

Shape from Shading: A Survey ^{*}

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer and Mubarak Shah

Computer Vision Lab

School of Computer Science

University of Central Florida

Orlando, FL 32816

shah@cs.ucf.edu

Abstract

Since the first shape-from-shading (SFS) technique was developed by Horn in the early 1970s, many different approaches have emerged. In this paper, six well-known SFS algorithms are implemented and compared. The performance of the algorithms was analyzed on synthetic images using mean and standard deviation of depth (Z) error, mean of surface gradient (p, q) error and CPU timing. Each algorithm works well for certain images, but performs poorly for others. In general, minimization approaches are more robust, while the other approaches are faster. The implementation of these algorithms in C, and images used in this paper, are available by anonymous ftp under the pub/tech_paper/survey directory at eustis.cs.ucf.edu (132.170.108.42). These are also part of the electronic version of paper.

Keywords: Shape from shading, analysis of algorithms, Lambertian model, survey of shape from shading algorithms.

^{*}This work was supported by NSF grants CDA-9122006 and CDA-9222798.

1 Introduction

Shape recovery is a classic problem in computer vision. The goal is to derive a 3-D scene description from one or more 2-D images. The recovered shape can be expressed in several ways: depth $Z(x, y)$, surface normal (n_x, n_y, n_z) , surface gradient (p, q) , and surface slant, ϕ , and tilt, θ . The depth can be considered either as the relative distance from camera to surface points, or the relative surface height above the x-y plane. The surface normal is the orientation of a vector perpendicular to the tangent plane on the object surface. The surface gradient, $(p, q) = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y})$, is the rate of change of depth in the x and y directions. The surface slant, ϕ , and tilt, θ , are related to the surface normal as $(n_x, n_y, n_z) = (l \sin \phi \cos \theta, l \sin \phi \sin \theta, l \cos \phi)$, where l is the magnitude of the surface normal.

In computer vision, the techniques to recover shape are called shape-from-X techniques, where X can be shading, stereo, motion, texture, etc. Shape-from-shading (SFS) deals with the recovery of shape from a gradual variation of shading in the image. Artists have long exploited lighting and shading to convey vivid illusions of depth in paintings. To solve the SFS problem, it is important to study how the images are formed. A simple model of image formation is the Lambertian model, in which the gray level at a pixel in the image depends on the light source direction and the surface normal. In SFS, given a gray level image, the aim is to recover the light source and the surface shape at each pixel in the image. However, real images do not always follow the Lambertian model. Even if we assume Lambertian reflectance and known light source direction, and if the brightness can be described as a function of surface shape and light source direction, the problem is still not simple. This is because if the surface shape is described in terms of the surface normal, we have a linear equation with three unknowns, and if the surface shape is described in terms of the surface gradient, we have a non-linear equation with two unknowns. Therefore, finding a unique solution to SFS is difficult; it requires additional constraints.

This paper is about the comparison and performance analysis of SFS techniques. We have implemented six well-known SFS algorithms, and compared them in terms of timing (CPU time) and accuracy (mean and standard deviation of the depth error and mean of the surface gradient error), in order to analyze the advantages and disadvantages of these approaches.

Two synthetic images with two different light sources each and three real images were used in our experiments. We found that none of the algorithms has consistent performance for all images, since they work well for certain images, but perform poorly for others. In general, minimization approaches are more robust, while the other approaches are faster. The implementation of these algorithms in C, and images used in this paper, are available by anonymous ftp.

2 Literature Review

Shading plays an important role in human perception of surface shape. Researchers in human vision have attempted to understand and simulate the mechanisms by which our eyes and brains actually use the shading information to recover the 3-D shapes. Ramachandran [45] demonstrated that the brain recovers the shape information not only by the shading, but also by the outlines, elementary features, and the visual system's knowledge of objects. The extraction of SFS by visual system is also strongly affected by stereoscopic processing. Barrow and Tenenbaum discovered that it is the line drawing of the shading pattern that seems to play a central role in the interpretation of shaded patterns [2]. Mingolla and Todd's study of human visual system based on the perception of solid shape [30] indicated that the traditional assumptions in SFS—Lambertian reflectance, known light source direction, and local shape recovery—are not valid from psychology point of view. One can observe from the above discussion that human visual system uses SFS differently than computer vision normally does.

Recently, Horn, Szeliski and Yuille [19] discovered that some impossibly shaded images exist, which could not be shading images of any smooth surface under the assumption of uniform reflectance properties and lighting. For this kind of image, SFS will not provide a correct solution, so it is necessary to detect impossibly shaded images.

SFS techniques can be divided into four groups: minimization approaches, propagation approaches, local approaches and linear approaches (see figure 1). Minimization approaches obtain the solution by minimizing an energy function. Propagation approaches propagate the shape information from a set of surface points (e.g., singular points) to the whole image. Local approaches derive shape based on the assumption of surface type. Linear approaches

Minimization	Propagation	Local	Linear
Ikeuchi & Horn(81)	Horn (70)	Pentland (84)	Pentland(88)
Brooks & Horn(85)	Rouy & Tourin(92)	Lee & Rosefeld(85)	Tsai & Shah (92)
Frankot & Chellappa (88)	Dupuis & Oliensis(92)		
Horn(89)	Kimmel & Bruckstein (92)		
Malik & Maydan(89)	Bichsel & Pentland(92)		
Szeliski(91)			
Zheng & Chellappa(91)			
Lee & Kuo(91)			
Leclerc & Bobick(93)			
Vega & Yang(93)			

Figure 1: Four categories of shape from shading approaches. The number in parentheses besides authors' names is the year of publication.

compute the solution based on the linearization of the reflectance map.

2.1 Minimization Approaches

One of the earlier minimization approaches, which recovered the surface gradients, was by Ikeuchi and Horn [21]. Since each surface point has two unknowns for the surface gradient and each pixel in the image provides one gray value, we have an underdetermined system. To overcome this, they introduced two constraints: the brightness constraint and the smoothness constraint. The brightness constraint requires that the reconstructed shape produce the same brightness as the input image at each surface point, while the smoothness constraint ensures a smooth surface reconstruction. The shape was computed by minimizing an energy function which consists of the above two constraints. To ensure a correct convergence, the shape at the occluding boundary was given for the initialization. Since the gradient at the the occluding boundary has at least one infinite component, stereographic projection was used to transform the error function to a different space. Also using these two constraints, Brooks and Horn [5] minimized the same energy function, in terms of the surface normal. Frankot and Chellappa [11] enforced integrability in Brooks and Horn's algorithm in order to recover integrable

surfaces (surfaces for which $z_{xy} = z_{yx}$). Surface slope estimates from the iterative scheme were expressed in terms of a linear combination of a finite set of orthogonal Fourier basis functions. The enforcement of integrability was done by projecting the nonintegrable surface slope estimates onto the nearest (in terms of distance) integrable surface slopes. This projection was fulfilled by finding the closest set of coefficients which satisfy integrability in the linear combination. Their results showed improvements in both accuracy and efficiency over Brooks and Horn’s algorithm [5]. Later, Horn also [18] replaced the smoothness constraint in his approach with an integrability constraint. The major problem with Horn’s method is its slow convergence. Szeliski [48] sped it up using a hierarchical basis pre-conditioned conjugate gradient descent algorithm. Based on the geometrical interpretation of Brooks and Horn’s algorithm, Vega and Yang [51] applied heuristics to the variational approach in an attempt to improve the stability of Brooks and Horn’s algorithm.

Instead of the smoothness constraint, Zheng and Chellappa [54] introduced an intensity gradient constraint, which specifies that the intensity gradients of the reconstructed image and the input image are close to each other in both the x and y directions.

All of the above techniques use variational calculus. Leclerc and Bobick [25] solved directly for depth by using a discrete formulation and employing a conjugate gradient technique. The brightness constraint and smoothness constraint were applied to ensure convergence, and a stereo depth map was used as an initial estimate. Recently, Lee and Kuo [28] also proposed an approach to recover depth using the brightness and the smoothness constraint. They approximated surfaces by a union of triangular patches. This approach did not require the depth initialization.

The approaches described so far deal with a single smooth surface. Malik and Maydan [29] developed a solution for piecewise smooth surfaces. They combined the line drawing and shading constraints in an energy function, and recovered both surface normal and line labeling through the minimization of the energy function.

2.2 Propagation Approaches

Horn’s characteristic strip method [17] is essentially a propagation method. A characteristic strip is a line in the image along which the surface depth and orientation can be computed if these quantities are known at the starting point of the line. Horn’s method constructs

initial surface curves around the neighborhoods of singular points (singular points are the points with maximum intensity) using a spherical approximation. The shape information is propagated simultaneously along the characteristic strips outwards, assuming no crossover of adjacent strips. The direction of characteristic strips is identified as the direction of intensity gradients. In order to get a dense shape map, new strips have to be interpolated when neighboring strips are not close to each other.

Rouy and Tourin [46] presented a solution to SFS based on Hamilton-Jacobi-Bellman equations and viscosity solutions theories in order to obtain a unique solution. A link between viscosity solutions and optimal control theories was given via dynamic programming. Moreover, conditions for the existence of both continuous and smooth solutions were provided.

Oliensis [33] observed that the surface shape can be reconstructed from singular points instead of the occluding boundary. Based on this idea, Dupuis and Oliensis [9, 34] formulated SFS as an optimal control problem, and solved it using numerical methods. Bichsel and Pentland [3] simplified Dupuis and Oliensis’s approach and proposed a minimum downhill approach for SFS which converged in less than ten iterations.

Similar to Horn’s, and Dupuis and Oliensis’s approaches, Kimmel and Bruckstein [23] reconstructed the surface through layers of equal height contours from an initial closed curve. Their method applied techniques in differential geometry, fluid dynamics, and numerical analysis, which enabled the good recovery of non-smooth surfaces. The algorithm used a closed curve in the areas of singular points for initialization.

2.3 Local Approaches

Pentland’s local approach [37] recovered shape information from the intensity, and its first and second derivatives. He used the assumption that the surface is locally spherical at each point. Under the same spherical assumption, Lee and Rosenfeld [26] computed the slant and tilt of the surface in the light source coordinate system using the first derivative of the intensity.

The approaches by Pentland, and Tsai and Shah are linear approaches, which linearize the reflectance map and solve for shape.

2.4 Linear Approaches

Pentland [38] used the linear approximation of the reflectance function in terms of the surface gradient, and applied a Fourier transform to the linear function to get a closed form solution for the depth at each point.

Tsai and Shah [50] applied the discrete approximation of the gradient first, then employed the linear approximation of the reflectance function in terms of the depth directly. Their algorithm recovered the depth at each point using a Jacobi iterative scheme.

Interreflections

None of the above methods deals with interreflections – the mutual illumination between surface facets. Nayar, Ikeuchi, and Kanade [31, 32] addressed the shape-from-interreflection problem using photometric stereo. They observed that the erroneous shape extracted by shape-from-photometric-stereo algorithms in the presence of interreflections was shallower than the real shape. Therefore, they proposed a method to iteratively refine the shape. Similar formulation of interreflection was also discussed by Forsyth and Zisserman [10].

Convergence, Uniqueness and Existence

Little work has been done on proving the uniqueness or existence of a solution to SFS. The uniqueness of SFS can be proven under the condition that the light source direction is equal to, or symmetric around, the viewing direction [33]. With an initial known curve, the method of characteristic strips yields a unique solution if the first derivative of surface depth is continuous. For general cases, the uniqueness is unknown. However, Lee and Kuo [28] showed that, given the depth at a reference point, the addition of the smoothness constraint and successive linearization of the reflectance map (based on the local gradients obtained from the previous iteration) provides a unique solution for their approach, in most cases. Blake & Zisserman, Bruss, and Rouy & Tourin also provided sufficient conditions for uniqueness [4, 6, 46]. Their conditions contain either the singular point or the occluding boundary.

If we consider local uniqueness instead of global uniqueness over the entire image, the uniqueness of a solution can be easily determined at singular points, and occluding boundaries, provided that the reflectance map is given. These are the points at which we can

determine the surface orientation directly from the image brightness. The brightness pattern in any arbitrary region could arise from an infinite number of different surfaces. However, the information at singular points and at occluding boundaries can be used to constrain the possible solutions.

The organization of the paper

The organization of the remainder of this paper is as follows: The next section introduces background knowledge related to reflectance models. Section four describes six SFS approaches studied in detail in this paper. Section five is devoted to the description of the synthetic and real images used in this study. We present our experimental results for six different SFS algorithms in section six. The error analysis is presented in section seven. Section eight summarizes the CPU timing of all six algorithms for different images. Finally, conclusions and future research are covered in section nine.

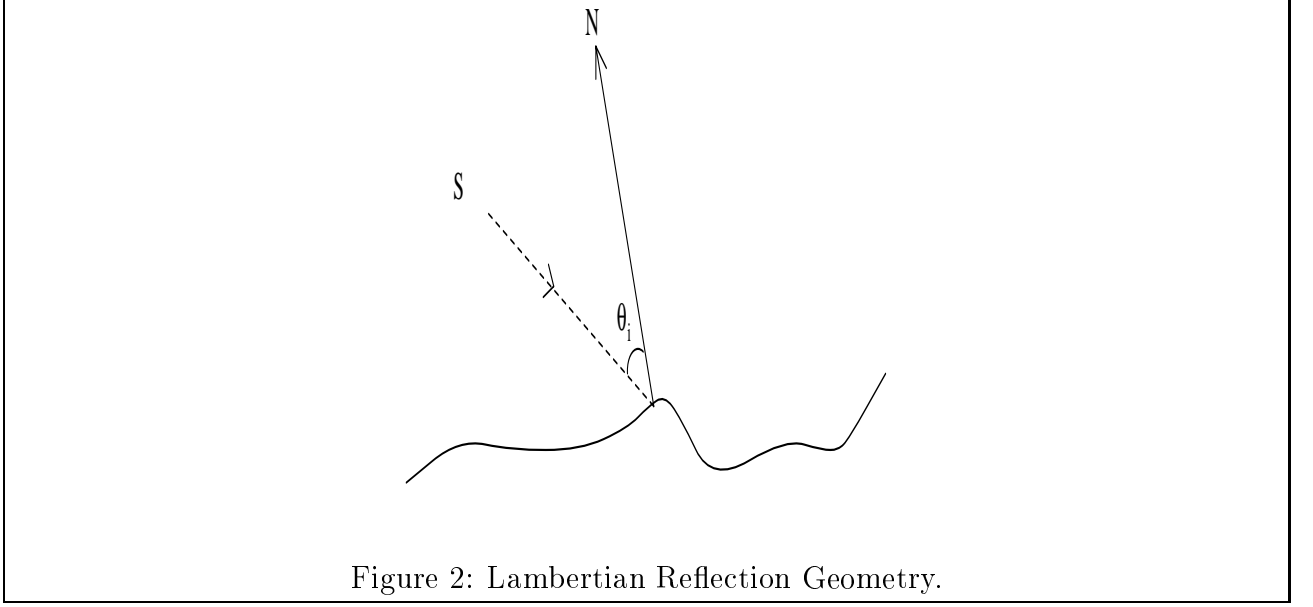
3 Reflectance Models

Depending on their physical properties, surfaces can be categorized as pure Lambertian, pure specular, hybrid, or more sophisticated surfaces. In this section, we will describe the reflectance models and discuss their properties related to shape from shading.

3.1 Lambertian and Specular Reflectance Models

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

$$I_L = R = A\rho \cos \theta_i, \quad (1)$$



where R is the reflectance map and θ_i is the angle between the surface normal $\vec{N} = (n_x, n_y, n_z)$ and the source direction $\vec{S} = (s_x, s_y, s_z)$ (See Figure 2). If we let the surface normal and the light source direction both be unit vectors, we can rewrite the above formula as:

$$I_L = A\rho\vec{N} \cdot \vec{S}, \quad (2)$$

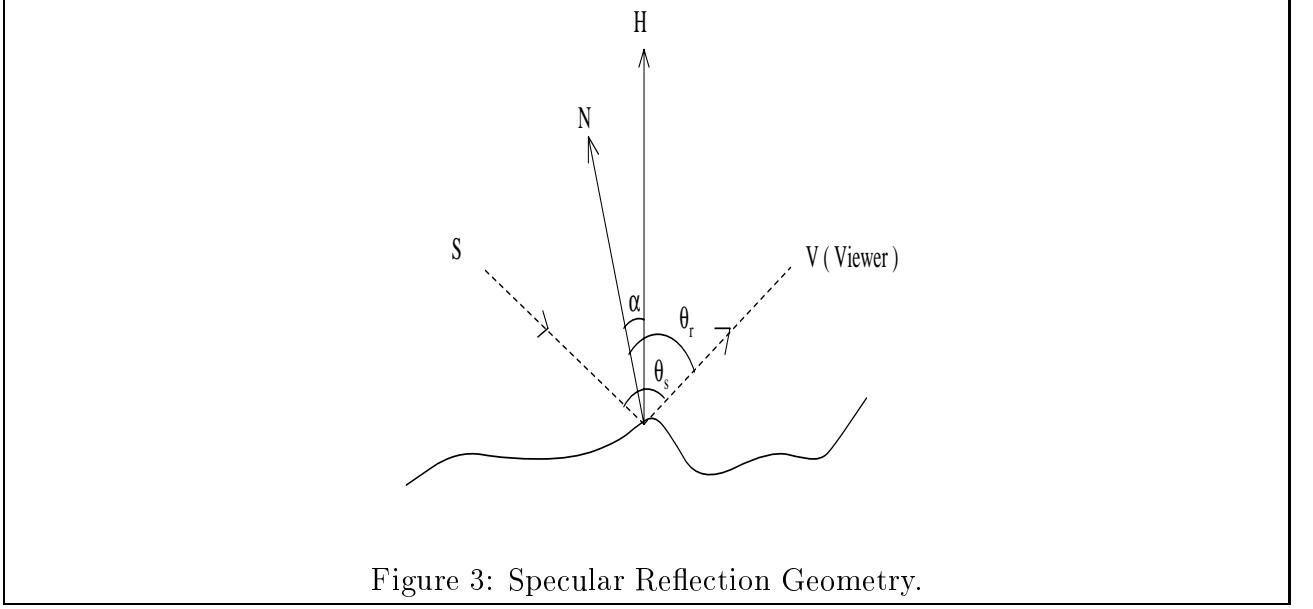
where ' \cdot ' represents dot product.

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

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

$$I_S = B\delta(\theta_s - 2\theta_r), \quad (3)$$

where I_S is the specular brightness, B is the strength of the specular component, θ_s is the angle between the light source direction and the viewing direction, and θ_r is the angle between the surface normal and the viewing direction. This model assumes that the highlight caused by specular reflection is only a single point, but in real life this assumption is not true. Another model developed by Phong [40] represents the specular component of reflection as powers of the cosine of the angle between the perfect specular direction and the viewing direction. This model is capable of predicting specularities which extend beyond



a single point; however, the parameters have no physical meaning. A more refined model, the Torrance-Sparrow model [49], assumes that a surface is composed of small, randomly oriented, mirror-like facets. It describes the specular brightness as the product of four components: energy of incident light, Fresnel coefficient, facet orientation distribution function, and geometrical attenuation factor adjusted for foreshortening. On the basis of the Torrance-Sparrow model, Healey and Binford [15] derived a simplified model by using the Gaussian distribution as the facet orientation function, and considering the other components as constants. It can be described as:

$$I_S = K e^{-(\frac{\alpha}{m})^2}, \quad (4)$$

where K is a constant, α is the angle between the surface normal \vec{N} and the bisector H of the viewing direction and source direction, and m indicates the surface roughness (Figure 3).

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

$$I = (1 - \omega)I_L + \omega I_S, \quad (5)$$

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

Nayar, Ikeuchi and Kanade [32] proposed a reflectance model which consists of three

components: diffuse lobe, specular lobe, and specular spike. The Lambertian model was used to represent the diffuse lobe, the specular component of the Torrance-Sparrow model was used to model the specular lobe, and the spike component of the Beckmann-Spizzichino model was used to describe the specular spike. The resulting hybrid model is given as:

$$I = K_{dl} \cos \theta_i + K_{sl} e^{-\frac{\beta^2}{2\sigma^2}} + K_{ss} \delta(\theta_i - \theta_r) \delta(\phi_r). \quad (6)$$

where K_{dl} , K_{sl} and K_{ss} are the strengths of the three components, β is the angle between the surface normal of a micro-facet on a patch and the mean normal of this surface patch, and σ is its standard derivation. If we consider the surface normal being in the Z direction, then, (θ_i, ϕ_i) is the direction of incidence light in terms of the slant and tilt in 3-D, (θ_r, ϕ_r) is the direction of reflected light.

Although the Lambertian model is widely used because of its simplicity, it is a poor approximation to the diffuse component of rough surfaces. See [20, 24, 7, 35] for more sophisticated approaches.

4 Selected Shape from Shading Algorithms

Most SFS algorithms assume that the light source direction is known. In the case of the unknown light source direction, there are algorithms [36, 26, 54] which can estimate the light source direction without the knowledge of the surface shape. However, some assumptions about the surface shape are required, such as the local spherical surface, uniform and isotropic distribution of the surface orientation. Once the light source direction is known, 3-D shape can be estimated.

We have implemented two minimization, one propagation, one local, and two linear methods. Ease of finding/making an implementation was an important criterion. In addition, this selection was guided by several other reasons. First, we have attempted to focus on more recent algorithms. Since authors improve their own algorithms or others' algorithms, new algorithms, in general, perform better than older algorithms. Second, some papers deal with theoretical issues related to SFS, but do not provide any particular algorithm. Such papers have not been dealt with in detail in this paper. Third, some approaches combine shape from shading with stereo, or line drawings, etc. We have mentioned such approaches in the

review section, for the sake of completeness, but have not discussed them in detail. Finally, since papers related to interreflection and specularly consider special and more complex situations of shape from shading, they have not been dealt with in detail.

4.1 Minimization Approaches

Minimization approaches compute the solution which minimizes an energy function over the 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 unit normal constraint. In this subsection, first, we briefly describe these constraints, and then discuss SFS methods which use these constraints.

The Brightness constraint is derived directly from the image irradiance (Equation 2).

It indicates the total brightness error of the reconstructed image compared with the input image, and is given by

$$\iint (I - R)^2 dx dy, \quad (7)$$

where I is the measured intensity, and R is the estimated reflectance map.

The Smoothness constraint ensures a smooth surface in order to stabilize the convergence to a unique solution, and is given by

$$\iint (p_x^2 + p_y^2 + q_x^2 + q_y^2) dx dy, \quad (8)$$

here p and q are surface gradients along the x and y directions. Another version of the smoothness term is less restrictive by requiring constant change of depth only in x and y directions:

$$\iint (p_x^2 + q_y^2) dx dy. \quad (9)$$

The smoothness constraint can also be described in terms of the surface normal \vec{N} :

$$\iint (\|\vec{N}_x\|^2 + \|\vec{N}_y\|^2) dx dy. \quad (10)$$

This means that the surface normal should change gradually.

The Integrability constraint ensures valid surfaces, that is, $Z_{x,y} = Z_{y,x}$. It can be described by either

$$\iint (p_y - q_x)^2 dx dy, \quad (11)$$

or

$$\iint ((Z_x - p)^2 + (Z_y - q)^2) dx dy. \quad (12)$$

The Intensity Gradient constraint requires that the intensity gradient of the reconstructed image be close to the intensity gradient of the input image in both the x and y directions:

$$\iint ((R_x - I_x)^2 + (R_y - I_y)^2) dx dy. \quad (13)$$

The Unit Normal constraint forces the recovered surface normals to be unit vectors:

$$\iint (\|\vec{N}\|^2 - 1) dx dy. \quad (14)$$

4.1.1 Zheng and Chellappa (91)

Zheng and Chellappa [54] applied the intensity gradient constraint, instead of a smoothness constraint (used by Ikeuchi and Horn [21], Brooks and Horn [5], and Horn [18]), therefore, their energy function became:

$$\iint ((I - R)^2 + ((R_x - I_x)^2 + (R_y - I_y)^2) + \mu((Z_x - p)^2 + (Z_y - q)^2)) dx dy. \quad (15)$$

The minimization of the above function was done through variational calculus. In variational calculus, the Euler equations, which are differential equations, are first computed, and then solved discretely to minimize the energy function. Zheng and Chellappa simplified the Euler equations by taking the first-order Taylor series of the reflectance map and representing the depth, gradient and their derivatives in discrete form. Then, a new iterative scheme was derived, which updates depth and gradients simultaneously. The algorithm was implemented using a hierarchical structure (pyramid) in order to speed up the computation. There was no special requirement for the initialization of the boundary. The initial values for both depth and gradient can be zero.

The implementation of Zheng and Chellappa's method is very straightforward. We used the forward difference approximation to compute the partial derivatives [54]. For the boundary points, where the forward approximation could not be applied, we switched to the backward difference approximation for the first order partial derivatives and set the second order partial derivatives to zero. In all the experiments, we set μ to 1 as suggested by the authors [54].

4.1.2 Lee and Kuo (91)

Lee and Kuo [28] used the brightness constraint and the smoothness constraint. In their approach, surfaces were approximated by the union of triangular surface patches. The vertices of the triangles were called nodal points, and only nodal depths were recovered. Depths at the pixels, which are not nodal points, were obtained through interpolation. For each triangular patch, the intensity of the triangle was taken as the average intensity of all pixels in the triangle, and the surface gradient of the triangle was approximated by the cross product of any two adjacent edges of the triangle. This established a relationship between the triangle's intensity and the depth at its three nodal points. Linearizing the reflectance map in terms of the surface gradient (p, q) , a linear relationship between the intensity and depth at the nodal points was derived. The surface depths at the nodal points were computed using optimization. The optimization problem was reduced to the solution of a sparse linear system, and, a multigrid computational algorithm was applied. There was no initialization required.

Lee and Kuo's algorithm was implemented using the V-cycle multigrid scheme to solve the linear system, as reported in [28]. We used Gauss-Seidel relaxation as the smoothing operator, and as the exact solver for the finest grid. Full-weighting restriction was applied to transfer the residual from finer grids to coarser grids, and bi-linear interpolation was applied to make the prolongation from the coarser grid to finer grids. The same template was used for the smoothness term as given in their paper. The nodal points in the finest grids were chosen to be the image pixels. Successive linearizations were done through a maximum of 10 successive iterations, and the number of V-cycles was set to 10 for the first iteration, 2 for the second, 1 for the rest. The initial values for depth of the finest grid, and corrections for the coarser grids, were all zero. Since the algorithm does not work for light source direction $(0, 0, 1)$, we used $(0.001, 0.001, 1.0)$ as the input light source direction instead.

We used smoothing factor of 2000, and computed the level of grids by $L = \log(M) - 1$, where M is the size of the image. Therefore, we have 7 levels for 256 by 256 images, and 6 levels for 128 by 128 images. The depth maps, after the first iteration, contain more detail but have a smaller range. After 10 iterations, details are smoothed out, but the depth range is wider. This means that more iterations will provide more low frequency information,

which overtakes the high frequency information from the initial iterations.

4.2 Propagation Approaches

Propagation approaches start from a single reference surface point, or a set of surface points where the shape either is known or can be uniquely determined (such as singular points), and propagate the shape information across the whole image. We discuss one algorithm in this section.

4.2.1 Bichsel and Pentland (92)

Following the main idea of Dupuis and Oliensis, Bichsel and Pentland [3] developed an efficient minimum downhill approach which directly recovers depth and guarantees a continuous surface. Given initial values at the singular points (brightest points), the algorithm looks in eight discrete directions in the image and propagates the depth information away from the light source to ensure the proper termination of the process. Since slopes at the surface points in low brightness regions are close to zero for most directions (except the directions which form a very narrow angle with the illumination direction), the image was initially rotated to align the light source direction with one of the eight directions. The inverse rotation was performed on the resulting depth map in order to get the original orientation back. Assuming the constraint of parallel slope, the surface gradient, (p, q) , was pre-computed by taking the derivative of the reflectance map with respect to q in the rotated coordinate system, setting it to zero, and then solving for p and q . The solutions for p and q were given by:

$$p = \frac{-s_x s_z \pm \sqrt{(1 - R^2)(R^2 - s_y^2)}}{R^2 - s_x^2 - s_y^2}, \quad (16)$$

$$q = \frac{p s_y s_x - s_y s_z}{R^2 - s_y^2}. \quad (17)$$

where (s_x, s_y, s_z) is the light source direction and R is the reflectance map as previously defined.

In the implementation of Bichsel and Pentland's algorithm the initial depth values for the singular points were assigned a fixed positive value (55 in our case; this number should be related to the maximum height of object), and the depth values for the other points

were initialized to large negative values (-1.0e10). Instead of computing the distance to the light source, only the local surface height is computed and maximized, in order to select the minimum downhill direction. This is based on the fact that the distance to the light source is a monotonically increasing function of the height when the angle between the light source direction and the optical axis (z-axis here) is less than 90 degrees. Height values are updated with a Gauss-Seidel iterative scheme and the convergence is accelerated by altering the direction of the pass at each iteration.

4.3 Local Approaches

Local approaches derive the shape by assuming local surface type. They use the intensity derivative information and assume spherical surface. Here, we describe Lee and Rosenfeld's [26] approach.

4.3.1 Lee and Rosenfeld (85)

Lee and Rosenfeld [26] approximated the local surface regions by spherical patches. The slant and tilt of the surface were first computed in the light source coordinate, then transformed back to the viewer coordinate. They proved that the tilt of the surface normal could be obtained from:

$$\tau = \arctan \frac{I_y \cos \tau_S - I_x \sin \tau_S}{I_x \cos \tau_S \cos \sigma_S + I_y \cos \sigma_S \sin \tau_S}, \quad (18)$$

where I_x and I_y are intensity derivatives along the x and y directions, σ_S is the slant of the light source, and τ_S is the tilt of the light source.

If the surface has uniform reflectance, and if the reflectance map is given by $I = \rho \vec{N} \cdot \vec{S}$ (Lambertian), then the brightest point has its surface normal pointing toward the light source, and the cosine value of surface slant can be obtained by the ratio of its intensity and ρ .

This approach is an improvement of Pentland's first approach, since it involves only the first derivatives of the intensity rather than the second derivatives. This makes it less sensitive to noise. However, the local spherical assumption of the surface limits its application.

The major part in the implementation of Lee and Rosenfeld's algorithm is the rotation of the image from the viewer coordinates to the light source coordinates, and the computation

of the intensity gradient in the light source coordinates. For this method, there are no parameters which needed to be determined.

4.4 Linear Approaches

Linear approaches reduce the non-linear problem into a linear through the linearization of the reflectance map. The idea is based on the assumption that the lower order components in the reflectance map dominate. Therefore, these algorithms only work well under this assumption.

4.4.1 Pentland (88)

Pentland's second approach [38] used the linear approximation of the reflectance map in p and q . The reflectance function can be expressed as follows:

$$I(x, y) = R(p, q) = \frac{\cos \sigma_S + p \cos \tau_S \sin \sigma_S + q \sin \tau_S \sin \sigma_S}{\sqrt{1 + p^2 + q^2}}, \quad (19)$$

where σ_S and τ_S are respectively the slant and tilt of light source. By taking the Taylor series expansion of the reflectance function about $p = p_0$, $q = q_0$, and ignoring the high order terms, we have

$$I(x, y) = R(p_0, q_0) + (p - p_0) \frac{\partial R}{\partial p}(p_0, q_0) + (q - q_0) \frac{\partial R}{\partial q}(p_0, q_0). \quad (20)$$

For Lambertian reflectance, the above equation at $p_0 = q_0 = 0$, reduces to

$$I(x, y) = \cos \sigma_S + p \cos \tau_S \sin \sigma_S + q \sin \tau_S \sin \sigma_S.$$

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

$$\frac{\partial}{\partial x} Z(x, y) \longleftrightarrow F_Z(\omega_1, \omega_2)(-i\omega_1) \quad (21)$$

$$\frac{\partial}{\partial y} Z(x, y) \longleftrightarrow F_Z(\omega_1, \omega_2)(-i\omega_2), \quad (22)$$

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

$$F_I = F_Z(\omega_1, \omega_2)(-i\omega_1) \cos \tau_S \sin \sigma_S + F_Z(\omega_1, \omega_2)(-i\omega_2) \sin \tau_S \sin \sigma_S, \quad (23)$$

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

This algorithm gives a non-iterative, closed-form solution using Fourier transform. The problem lies in the linear approximation of the reflectance map, which causes trouble when the non-linear terms are large. As pointed out by Pentland, when the quadratic terms in the reflectance map dominate, the “frequency doubling” occurs, in this case, the recovered shape will not be consistent with the illumination conditions.

The implementation of Pentland’s algorithm was done using the fast Fourier transform and the inverse fast Fourier transform (see equation 23).

4.4.2 Tsai and Shah (92)

Tsai and Shah [50] employed the discrete approximations of p and q using finite differences in order to linearize the reflectance map in terms of Z . The reflectance function for Lambertian surfaces is:

$$R(p_{i,j}, q_{i,j}) = \frac{-s_x p_{i,j} - s_y q_{i,j} + s_z}{\sqrt{1 + p_{i,j}^2 + q_{i,j}^2}}, \quad (24)$$

Using the following discrete approximations for p and q , $p = Z_{i,j} - Z_{i-1,j}$ and $q = Z_{i,j} - Z_{i,j-1}$, the reflectance equation can be rewritten as:

$$0 = f(I_{i,j}, Z_{i,j}, Z_{i-1,j}, Z_{i,j-1}) = I_{i,j} - R(Z_{i,j} - Z_{i-1,j}, Z_{i,j} - Z_{i,j-1}). \quad (25)$$

For a fixed point (i, j) and a given image I , a linear approximation (Taylor series expansion up through the first order terms) of the function f (Equation (25)) about a given depth map Z^{n-1} is:

$$\begin{aligned} 0 &= f(I_{i,j}, Z_{i,j}, Z_{i-1,j}, Z_{i,j-1}) \\ &\approx f(I_{i,j}, Z_{i,j}^{n-1}, Z_{i-1,j}^{n-1}, Z_{i,j-1}^{n-1}) + (Z_{i,j} - Z_{i,j}^{n-1}) \frac{\partial}{\partial Z_{i,j}} f(I_{i,j}, Z_{i,j}^{n-1}, Z_{i-1,j}^{n-1}, Z_{i,j-1}^{n-1}) \\ &\quad + (Z_{i-1,j} - Z_{i-1,j}^{n-1}) \frac{\partial}{\partial Z_{i-1,j}} f(I_{i,j}, Z_{i,j}^{n-1}, Z_{i-1,j}^{n-1}, Z_{i,j-1}^{n-1}) \\ &\quad + (Z_{i,j-1} - Z_{i,j-1}^{n-1}) \frac{\partial}{\partial Z_{i,j-1}} f(I_{i,j}, Z_{i,j}^{n-1}, Z_{i-1,j}^{n-1}, Z_{i,j-1}^{n-1}). \end{aligned} \quad (26)$$

For an M by M image, there are M^2 such equations, which will form a linear system. This system can be solved easily using the Jacobi iterative scheme, which simplifies equation (26) into the following equation:

$$0 = f(Z_{i,j}) \approx f(Z_{i,j}^{n-1}) + (Z_{i,j} - Z_{i,j}^{n-1}) \frac{d}{dZ_{i,j}} f(Z_{i,j}^{n-1}). \quad (27)$$

Then for $Z_{i,j} = Z_{i,j}^n$, the depth map at the n -th iteration, can be solved directly:

$$Z_{i,j}^n = Z_{i,j}^{n-1} + \frac{-f(Z_{i,j}^{n-1})}{\frac{d}{dZ_{i,j}}f(Z_{i,j}^{n-1})}. \quad (28)$$

The initial estimate of $Z_{i,j}^0$ is set to zero for all pixels, Gaussian smoothing is applied to the final depth map to get a smoother result. Note that depth is computed here by a simple division without any matrix inverse (Equation 28). This is a simple but efficient algorithm. However, it has problems with self-shadows.

The implementation of Tsai and Shah’s algorithm is straightforward. The depth is updated iteratively using equation 28. However, special care is needed to avoid division by zero [50].

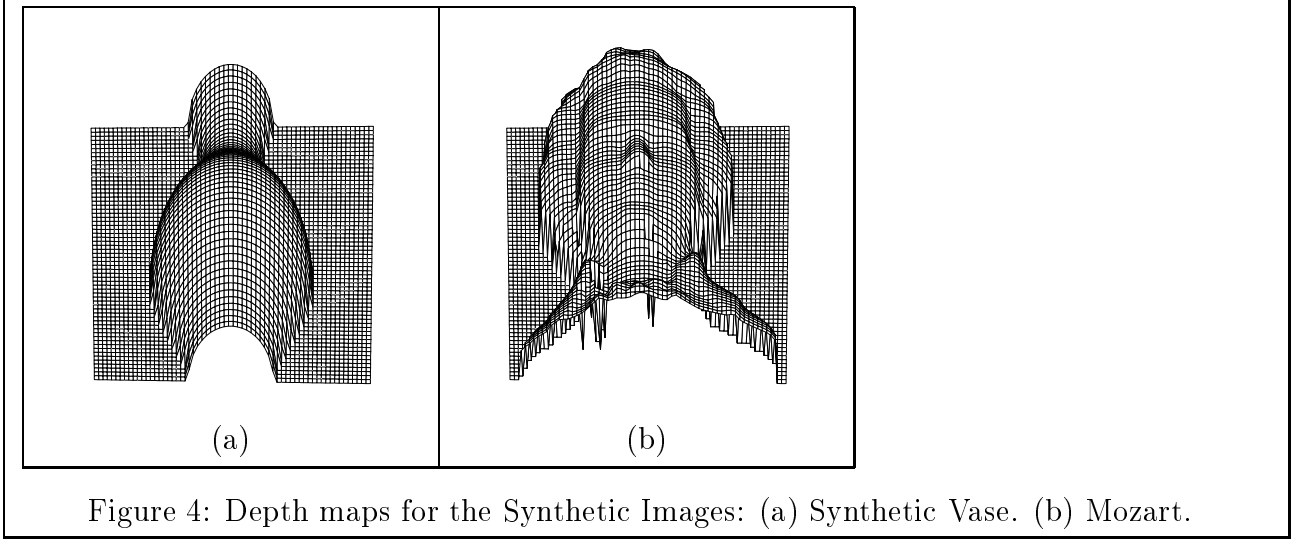
5 Experimental Images

It is very difficult to choose good test images for SFS algorithms. A good test image must match the assumptions of the algorithms, e.g. Lambertian reflectance model, constant albedo value, and infinite point source illumination. It is not difficult to satisfy these assumptions for synthetic images, but no real scene perfectly satisfies these conditions. In real images there will be errors to the extent that these assumptions are not matched. In this section, we describe the images chosen to test the SFS algorithms.

5.1 Synthetic Images

The synthetic images were generated using true depth maps, or range data obtained from a laser range finder. We simply computed the surface gradient ($p = \frac{\partial Z}{\partial x}$, $q = \frac{\partial Z}{\partial y}$) using the forward discrete approximation of the depth, Z , and generated shaded images using the Lambertian reflectance model. There are at least two advantages of using synthetic images. First, we can generate shaded images with different light source directions for the same surface. Second, with the true depth information, we can compute the error and compare the performance. However, the disadvantage of using synthetic images is that performance on synthetic images can not be used reliably to predict performance on real images.

In our study we used five synthetic surfaces (**Synthetic Vase**, **Sphere**, **Mozart**, **Penny** and **Sombrero**), with three different light sources $((0, 0, 1), (1, 0, 1), (5, 5, 7))$. These images



are available through our web site. In this paper, we focus our discussion on **Synthetic Vase** and **Mozart** with two light sources $(0, 0, 1)$ and $(1, 0, 1)$. See [53] for detailed discussion on all five images.

The **Synthetic Vase** was generated using the formula provided by Ascher and Carter [1] as follow:

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

where

$$\begin{aligned} f(y) &= 0.15 - 0.1 * y * (6y + 1)^2 * (y - 1)^2 * (3y - 2), \\ -0.5 &\leq x \leq 0.5 , \text{ and } 0.0 \leq y \leq 1.0 . \end{aligned}$$

This yields a maximum depth value of approximately 0.29. In order to generate a depth map with the proper size and scale, we map the x and y ranges to $[0, 127]$, and scale Z by a factor of 128.

The depth map for **Mozart** was provided by Professor Kuo of USC.

The depth maps for both synthetic surfaces are shown in Figure 4. The synthetic images generated from these surfaces, with light source directions $(0, 0, 1)$, and $(1, 0, 1)$ are shown in Figure 5. The images are either reduced or expanded to make their sizes power of 2.

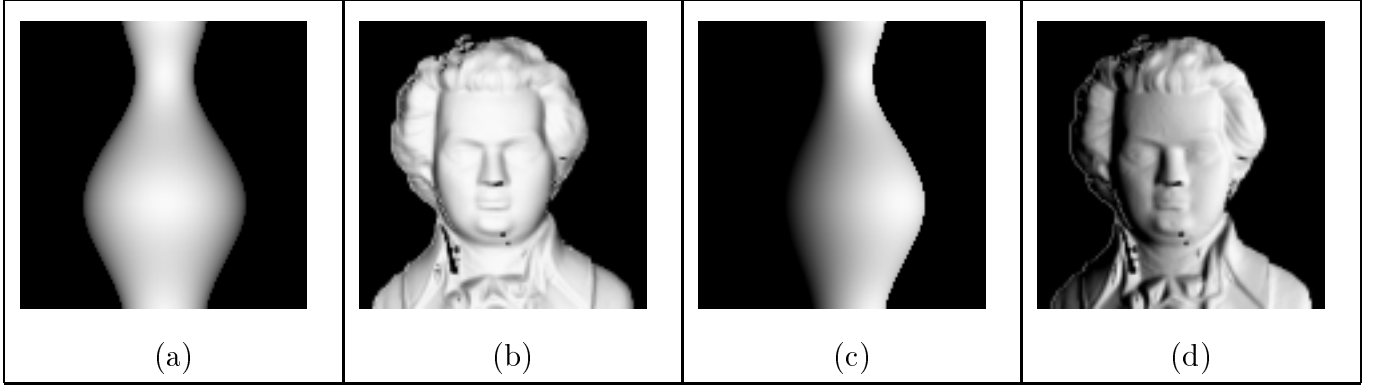


Figure 5: Synthetic images generated using two different light sources: (a) Synthetic Vase $(0, 0, 1)$. (b) Mozart $(0, 0, 1)$. (c) Synthetic Vase $(1, 0, 1)$. (d) Mozart $(1, 0, 1)$.

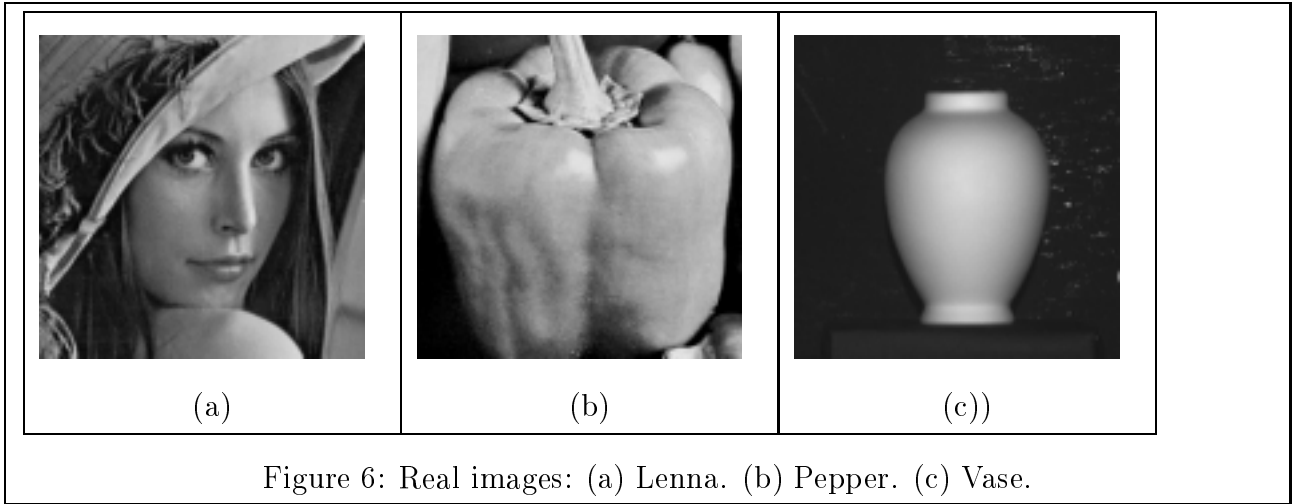


Figure 6: Real images: (a) Lenna. (b) Pepper. (c) Vase.

5.2 Real Images

In our study we used five real images (**Vase**, **Lenna**, **Pepper**, **Mannequin** and **Vase**). These images are available through our web site. In this paper, we focus our discussion on **Lenna**, **Pepper** and **Vase**, shown in Figure 6. See [53] for detailed discussion on all five images.

The light source direction for Vase was estimated by the Lee and Rosenfeld method, and the rest were provided with the images:

- **Lenna:** Light source direction is $(1.5, 0.866, 1)$.
- **Pepper:** Light source direction is $(0.766, 0.642, 1)$.
- **Vase:** Light source direction is $(-0.939, 1.867, 1.0)$.

The **Vase** image was provided by Professor Woodham of UBC. The other two images were provided by Professor Kuo of USC.

Note that the light source direction is one possible source of uncertainty in these algorithms.

6 Experimental Results

It is difficult to choose the proper control parameters for each algorithm because they may vary from image to image. It is even harder to choose similar parameters for different algorithms. We manually selected parameters for each algorithm in order to obtain the best results. This causes the variation of parameters for different algorithms and different images. All algorithms were treated equally when tuning the parameters.

With respect to the initial conditions for different algorithms, we tried to provide the same initial condition for all the algorithms in order to obtain a fair comparison. Therefore, only singular points (points with the highest gray level) were used in the initialization of Bichsel and Pentland's algorithm, since singular points can be automatically detected.

The depth maps computed from both synthetic and real images using each algorithm are shown in Figures 7 through 18. Below we analyze the results using the 3-D plots of depth maps.

6.1 Zheng and Chellappa

The results for synthetic images are shown in Figure 7, and the results for real images are shown in Figure 13. The basic shapes for Lenna and Pepper are recovered with enough details, however, some errors can be seen around the mouth and on the cheeks in Lenna. Their method also has a problem with light source $(0, 0, 1)$, which will zero out most of the terms in the iterative equations ([54] page 688). In order to get reasonable results for the images with light source $(0, 0, 1)$, we used $(0.01, 0.01, 1)$ instead as the light source direction. Their results also showed some error along the light source direction.

6.2 Lee and Kuo

The results for synthetic images are shown in Figure 8, and the results for real images are shown in Figure 14. It can be seen that the algorithm performs ok, even when the light source is from the side, except in the cases of Vase which create the self-shadows. The recovered surfaces are well outlined, but lack details and have a tendency to be over-smoothed. Although different smoothing factors can be used for different images in order to get the best results, small changes in the smoothing factor will not affect the results very much.

6.3 Bichsel and Pentland

The results for synthetic images are shown in Figure 9, and the results for real images are shown in Figure 15. All results were obtained after 8 iterations. The algorithm provides the best results for the cases when the light source is on the side. However, the algorithm does not give good results for real images considered in this study except for Pepper. This, we think, is due to the inaccuracy of the initial singular points. The algorithm is very fast; usually only 5 iterations are required to provide reasonable results.

6.4 Lee and Rosenfeld

The results for synthetic images are shown in Figure 10, and the results for real images are shown in Figure 16. Their method estimates the depth using local spherical assumption and intensity derivatives. This makes the algorithm unsuitable for non-spherical surfaces, and very sensitive to noise, which is clear from the depth maps obtained for the real images and some synthetic images, such as Mozart. The intensity of the real images varies slightly, causing the depth estimation to falter, while the synthetic images yield good depth maps, due to the smooth surfaces. However, the results can be improved by pre-filtering the input images in order to reduce the noise.

6.5 Pentland

The results for synthetic images are shown in Figure 11, and the results for real images are shown in Figure 17. Pentland's algorithm produces reasonable results on most surfaces

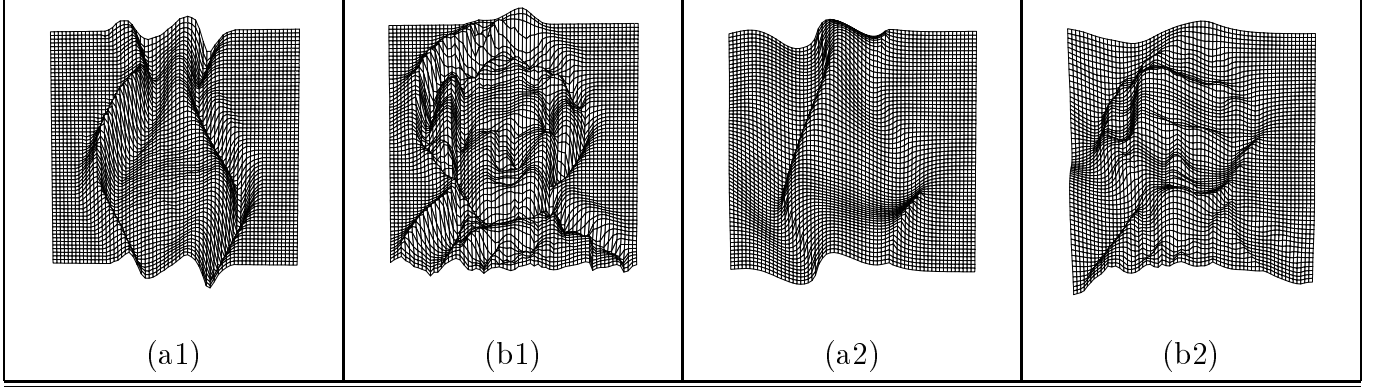


Figure 7: Results for Zheng and Chellappa’s method on synthetic images: (a) Vase. (b) Mozart. (a1) and (b1) show the results for test images with light source $(0, 0, 1)$. (a2) and (b2) show the results for test images with light source $(1, 0, 1)$.

where the reflectance changes linearly with respect to the surface shape, even if the surface is naturally varying such as a person’s face. However, this algorithm has difficulty when the reflectance changes in a non-linear manner. For real images, the algorithm produces adequate results for all images except for Vase.

6.6 Tsai and Shah

The results for synthetic images are shown in Figure 12, and the results for real images are shown in Figure 18. From Figure 12, we can clearly see that their method works well on smooth objects with the light source close to the viewing direction. However, it is sensitive to the noise, such as the black hole on the nose of Mozart image or the shadow areas.

The results for real images are reasonable for Vase, but noisy for Lenna and Pepper, especially in the top and bottom regions of Pepper, and the nose, eyes, and hat regions of Lenna.

7 Error Analysis

In the previous section, we reported results for synthetic and real images, and qualitatively analyzed the results by considering the 3-D plots of the depth map. In this section, we will quantitatively analyze the results for the synthetic images for which the true depth maps are available to be used in error measurements. There are several ways to report the error

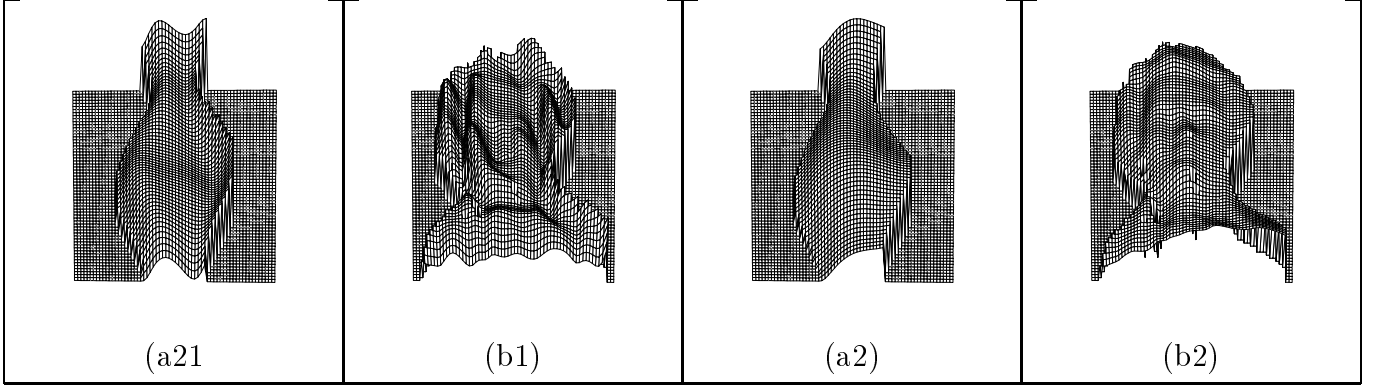


Figure 8: Results for Lee and Kuo's method on synthetic images: (a) Vase. (b) Mozart. (a1) and (b1) show the results for test images with light source $(0,0,1)$. (a2) and (b2) show the results for test images with light source $(1,0,1)$.

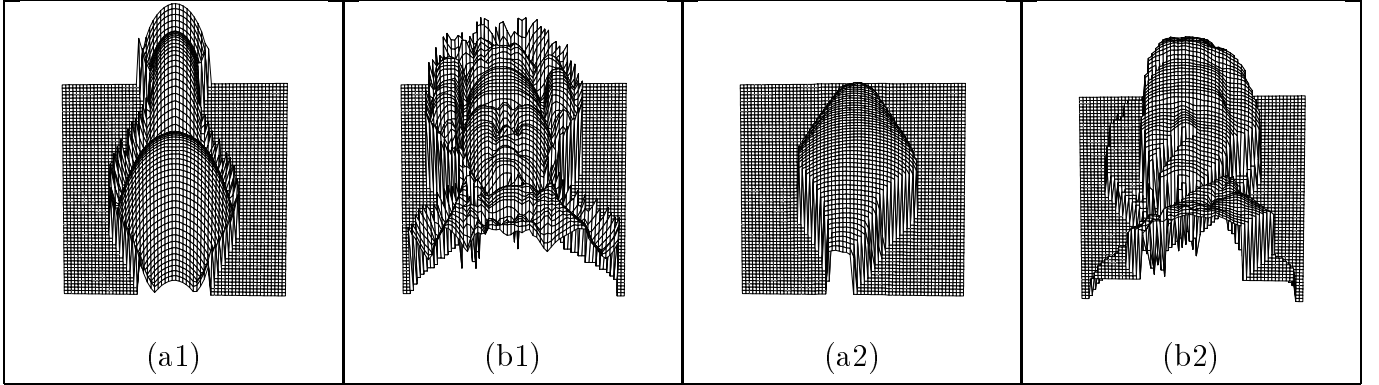


Figure 9: Results for Bichsel and Pentland's method on synthetic images: (b) Vase. (c) Mozart. (a1) and (b1) show the results for test images with light source $(0,0,1)$. (a2) and (b2) show the results for test images with light source $(1,0,1)$.

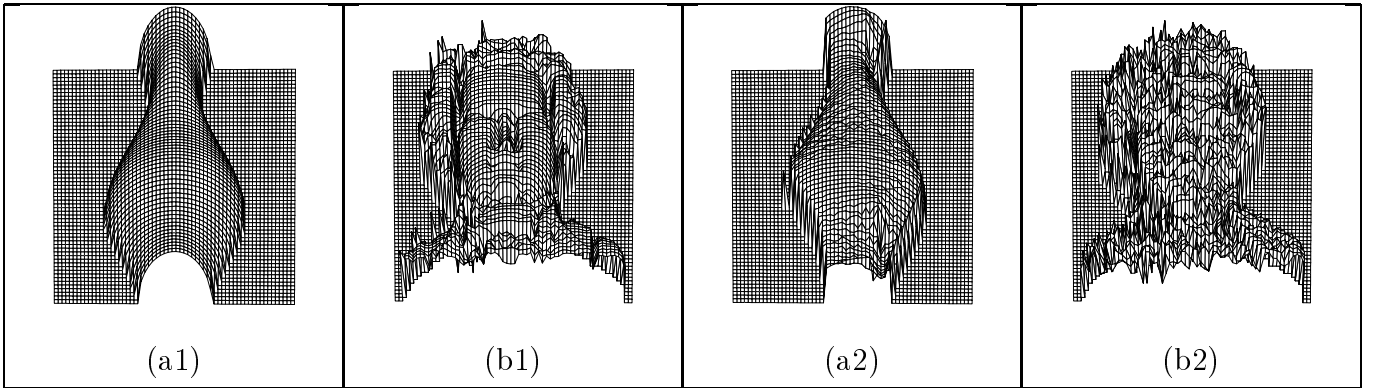


Figure 10: Results for Lee and Rosenfeld's method on synthetic images: (a) Vase. (b) Mozart. (a1) and (b1) show the results for test images with light source $(0,0,1)$. (a2) and (b2) show the results for test images with light source $(1,0,1)$.

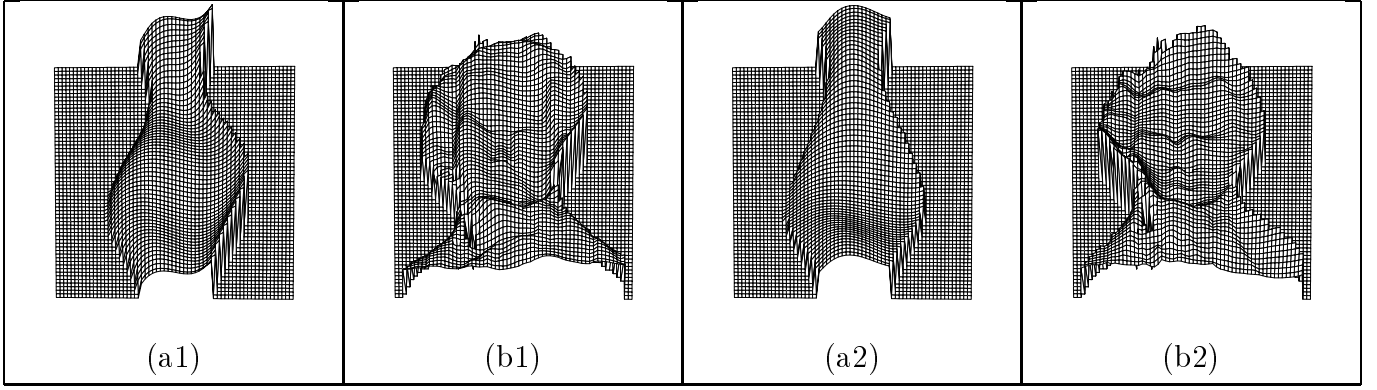


Figure 11: Results for Pentland's method on synthetic images: (b) Vase. (c) Mozart. (a1) and (b1) show the results for test images with light source $(0,0,1)$. (a2) and (b2) show the results for test images with light source $(1,0,1)$.

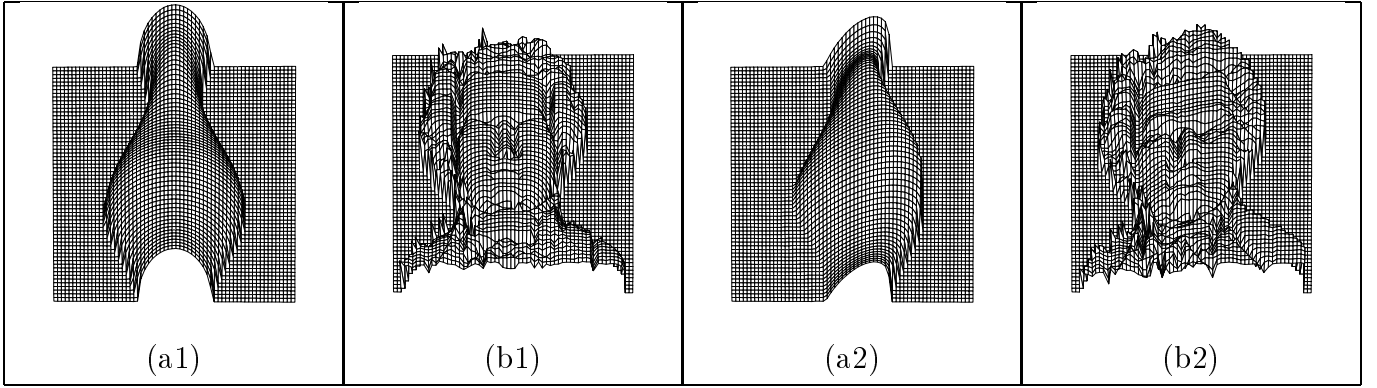


Figure 12: Results for Tsai and Shah's method on synthetic images: (a) Vase. (b) Mozart. (a1) and (b1) show the results for test images with light source $(0,0,1)$. (a2) and (b2) show the results for test images with light source $(1,0,1)$.

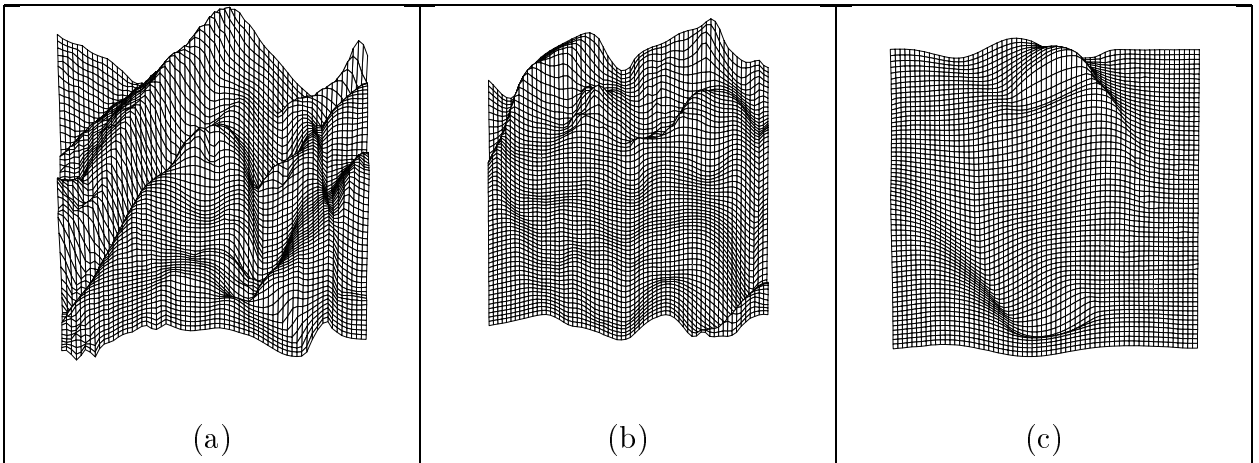


Figure 13: Results for Zheng and Chellappa's method on real images: (a) Lenna. (b) Pepper. (c) Vase.

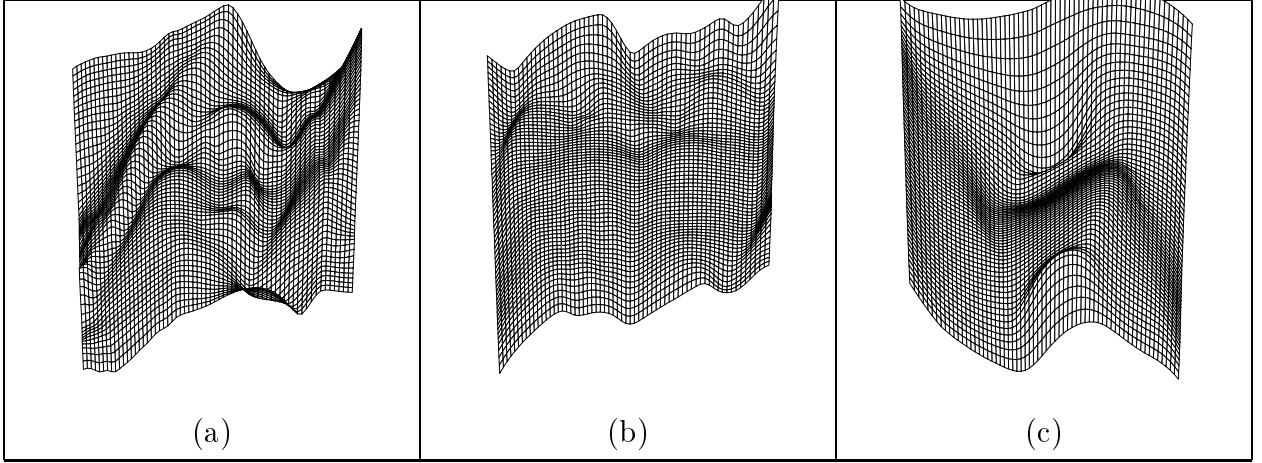


Figure 14: Results for Lee and Kuo's method on real images: (a) Lenna. (b) Pepper. (c) Vase.

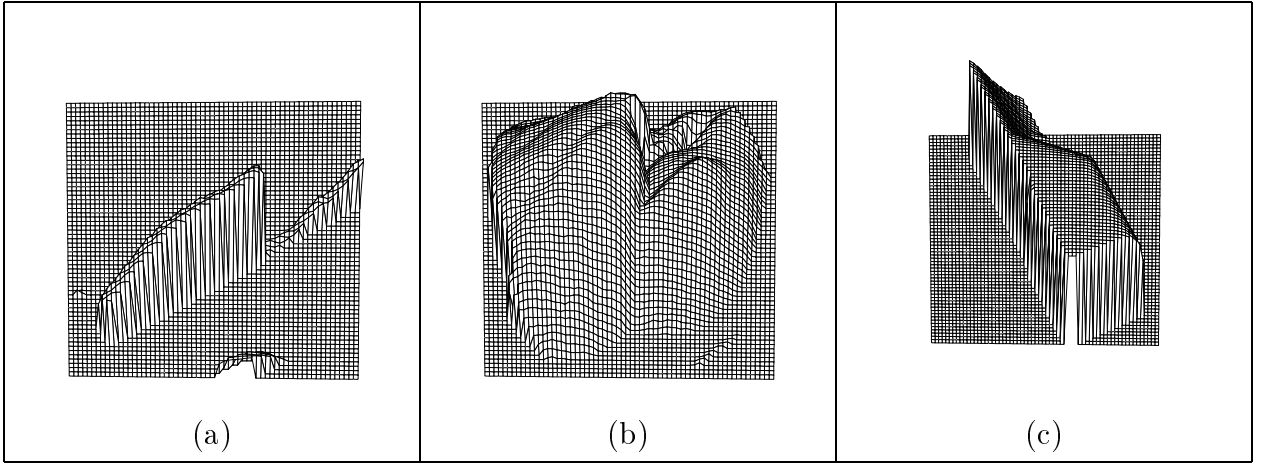


Figure 15: Results for Bichsel and Pentland's method on real images: (a) Lenna. (b) Pepper. (c) Vase.

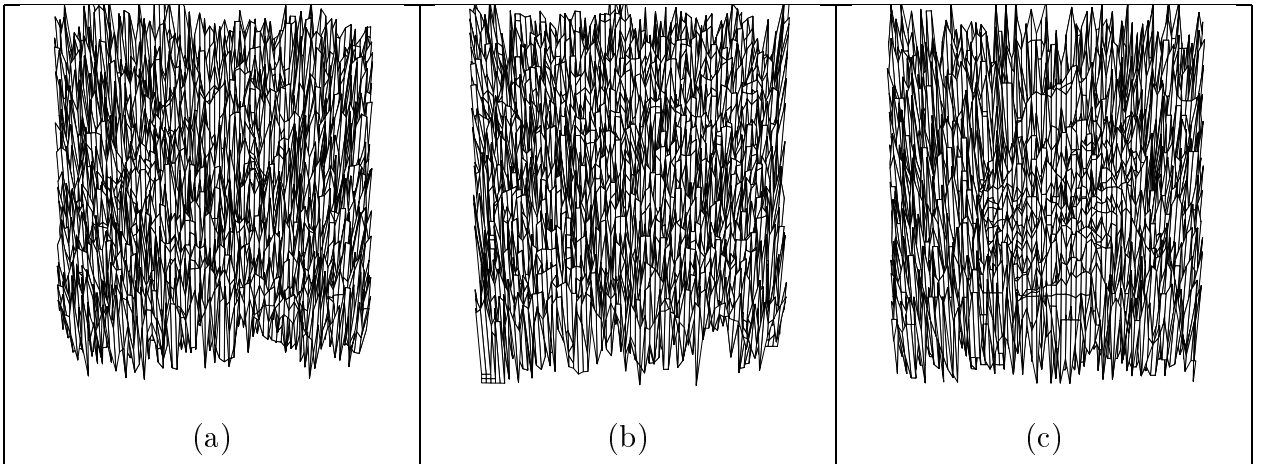


Figure 16: Results for Lee and Rosenfeld's method on real images: (a) Lenna. (b) Pepper. (c) Vase.

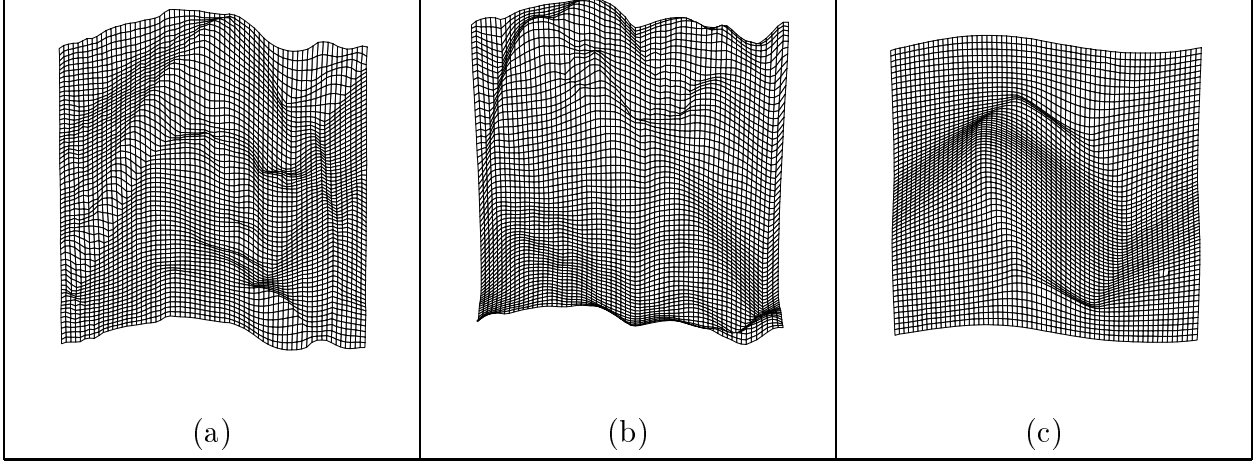


Figure 17: Results for Pentland's method on real images: (a) Lenna. (b) Pepper. (c) Vase.

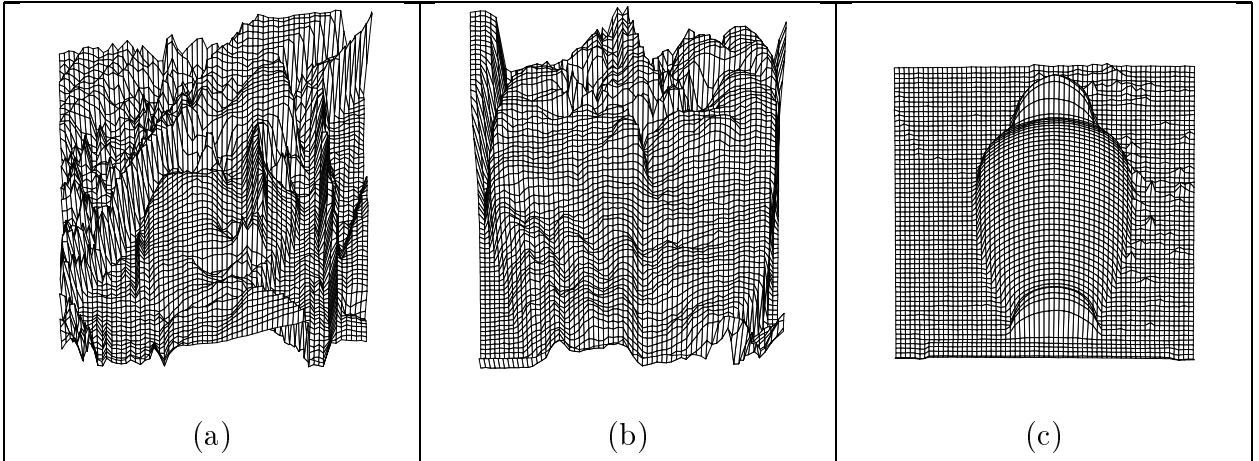


Figure 18: Results for Tsai and Shah's method on real images: (a) Lenna. (b) Pepper. (c) Vase.

behavior. We use the following two:

- **Mean and standard deviation of depth error** (Tables 1 and 2). For each algorithm, we compared the recovered depth with the true depth from the range image. The output depth from each algorithm was first normalized according to the true depth, then compared with the true depth for mean and standard deviation of depth error.
- **Mean gradient error** (Table 3). This indicates the error in the surface orientation. The standard deviation is not used here, since it does not have any physical meaning. The forward discrete approximation was used to compute the gradient from the recovered depth.

For those algorithms which compute the surface gradient together with the depth, we still use the discrete approximation of the depth to calculate the surface gradient in the gradient error table, in order to be consistent with the other algorithms.

There are several ways to analyze the error tables 1–3. One way to is to consider for each algorithm the sum of errors for four images, and rank algorithms based on this measure. According to the sum of average error the ranking of algorithms (table 4) is as follows: Lee and Kuo, Zheng and Chellappa, Bichsel and Pentland, Pentland, Lee and Rosenfeld, and Tsai and Shah. We have also tried to rank these algorithms based on sum of standard deviation of depth errors, sum of gradient (p-q) errors, and finally based on their performance on each image, the ranking is pretty much similar to the above ranking.

8 Timing

CPU timing (Table 5) is computed on a SUN SPARC 4. Disk I/O time is not included, and only computational time is considered. All the synthetic images are 128 by 128. All the real images are 256 by 256. From the timing table (Table 5), we see that the minimization approaches are significantly slower than other approaches. For the minimization approaches, time depends not only on the size of the input image, but also varies from scene to scene. For the other approaches, time depends only on the size of the input image. The order of the algorithms according to CPU time, from the slowest to the fastest, is Lee & Kuo’s algorithm

Table 1: Average Z error for synthetic images.

Methods	Images			
	Vase		Mozart	
	S1	S2	S1	S2
<i>Zheng & Chellappa</i>	8.5	8.5	15.1	10.56
<i>Lee & Kuo</i>	10.0	7.9	13.7	9.77
<i>Bichsel & Pentland</i>	10.08	7.9	20.5	7.7
<i>Lee & Rosenfeld</i>	8.4	18.3	17.8	11.3
<i>Pentland</i>	11.2	9.0	15.7	19.7
<i>Tsai & Shah</i>	8.3	12.7	18.5	20.0

then followed by Zheng & Chellappa’s, Bichsel & Pentland’s, Pentland’s, Lee & Rosenfeld’s, and Tsai & Shah.

9 Conclusions and Future Research

SFS techniques recover the 3-D description of an object from a single view of the object. In this paper, we discussed several existing algorithms and grouped them into four different categories: Minimization techniques, propagation techniques, local techniques, and linear techniques. These groupings are based on the conceptual differences among the algorithms. Six representative algorithms were implemented in order to compare their performance in terms of accuracy and time. This comparison was carried out on two synthetic surfaces, each was used to generate two synthetic images using different light source directions.

To analyze the accuracy, the output for the synthetic images were compared with the true surface shapes and the results of comparison were shown in the forms of the average depth error, the average gradient error and the standard deviation of depth error. The output for real images was only analyzed and compared visually. The conclusion drawn from the analysis are as follows:

1. all the SFS algorithms produce generally poor results when given synthetic data,
2. results are even worse on real images, and

Table 2: Standard deviation of Z error for synthetic images.

	Images			
Methods	Vase		Mozart	
	S1	S2	S1	S2
<i>Zheng & Chellappa</i>	11.1	13.9	18.4	15.9
<i>Lee & Kuo</i>	13.2	15.39	19.2	22.1
<i>Bichsel & Pentland</i>	13.8	16.9	37.4	14.6
<i>Lee & Rosenfeld</i>	14.6	22.3	33.0	30.3
<i>Pentland</i>	12.6	11.1	18.2	20.5 6
<i>Tsai & Shah</i>	15.0	19.7	33.3	30.5

S1, and S2 stand for two different light sources, $(0, 0, 1)$, and $(1, 0, 1)$.

Table 3: Average p-q error for synthetic images.

	Images			
Methods	Vase		Mozart	
	S1	S2	S1	S2
<i>Zheng & Chellappa</i>	2.2	1.3	2.3	1.1
<i>Lee & Kuo</i>	1.6	0.9	1.7	0.6
<i>Bichsel & Pentland</i>	2.7	1.9	3.1 1	1.9
<i>Lee & Rosenfeld</i>	3.3	6.8	13.7	4.3
<i>Pentland</i>	1.8	1.2	1.3	1.3
<i>Tsai & Shah</i>	1.4	6.7	5.5	4.2

Table 4: Ranking based on sum of average depth errors (using Table 1) for synthetic images.

Methods	Sum of error	ranking
<i>Zheng & Chellappa</i>	42.6	2
<i>Lee & Kuo</i>	41.3	1
<i>Bichsel & Pentland</i>	46.1	3
<i>Lee & Rosenfeld</i>	55.8	5
<i>Pentland</i>	55.6	4
<i>Tsai & Shah</i>	59.5	6

Table 5: CPU time (in seconds) for synthetic images.

Methods	Images			
	Vase		Mozart	
	S1	S2	S1	S2
<i>Zheng & Chellappa</i>	18.1	70.8	49.6	409.5
<i>Lee & Kuo</i>	188.4	2584.9	771.5	20249.5
<i>Bichsel & Pentland</i>	7.3	6.1	27.6	26.9
<i>Lee & Rosenfeld</i>	1.5	1.7	6.2	6.9
<i>Pentland</i>	4.2	4.2	17.7	18.4
<i>Tsai & Shah</i>	0.9	1.2	3.8	4.8

S1 and S2 stand for two different light sources, $(0, 0, 1)$ and $(1, 0, 1)$.

3. results on synthetic data are not generally predictive of results on real data.

There are several possible directions for future research. As we noted, reflectance models used in SFS methods are too simplistic; recently, more sophisticated models have been proposed ([20, 24, 7, 35]). This not only includes more accurate models for Lambertian, specular, and hybrid reflectance, but also includes replacing the assumption of orthographic projection with perspective projection, which is a more realistic model of cameras in the real world. The traditional simplification of lighting conditions, assuming an infinite point light source, can also be eliminated by either assuming a non-infinite point light source, or simulating lighting conditions using a set of point sources. This trend will continue. SFS methods employing more sophisticated models will be developed to provide more accurate, and realistic, results.

Another direction is the combination of shading with some other cues. One can use the results of stereo or range data to improve the results of SFS (such as [25] and [41]), or use the results of SFS or range data to improve the results of stereo. A different approach is to directly combine results from shading and stereo (such as [8]).

Multiple images can also be employed by moving either the viewer (as in [16]) or the light source (as in [52]) in order to successively refine the shape. The successive refinement can improve the quality of estimates by combining estimates between image frames, and reduce the computation time since the estimates from the previous frame can be used as the initial values for the next frame, which may be closer to the correct solution. By using successive refinement, the process can be easily started at any frame, stopped at any frame, and restarted if new frames become available. The advantage of moving the light source over moving the viewer is the elimination of the mapping of the depth map (warping) between image frames.

One problem with SFS is that the shape information in the shadow areas is not recovered, since shadow areas do not provide enough intensity information. This can be solved if we make use of the information available from shape-from-shadow (shape-from-darkness) [12, 13, 14, 22, 43, 44, 47] and combine it with the results from SFS. The depth values on the shadow boundaries from SFS can be used either as the initial values for shape-from-shadow, or as constraints for the shape-from-shadow algorithm. In the case of multiple image frames, the information recovered from shadow in the previous frame can also be used for SFS in

the next frame.

Acknowledgments

Thanks to Dr. Bichsel, Prof. Dupuis, Dr. Lee, Prof. Healey, Prof. Kuo, and Dr. Zheng for their helpful discussions.

References

- [1] U. M. Ascher and P. M. Carter. In *SIAM J. Numer. Anal.*, Vol. 30, No. 1, pages 102–115, Feb., 1993.
- [2] H. G. Barrow and J. M. Tenenbaum. Retrospective on “Interpreting line drawings as three-dimensional surfaces”. *Artificial Intelligence*, Vol. 59, pages 71–80, 1993.
- [3] 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.
- [4] A. Blake and A. Zisserman. Surface descriptions from stereo and shading. *Image and Vision Computing*, 3(2):183–191, 1985.
- [5] M. J. Brooks and B. K. P. Horn. Shape and source from shading. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 932–936, 1985.
- [6] A. R. Bruss. The Eikonal equation: Some results applicable to computer vision. *Journal of Mathematical Physics*, 23(5):890–896, 1982.
- [7] J. J. Clark. Active photometric stereo. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 29–34, 1992.
- [8] J. Cryer, P. Tsai, and M. Shah. Integration of shape from x modules: Combining stereo and shading. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 720–721, June 1993.
- [9] P. Dupuis and J. Oliensis. Direct method for reconstructing shape from shading. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 453–458, 1992.
- [10] D. Forsyth and A. Zisserman. Mutual illumination. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 466–473, 1989.
- [11] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:439–451, 1988.

- [12] L. N. Hambrick and M. H. Loew. Entry-Exit Method of Shadow Boundary Segmentation. *IEEE Proceedings of Computer Vision and Pattern Recognition*, 656–658, 1985.
- [13] M. Hatzitheodorou and J. R. Kender. An Optimal Algorithm for the Derivation of Shape from Shadows. *IEEE Proceedings of Computer Vision and Pattern Recognition*, 486–491, 1988.
- [14] M. Hatzitheodorou. The Derivation of 3-D Surface Shape from Shadows. *Image Understanding Workshop*, 1012–1020, 1989.
- [15] G. Healey and T. O. Binford. Local shape from specularity. *Computer Vision, Graphics, and Image Processing*, 42:62–86, 1988.
- [16] J. Heel. Dynamic motion vision. *Robotics and Autonomous Systems*, 6(1), 1990.
- [17] B. K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, MIT, 1970.
- [18] B. K. P. Horn. Height and gradient from shading. *International Journal of Computer Vision*, pages 37–75, 1989.
- [19] B. K. P. Horn, R. S. Szeliski, and A. L. Yuille. Impossible shaded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(2):166–170, 1993.
- [20] D. R. Hougen and N. Ahuja. Estimation of the light source distribution and its use in integrated shape recovery from stereo and shading. In *Proceedings of International Conference on Computer Vision*, pages 29–34, 1993.
- [21] K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1-3):141–184, 1981.
- [22] C. Jiang and M. O. Ward. Shadow Identification. *IEEE Proceedings of Computer Vision and Pattern Recognition*, 606–612, 1992.
- [23] R. Kimmel and A. M. Bruckstein. Shape from shading via Level Sets. *Israel Institute of Technology, CIS Report 9209*, 1992.

- [24] M. S. Langer and S. W. Zucker. Diffuse shading, visibility fields, and the geometry of ambient light. In *Proceedings of International Conference on Computer Vision*, pages 29–34, 1993.
- [25] Y. G. Leclerc and A. F. Bobick. The direct computation of height from shading. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 552–558, 1991.
- [26] C.H. Lee and A. Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence*, 26:125–143, 1985.
- [27] D. Lee. A provably convergent algorithm for shape from shading. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 478–485, 1988.
- [28] K. M. Lee and C. C. J. Kuo. Shape from shading with a linear triangular element surface model. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(8):815–822, 1993.
- [29] J. Malik and D. Maydan. Recovering three dimensional shape from a single image of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 1989.
- [30] E. Mingolla and J. T. Todd. Perception of solid shape from shading. *Biological Cybernetics*, Vol. 53, pages 137–151, 1986.
- [31] S. K. Nayar, K. Ikeuchi and T. Kanade. Shape from interreflections. In *Proceedings of International Conference on Computer Vision*, pages 1–11, 1990.
- [32] S. K. Nayar, K. Ikeuchi and T. Kanade. Surface reflection: physical and geometrical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):611–634, 1991.
- [33] J. Oliensis. Shape from shading as a partially well-constrained problem. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54:163–183, 1991.
- [34] J. Oliensis and P. Dupuis. A global algorithm for Shape from shading. In *Proceedings of International Conference on Computer Vision*, pages 692–701, 1993.

- [35] M. Oren and S. K. Nayar. Diffuse reflectance from rough surfaces. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 763–764, 1993.
- [36] A. P. Pentland. Finding the illuminant direction. *J. Optical Society of America*, pages 448–455, 1982.
- [37] A. P. Pentland. Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:170–187, 1984.
- [38] A. Pentland. Shape information from shading: a theory about human perception. In *Proceedings of International Conference on Computer Vision*, pages 404–413, 1988.
- [39] A. Pentland. Photometric motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:879–890, 1991.
- [40] B. T. Phong. Illumination for computer generated pictures. *Comm. of ACM*, 18:311–317, 1975.
- [41] H. Pien and J. Gauch. A variational approach to sensor fusion using registered range and intensity data. In *SPIE OE/Aerospace Sensing*, April 1993.
- [42] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1990.
- [43] D. Raviv and Y. H. Pao and K. A. Loparo. Segmentation Between Overlapping Parts: The Moving Shadows Approach. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4):880–883, 1989.
- [44] D. Raviv and Y. H. Pao and K. A. Loparo. Reconstruction of Three-Dimensional Surfaces from Two-Dimensional Binary Images. *IEEE Transactions on Robotics and Automation*, 5(5):701–710, 1989.
- [45] V. S. Ramachandran. Perceiving shape from shading. *Scientific American*, 159:76–83, 1988.
- [46] E. Rouy and A. Tourin. A Viscosity solutions approach to shape from shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992.

- [47] S. A. Shafer and T. Kanade. Using Shadows in Finding Surface Orientations. *Computer Vision, Graphics, and Image Processing*, 22:145–176, 1983.
- [48] R. Szeliski. Fast shape from shading. *Computer Vision, Graphics, Image Processing: Image Understanding*, 53:129–153, 1991.
- [49] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *J. Of Optical Society Of America*, 57:1105–1114, 1967.
- [50] P.S. Tsai and M. Shah. Shape from shading using linear approximation. *Image and Vision Computing Journal*, 12(8):487–498, 1994.
- [51] O. E. 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, 1993.
- [52] Zhang, R., Tsai, P., and Shah, M. Photomotion. *Computer Vision and Image Understanding*, 63(2), March 1996.
- [53] Zhang, R., Tsai, P., Cryer, J., and Shah, M. A survey of shape from shading methods. Technical Report CS-TR-97-15, Computer Science Department University of Central Florida, 1997.
- [54] 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.

Ruo Zhang received her Ph.D. in 1995 from University of Central Florida. Her research interests are shape from shading, shape from photomotion, software engineering, and computer networks. Currently she is working in the industry.

Ping-Sing Tsai was born in Taipei, Taiwan, R.O.C.. He received the B.S. degree in information and computer engineering from Chung-Yuan Christian University in 1985, and his Ph.D. in computer science at the University of Central Florida in 1995. Dr. Tsai is currently working at Intel Corporation, Digital Imaging and Video Division, as a senior system engineer.

James Cryer received his B.S and M.S. degrees from University of Central Florida. He participated in Research Experience for Undergraduates (REU) during 1989-90. His research interests are computer vision, image processing, and parallel processing.

Mubarak Shah received his BE degree in 1979 in electronics from Dawood College of Engineering and Technology, Karachi, Pakistan. He spent 1980 at Philips International Institute of Technology, Eindhoven, The Netherlands, where he completed the EDE diploma. Dr. Shah received his MS and Ph.D. degrees both in computer engineering from Wayne State University, Detroit, Michigan, in 1982 and 1986, respectively. Since 1986, he has been with the University of Central Florida, where he is currently professor of computer science and the director of the Computer Vision Lab. Prof. Shah spent 1998-99 year at the Information Systems Division of Harris Corporation, working with their Video Research and Development group as a technical consultant. Dr. Shah has received the Teaching Incentive Program Award, IEEE Distinguished Visitors Program Speaker Award, IEEE Outstanding Engineering Educator Award, and two TOKTEN awards (in 1995 and 1997) by UNDP.

Prof. Shah has published one book with Ramesh Jain (Motion-Based Recognition, Kluwer Academic Publishers) and over 70 research papers in refereed journals and conferences on topics including optical flow, structure from motion, motion-based recognition, gesture recognition, activity recognition, lipreading, snakes, object recognition, edge and contour detection, multisensor fusion, shape from shading and stereo, and hardware algorithms for computer vision. Prof. Shah is an associate editor of IEEE Transactions on PAMI and Pattern Recognition journal. He has served on the program committees, and chaired

sessions of several conferences. He was an organizer of "Workshop on Recent Advances in Computer Vision," held in Karachi, Pakistan, Jan 1-2, 1998, and "Workshop on Computer Vision" held in Islamabad, Pakistan, January 3-5, 1995. Both workshops were supported by grants from NSF.

Prof. Shah has served as a project director for the national site for REU, Research Experience for Undergraduates in Computer Vision, funded by the National Science Foundation since 1987. He has received over 2 millions in grants from NSF, DARPA, Army and the State of Florida.