11
2.2 Local	12
2.3 Linearization	12
2.4 Propagation	13
2.5 Interactive Shape from Shading.....	15
2.6 Supporting Material for Hybrid Methods	17
2.6.1 Stereopsis	17
2.6.2 Image Segmentation.....	19
3 Algorithm.....	21
3.1 Solving the Challenge of Ambiguity	21
3.2 Solving the Challenge of Multiple Materials.....	24
3.2.1 Computing the Maximum Intensity for a Material	26

3.3 Solving the Challenge of Specularity	28
3.4 Summary of Algorithm	30
4 Sample Acquisition	32
4.1 Additional Acquisitions	35
5 Conclusion and Future Work	38
Bibliography	40
Appendix A: Deriving a technique for SFS	44
Appendix B: Proof of equivalence between Eikonal Equation and Derived Equation	49

List of Figures

Figure 1.1 SFS Example	1
Figure 1.2 Phong Model	2
Figure 1.3 Challenge of Ambiguity Example	6
Figure 1.4 Ambiguous Surface Example	7
Figure 1.5 Challenge of Multiple Materials Example	8
Figure 2.1 FM-SFS Results.....	14
Figure 2.2 Results from Interactive SFS.....	16
Figure 2.3 Stereopsis Diagram.....	18
Figure 3.1 Using Stereopsis to Solve the Challenge of Ambiguity	23
Figure 3.2 Two Images with Different Intensity Ranges.....	25
Figure 3.3 Material with and without a Specular Component	29
Figure 3.4 Improving the Challenge of Specularity.....	29
Figure 4.1 Stereo View of Vase	32
Figure 4.2 Constructed Surface Using FM-SFS	33
Figure 4.3 Constructed Surface with Handling of Multiple Materials	34
Figure 4.4 Identified Materials on Vase	35
Figure 4.5 Face Acquisition Example.....	36
Figure 4.6 Real Image Acquisition Example.....	37
Figure A.1 Phong Model Diagram.....	45
Figure A.2 Calculating Change in Depth.....	46
Figure A.3 Polar Coordinates	47

Chapter 1

Introduction

The Shape from Shading (SFS) problem can be described as acquiring 3D information about a surface from the intensity values of a single image [6]. The figure seen below is an example of an acquired surface using a SFS technique.

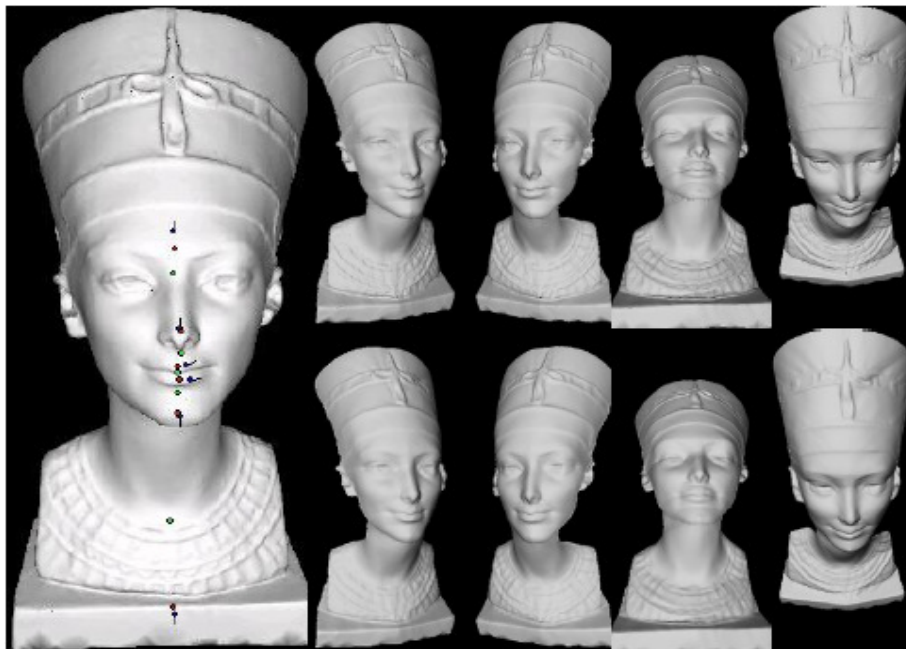


Figure 1.1 SFS Example: Multiple views of a constructed surface from a single image using SFS [4].

Three dimensional surface acquisition has applications in the computer vision, archaeological, medical, and entertainment fields because it provides a way to study, interact, visualize, and manipulate data that cannot be matched by two dimensional images.

One solution is to use Laser Range scanners which provide a way to digitally scan a 3D model to capture the information like a 3D fax machine [19]. However these machines are expensive, sometimes costing several hundred thousand dollars making their availability for more general applications limited. They also take several hours of post processing time [19] making them infeasible for real time applications like computer vision. It is because of this that a fast, autonomous, inexpensive method for acquiring 3D information is needed [1] [4].

SFS has been the topic of much research for over thirty years ever since the first technique was proposed in the early 70's by Horn [5]. Since then, the field of SFS has branched into multiple directions, but the goal remained the same. The simple, cheap, fast features of SFS techniques make it very attractive to sectors of industry that cannot afford expensive range scanners [19].

In general, SFS techniques work by using intensity values of an image to infer 3D information. This is based on light models that take 3D information and produce intensity values. A common light model used is the Phong Model. **Figure 1.2** gives a graphical depiction of how the Phong model works.

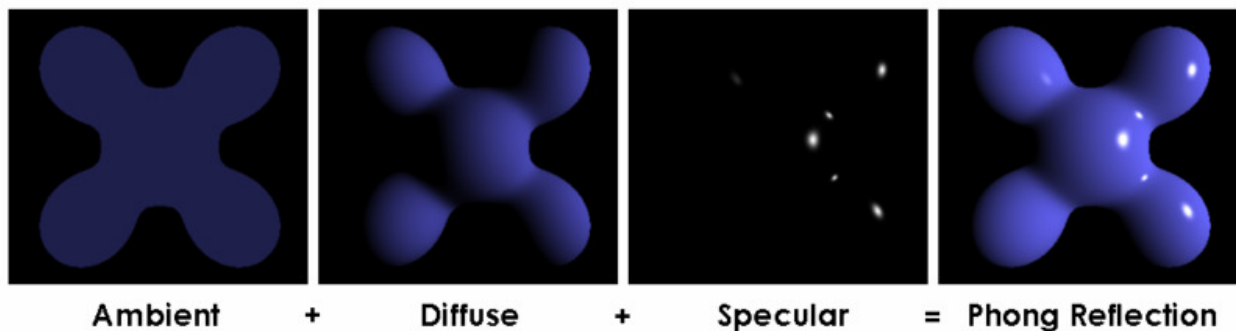


Figure 1.2 Phong Model: Graphical depiction of the Phong Model of Reflection. Created by Brad Smith, August 7, 2006.

The Phong model separates the total intensity visible on an object into an ambient glow, diffuse shading, and specular highlights [27].

- The ambient component is a constant intensity seen on every pixel that is caused by other light sources, reflections off other surfaces etc. This is often safe to ignore in SFS since the problem involves only a single light source.
- The diffuse component that goes into the final intensity seen in an image of a surface is only dependent on the angle at which the light hits the surface. It is independent of the viewing direction.
- The specular component causes the bright spots on a surface. This component is seen when the light rays are reflected near the view direction. We experience specular highlights every time we are blinded by the sun's rays after they have reflected off a smooth surface.

In order to simplify the problem most SFS techniques only worry about the diffuse component. Surfaces that only experience diffuse light are known as lambertian reflectors [30]. Using this constraint the intensity value at any point is given by the simple equation seen below.

$$I = \vec{L} \cdot \vec{N} \quad (1)$$

Where I is the final intensity value, L is a 3D vector representing the direction the light is facing, and N is the normal at that location on the surface. Supposing we can set up a situation where the light direction and at least one normal is known, we can use the light direction and the intensity value from an image to solve for the surface normal at a point neighboring the known normal. We can then continue solving for neighboring normals until we have a normal for every point.

Then using the computed normal for a given point, we can compute the change in depth. Intuitively, when we experience a bright region on an image we know the surface at that location is flat; when the image is shaded, the surface must be sloped. Once the changes in depth are computed for every pixel in the image, they can be used to construct the 3D surface. (Refer to **Section 2.4** for more information on this process.) However, techniques that acquire 3D information autonomously from a single 2D image are not without their challenges.

1.1 Challenges with Shape from Shading

There are five major challenges that modern SFS techniques face: light orientation with respect to the camera, perspective versus orthographic cameras, ambiguity (whether a surface region is convex or concave), multi material surfaces, and specular highlights.

Most SFS techniques simplify their input scene with the assumption that the light is at the optical center of a camera or apply a transformation to the problem so they can treat it as if the light was at the center [1]. This constraint of having the light at the optical center removes the need to handle shadows in an image but at the cost of having more flexible lighting conditions. However, due to the fact that it is common to have the light source near the optical center (consider a camera with a mounted flash), this assumption is acceptable.

Abundant research has focused on perspective versus orthographic cameras [7] [28] [29]. An orthographic view is comparable to taking a picture with a box as a camera, while the perspective view is like looking through a pinhole (e.g. in an orthographic projection parallel lines never meet, while in perspective they will meet at a vanishing point on the horizon). Most SFS techniques rely on an orthographic view. The challenge with using these techniques is that cameras take pictures with a perspective view. This difference in view causes variations in

calculated depths of a surface. In most cases the distance variations caused by perspective versus orthographic views of a surface are ignored [8]. This assumption does limit the amount of real applications a SFS technique can have since cameras in the real world are perspective. However, the problem can be solved easily with a simple transformation if the distance from object to camera is known [7].

The two previous challenges have been addressed in previous research and their general assumptions are acceptable for a number of SFS techniques. Due to this fact, this paper focuses on solving the three open challenges of ambiguity, multiple materials, and specularity.

1.1.1 Challenge of Ambiguity

The challenge of ambiguity to SFS is: given a single image with a single light source, how can one determine if a specific section of the surface is concave or convex? Belhumeur, Kriegman, and Yuille prove that when the lighting direction is unknown and specular highlights are negligible then the same image can be obtained by a continuous family of surfaces (depending linearly on three parameters)[9]. A concrete example of this can be seen in **Figure 1.3**, where the image could represent either a concave or convex sphere but there is no way to know for sure with the information provided in the scenario.

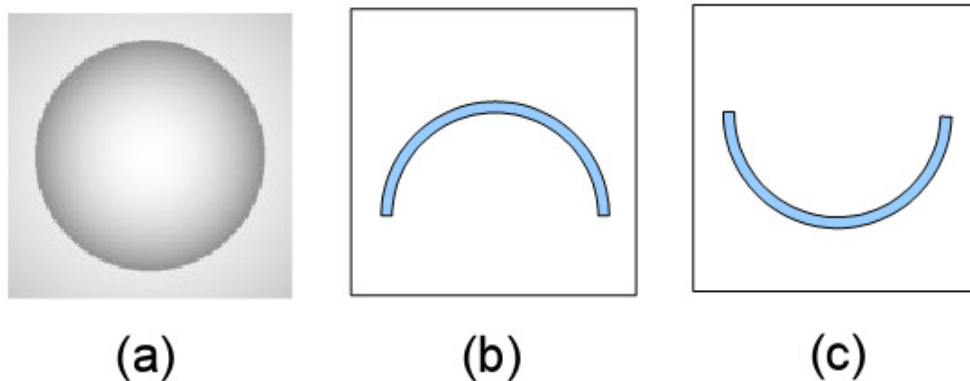


Figure 1.3 Challenge of Ambiguity Example: (a) An image of a hemisphere. (b) Slice of a surface that could produce the image in (a). (c) Slice of another surface that could produce the same image in (a).

In order to remove this ambiguity we have to make, further assumptions about our scene. For instance, the Fast Marching SFS (FM-SFS) (**Section 2.5**) solution uses a downhill approximation that assumes the surface is concave [10]. This downhill assumption removes the convex versus concave ambiguity but limits the technique to only spherical objects. Later works by Tankus, Sochen, and Yeshurun propose a method for a global solution by assuming the surface varies from convex to concave sections [11]. However this is limited to a particular type of surface called a Morse surface [12]. Even with the alternating up hill and down hill assumption, for complex solutions that show up in real life there are so many possible solutions that methods that make such assumptions are not very useful (refer to **Figure 1.4**). To get around such SFS ambiguities more information about the surface is required.

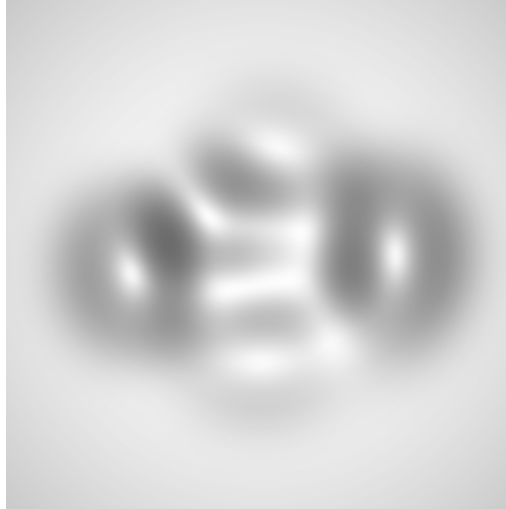


Figure 1.4 Ambiguous Surface Example: Example of a more complex image with 512 underlying surface possibilities.

1.1.2 Challenge of Multiple Materials

Most SFS research has been done on surfaces with a single material, but little research has been done on surfaces with multiple materials. Most SFS techniques assume that the objects in a given scene are all lambertian reflectors [30]. That is to say they are ideal surfaces: bright spots giving full light intensity and dark spots having no intensity. The problem is that in reality surfaces are not ideal. They typically absorb some of the light and some even absorb light differently in different areas of the surface. A surface is not always perfectly white or at least consistent; sometimes it is made up of multiple types of materials or multiple colors.

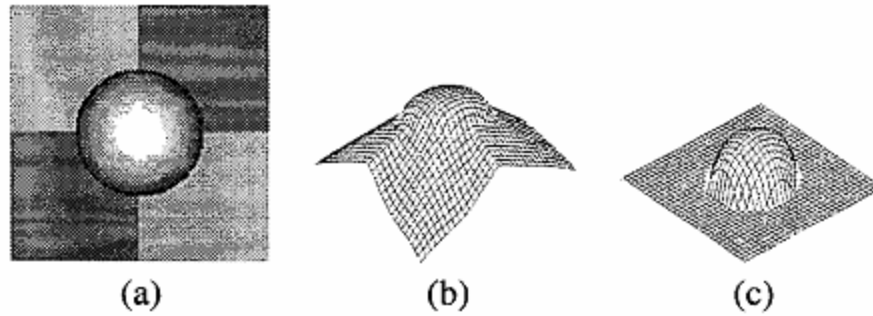


Figure 1.5 Challenge of Multiple Materials Example: Example of a surface with multiple materials [30]. (a) Input image. (b) Constructed surface using traditional SFS techniques. (c) Correct construction of the surface.

Traditional SFS approaches would not provide good solutions for images like the one in **Figure 1.5**. They would treat the line between the light and dark region in a surface to be a crease, rather than a flat region. Some research has been done to solve this problem but it is still in its early stages [30].

1.1.3 Challenge of Specularity

The challenge of specularity is similar to the challenge of multiple materials. As stated in the previous section, most SFS techniques assume that a surface is perfectly lambertian. This not only means it is of consistent material, but this also means it has no specular component.

Specular highlights are controlled when we constrain the problem to have the light source at the optical center of the camera since the bright spots that are experienced are in the same locations as bright regions caused by the diffuse component. This allows a SFS technique to ignore the specular highlights and treat the surface as lambertian. However, since SFS techniques use the assumption that high intensity regions correspond to flat regions, without knowing how strong the specular component is on a material, we often get extreme flat spots in a constructed surface when the actual surface is curved.

The rest of this paper will proceed as follows. First, we will explore previous approaches to SFS. We will then explore some techniques used in other methods of 3D data acquisition and computer vision to learn how they can be used to overcome the three open challenges of ambiguity, multiple materials, and specularity. Finally, we will apply the tools of other domains to create a new algorithm that does the following:

1. Captures depth information from a single object using a single light source and two cameras.
2. Captures depth information in real-time.
3. Operates autonomously (i.e. with no user input).
4. Allows for surfaces with multiple local minima and local maxima.
5. Allows for surfaces with multiple materials.
6. Allows for surfaces with specular components.

This new algorithm addresses the open challenges of SFS and allows for the construction of accurate 3D data from a single light source with minimal additional information.

Chapter 2

Previous Work

SFS was started by Berthold Horn in the early 70's with his discovery that the shape of a surface could be inferred from intensity information found in an image of a surface. Horn originally used a technique that started from one edge of an image and marched along the image constructing a surface based on the observed intensity values at any given point [24]. His approach later evolved to realize that solving for shape can be done by finding the solution of a nonlinear first-order Partial Differential Equation (PDE) called the brightness equation [5]. This equation, seen below, is known as an Eikonal equation.

$$\Delta z(x, y) = \sqrt{\frac{1}{I^2(x, y)} - 1} \quad (2)$$

This equation directly relates the change in depth of a surface to its intensity values, with the depth represented as the magnitude of the gradient of $z(x,y)$ and the intensity values as $I(x,y)$. A brief derivation of this equation as it relates to SFS can be found in [4].

Since its origins with Horn, the field of SFS has branched into four different directions: minimization, local, linearization, and propagation. A brief description of these approaches to the SFS problem is given below. The inclined reader will find detailed descriptions of these techniques as well as some of the results from them in recent surveys [1] [13] [26].

2.1 Minimization

Minimization or optimization approaches work by solving for the depth values of a surface by minimizing an error function. Recent surveys of minimization approaches have concluded that each approach encompasses a selection of energy function and minimization technique [13].

The error function is built up of many different components, called constraints, which allow the function to be used to describe an entire image. "The function can involve the brightness constraint, and other constraints, such as the smoothness constraint, the integrability constraint, the gradient constraint, and the normal constraint [1]." The brightness constraint represents the total brightness error of the acquired surface from the original image. The smoothness constraint ensures the final surface converges to a unique solution. The integrability constraint ensures that surfaces are valid, that is, the surface is one-to-one. The intensity gradient constraint insures that the distribution of intensities on a constructed surface stays close to the actual image. The unit normal constraint forces the normals on the acquired surface to be unit length.

In addition to choosing an energy function, a minimization technique has to be chosen that solves the energy function. As in Richard Szeliski's paper, Fast SFS [21], there are two ways to minimize an energy function. The first is to directly minimize the energy function E . The second is to solve a new system of equations produced by $\Delta E = 0$.

Although a majority of research in SFS uses minimization techniques, the approaches are still quite limited in the variety of surfaces that a given method can work on [13] (e.g. surfaces

with multiple local minima and maxima). This has motivated some researchers to take radically different approaches to the problem.

2.2 Local

Local approaches work by assuming an underlying shape (e.g. sphere) for a region, and then compute the normal at each point in the region. This process is repeated until the complete surface is generated. These methods rely on strong assumptions about surface shape [13].

Generally, this assumption is that the surface is locally spherical [22]. Because of this assumption these solutions perform extremely well on surfaces that are spherical in nature, or that are part of spheres. However, they perform exceptionally bad on surfaces that do not fit their assumption. While these methods are efficient and easy to implement, the assumption on underlying shape is quite limiting.

2.3 Linearization

Linear approaches take the non-linear problem introduced by the Eikonal equation and reduce it to a linear equation through the linearization of the reflectance map. A reflectance map is an equation that maps normals on a surface to intensities in an image.

$$I(x, y) = R(p, q) \quad (3)$$

Equation 3 gives an example of a reflectance equation where $I(x, y)$ is the intensity at a given pixel location, R is the reflectance map equation where (p, q) represents the normal at the corresponding pixel in polar coordinates.

Tsai and Shah's work in 1992 [23] still remains one of the best examples of linearization [6]. They operate under the assumption that a surface is lambertian. The intensity values on an image of a lambertian surface are only dependent on the angle at which a light reflects off a surface and not dependent, in any way, on the position of the camera. In other words, the surfaces are not shiny. Tsai and Shah's technique improves on previous work by first approximating the normals (p, q) from the input image, then linearizing a depth function $Z(x, y)$ directly rather than linearizing a reflectance map. This linearization is performed by first using a Taylor series to linearly approximate a given reflectance function. This generates a linear system of equations which is then solved using a Jacobi iterative method [23]. Tsai and Shah's technique works extremely well on synthetic images. However, because of their lambertian assumption, it suffers from the challenge of specularly and multiple materials. It therefore, performs poorly on real images since they typically are not ideal lambertian reflectors.

2.4 Propagation

Propagation approaches work by propagating solutions from known points. Propagation techniques start from a single or set of known points, and then propagate outward across the image to form the surface using intensity information from the input image as it marches outward from the given point.

The first SFS technique that was developed by Horn in 1970 was actually a propagation technique [24]. This first method worked by starting at one edge of a surface where the slope of the surface can be assumed to be at its maximum. It then linearly marching along the image using a change function to determine how the surface would change at each step based on the intensity at a given point (high intensity relates to flat region, low intensity relates to sloped

region). This solution worked well on some surfaces but had the disadvantage of needing to know the depths at the edges. This technique was later expanded upon by Oliensis when he observed that surface shape can be reconstructed from singular points rather than from edges or known curves [26].

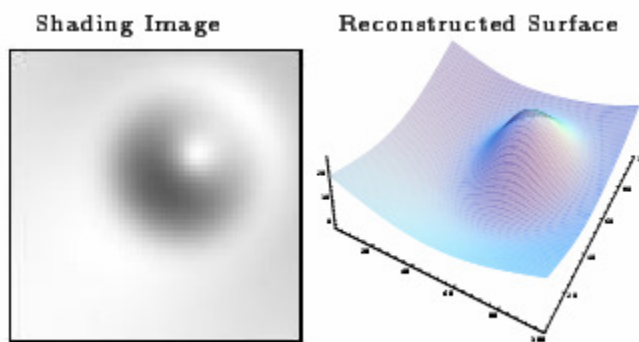


Figure 2.1 FM-SFS Results: Results from FM-SFS with the input image on the left and the reconstructed surface on the right [10].

The most well known propagation technique was developed by Kimmel and Sethian. Their method, dubbed Fast Marching SFS (FM-SFS) [10], is able to achieve extremely promising (refer to **Figure 2.1**) results with complexity of $O(N \log N)$ where N is the number of pixels in an image [10]. The algorithm works as follows:

1. The algorithm is initialized by adding a single point to a minimum depth sorted priority queue. This initial point is said to be the peak which is the highest point on the surface.

The algorithm then loops through the remaining steps until the queue is empty.

2. Remove the head of the queue and compute the depths of its unprocessed neighbors by adding the change in depth of the neighboring pixel to the depth of the head. Where the change in depth is computed by **Equation 2**.

$$\Delta z(x, y) = \sqrt{\frac{1}{I^2(x, y)} - 1} \quad (2)$$

3. The newly computed neighbors are then added to the queue to be processed later.

One disadvantage to this approach is that it works from the assumption that the surface we are trying to compute has only one maximum. In other words the solution has a single peak, and everything around that peak can be computed by using a downhill assumption. For example, if you had an image of a face, you would get the rounded region around the nose, but would not capture the local maxima at the cheeks, lips, etc. Weakness aside this method is popular because it efficiently provides a solution to the Eikonal equation by simply marching around an image. In the most recent survey to date FM-SFS gave an overall performance on both synthetic and real images that proved to be better than all other top SFS techniques [13].

2.5 Interactive Shape from Shading

In the work Interactive SFS (Zeng, Matsushita, Quan, and Shum)[4] a new SFS technique is proposed that uses human knowledge to solve for concave versus convex ambiguity. Zeng and colleagues base their work on SFS and interactive modeling. Their method uses FM-SFS to form local solutions. Because FM-SFS only works for surfaces with a single maximum (or minimum) point it cannot be used, by itself, to form an accurate solution to a complex surface with many peaks (or ruts). Due to this limitation they have to first locate these maxima then compute the relative altitudes between the maxima and feed this information as inputs into the FM-SFS algorithm.

In order to locate the maxima they rely on user input to make the problem feasible. The user provides a number of surface normals and the algorithm takes those normals and traces them back to peaks. Once the peaks are known the algorithm computes the relative altitudes of

each peak. The FM-SFS algorithm is then used with the initial peak altitudes as its inputs and produces a surface. The details of this method can be found in their paper [4].

One of the major areas of interest to this paper is their choice of having the user input a series of normals rather than indicating depths at certain points. They choose to do it this way because they felt it easier for a user to recognize surface orientation rather than its depth. However, because a user only specifies orientation at points of their choosing, the number of peaks is often lower than the actual number of peaks. This weakness is overcome by having the user look at the constructed surface, then looping back and entering in more normals to iteratively construct a solution that is to their satisfaction. Results of this interactive technique can be seen in **Figure 2.2**.

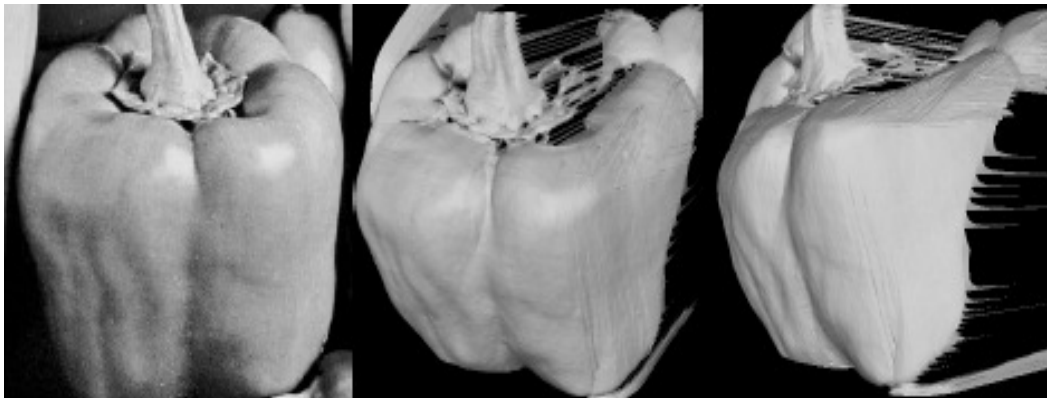


Figure 2.2 Results from Interactive SFS: Results from a real world scene reconstruction. From left to right, the input image followed by two views of the constructed image

This technique provides aesthetically pleasing results. However, this solution does not lead to an easy way to remove the human factor limiting its application to domains such as computer vision. The method presented in this thesis builds on the work of Zeng and colleagues by using a FM-SFS technique that allows for multiple peaks, but without user intervention.

2.6 Supporting Material for Hybrid Methods

Even with the excellent performance of recent SFS techniques they still have faults based on false assumptions. Therefore, it is important to review other techniques used in 3D data acquisition and computer vision to see how they overcome the challenges that face SFS in order to satisfy the goals of working autonomously, providing an unambiguous solution, and being able to handle multiple materials including those with specular components

The research and algorithms used in stereopsis and image segmentation can be used to help solve the challenges of SFS. Stereopsis allows our method to be autonomous and help differentiate ambiguous cases. Image segmentation allows our algorithm to handle images with multiple materials including those with specular components, by breaking the SFS problem into smaller pieces.

2.6.1 Stereopsis

Stereopsis, or stereo vision, is another popular approach to acquire 3D data. Stereopsis is modeled after the human vision system.

Our eyes capture different 2-D images of the objects around us and our brain uses these images to recover a description of the 3-D structure of the environment. Stereopsis is the process responsible for this reconstruction of the depth [17].

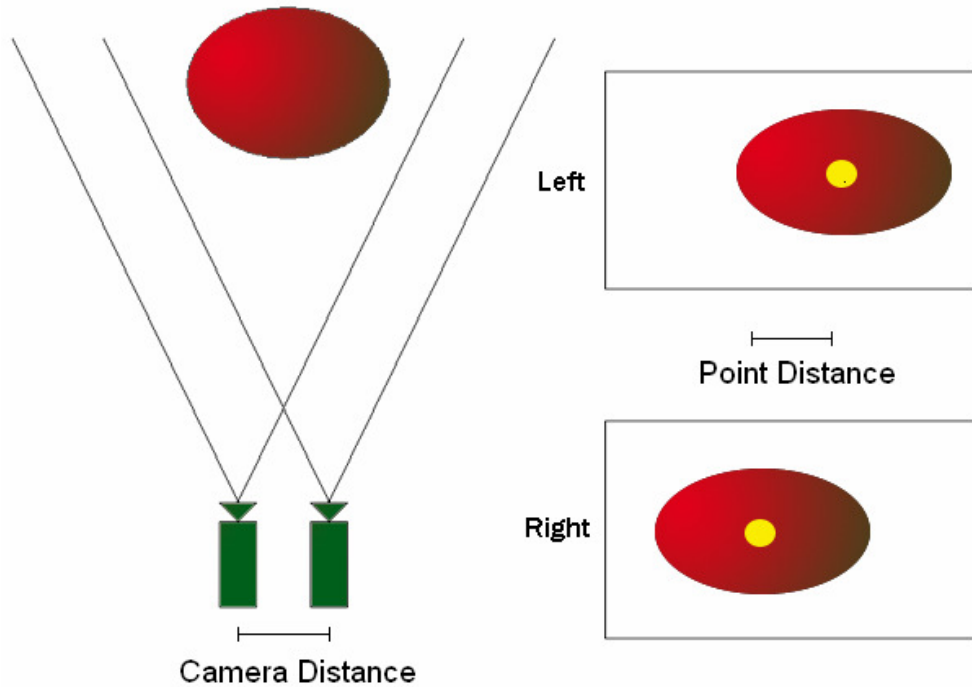


Figure 2.3 Stereopsis Diagram: Example of a Stereopsis setup with two cameras looking at the same object.

Depth can likewise be solved for computationally. Given two images of the same object from two different viewpoints you can compute the depth of the object can be computed by knowing the distance and angle between the cameras and the distance between the same points in an image. In stereopsis, depth is directly calculated by triangulation of two corresponding points in two images taken by different cameras under different light conditions.

Two cameras which capture the same scene have a common field of view. A point in the scene inside the common field of view is projected on the images of the two cameras. Due to the different points of view of the cameras, the coordinates of this point in the two images are different. The different coordinates are described in a horizontal and a vertical shift, called the disparities. Disparities can be relative to the coordinates of the image point of the first camera and of the second camera [16].

The main challenge of stereopsis is that it is difficult to find the correspondence of image points within the shared field of view between the two cameras; this is known as the correspondence problem. This problem is often simplified when the camera positions are close together since there is a greater number of overlapping points. However, even though the correspondence problem can be simplified, it is still computationally intense, taking hours rather than minutes like SFS solutions [17].

Some researchers have overcome the correspondence problem by either user input or by overlaying a surface with a known pattern then performing correspondence on the pattern. This is extremely effective as can be seen in work by Paul Debevec with his program called Facade which allows users to construct an underlying model of a building then uses multiple images to fine tune the model [18]. However, this has constrained the problem to the point where user input is required.

While stereopsis provides results that rival range scanners with far less equipment cost, the algorithms still require a significant amount of post processing time that limits its uses in real-time applications like computer vision. However, while performing stereopsis on an entire image would be slow, using stereopsis to solve for depths at individual points is still feasible for use in real-time systems. We present our use of stereopsis to assist in solving the challenge of ambiguity in **Section 3.1**.

2.6.2 Image Segmentation

Image segmentation is typically used to distinguish objects in an image from the background [31]. There are many different types of image segmentation but the four most

popular techniques are: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods.

Threshold techniques are based on local pixel information. If the difference in intensity from one pixel to the next is past a predetermined threshold then an edge is marked between them. These techniques work well when object edges are well defined but fail when object edges blend in with the background.

Edge-based methods rely heavily on edge detection. Once an edge is detected it is then used to segment an image. These methods are again very prone to problems when images are blurred since it is difficult to identify hard edges or breaks in the image.

Region-based methods work by starting at a given pixel then grouping it with neighboring pixels that are of similar intensity values. Once individual regions are created they are then merged by some criterion. If the merging mechanism is too stringent fragmentation occurs. For looser merging mechanisms blurring regions become troublesome. These methods often use the two techniques above to aid the segmentation.

Connectivity-preserving relaxation techniques segment by first generating a spline curve over an initial contour region. It then expands and shrinks the spline curve according to some energy function until that function is minimized and the curve accurately describes an edge. The most common problem with this technique is getting stuck at local minima [31].

Image segmentation methods have proven to be an efficient way to separate an image, and thus the problem domain, into a set of simpler tasks [31]. This advantage will prove to be useful in our later discussion on solving the problem of multiple materials in **Section 3.2**.

Chapter 3

Algorithm

This chapter describes our new algorithm for SFS that expands upon the work of Zeng and colleagues [4], and uses stereopsis and image segmentation. Our algorithm satisfies the following criteria:

1. Captures depth information from a single object using a single light source and two cameras.
2. Captures depth information in real-time.
3. Operates autonomously (i.e. with no user input).
4. Allows for surfaces with multiple local minima and local maxima.
5. Allows for surfaces with multiple materials.
6. Allows for surfaces with specular components.

We describe how each of the three open challenges of SFS will be handled by our new hybrid algorithm then present the details of the algorithm.

3.1 Solving the Challenge of Ambiguity

As we saw with the work of Zeng and colleagues [4], the challenge of the convex versus concave ambiguity can be solved with the help of a user. However, if we want to construct a SFS

technique that can be used in fields where user interaction is not feasible (e.g. computer vision), then we must remove the user from the process.

In Zeng's work, a user specified a series of normals, then those normals were used to determine the location of peaks. This was done by tracing up hill to locate high intensity regions in the input image. These high intensity regions were then marked as peaks [4]. Instead of having the user input surface normals then solving for surface peaks from the input normals, why not have the user indicate peaks directly? As stated before, they choose the input of normals because it is easier for a user to recognize surface orientation instead of indicating depth. However, it is easier for a machine to indicate depths of certain points using stereopsis rather than specifying a handful of normals.

With this in mind, we propose a new SFS technique that would work by first capturing two images of an object from slightly different view angles in order to use stereopsis to solve for depth for only certain pixels. The algorithm then proceeds by first locating the pixels in one of the images that have greater intensities than all of its neighboring pixels in the same image. These pixels are marked as possible peaks. Stereopsis is then used to calculate the depth at each of the marked possible peaks. Finally, FM-SFS is performed from these peaks. The advantage this technique has over the work of Zeng is that the user is eliminated from the process.

To summarize, once the images are captured the algorithm proceeds as follows.

1. Iterate through all intensity values in an image and determine which values are possible peaks. That is to say

for all x, y pixel coordinates in an image

if $\text{Intensity}_{x-1, y} \leq \text{Intensity}_{x, y}$ **and** $\text{Intensity}_{x+1, y} \leq \text{Intensity}_{x, y}$ **and**

Intensity_{x, y-1} <= Intensity_{x, y} **and** Intensity_{x, y+1} <= Intensity_{x, y} **then**

Mark x, y coordinate as possible peak

2. Use stereopsis to determine the altitudes at each of the possible peaks.
3. Perform FM- SFS using the computed depths as initial values into the depth sorted priority queue.

Because this method uses the FM-SFS algorithm in the same way as [4], it would yield the same results that are shown in the work of Zeng and colleagues. The only difference between the Zeng's technique and our technique is the way we determine the initial depths of the input points. Our technique could potentially identify more peaks than [4], however, since peaks will already have their depths computed, these values will not have to be recomputed during FM-SFS speeding up this stage of the algorithm.

The use of stereopsis to remove the user from the SFS problem allows the technique to be used in a wider number of applications. Some results of this autonomous version of SFS can be seen in **Figure 3.1**.

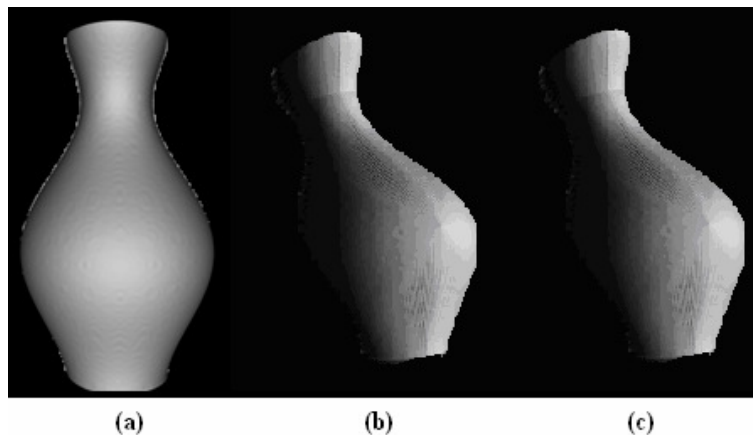


Figure 3.1 Using Stereopsis to Solve the Challenge of Ambiguity: (a) Input image of a vase. (b) Surface constructed through the help of user input. (c) Surface constructed autonomously using stereopsis to determine peak depths.

As can be seen in the figure the surface constructed using stereopsis to determine peak depths is virtually identical to the surface constructed with the help of user input. We now have an autonomous solution to the challenge of ambiguity.

From experimentation the user is useful since the current state of peak detection yields an excessively large number of possible peaks and since correspondence in this algorithm is poor, completely autonomous results are sometimes unreliable. Once peak detection and correspondence algorithms improve this will not be an issue. However, for some objects like a face (shown later in **Section 4**) we use minimal user input to select which points out of the marked possible peaks stereopsis should be run on.

3.2 Solving the Challenge of Multiple Materials

Since most SFS techniques assume the input image is of a continuous surface it is important to have a way to distinguish between different materials on a given surface. Image segmentation provides us a way to separate a surface into multiple regions. We can then modify our SFS algorithm for each region to insure that the transitions between materials are smooth.

The FM-SFS algorithm used by our SFS algorithm relies on an equation that relates intensity values to change in depth. **Equation 2** is repeated below.

$$\Delta z(x, y) = \sqrt{\frac{1}{I^2(x, y)} - 1} \quad (2)$$

Intensity values in this equation can be constrained to range from one to zero. For completely lambertian surfaces this equation works well since one corresponds to a flat region. However, for darker materials the *maximum intensity* is less than one. This poses a problem for

SFS algorithms since an intensity less than one implies that the region is sloped. Our technique solves this problem by shifting the intensities for a given region to always have a range from one to zero. This is done by dividing the intensity of a given pixel by the intensity of the *brightest pixel in the material*. This forces the brightest point of any material to correlate to a flat region.

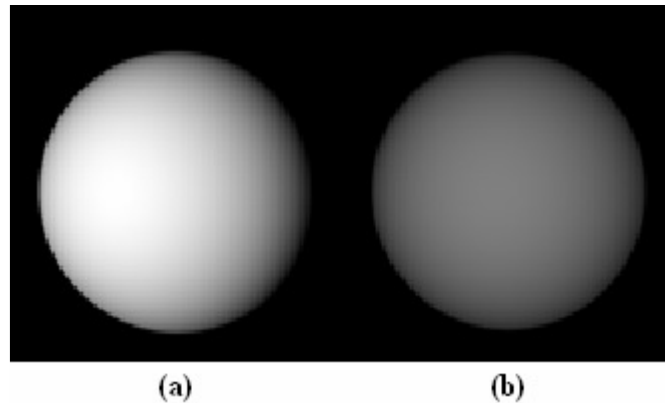


Figure 3.2 Two Images with Different Intensity Ranges: (a) An image of a sphere with intensity values ranging from 1 to 0. (b) An image of the same sphere with intensity values ranging from .75 to 0.

For example, if an image of a sphere like that seen in **Figure 3.2** has an intensity range from .75 to 0 we modify this image by dividing each intensity value by .75 to give the intensities the range from one to zero. After this transformation is complete we can now continue to perform our SFS technique using **Equation 1**.

This preprocessing of our input image can be done for each region of the image to insure that all intensity values on an image are based on a range from one to zero before performing SFS. The process proceeds as follows.

1. Perform image segmentation to separate the image into materials.
2. Loop through all materials computing the brightest point of each material.

3. Identify the global brightest point from all the materials and mark this intensity as the maximum intensity for the material.
4. Then loop through all remaining materials and compute the maximum intensity by inferring it from looking at neighboring materials (Refer to **Section 3.2.1**).
5. Use the brightest point for each material and compute transformed intensity values for the entire image.

Once this process is completed the resulting image will have all intensity values based on a range from one to zero. We can now use our SFS algorithm just as before.

3.2.1 Computing the Maximum Intensity for a Material

Computing the maximum intensity for a material that contains a peak is straight forward. Simply iterate over every pixel to determine the maximum intensity value. However, if a region does not have a peak, the point with the highest intensity value will not correlate to a flat area as when a peak is present.

To determine maximum intensity values for regions without peaks we have to use information about our neighbors. For example,

Preconditions:

1. Region **A** neighbors region **B**.
2. Region **A** has a computed maximum intensity, region **B** does not.
3. Assume that the transition between region **A** and region **B** is continuous.

Algorithm:

1. Find two border points on the transition between region **A** and region **B**.
2. Since the transition between the regions is continuous we can formulate the following constraint about the intensity values of the points.

$$\frac{\text{Border Point on A}}{\text{Brightest Point on A}} = \frac{\text{Border Point on B}}{\text{Brightest Point on B}}$$

3. Determine the ratio between the border point in region **A** and the brightest point on region **A**.
4. Since the intensity value of each border point are known and since the brightest point on **A** is known. We can solve for what the brightest point on **B** would be.

Sample:

Known:

Border point on A = .5

Border point on B = .3

Brightest point on A = 1.0

Brightest point on B = X

Solving for X:

$$\frac{0.5}{1.0} = \frac{0.3}{X}$$

$$0.5X = 1.0(0.3)$$

$$X = 0.6$$

With this preprocessing step we can now eliminate the challenge of multiple materials. This process does add a considerable processing time to our technique but **Section 3.4** will show that this process can be combined into the FM-SFS technique reducing the execution cost. We can even extend this process to use stereopsis to compute depths at points on given materials to determine whether the materials are part of the same object or part of a separate object. This expansion is left for future work.

3.3 Solving the Challenge of Specularity

The challenge of specularity is not that far from the challenge of multiple materials. Like solving for multiple materials we can think of solving for specularity as a preprocessing step. Also like multiple materials, the problem lies within the intensity range.

Again our desired intensity range is from one to zero for a lambertian surface with only a diffuse component in the material. If we are dealing with a material with a specular component, this can cause bright regions that are much larger than would have been perceived without a specular component, our range would be less. An example of this can be seen in **Figure 3.3**.

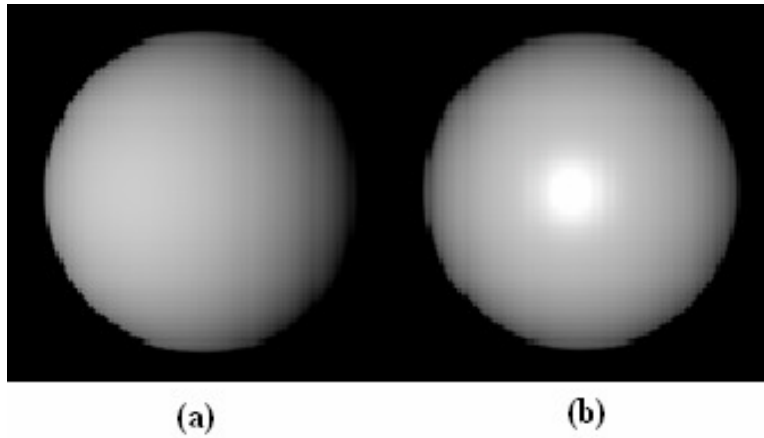


Figure 3.3 Material with and without a Specular Component: (a) An image of a sphere without a specular component with intensity range from .75 to 0. (b) An image of the same sphere with the same diffuse component in the material but with an added specular component causing the brightest point to have an intensity value of 1 instead of .75.

In addition to dealing with multiple materials, we need to adjust for the specular components of the image. Recall from **Figure 1.2** that our intensity function normally contains a specular component.

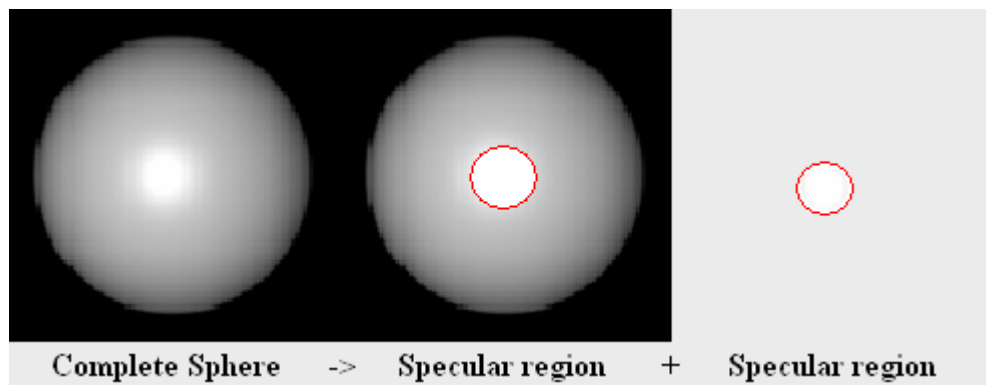


Figure 3.4 Improving the Challenge of Specularity: A diagram of how the region affected by the specular component is separated.

Our solution for handling the specular component is to break the material into two sections. The section affected by the specular component and the section not affected by the specular component. We can then treat these sections as different materials and solve them

separately. While this method is still an approximation, it is an improvement over most SFS techniques which simply ignore specular components.

3.4 Summary of Algorithm

Now that we have determined solutions to each of the challenges that SFS faces we must combine them to form a complete solution to the SFS problem. The following is an outline for the combined technique that takes advantage of all the solutions discussed previously.

Preconditions:

1. The surface to be modeled must be captured into two images from two cameras from slightly different view directions.
2. The distance between the cameras as well as the focal length must be known.
3. There is a single light source placed between the two cameras.
4. The scene must be captured from a far enough distance away such that the light source is effectively at the optical center of each camera.

Algorithm:

1. Process one of the two images for possible peaks as described in **Section 3.1**.

Note: this will still work for multiple materials. Borders will not be a problem since peaks are determined by looking at all neighboring pixels and not just pixels to any one side.

2. Use stereopsis to compute depths of the possible peaks as in **Section 3.1**.
3. Mark each peak as coming from a different material and store the intensity of the pixel as the maximum intensity of the material as discussed in step 3 in **Section 3.2**.

4. Perform FM-SFS with the following modifications:
 - a. When neighboring pixels are placed into the priority queue compute the difference between color values of the computed pixel and the pixel to be inserted into the queue. Check this difference against a threshold performing threshold based image segmentation (refer to **Section 2.6.2**).
 - i. If the difference is greater than the threshold then mark the pixel as part of a new material and compute the maximum intensity for the material as discussed in step 4 in **Section 3.2**.
 - ii. Otherwise mark the pixel to be inserted as coming from the same material as the computed pixel.
 - b. When pixels are removed from the priority queue modify their intensity values to fit in the intensity range of their material as described in **Section 3.2**.

Note: As discussed in **Section 3.3** the handling of multiple materials will handle specular highlights since we can treat specular regions as different materials.

The final result will be a surface constructed of an object that can contain multiple peaks, multiple materials, and specular regions. The accuracy of this technique will be demonstrated in future work using data from range scanners as an oracle.

Chapter 4

Sample Acquisition

The implementation of our algorithm was done in C# using OpenGL. C# was used not for its performance but for its rapid GUI development allowing for increased focus on validating the algorithm. Tests were run on an AMD Athlon XP 3200+ 2.0 GHz processor using 320 x 240 input images. Below are step by step captures of the acquisition of a simulated vase.

1. Two images were captured from a test scenario of a vase containing multiple materials.

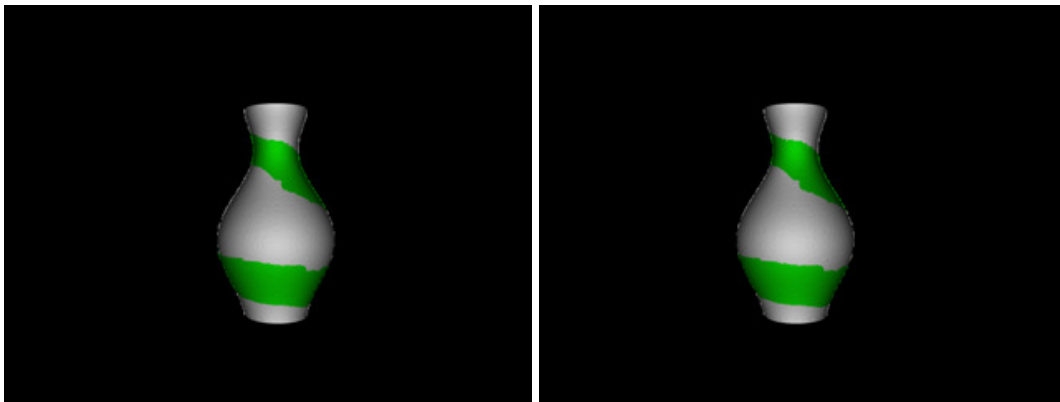


Figure 4.1 Stereo View of Vase: Two captured images from slightly different view directions.

2. Once our input images were captured we ran our autonomous SFS technique with handling of specular highlights and multiple materials disabled. Stereopsis was used on multiple points to compute absolute depth then FM-SFS was executed to construct the surface.

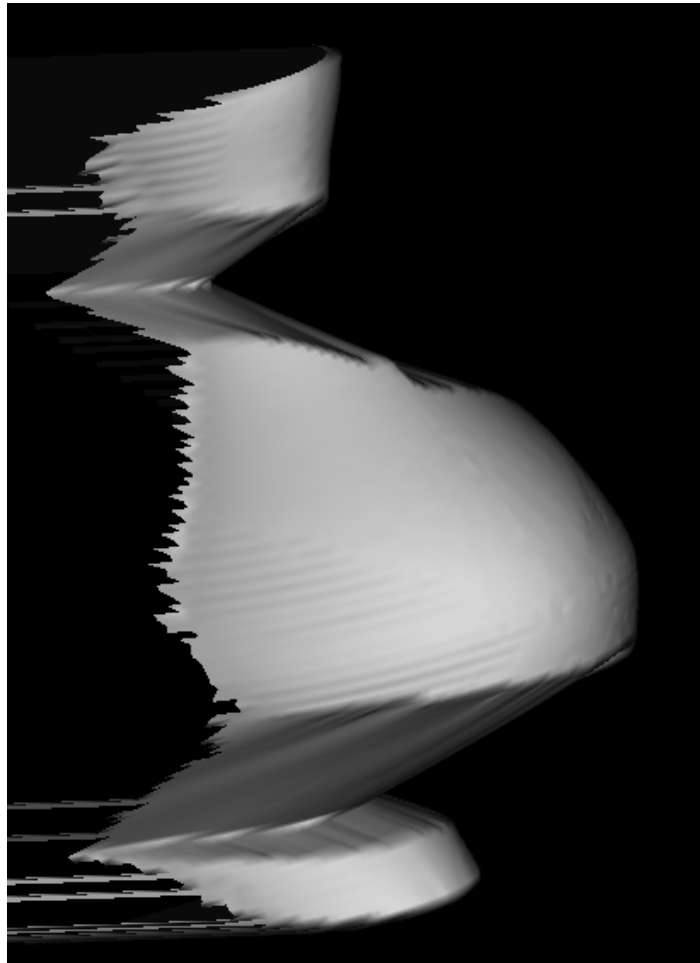


Figure 4.2 Constructed Surface Using FM-SFS: Surface constructed with handling of specular highlights and multiple materials disabled. Surface was constructed in approximately 0.3s.

As can be seen in **Figure 4.2**, without handling multiple materials the green material in the input images produce extremely sloped regions on the constructed surface. This is due to the fact that its intensity range is not based on a one to zero scale.

3. Next, we ran the algorithm again with all features enabled. The final surface was constructed in approximately 0.33s, just slightly slower than with handling of multiple materials disabled.

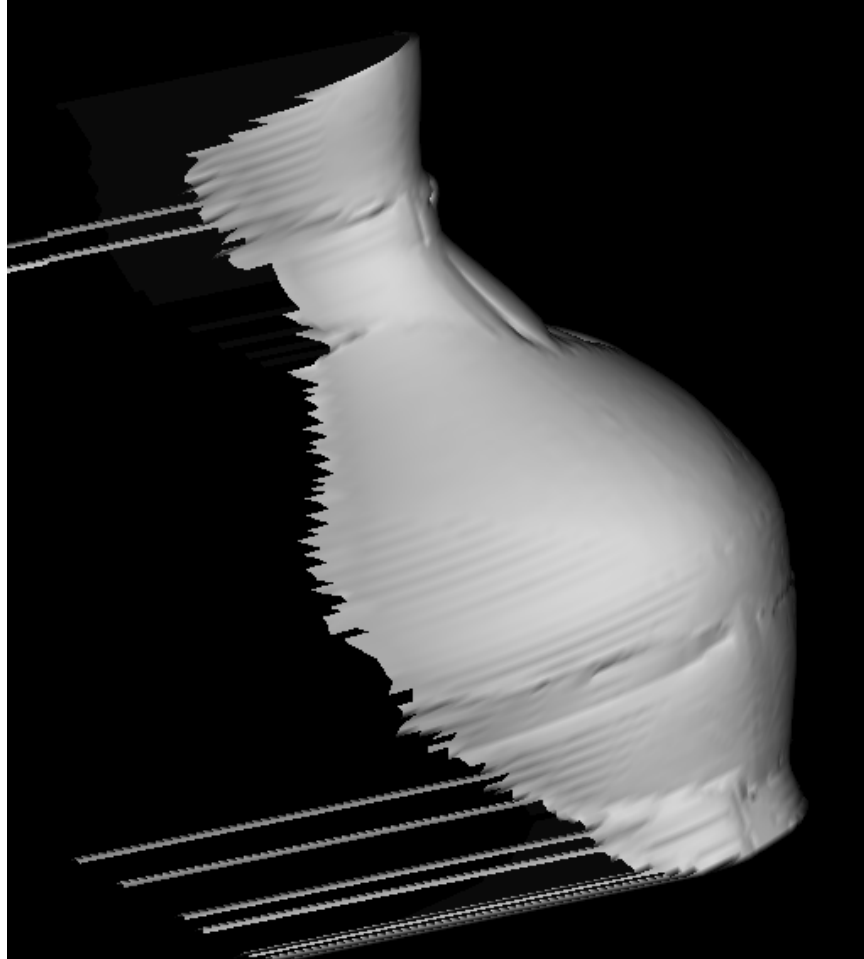


Figure 4.3 Constructed Surface with Handling of Multiple Materials: Surface constructed with handling of specular highlights and multiple materials disabled. Surface was constructed in approximately 0.33s.

Since the handling of multiple materials modified the green materials intensity range to use a one to zero scale, the surface closely resembles the vase where the input images were captured from.

4. Finally, we displayed to segmentation of the surface to determine where our algorithm perceived different materials.

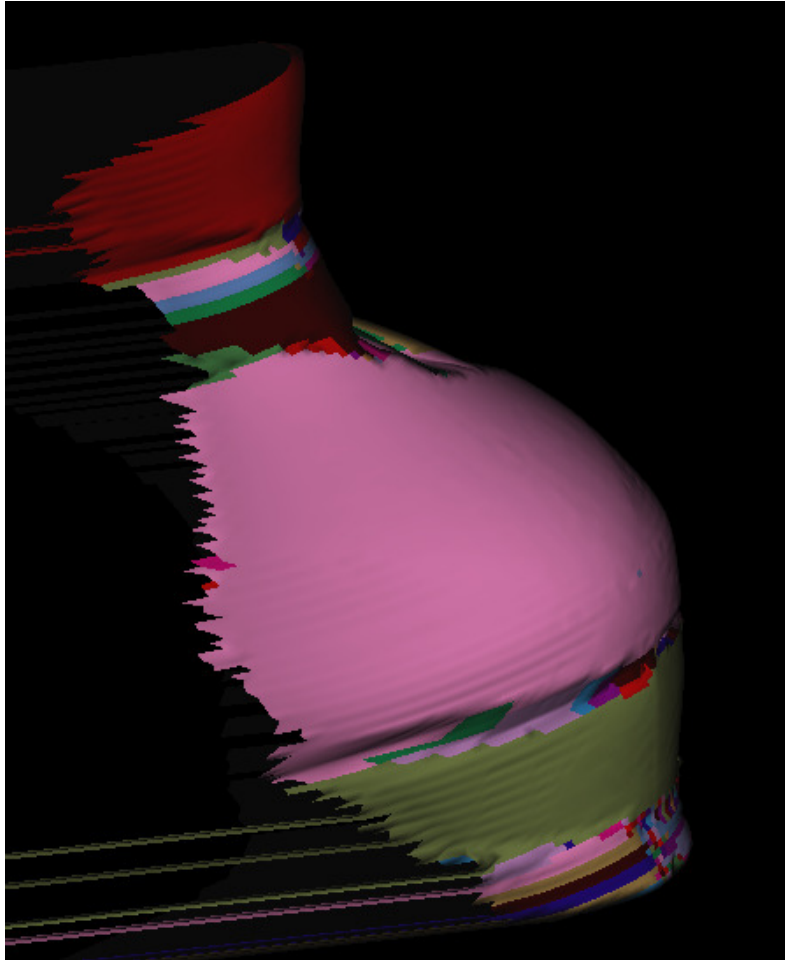


Figure 4.4 Identified Materials on Vase: Surface in Figure 4.3 showing were the algorithm perceived different materials.

It can be seen in **Figure 4.4** that our algorithm perceived many more material changes than actually occurred in the input images. This can be attributed to the threshold image segmentation technique used.

4.1 Additional Acquisitions

In addition to the vase acquisition we provide two more acquisitions with discussion on

there strengths and weaknesses.

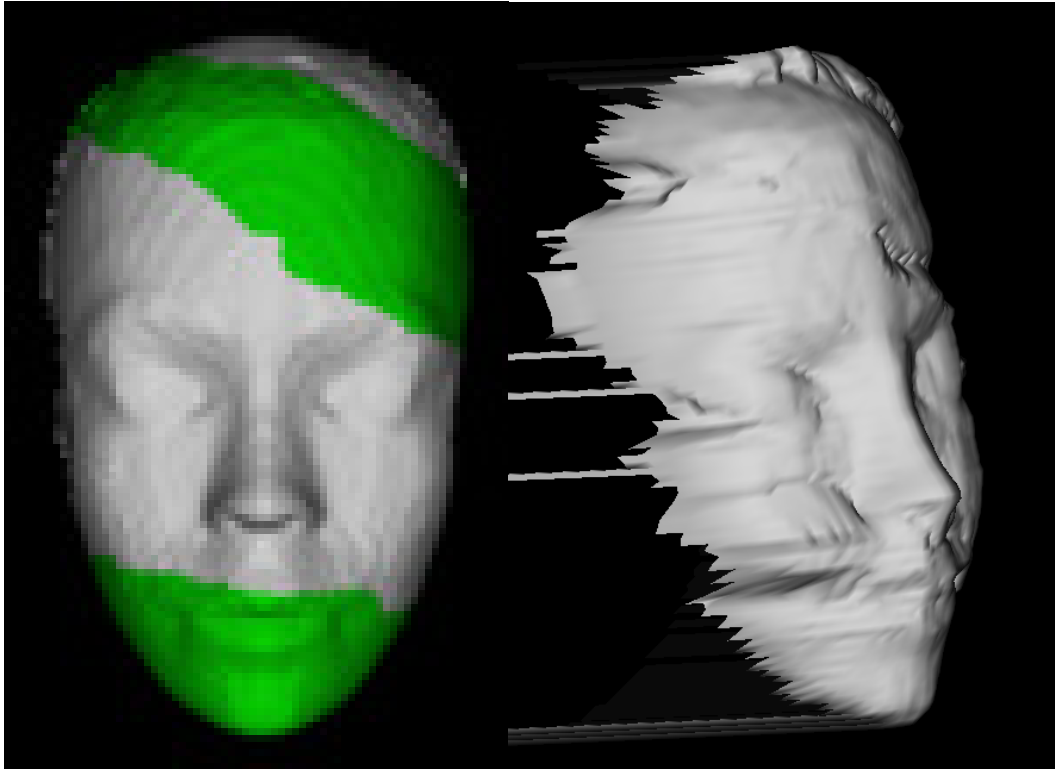


Figure 4.5 Face Acquisition Example: (Left) Input image of a face containing multiple materials. (Right) Constructed surface.

Similar to **Figure 4.3** we can see in **Figure 4.5** various imperfections in the regions surrounding the boundaries of the input image. This problem is caused by two weaknesses with the implementation of our algorithm.

First, since the segmentation technique is integrated into the Fast Marching Method the technique perceives many more materials than it should. When the Fast Marching Method jumps around the surface it marks new materials and there is no process to merge two materials that are the same on different regions of a surface. This could be fixed by changing the segmentation to be a preprocessing step.

The second problem is with the type of image segmentation technique used. As mentioned before, this could be improved by moving from a threshold based technique to an

edge based technique.

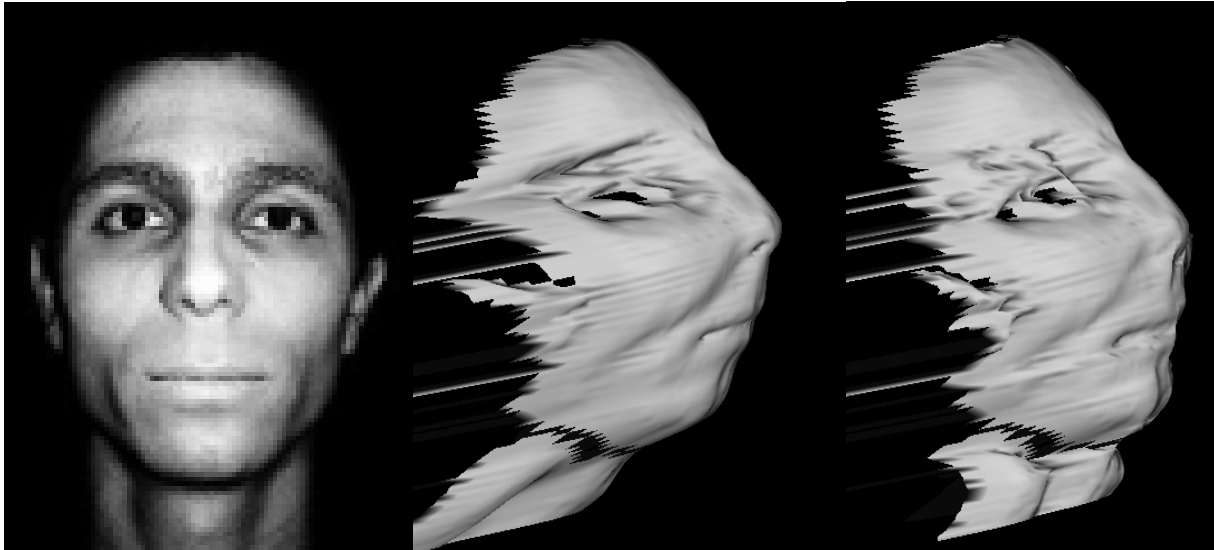


Figure 4.6 Real Image Acquisition Example: (Left) Input image of a real face. (Center) Surface constructed using the FM-SFS (Right) Surface constructed using the FM-SFS with handling of multiple materials.

When running our FM-SFS with multiple materials on real images like in **Figure 4.6** we see an improvement over unmodified FM-SFS but we still suffer from the problems with the image segmentation method used. Our technique more closely resembles that face the image was acquired from, however there still are rough regions around the eye brows caused by the image segmentation technique used.

Initial results of the presented algorithm are promising, but some weaknesses can be seen that leave room for improvement.

Chapter 5

Conclusion and Future Work

We have shown how techniques from other fields can be used to overcome the weakness of SFS. To achieve our goal of an autonomous SFS solution that performs well on real images we have identified the major challenges that occur in real images. We chose to address the challenges of ambiguity, multiple materials, and specularities because previous assumptions that address these problems have been too limiting. Our method uses stereopsis and image segmentation to solve the three open challenges of SFS. We propose a hybrid technique that allows for an autonomous solution to the SFS problem that can handle multiple peaks, multiple materials, and specular components.

Our method performs reasonably well when objects are simple enough that our correspondence succeeds and when breaks between materials are harsh. While we have greatly reduced the reliance on traditional SFS assumptions we have introduced the need for two images rather than one. This restriction is acceptable for newly captured data, but for previous images where only a single image is available the use of stereopsis will not be possible. However, our handling of multiple materials is not coupled with stereopsis and could still be applied to these scenarios.

The following is a list of future work:

1. Validate the accuracy of this technique using data acquired from a Laser Range Scanner as an oracle.
2. Expand the algorithm to handle scenes with multiple objects (refer to **Section 3.2**).
3. Improve the algorithm's correspondence technique. This would improve the algorithms autonomous performance and allow our technique to operate on more complex surfaces.
4. Improve the algorithm's image segmentation technique to use edge segmentation or another more reliable technique.

Once these modifications are made this algorithm will be able to be applied to the entertainment industry, medical fields, and also allow autonomous vision systems to recognize and visualize scenes in full 3D.

Our hybrid approach of combining SFS, stereopsis, and image segmentation will allow us to move away from traditional per pixel stereopsis and move towards a much more time efficient per region stereopsis. The idea of per region stereopsis should be researched further using this work as its foundation.

Bibliography

- [1] R. Zhang, P. S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 690-706, 1993.
- [2] M. Bichsel and A. P. Pentland. A simple algorithm for shape from shading. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 459-465, 1992.
- [3] E. Prados and S Soatto. Fast Marching Method for Generic Shape from Shading. In *Proceedings of VLISM 2005*, pages 320-331, 2005.
- [4] G. Zeng, Y. Matsushita, L. Quan, and H. Y. Shum. Interactive Shape from Shading. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 690-706, August 1999.
- [5] B. Horn. Obtaining shape from shading information. In P. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.
- [6] E. Prados and O. Faugeras, Shape from Shading. *Mathematical Models in Computer Vision: The Handbook*, N. Paragios, Y. Chen, and O. Faugeras (editors), edition 2005, Springer, 2005.
- [7] E. Prados and O. Faugeras. Perspective shape from shading and viscosity solutions. *Proceedings of the Intl. Conference on Computer Vision, volume 2*, pages 826–831, Oct. 2003.
- [8] I. Seong, S. Hideo, and O. Shinji. A Divide-and-conquer Strategy in Shape from Shading Problem. In *Computer Vision and Pattern Recognition*, pages 413–419, 1997.
- [9] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *IJCV*, 35(1):33-44, 1999.

- [10] R. Kimmel and J. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, pages 237-244, 2001.
- [11] A. Tankus, N. Sochen, and Y. Yeshurun. Perspective shape-from-shading using the light source coordinate system. *IEEE Proceedings of Computer Vision and Pattern Recognition (CPVR'04)*, volume 1, pages 43-49, Jun. 2004.
- [12] H. Griffiths. *Surfaces*. Cambridge University Press, 1981.
- [13] J. Durou, M. Falcone, and M. Sagona. A Survey of Numerical Methods for Shape from Shading. Rapport de recherche 2004-2-R, IRIT, janvier 2004.
- [14] B. K. P. Horn, R. S. Szeliski, and A. L. Yuille. Impossible shaded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 166-170, 1993.
- [15] J. P. Lavelle, S. R. Schuet, and D. J. Schuet. High Speed 3D Scanner with Real-Time 3D Processing. 2004 IEEE International Workshop on Imaging Systems and Techniques, pages 13-17. 2004.
- [16] H. Lange. Advances in the cooperation of shape from shading and stereo vision. Second International Conference on 3D Digital Imaging, pages 46-58, 1999.
- [17] G. F. Poggio and T. Poggio. The analysis of stereopsis. *Annual Review of Neuroscience*, pages 7:379-412. 1984.
- [18] P. Debevec. FACADE: modeling and rendering architecture from photographs and the campanile model. *ACM SIGGRAPH 97 Visual Proceedings: The art and interdisciplinary programs of SIGGRAPH '97*, page 254. 1997.
- [19] M. Levoy, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D Scanning of

Large Statues. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 343-352. 2000.

- [20] S. Zhang and P. Huang. High-resolution, Real-time 3D Shape Acquisition. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 3:28–37. 2004.
- [21] R. Szeliski. Fast Shape from Shading. *Computer Vision, Graphics, and Image Processing: Image Understanding*, pages 129–153, March 1991.
- [22] B. Horn. Height and Gradient from Shading. *International Journal of Computer Vision*, pages 37–75, August 1990.
- [23] P. S. Tsai and M. Shah. Shape from Shading Using Linear Approximation. *Image and Vision Computing*, pages 487–498, October 1994.
- [24] B. Horn. Shape from shading: A Method for obtaining the Shape of a Smooth Opaque Object from One View. Height and gradient from shading. PhD thesis, MIT, 1970.
- [25] J. Oliensis. Shape from shading as a partially well-constrained problem. *Computer Vision, Graphics, and Image Processing: Image Understanding*, pages 163-183, 1991.
- [26] R. Kozera. An overview of the shape from shading problem. *Machine Graphics and Vision*, pages 291-312. 1998.
- [27] B. T. Phong. Illumination for computer generated images. *Comm. ACM 18*, pages 311-317, June 1975.
- [28] S. Y. Yuen, Y. Y. Tsui, Y. W. Leung, and R. M. M. Chen. Fast marching method for shape from shading under perspective projection. In *Proceedings of VIIP'02*, pages 584–589, 2002.

- [29] A. Tankus, N. Sochen, and Y. Yeshurun. Perspective Shape-from-Shading by Fast Marching. In *Proceedings of CVPR'04*. 2004.
- [30] A. Ortiz and G. Oliver. Shape from Shading for Multiple Albedo Images. In *Proceedings of the 15th International Conference in Pattern Recognition*, 2000.
- [31] T. Asano, D. Z. Chen, N. Katoh, and T. Tokuyama. Polynomial-time solutions to image segmentation. In *Proceedings of the 7th Ann. SIAM-ACM Conference on Discrete Algorithms*, pages 104-113, January 1996.

Appendix A: Deriving a technique for SFS

In order to gain a greater understanding of SFS and its challenges it is useful to walk through the creation of a SFS technique to see where some of the concepts come from and where some of the challenges arise. The following section will give you a brief walk through of my attempt at creating a propagation approach to the SFS problem.

My SFS technique had its start from one of the basic models of 3D lighting: the Phong model. The Phong model shows us that the shading of an image can be computed by the angle at which light hits a surface [27]. This model also deals with the specularity of a surface but for the simplicity of this example we will assume a lambertian surface and ignore the specular component of his equation.

Phong's basic equation for the intensity value preconceived by a viewer is given by the following equation,

$$I = L \cdot N \quad (1)$$

where I is the intensity perceived, L is the light vector, and N is the normal of the surface at a given point. If we normalize the light vector and normal, and recognize that the dot product is simply the product of the lengths of the two vectors times by the cosine of the angle between them, we can rewrite **Equation 1** to be as follows,

$$I = \cos(\theta) \quad (4)$$

where θ is the angle between the light vector and the normal of the surface. A graphical description of this equation can be seen in **Figure A.1**.

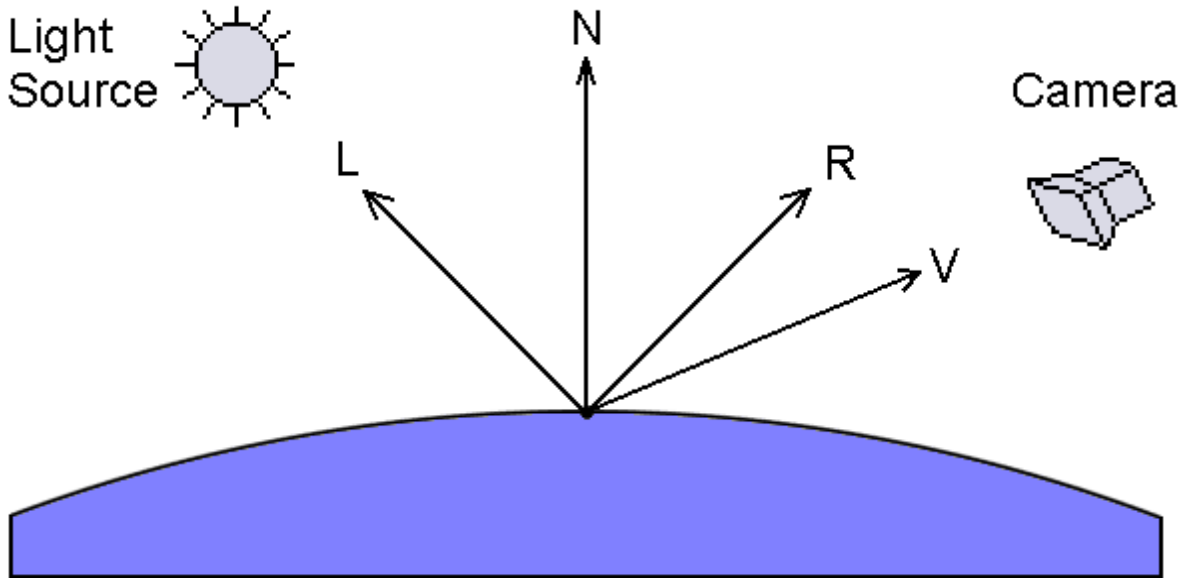


Figure A.1 Phong Model Diagram: Graphical example of the Phong model. L is the light vector, N is the surface normal, R is the reflected vector, and V is the view vector. For the purposes of this paper, R and V are ignored.

If we rewrite **Equation 4** in terms of I , we see that the angle between the light vector and the normal can be computed as follows,

$$\theta = \arccos(I) \quad (5)$$

The angle alone does not help us much, but if we know the light vector we can use the angle to get the normal of the surface. The light vector can either be given, or we can assume, as most SFS methods do, that the light is at the optical center of the camera. Now that we have the light vector we can compute the normal using our computed value for θ .

Once we have our normals we can compute depth by recognizing that our information comes from a square grid of intensity values. If we make the assumption that our surface is

continuous we can compute the depth at any point by using the depth of our neighbor plus the change in depth over the pixel we are currently working on. This change in depth can be computed using some simple geometry.

$$\text{Change in Depth} = \text{Pixel Length} * \tan(\theta)$$

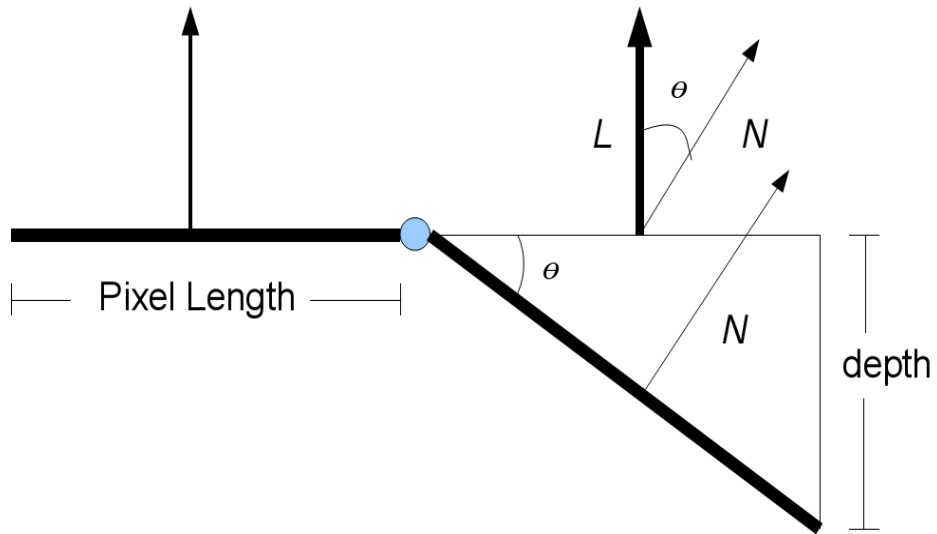


Figure A.2 Calculating Change in Depth: Geometric derivation of change in depth

However, there is still a problem. Given that we know the light vector and the angle between the light and normal we have a continuous circle of possible normals. If we think about the problem in terms of sphere coordinates (refer to **Figure A.3**) we realize that we have only solved for the θ . We still need to find the Φ .

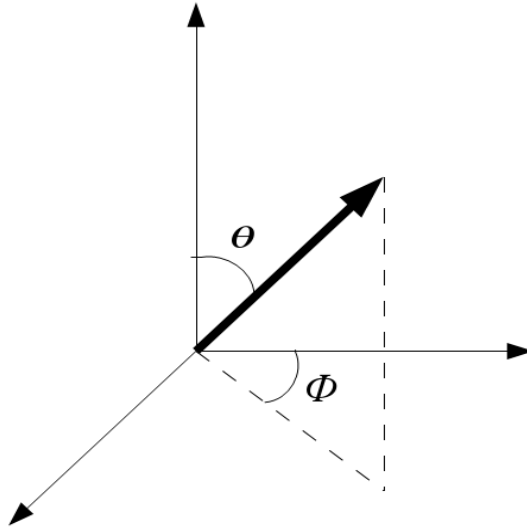


Figure A.3 Polar Coordinates: Illustration of polar coordinates

Now, we could introduce a constraint that if we recognize that our data comes from square pixels on an image, we can constrain Φ based on our neighbors. During our computation of depth we will need to march from some starting point outward. As we are marching we can set our Φ with respect to the direction we are traveling. For example, if we are traveling down an image we can say Φ is -90 degrees with Φ zero when traveling to the right. However, if we remember that we are trying to compute depth and do not really care about our normals we will realize that Φ is not used in the computation of change in depth (refer to **Figure A.2**).

Now it seems we have all the information we need to construct a surface from an image of a lambertian surface. We have computed the important component of our normal in sphere coordinates, the θ , and we have derived a way to compute the depth from this θ . Now our combined solution looks like this,

$$\Delta depth = \tan(\arccos (I)) \quad (6)$$

So, how well did we do? Well if we rewrite the equation we derived we realize that it is exactly the same as the Eikonal equation (**Equation 2**) discussed in **Chapter 2**. A proof of this

can be found in **Appendix C**. Also our downhill marching from a single start point is the exact same technique that FM-SFS uses. So, with some simple reasoning we have derived the state of the art in SFS techniques.

Appendix B: Proof of equivalence between Eikonal Equation and Derived Equation

$$\textit{Derived Equation} = \mathbf{\tan} (\mathbf{\arccos} (I))$$

$$\mathbf{\tan} (\mathbf{\arccos} (I)) = \sqrt{\frac{1 - I^2}{I}}$$

$$\sqrt{\frac{1}{I^2} - \frac{I^2}{I^2}} = \sqrt{\frac{1}{I^2} - 1}$$

$$\sqrt{\frac{1}{I^2} - 1} = \textit{Eikonal Equation}$$