Analysis of Shape from Shading Techniques

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer and Mubarak Shah
Computer Science Department
University of Central Florida
Orlando, FL 32816
shah@cs.ucf.edu

Abstract

Since the first shape-from-shading technique was developed by Horn in the early 1970s, different approaches have been continuously emerging in the past two decades. Some of them improve existing techniques, while others are completely new approaches. However, there is no literature on the comparison and performance analysis of these techniques. This is exactly what is addressed in this paper. A total of eight well-known shape-from-shading algorithms are implemented and compared in terms of timing and accuracy, in order to analyze the advantages and disadvantages of these approaches. The experiments were performed on five synthetic images with three different light sources each, and five real images. The performance of the algorithms was analyzed using mean and standard deviation of depth (Z) error, mean of surface gradient (p, q) error and CPU timing. The comparison showed that all of them have some limitations. None of the algorithms has consistent performance for all images, since they work well for certain images, but perform poorly for others. In general, global approaches are more robust, while local 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).

Contents

1 Introduction

Shape recovery is a classic problem in computer vision. The goal in shape recovery 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, surface normal (n_x, n_y, n_z), surface gradient (p, q), and surface slant, ϕ, and tilt, θ. The depth is the relative height of the surface. The surface normal is the orientation of a vector perpendicular to the tangent plane on the object surface, which is usually a unit vector. The surface gradient, $(p, q) = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y})$, 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. 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. In SFS, it is important to study how the images are formed. A simple model of image formation is the Lambertian model. According to the Lambertian model, the gray level at a pixel in the image depends on the light source location, and the surface normal. In SFS, given a gray level image, the aim is to recover the light source and a surface normal at each pixel in the image.

Shape-from-photometric-stereo [37] is another method for shape recovery. The difference between shape-from-photometric-stereo and shape-from-shading (SFS) is in the number of input images. Shape-from-photometric-stereo recovers shape from *multiple* intensity images of the same scene generated using a fixed viewing direction and different light source directions; while SFS provides the shape estimate from a *single* intensity image.

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

One of the earlier global minimization approaches was by Ikeuchi and Horn [15]. Since each surface point has two unknowns for the surface normal, and each pixel in the image provides one gray value, therefore image gray levels alone are not enough to recover the shape. To overcome this, Ikeuchi and Horn introduced two constraints: The brightness constraint and the smoothness constraint. The brightness constraint requires that the reconstructed shape shall produce the same brightness as the input image at each surface point, while the smoothness constraint forces the gradient of the surface to change smoothly. The shape was computed by minimizing an energy function which consists of the above two constraints. Also using these two constraints, Brooks and Horn [3] minimized the same energy function, in terms of surface normal instead of surface gradient. Frankot and Chellappa [8] enforced the integrability in Brooks and Horn's algorithm in order to recover integrable surfaces (surfaces for which $z_{xy} = z_{yx}$). Surface slope estimates from the iterative scheme were expressed in terms of a linear combination of a finite set of orthogonal Fourier basis functions. The enforcement of integrability was done by projecting the nonintegrable surface slope estimates onto the nearest (in terms of distance) integrable surface slopes. This projection was fulfilled by finding the closest set of coefficients which satisfy integrability in the linear combination. Their results showed improvements in both accuracy and efficiency. Later, Horn also [12] replaced the smoothness constraint in his approach with an integrability constraint. The major problem with Horn's method is its slow convergence. Szeliski [33] 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 [36] applied heuristics to the variational approach so that the stability of Brooks and Horn's algorithm was improved.

Instead of the smoothness constraint, Zheng and Chellappa [39] 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. Leclerc and Bobick [17] 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 [19] also proposed an approach to recover depth using the brightness and the smoothness constraint. They approximated surfaces by a union of triangular patches. Unlike Leclerc and Bobick's method, this approach did not require the depth from stereo as an initial value.

All of the above approaches deal with a single smooth surface. Malik and Maydan [20] developed the first solution for piecewise smooth surfaces. They combined the line drawing and shading constraints in an energy function, and recovered both surface normal and line labelling through the minimization of the energy function.

The first global propagation approach was the characteristic strip technique by Horn [11]. 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 are identified as the direction of intensity gradients. In order to get a dense shape map, new strips have to be interpolated when neighboring strips separate too much.

Oliensis [23] observed that the smoothness constraint is only needed at the boundaries if we have initial values at the singular points. Based on this idea, Dupuis and Oliensis [6, 24] formulated SFS as an optimal control problem, and solved it using numerical methods. Bichsel and Pentland [2] simplified Dupuis and Oliensis's approach and proposed a minimum downhill approach for SFS which converged in less than ten iterations.

Among the local approaches, two are by Pentland, one by Lee and Rosenfeld, and one by Tsai and Shah. Pentland [27] 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 [18] computed the slant and tilt of the surface in the light source coordinate system through the first derivative of the intensity. A later approach by Pentland [28] 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. Similar to Pentland's method, Tsai and Shah [35] 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 iteratively recovered the depth at each point without using any global information.

Pentland's linear shape from shading has problems with images of quadratic surface reflectance. Therefore, Pentland [29] proposed photometric motion to solve for shape and reflectance. The images needed in his approach were taken at different time frames while the object was rotated. The quadratic component of the surface reflectance function was factored out by subtracting two of the images. The ratio of one of the images and the difference image was used to cancel out the albedo and obtain the surface shape. Therefore, at least two images were required for the shape recovery. This approach was also extended to three-image photometric motion by considering second derivatives in the discrete form.

None of the above methods deal with interreflections – the mutual illumination between surface facets. Nayar, Ikeuchi, and Kanade [21] 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. Their formulation of interreflection was based on Forsyth and Zisserman's result [7].

There is no literature on the comparison and performance analysis of these techniques. This is exactly what is addressed in this paper. A total of eight well-known shape-from-shading algorithms are implemented and compared in terms of timing and accuracy, in order to analyze the advantages and disadvantages of these approaches. The experiments were performed on five synthetic images with three different light sources each, and five real images. The performance of the algorithms was analyzed using mean and standard deviation of depth (Z) error, mean of surface gradient (p, q) error and CPU timing. The comparison showed that all of them have some limitations. None of the algorithms has consistent performance for all images, since they work well for certain images, but perform poorly for others. In general, global approaches are more robust, while local approaches are faster. The implementation of these algorithms in C, and images used in this paper, are available by anonymous ftp.

The organization of the remainder of this paper is as follows: The next section introduces background knowledge related to reflectance models. In section three, which deals with recovery of the light source location, we explain Pentland's original approach for light source computation, its refinements by Lee and Rosenfeld, and improvements of both methods by Zheng and Chellappa. SFS approaches, which are divided into global and local approaches,

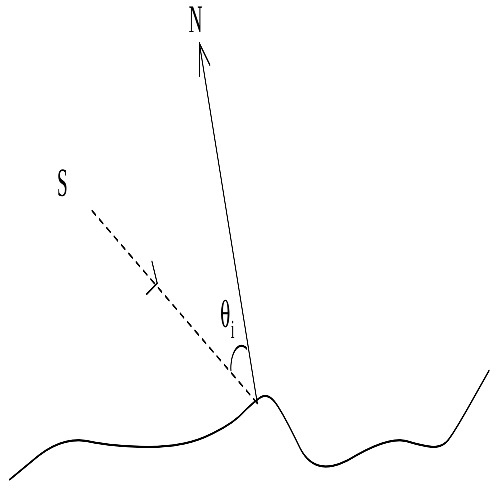


Figure 1: Lambertian Reflection Geometry.

are discussed in section four. Global approaches are further divided into global minimization approaches and global propagation approaches. A total of twelve SFS methods are summarized in this section. Section five is devoted to the description of the synthetic and real images used in this study. We present our experimental results on eight different SFS algorithms in section six. The error analysis is presented in section seven, where we compare different methods using mean and standard deviation of depth (Z) error, mean of surface gradient (p, q) error, difference images of absolute depth error, and histograms of percentage depth error. Section eight summarizes the CPU timing of all eight algorithms for different images. Section nine presents a general discussion of convergence and uniqueness. Finally, conclusions and future research are covered in section ten.

2 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.

2.1 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 B, the albedo of the surface ρ, and the foreshortened area $\cos\theta_i$ as follows:

$$I_L = R = B\rho\cos\theta_i, \tag{1}$$

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

$$I_L = B\rho\vec{N}\cdot\vec{S}, \tag{2}$$

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:

$$I_S = B\delta(\theta_s - 2\theta_r), \tag{3}$$

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 was developed by Phong [30]. It represents the specular component of reflection as powers of the cosine of the angle between the perfect specular direction and the viewing direction. This model is capable of predicting specularities which extend beyond a single point; however, the parameters have no physical meaning. A more refined model, the Torrance-Sparrow model [34], assumes that a surface is composed of small, randomly oriented, mirror-like facets. It describes the specular brightness as the product of

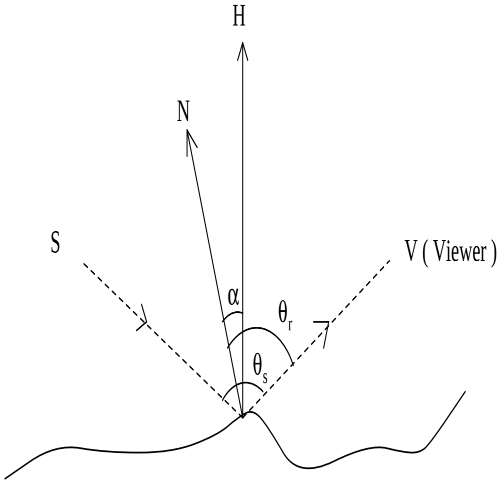


Figure 2: Specular Reflection Geometry.

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 [9] 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:

$$I_S = Ke^{-(\frac{\alpha}{m})^2}, \tag{4}$$

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 2).

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:

$$I = (1-\omega)I_L + \omega I_S, \tag{5}$$

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 [22] 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:

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

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.

2.2 More Sophisticated Reflectance Models

Although the Lambertian model is widely used because of its simplicity, it is a poor approximation to the diffuse component of rough surfaces. For a rough surface, the radiance increases as the viewer approaches the source direction. Oren and Nayar [25] derived a reflectance model for rough diffuse surfaces, taking into account complex geometrical effects of masking, shadowing, and interreflection. Modeling rough surfaces as a collection of V-cavities, they developed a functional approximation of rough surfaces for both isotropic and non-isotropic surfaces, including uni-directional single-slope distribution, isotropic single-slope distribution and Gaussian distribution. A simplified qualitative model was derived by considering the relative significance of the various terms in the functional approximation:

$$I \;=\; \cos\theta_i(A + BMax[0, \cos(\phi_r - \phi_i)]\sin\alpha\tan\beta), \tag{7}$$

where $A \approx \rho(\frac{1}{\pi} - 0.09\frac{m^2}{m^2+0.4})$, $B \approx \rho(0.05\frac{m^2}{m^2+0.18})$, (θ_i, ϕ_i) and (θ_r, ϕ_r) are the same as in the previous section, $\alpha = Max[\theta_i, \theta_r]$, $\beta = Min[\theta_i, \theta_r]$, ρ is the albedo value, and m is the surface roughness. This model reduces to the Lambertian model when $m = 0$.

Clark [4] used perspective, rather than orthographic, projection in modeling reflectance. In his model, there is no requirement for the light source to be at infinity. Therefore, the reflectance function is:

$$I(\vec{x}) = K\frac{R(\vec{S})}{|Z\vec{\chi} + \vec{t}|^2|Z\vec{\chi}|^2},$$

where $\vec{x} = (x, y)$ is the image coordinate vector, K is a constant, R is the reflectance map, $\vec{S} = \frac{Z\chi+\vec{t}}{|Z\vec{\chi}+\vec{t}|}$, which indicates the direction from the surface point to the light source, $\vec{t}$ is the location of the light source with respect to the coordinate system centered on the focal point of the camera, $\vec{\chi} = (\frac{x}{f}, \frac{y}{f}, -1)^T$, f is the focal length of the camera, and Z is the depth. Since the distances from surface to camera and from light source to surface are both considered in this model, it yields a more realistic description of reflectance.

Hougen and Ahuja [14] have observed that the assumption of a single point source overly simplifies the model of the light source distribution. They approximated the light source distribution by a set of m distinct light source vectors, $\vec{S}_1, \vec{S}_2, \ldots, \vec{S}_m$, where $\vec{S}_k$ is the average value of $\vec{S}$ over a neighborhood angle of $\vec{S}_k$. By writing $\vec{S}_k$ as a product of its magnitude λ_k and unit direction $\vec{S}'_k$, the brightness equation can be expressed by:

$$I = \rho(\lambda_0 R_0 + \sum_{k=1}^{m} \lambda_k R(\vec{N}, \vec{S}'_k)), \tag{8}$$

where $\rho\lambda_0 R_0$ is due to the contribution of ambient light, and R is the reflectance map, which is independent of the magnitude of the light source.

Unlike the classical Lambertian reflectance model, Langer and Zucker [16] introduced the concept of "Shape from Shading on a Cloudy Day". They claimed that under diffuse lighting, the radiance depends primarily on the amount of the diffuse source visible from each surface element, with the surface normal of secondary importance. Assuming the effect of mutual illumination can be ignored, the brightness at image point $\vec{x} = (x, y)$ is described as:

$$I(\vec{x}) = \rho I_D \frac{1}{\pi} \int_{\nu(\vec{x})} \vec{N}(\vec{x}) \cdot \vec{S} d\Omega, \tag{9}$$

where ρ is the albedo, I_D is the illuminance from a uniform hemispheric light source, $\nu(\vec{x})$ is the set of unit directions in which the sky is visible from $\vec{x}$, and $d\Omega$ denotes an infinitesimal solid angle.

The above reflectance models attempt to remove one or more of the following constraints used in the simplification of the classical Lambertian model:

- The brightness is independent of the viewing direction,
- The illumination is from an infinite point source,
- The projection of the object onto the image plane is perspective.

Oren and Nayar's, and Langer and Zucker's models emphasized the important effect of the viewing direction in the reflectance model. Hougen and Ahuja considered background illumination and more realistic lighting as compared to a single point source illumination widely used. Clark used the perspective projection in the shape recovery, and his model does not even depend on any specific reflectance model. In short, all of those models overcome the over-simplification of the traditional Lambertian model, which results in the elimination of the modeling error.

3 Source From Shading

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

Several techniques to estimate source orientation have been developed. The first one, by Pentland [26], estimates the light source direction from the distribution of image derivatives. By assuming an umbilical surface and isotropic surface normal, a maximum-likelihood analysis was performed to estimate the slant and tilt angles of the light source. The basic idea underlying this approach is simple, since surface orientation is a random variable over the whole image for most scenes, so both surface normal $\vec{N}$ and the change of the surface normal $d\vec{N}$ are isotropically distributed. This means that if we consider any direction (dx_I, dy_I) on the image plane, the z component of the expected value of $d\vec{N}$, $d\bar{n}_z$, is zero. Therefore, the derivative of the brightness equation

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

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

$$E[dI] = d\bar{I} = B(s_x \; d\bar{n}_x + s_y \; d\bar{n}_y),$$

where E indicates the expected value.

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

$$\begin{pmatrix} d\bar{I}_1 \\ d\bar{I}_2 \\ \vdots \\ d\bar{I}_{m'} \end{pmatrix} = \begin{pmatrix} dx_1 & dy_1 \\ dx_2 & dy_2 \\ \vdots & \vdots \\ dx_{m'} & dy_{m'} \end{pmatrix} \begin{pmatrix} \tilde{s_x} \\ \tilde{s_y} \end{pmatrix},$$

where $d\bar{I}_i$ is the average of the intensity change along the image direction (dx_i, dy_i), $\tilde{s_x} = B\bar{k}s_x$ and $\tilde{s_y} = B\bar{k}s_y$. A typical choice for the (dx_i, dy_i) are the eight directions in the image plane: Two in the horizontal direction, two in the vertical direction, and four along the diagonals.

Solving the above system by least squares, we get:

$$\begin{pmatrix} \tilde{s_x} \\ \tilde{s_y} \end{pmatrix} = (\beta^T\beta)^{-1}\beta^T \begin{pmatrix} d\bar{I}_1 \\ d\bar{I}_2 \\ \vdots \\ d\bar{I}_{m'} \end{pmatrix}, \tag{10}$$

where β is the matrix of directions (dx_i, dy_i).

The tilt, τ_S, of the light source direction is given by:

$$\tau_S = \arctan(\frac{\tilde{s_y}}{\tilde{s_x}}), \tag{11}$$

and the slant, σ_S, of the light source direction is:

$$\sigma_S = \arccos\sqrt{1 - {s_x}^2 - {s_y}^2}.$$

Taking the expected value of the square of intensity derivative $E[dI^2]$, and cancelling out the common terms between $E[dI]^2$ and $E[dI^2]$ by subtracting one from the other, we have the relation $E[dI^2] - E[dI]^2 = B^2\bar{k}^2$. Since $\tilde{s_x} = B\bar{k}s_x$, $\tilde{s_y} = B\bar{k}s_y$, by introducing $k = B\bar{k} = \sqrt{E[dI^2] - E[dI]^2}$, the equation for the slant of the light source can be simplified to:

$$\sigma_S = \arccos\sqrt{1 - \frac{\tilde{s_x}^2 + \tilde{s_y}^2}{k^2}}. \tag{12}$$

Instead of taking intensity derivatives along a number of directions, Lee and Rosenfeld [18] considered only the derivatives along the x and y directions. They approximated the surface geometry by a spherical patch in a local region, so their method was also based on an isotropic distribution of the surface orientation. Since the image of a sphere is symmetric about the projection of the light source vector in the image plane, the average direction of the intensity gradient must be parallel to this projection. This gives $\frac{E(I_y)}{E(I_x)} = \frac{S_y}{S_x}$, so we have,

$$\tau_S = \arctan(\frac{E(I_y)}{E(I_x)}), \tag{13}$$

where the expectations are taken over the given image region.

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

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

The expectation here is taken over the whole image.

Zheng and Chellappa [39] modified Lee and Rosenfeld's method by considering not only the area of the illuminated portion in the integral, but also the area of the portion in shadow. Although the shadow does not contribute to the total intensity, it does contribute to the area computation in order to correctly calculate the mean intensity value over the whole image. After the modification, the computation for slant, σ_S, became:

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

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

$$\tau_S = \arctan(\frac{E(\frac{\tilde{s_x}}{\sqrt{\tilde{s_x}^2+\tilde{s_y}^2}})}{E(\frac{\tilde{s_y}}{\sqrt{\tilde{s_x}^2+\tilde{s_y}^2}})}),$$

where $(\tilde{s_x}, \tilde{s_y})$ is the same as given by Pentland's method.

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

$$\tau_S = \arctan(\frac{x_2}{x_1}),$$

where

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (\beta^T\beta)^{-1}\beta^T \begin{pmatrix} \bar{I}_1 \\ \bar{I}_2 \\ \vdots \\ \bar{I}_{m'} \end{pmatrix},$$

and β is the same as in equation 10.

The slant of the light source σ_S is estimated by:

$$\sigma_S = \frac{1}{f_1^2(\gamma) + f_2(\gamma)}(E(I)f_1(\gamma) + \sqrt{E(I^2)f_2(\gamma)}),$$

where $E(I)$ and $E(I^2)$ are the ensemble averages of the image intensities, and the square of the image intensities, γ, can be solved by:

$$f_3(\gamma) = \frac{E(I)}{\sqrt{E(I^2)}}.$$

Here, $f_1(\gamma)$, $f_2(\gamma)$ and $f_3(\gamma)$ are three known seventh-order polynomials [39] in $\cos\gamma$.

All three methods are similar. For example, for the estimation of tilt, if we take the derivatives only in the x and y directions in Pentland's regression model, it reduces to Lee and Rosenfeld's method.

In [39], Zheng and Chellappa tested the above methods on a set of three different images. The results showed that for the estimation of tilt, all algorithms work almost perfectly for a sphere without background. However, background and noise will degrade the performance of both Lee and Rosenfeld's and Pentland's methods. Consequently, Zheng and Chellappa's method is more robust to background and noise in most of the cases. For the estimation of slant, Pentland's method is very sensitive to noise, but Lee and Rosenfeld's and Zheng and Chellappa's methods are robust to Gaussian noise. If a uniform background is included, results for all three methods are degraded. In terms of the computation time, since m' is usually greater than 2, Lee and Rosenfeld's is the most efficient, Pentland's is the second, and Zheng and Chellappa's is a bit slower than Pentland's as it needs to solve seventh-order polynomials.

In the following sections, we assume the light source is known.

4 Shape From Shading

Once the light source direction is known, 3-D shape can be estimated. In this section, we discuss two classes of SFS algorithms: Global and local. We have surveyed eight global methods and three local methods.

4.1 Global Minimization Approaches

Global 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 to ensure the correct convergence. In this subsection, first, we briefly describe these constraints, and then discuss eight global SFS methods.

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

$$\int\int (I - R)^2 \, dx \, dy, \tag{14}$$

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

$$\int\int (p_x^2 + p_y^2 + q_x^2 + q_y^2) \, dx \, dy, \tag{15}$$

here p and q are surface gradients along the x and y directions. Physically, this constraint means that the surface can be approximated by a small facet in a close neighborhood. Another version of the smoothness term is less restrictive by requiring constant change of depth only in x and y directions:

$$\iint (p_x^2 + q_y^2)\, dx\, dy. \tag{16}$$

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

$$\iint (\|\vec{N}_x\|^2 + \|\vec{N}_y\|^2)\, dx\, dy. \tag{17}$$

This means that the surface normal should change gradually.

Although constraints (16) and (17) look alike, in reality, constraints (15) and (17) are similar if we consider the relationship between surface normal, $\vec{N}$, and surface gradient, (p, q). Both (15) and (17) are more restrictive than (16) in terms of the smoothing directions. They are more frequently used, since (16) tends to lead to excessively flattened surfaces.

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

$$\iint (p_y - q_x)^2\, dx\, dy, \tag{18}$$

or

$$\iint ((Z_x - p)^2 + (Z_y - q)^2))\, dx\, dy. \tag{19}$$

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:

$$\iint ((R_x - I_x)^2 + (R_y - I_y)^2)\, dx\, dy. \tag{20}$$

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

$$\iint (\|\vec{N}\|^2 - 1)\, dx\, dy. \tag{21}$$

4.1.1 Ikeuchi and Horn

Ikeuchi and Horn [15] minimize the following energy function to estimate the shape (p, q):

$$E = \iint (I(x,y) - R(p,q))^2 + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2)\, dx\, dy. \tag{22}$$

The first term is the brightness constraint, and the second term is the surface smoothness constraint. Differentiating the above equation with respect to p and q, setting the resulting equations to zero, and solving for p and q we get:

$$p(x,y) = \bar{p}(x,y) + T(x,y,p,q)\frac{\partial R}{\partial p}, \tag{23}$$

$$q(x,y) = \bar{q}(x,y) + T(x,y,p,q)\frac{\partial R}{\partial q}, \tag{24}$$

where $T(x,y,p,q) = \frac{I(x,y)-R(p,q)}{\lambda}$, $\bar{p}$ and $\bar{q}$ are the local averages of p and q. The problem with this approach is that it requires the complete gradient information, along the occluding boundary and at the singular points, as initial values.

4.1.2 Brooks and Horn

Brooks and Horn [3] combined the brightness constraint, the smoothness constraint and the unit normal constraint, and minimized the following energy function:

$$\iint ((I-R)^2 + \lambda(\|\vec{N}_x\|^2 + \|\vec{N}_y\|^2) + \mu(\|\vec{N}\|^2 - 1))\,dx\,dy, \tag{25}$$

where $\vec{N}$ is the surface normal, λ is a scalar that weighs the relative importance of the smoothness term, and μ is a Lagrangian multiplier. The first term in the equation is the brightness constraint, the second term is the smoothness constraint, and the third term is the unit normal constraint.

The minimization of the above function was done through variational calculus. In general, the functional

$$\iint F(x,y,\vec{N},\vec{N}_x,\vec{N}_y)\,dx\,dy, \tag{26}$$

has the Euler equation

$$F_{\vec{N}} - \frac{\partial}{\partial x}F_{\vec{N}_x} - \frac{\partial}{\partial y}F_{\vec{N}_y} = 0. \tag{27}$$

Therefore, the function given in equation (25) has the Euler equation:

$$(I - \vec{N}\cdot\vec{S})\vec{S} + \lambda\nabla^2\vec{N} - \mu\vec{N} = 0. \tag{28}$$

A discrete approximation to the Laplacian operator

$$\nabla\vec{N}^2_{ij} \approx \frac{4}{\epsilon^2}(\bar{\vec{N}}_{ij} - \vec{N}_{ij}), \tag{29}$$

was used in order to change the Euler equation into the following discrete form:

$$(I_{ij} - \vec{N}_{ij} \cdot \vec{S}) + \frac{4\lambda}{\epsilon^2}(\bar{\vec{N}}_{ij} - \vec{N}_{ij}) - \mu_{ij}\vec{N}_{ij} = 0. \tag{30}$$

Here $\bar{\vec{N}}_{ij}$ is the average of the normals in the neighborhood around the point (i, j).

Assuming known light source direction, $\vec{S}$, the iterative scheme for $\vec{N}$ was developed by rearranging the above equation and taking only the direction of the vector by dropping the constant term $\frac{1}{1+\mu_{i,j}(\epsilon^2/4\lambda)}$. The iterative scheme for $\vec{S}$ was derived by assuming $\vec{N}$ is known, then setting the partial derivative of the energy function (25), with respect to $\vec{S}$, to zero and solving it. The final iterative equations are:

$$\left\{ \begin{array}{ll} m_{i,j}^{(k+1)} & = \bar{\vec{N}}_{i,j}^{(k)} + \frac{\epsilon^2}{4\lambda}(I_{i,j} - \vec{N}_{i,j}^{(k)} \cdot \vec{S}^{(k)})\vec{S}^{(k)} \\ \vec{N}_{i,j}^{(k+1)} & = m_{i,j}^{(k+1)} / \|m_{i,j}^{(k+1)}\| \\ \vec{S}^{(k+1)} & = (\sum_{i,j\in\Omega} \vec{N}_{i,j}^{(k+1)} \cdot \vec{N}_{i,j}^{(k+1)T})^{-1} \sum_{i,j\in\Omega} I_{i,j}\vec{N}_{i,j}^{(k+1)} \end{array} \right.$$

where ϵ is the distance between adjacent pixels in the image.

Later, Horn [12] applied another variation of the smoothness constraint, which is similar to the one used by Ikeuchi and Horn in [15],

$$\iint (p_x^2 + p_y^2 + q_x^2 + q_y^2)\, dx\, dy. \tag{31}$$

Depth and gradient were recovered simultaneously by combining this smoothness constraint with the brightness and the integrability constraints.

Both versions of Horn's algorithms require that the shape at occluding boundaries (either the surface normal, or the surface gradient) be known. However, information at occluding boundaries is difficult to obtain, especially for the surface gradient, since the surface gradient at the occluding boundary has at least one infinite component. Another disadvantage is the slow convergence of the algorithms.

4.1.3 Szeliski

Horn's algorithms, discussed in the previous sections, both converge slowly. Szeliski [33] used the same constraints as Horn did (see section 4.1.2) in his recent approach, with discretization of the gradient using depth. By combining the conjugate gradient descent and a hierarchical technique, Szeliski provides a faster solution; however, his algorithm needs the same initial values as Horn's algorithm.

4.1.4 Zheng and Chellappa

Zheng and Chellappa [39] applied the intensity gradient constraint, instead of a smoothness constraint, therefore, their energy function became:

$$\iint ((I-R)^2 + \lambda((R_x - I_x)^2 + (R_y - I_y)^2) + \mu((Z_x - p)^2 + (Z_y - q)^2))\,dx\,dy. \tag{32}$$

The Euler equations were simplified by taking the Taylor series of the reflectance map and representing the depth, gradient and their derivatives in discrete form. Then, a new iterative scheme, which updates depth and gradients simultaneously, was derived. The algorithm was implemented using a hierarchical structure (pyramid) in order to speed up the computation. There was no special requirement for the initialization of the boundary. The initial values for both depth and gradient can be zero.

4.1.5 Leclerc and Bobick

Leclerc and Bobick dropped the integrability constraint, and minimized the following function in the discrete domain

$$E = \sum_{i,j}((1-\lambda)(R(p_{i,j}, q_{i,j}) - I_{i,j})^2 + \lambda(u_{i,j}^2 + v_{i,j}^2)), \tag{33}$$

where

$$R(p_{i,j}, q_{i,j}) = \frac{-s_x p_{i,j} - s_y q_{i,j} + s_z}{\sqrt{1 + p_{i,j}^2 + q_{i,j}^2}}, \tag{34}$$

and (s_x, s_y, s_z) is the unit light source vector. The following discretizations, using central difference approximation, were used: $p_{i,j} = \frac{1}{2}(Z_{i+1,j} - Z_{i-1,j})$, $q_{i,j} = \frac{1}{2}(Z_{i,j+1} - Z_{i,j-1})$, $u_{i,j} = Z_{i+1,j} - 2Z_{i,j} + Z_{i-1,j}$, and $v_{i,j} = Z_{i,j+1} - 2Z_{i,j} + Z_{i,j-1}$.

The depth was computed by assuming a known light source, and setting the derivative of equation 33, with respect to depth, to zero. When the depth is known, the light source direction can be estimated by differentiating equation (33) with respect to the three components of the light source, setting them to zeros, then solving the three simultaneous linear equations.

The approach used a conjugate gradient together with a hierarchical structure to solve simultaneous equations. Initially, the weight of the smoothness term, λ, was set to 1, and it was gradually reduced to zero. An object mask was used to eliminate the background, boundaries, and regions with significant albedo changes. Since the central approximation was not suitable on the boundary, the discretization of the boundary points was changed into either a forward or backward approximation. The initial depth values were obtained from stereo.

4.1.6 Lee and Kuo

Lee and Kuo [19] 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.

Although the combination of the linear approximation of the reflectance map and triangular surface approximation resulted in a linear relationship between the reflectance map and the depth, it also introduced an approximation error.

4.2 Global Propagation Approaches

Global propagation approaches start from the 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 present two algorithms in this section.

4.2.1 Dupuis and Oliensis

Most of the approaches described above make use of the smoothness constraint; however, the smoothness constraint sometimes leads to an over-smoothed surface. Oliensis [23] discovered that the smoothness constraint is often unnecessary, since shading determines shape with little ambiguity for a typical image. He also observed that while the occluding boundary does not strongly constrain the surface solution, a singular point does. Therefore, the surface should be reconstructed from the interior of the image outward, instead of from the boundary inward. Since both singular points and characteristic strips are independent of the viewing direction, and characteristic strips correspond to steepest ascent curves on the object, the shape of a surface can be constructed strip by strip given the information at the singular points.

Considering the importance of singular points, Dupuis and Oliensis [6, 24] developed an iterative algorithm to recover depth using discretized optimal control and dynamic programming. The proof of equivalence between the optimal control representation and SFS was illustrated. At first, they required a vertical light source, and only one singular point, then, they removed these restrictions and allowed for a general light source, and multiple singular points. However, their initial algorithm [6] requires priori depth information for all the singular points. A later extension [24] can determine this information automatically by assuming twice differentiable depth, isolated singular points and nonzero curvature at singular points.

4.2.2 Bichsel and Pentland

Following the main idea of Dupuis and Oliensis's, Bichsel and Pentland [2] 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 precomputed by taking the derivative of the reflectance map (34) 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:

$$\begin{aligned} p &= \frac{-s_x s_z \pm \sqrt{(1-R^2)(R^2-s_y^2)}}{R^2 - s_x^2 - s_y^2}, && (35) \\ q &= \frac{p s_y s_x - s_y s_z}{R^2 - s_y^2}. && (36) \end{aligned}$$

One drawback to this approach is the requirement for singular points. When singular points do not have the same depth, the algorithm will have trouble initializing their depths.

4.3 Local Approaches

Local approaches use only local intensity information around the current pixel to derive shape. Here, we describe four approaches, two use linear approximations of the reflectance function, and two uses the intensity derivative information and surface spherical assumption.

4.3.1 Pentland

Pentland's first approach [27] solved for the surface slant and tilt, the radius of curvature, and the light source direction through six equations obtained from the intensity, as well as the first and second derivatives of the intensity. His approach can classify a surface into planar, cylindrical, convex, concave, or saddle surface. However, it is limited to surfaces with equal-magnitude principal curvatures. The use of the second derivatives also makes the algorithm very sensitive to noise.

Pentland's second approach [28] used the linear approximation of the reflectance map in p and q. By taking the Taylor series expansion of the reflectance function

$$R(p,q) = \frac{-s_x p - s_y q + s_z}{\sqrt{1+p^2+q^2}}, \qquad (37)$$

about $p = p_0$, $q = q_0$, and ignoring the high order terms, we have

$$I(x,y) = R(p_0,q_0) + (p-p_0)\frac{\partial R}{\partial p}(p_0,q_0) + (q-q_0)\frac{\partial R}{\partial q}(p_0,q_0). \qquad (38)$$

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:

$$\frac{\partial}{\partial x}Z(x,y) \longleftrightarrow F_Z(\omega_1,\omega_2)(-i\omega_1) \tag{39}$$

$$\frac{\partial}{\partial y}Z(x,y) \longleftrightarrow F_Z(\omega_1,\omega_2)(-i\omega_2), \tag{40}$$

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

$$F_I = F_Z(\omega_1,\omega_2)(-i\omega_1)\cos\tau_S\sin\sigma_S + F_Z(\omega_1,\omega_2)(-i\omega_2)\sin\tau_S\sin\sigma_S, \tag{41}$$

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.

4.3.2 Lee and Rosenfeld

Lee and Rosenfeld [18] 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:

$$\tau = \arctan\frac{I_y\cos\tau_S - I_x\sin\tau_S}{I_x\cos\tau_S\cos\sigma_S + I_y\cos\sigma_S\sin\tau_S}, \tag{42}$$

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 $I = \rho\vec{N}\cdot\vec{S}$, then the brightness 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.

4.3.3 Tsai and Shah

Tsai and Shah [35] 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 the same as equation (34). 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:

$$0 = f(I_{i,j}, Z_{i,j}, Z_{i-1,j}, Z_{i,j-1}) = I_{i,j} - R(Z_{i,j} - Z_{i-1,j}, Z_{i,j} - Z_{i,j-1}). \tag{43}$$

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 (43)) about a given depth map Z^{n-1} is:

$$\begin{aligned}
0 &= f(I_{i,j}, Z_{i,j}, Z_{i-1,j}, Z_{i,j-1}) \\
&\approx f(I_{i,j}, Z^{n-1}_{i,j}, Z^{n-1}_{i-1,j}, Z^{n-1}_{i,j-1}) + (Z_{i,j} - Z^{n-1}_{i,j})\frac{\partial}{\partial Z_{i,j}} f(I_{i,j}, Z^{n-1}_{i,j}, Z^{n-1}_{i-1,j}, Z^{n-1}_{i,j-1}) + (Z_{i-1,j} - Z^{n-1}_{i-1,j}) \\
&\quad \frac{\partial}{\partial Z_{i-1,j}} f(I_{i,j}, Z^{n-1}_{i,j}, Z^{n-1}_{i-1,j}, Z^{n-1}_{i,j-1}) + (Z_{i,j-1} - Z^{n-1}_{i,j-1})\frac{\partial}{\partial Z_{i,j-1}} f(I_{i,j}, Z^{n-1}_{i,j}, Z^{n-1}_{i-1,j}, Z^{n-1}_{i,j-1}).
\end{aligned} \tag{44}$$

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 (44) into the following equation:

$$0 = f(Z_{i,j}) \approx f(Z^{n-1}_{i,j}) + (Z_{i,j} - Z^{n-1}_{i,j})\frac{d}{dZ_{i,j}} f(Z^{n-1}_{i,j}). \tag{45}$$

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

$$Z^n_{i,j} = Z^{n-1}_{i,j} + \frac{-f(Z^{n-1}_{i,j})}{\frac{d}{dZ_{i,j}} f(Z^{n-1}_{i,j})}. \tag{46}$$

The initial estimate of $Z^0_{i,j}$ is set to zero for all pixels, Gaussian smoothing is applied to the final depth map to get a smoother result. This is a simple but efficient algorithm. However, self-shadows will break it down.

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. 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.

We used five synthetic surfaces. The true depth maps of the first two surfaces, Sphere and Vase, were generated mathematically:

- The **Sphere** was generated using the following formula:

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

  where $r = 52$, and $-63 \le x,\ y \le 64$.

- 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

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

  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 maps for the other three synthetic images, **Mozart**, **Penny** and **Sombrero**, were provided by Professor Kuo of USC.

The depth maps for all five synthetic surfaces are shown in Figure 3. The synthetic images generated from these surfaces, with light source directions $(0,0,1)$, $(1,0,1)$, $(5,5,7)$,

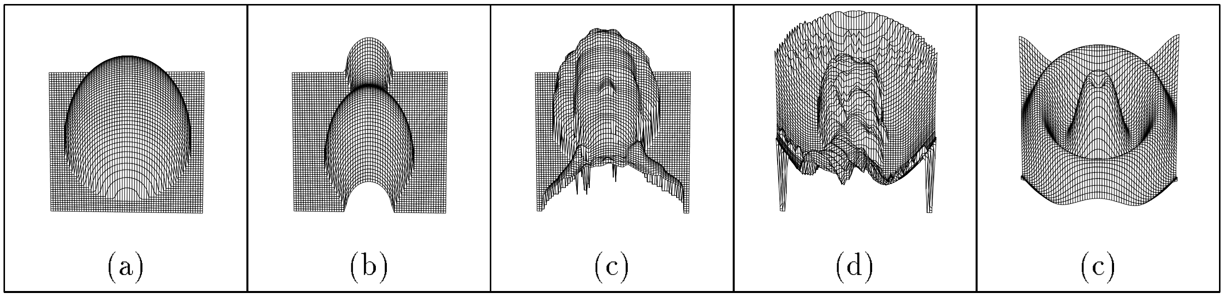


Figure 3: Depth maps for the Synthetic Images: (a) Sphere. (b) Synthetic Vase. (c) Mozart. (d) Penny. (e) Sombrero.

are shown in Figure 4. The images are either reduced or expanded to make their sizes power of 2. Here, the only image which has the convex/concave ambiguity is the **Sombrero** with light source $(0, 0, 1)$.

5.2 Real Images

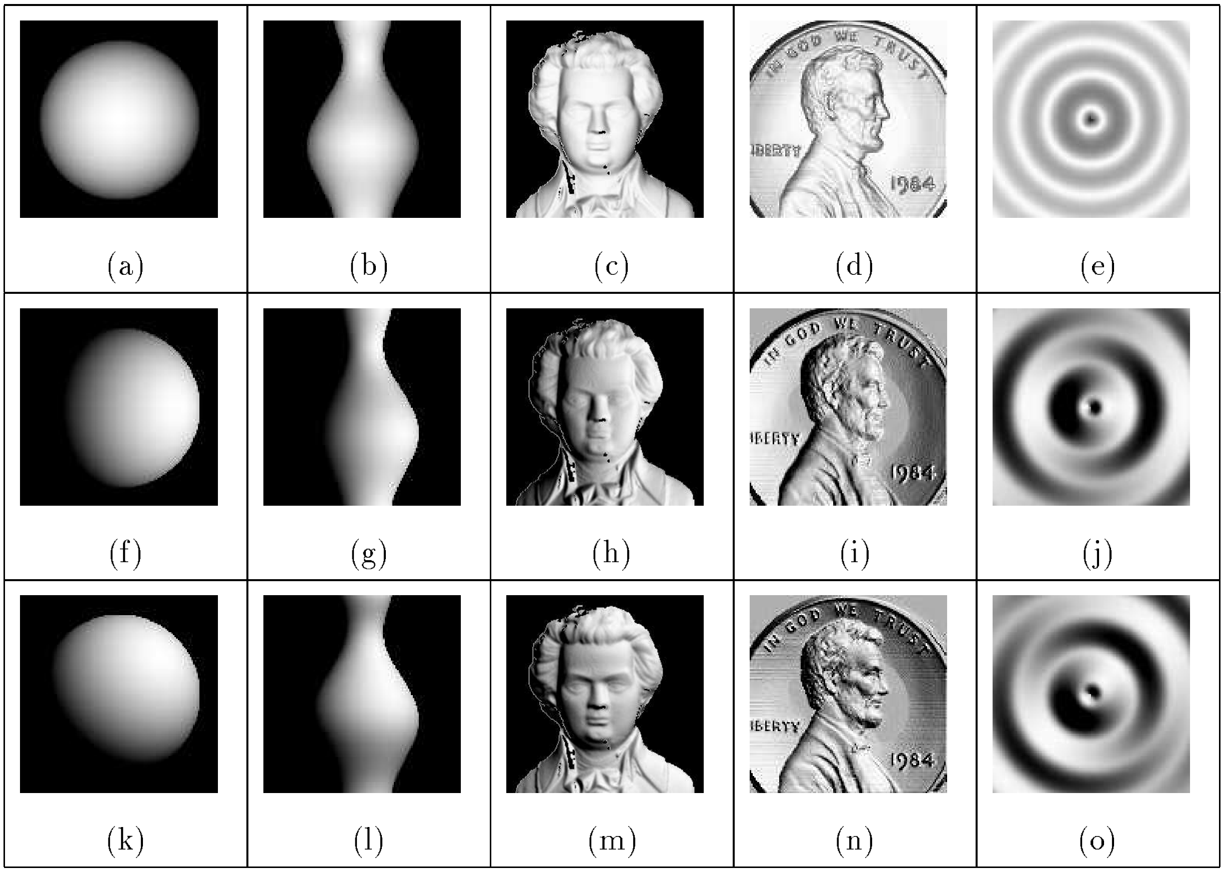
Five real images, shown in Figure 5, were also used. The light source directions, given below, either were estimated by the Lee and Rosenfeld method or provided with the images:

- **Lenna:** Estimated light source direction is $(1.5, 0.866, 1)$.
- **Mannequin:** Estimated light source direction is $(-0.345, 0.345, 0.875)$.
- **Pepper:** Estimated light source direction is $(0.766, 0.642, 1)$.
- **David:** Estimated light source direction is $(-0.707, 0.707, 1)$.
- **Vase:** Estimated light source direction is $(-0.939, 1.867, 1.0)$.

The **Vase** image was provided by Professor Woodham of UBC, and **Mannequin** by Dr. Leclerc of Artificial Intelligence Center, SRI International. The rest of the images were provided by Professor Kuo of USC.

6 Experimental Results

We implemented eight of the twelve algorithms discussed in this paper. Szeliski's algorithm was not implemented since it is a faster version of Brooks and Horn's algorithm. We did

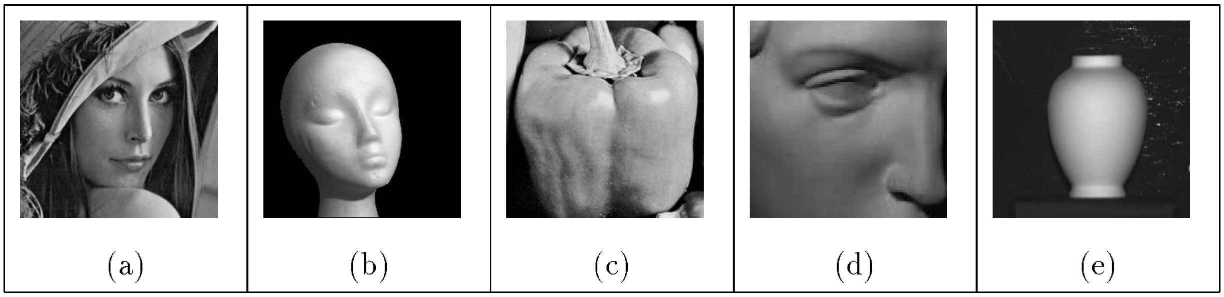


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

Figure 4: Synthetic images generated using different light sources: (a) Sphere $(0,0,1)$. (b) Synthetic Vase $(0,0,1)$. (c) Mozart $(0,0,1)$. (d) Penny $(0,0,1)$. (e) Sombrero $(0,0,1)$. (f) Sphere $(1,0,1)$. (g) Synthetic Vase $(1,0,1)$. (h) Mozart $(1,0,1)$. (i) Penny $(1,0,1)$. (j) Sombrero $(1,0,1)$. (k) Sphere $(5,5,7)$. (l) Synthetic Vase $(5,5,7)$. (m) Mozart $(5,5,7)$. (n) Penny $(5,5,7)$. (o) Sombrero $(5,5,7)$.



(a) (b) (c) (d) (e)

Figure 5: Real images: (a) Lenna. (b) Mannequin. (c) Pepper. (d) David. (e) Vase.

not implement Ikeuchi and Horn's algorithm because it resembles Brooks and Horn's. Since Bichsel and Pentland's algorithm is a simplification of Dupuis and Oliensis's, we implemented Bichsel and Pentland's algorithm only.

The depth maps computed from both synthetic and real images by each algorithm are shown in Figures 7 through 19. Below we discuss some important points about the implementation of each algorithm, and analyze the results using the 3-D plots of depth maps.

6.1 Brooks and Horn

Brooks and Horn's approach requires occluding boundary information for the input image. Their algorithm computes the shape in terms of surface normals. In order to reconstruct the surface of the object, an integration step must be applied to compute the depth. However, neither finding the occluding boundary from an image nor integrating the surface normals are easy tasks. Since the primary objective of this survey paper is to study SFS algorithms, we only tested two synthetic images, sphere and synthetic vase with light source $(0, 0, 1)$, for which the occluding boundary information was available. The results of the reconstructed gray level images from the computed shape with the same light source are shown in Figure 6. We can see that the reconstructed gray level images closely resemble the input images. A well-known problem of Brook and Horn's method is the slow convergence rate. In our implementation, we forced the algorithm to terminate when the error in the energy function starts to increase, or the error is less than some threshold. It took at least 400 iterations to achieve convergence for our test images.

6.2 Zheng and Chellappa

The implementation of Zheng and Chellappa's method is very straightforward. We used the forward difference approximation to compute the partial derivatives. 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. This method is also very robust, since no parameters have to be tuned.

The results for synthetic images are shown in Figure 7, and the results for real images are shown in Figure 14. Our implementation works well for most of the real images, except

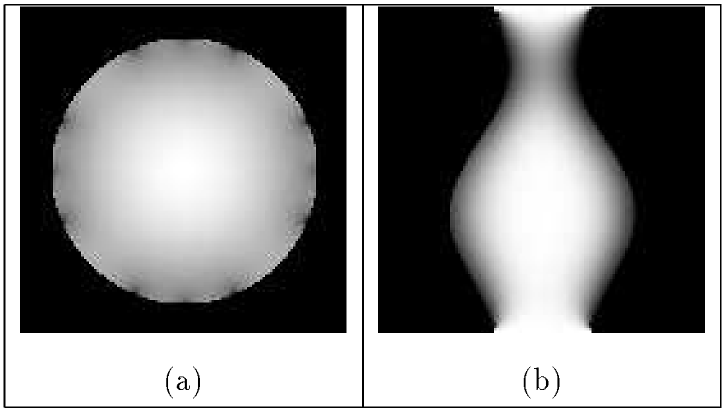

(a) (b)

Figure 6: The results for Brook and Horn's method. (a)The reconstructed gray level image with light source $(0, 0, 1)$ for Sphere. (b)The reconstructed gray level image with light source $(0, 0, 1)$ for Vase.

for Mannequin and Vase. This is due to the dark background in these two images, which violates the uniform albedo assumption used in their algorithm. The basic shape 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 approximation equation of the iterative method. 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. This can be clearly observed in the result of the Mannequin image (as shown in Figure 14(b)). We can see that there is a crevice on the face along the light source direction,$(-0.345, 0.345, 0.875)$. We think this is due to the use of the intensity gradient constraint instead of the smoothness constraint used in their energy function, and the discrete approximation used for computing the partial derivatives.

6.3 Leclerc and Bobick

Leclerc and Bobick's approach was implemented without hierarchical structure, using the conjugate gradient routine given in [32]. On the occluding boundary, the discrete approximations for the first order partial derivatives were changed from the central difference approximation to either the forward approximation or the backward approximation, and the second order partial derivatives were set to zero. Their approach requires the output from stereo as the initial estimate for the conjugate gradient method. Since we do not have stereo pairs for the test images, we used the true depth with ±5% uniform random noise as the initial estimate, and tested the algorithm on synthetic images only.

The results for synthetic images are shown in Figure 8. Their results depend heavily on the initial estimate and the initial weight of the smoothness term, λ. It can be seen that the algorithm works well on two sets of images, Sphere and Mozart, even when the light source is from the side. The initial value of the smoothness term was 0.25 for both sets of images, which was progressively reduced to 0.01 by a factor of 0.7. The maximum number of iterations for the conjugate gradient routine was set to 200. We do not report the result for the Vase image (with light source $(1,0,1)$), Penny and Sombrero images (with light source $(5,5,7)$ and $(1,0,1)$) here, because we could not find a good initial value for the smoothness term, λ, which would make the conjugate gradient method converge. We feel that this may be due to two reasons: Large self-shadow areas in the images (especially in the Sombrero image), and inadequate initial estimate of the depth map. This method is basically a combination of stereo and shape from shading. It is hard to compare the performance of this method with other SFS methods. Because it heavily relies on the initial estimate from stereo, one has to take into account the computation and accuracy involved in stereo as well.

6.4 Lee and Kuo

Lee and Kuo's algorithm was implemented using the V-cycle multigrid scheme to solve the linear system, as they reported in their paper. 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 stencil 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 zeros. Since the algorithm does not work for light source direction $(0,0,1)$, we used $(0.0001, 0.0001, 1.0)$ as the input light source direction instead. This light source approximation is different from the $(0.01, 0.01, 1)$ used in Zheng and Chellappa's algorithm. The implementation of Zheng and Chellappa's algorithm does not work with $(0.0001, 0.0001, 1.0)$.

For most of the images, the smoothing factor was 2000, and the level of grids was computed 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. However, to eliminate the effect of over-smoothing, we used 200 as the smoothing factor for Sombrero and David, and we ran only 1 iteration for David, Mannequin and Penny. 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.

The results for synthetic images are shown in Figure 9, and the results for real images are shown in Figure 15. It can be seen that the algorithm works well, even when the light source is from the side, except in the cases of Sphere and Vase which create the most 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.

6.5 Bichsel and Pentland

In the implementation of Bichsel and Pentland's algorithm the initial depth values for the singular points were assigned a fixed positive value, and the depth values for the other points were initialized a large negative value. 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.

The results for synthetic images are shown in Figure 16, and the results for real images are shown in Figure 10. All results were obtained after 8 iterations. The algorithm provides the best results for the cases when the light source is on the side; even the sphere can be recovered very well when the light source comes from the side. However, the algorithm does not give good results for real images except for Pepper. This, we think, is due to the inaccuracy of the initial singular points, and noise in the real images. The algorithm is very fast; usually only 5 iterations are required to provide reasonable results.

6.6 Lee and Rosenfeld

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.

The results for synthetic images are shown in Figure 11, and the results for real images are shown in Figure 17. Their method estimates the depth of an image 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 Penny or 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.

6.7 Pentland

The implementation of Pentland's algorithm was done using the fast Fourier transform and the inverse fast Fourier transform (see equation 41).

The results for synthetic images are shown in Figure 12, and the results for real images are shown in Figure 18. His algorithm produces good results on most surfaces that change linearly, even if the surface has a naturally varying surface such as a person's face. However, this algorithm falls apart when the surface changes in a non-linear manner; this can be seen clearly from the results of Sphere. For real images, the algorithm produces the best results except for Vase. The details of Mannequin are not recovered and inaccuracy is high around the eye regions in David and Lenna.

6.8 Tsai and Shah

The implementation of Tsai and Shah's algorithm is straight forward. The depth is updated iteratively using equation 46. There is no special care needed for the boundary.

The results for synthetic images are shown in Figure 13, and the results for real images are shown in Figure 19. From Figure 13, we can clearly see that their method works very well on smooth objects with the light source close to the viewing direction. However, it is sensitive to the intensity noise, such as the black hole on the nose of Mozart image or the shadow areas. The problem of the convex/concave ambiguity [1] is clearly shown in the result for the Sombrero image with light source $(0, 0, 1)$ (see the Figure 13.(e1)). The results for real images are good for Mannequin, David and 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. These are regions where there are sudden intensity changes, which cause roughness in the depth estimate due to the relationship between depth and intensity.

7 Error Analysis

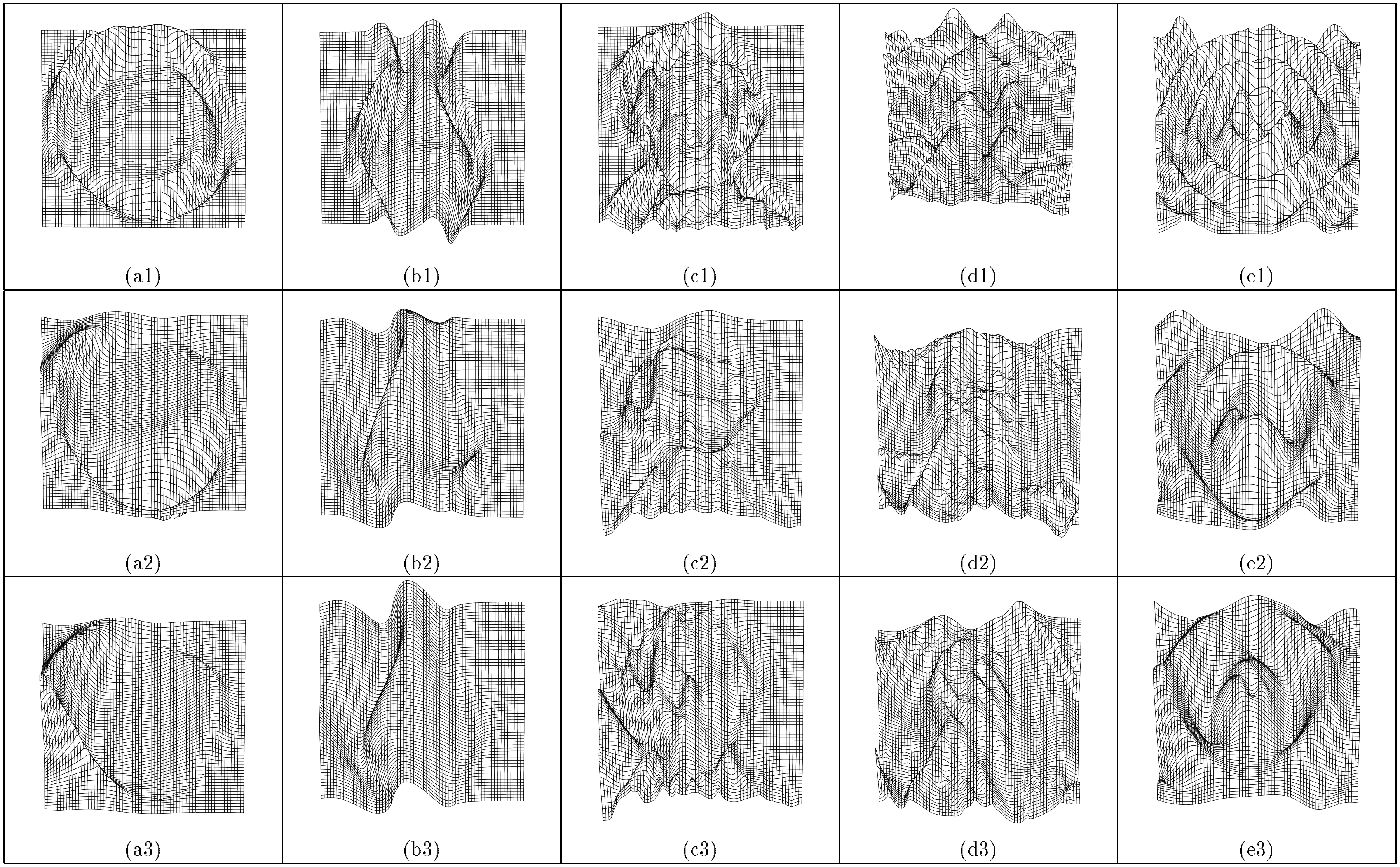
In the previous section, we reported results for synthetic and real images, and qualitatively analyzed the results by considering the 3-D plots of the depth map. In this section, we will quantitatively analyze the results for the synthetic images for which the true depth maps are available by using some error measures. There are several ways to report the error behavior. In this section, we use the following:

- **Mean and standard deviation of depth error** (Tables 1-2). 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 range data, then compared with the range data 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

---

[1] The convex/concave ambiguity is due to fact that for each convex surface, there exists a concave surface which produces the same shading image as the convex one.

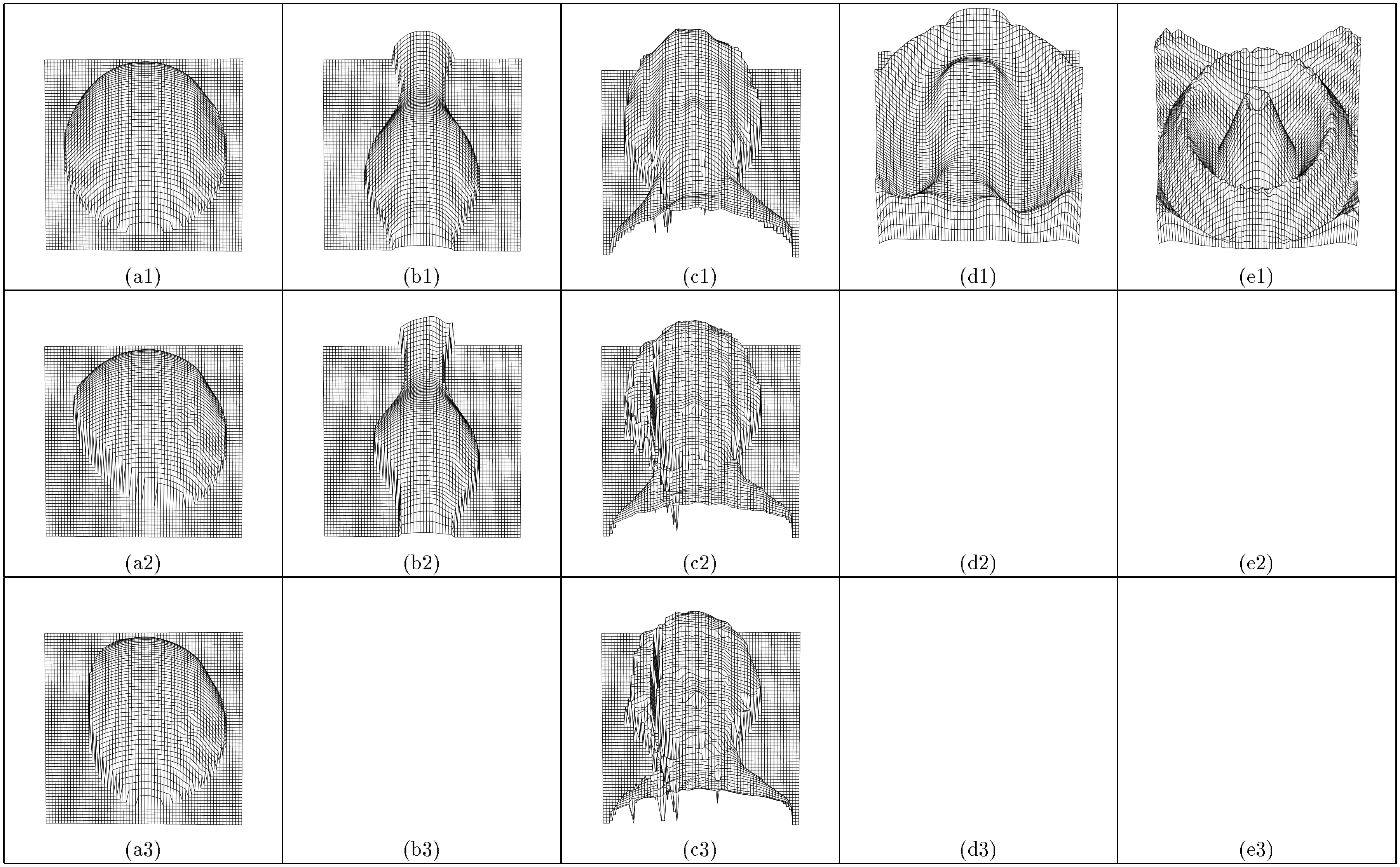


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 7: Results for Zheng and Chellappa's method on synthetic images: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The first row shows the results for test images with light source $(0, 0, 1)$. The second row shows the results for test images with light source $(5, 5, 7)$. The third row shows the results for test images with light source $(1, 0, 1)$.



(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 8: Results for Leclerc and Bobick's method on synthetic images: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The first row shows the results for test images with light source $(0, 0, 1)$. The second row shows the results for test images with light source $(5, 5, 7)$. The third row shows the results for test images with light source $(1, 0, 1)$. The results for the Vase image with light source $(1, 0, 1)$,(b3), Penny and Sombrero images with light source $(5, 5, 7)$ and $(1, 0, 1)$,(d2, d3, e2, and e3) are not reported here, because we could not find a good initial value for the smoothness term.

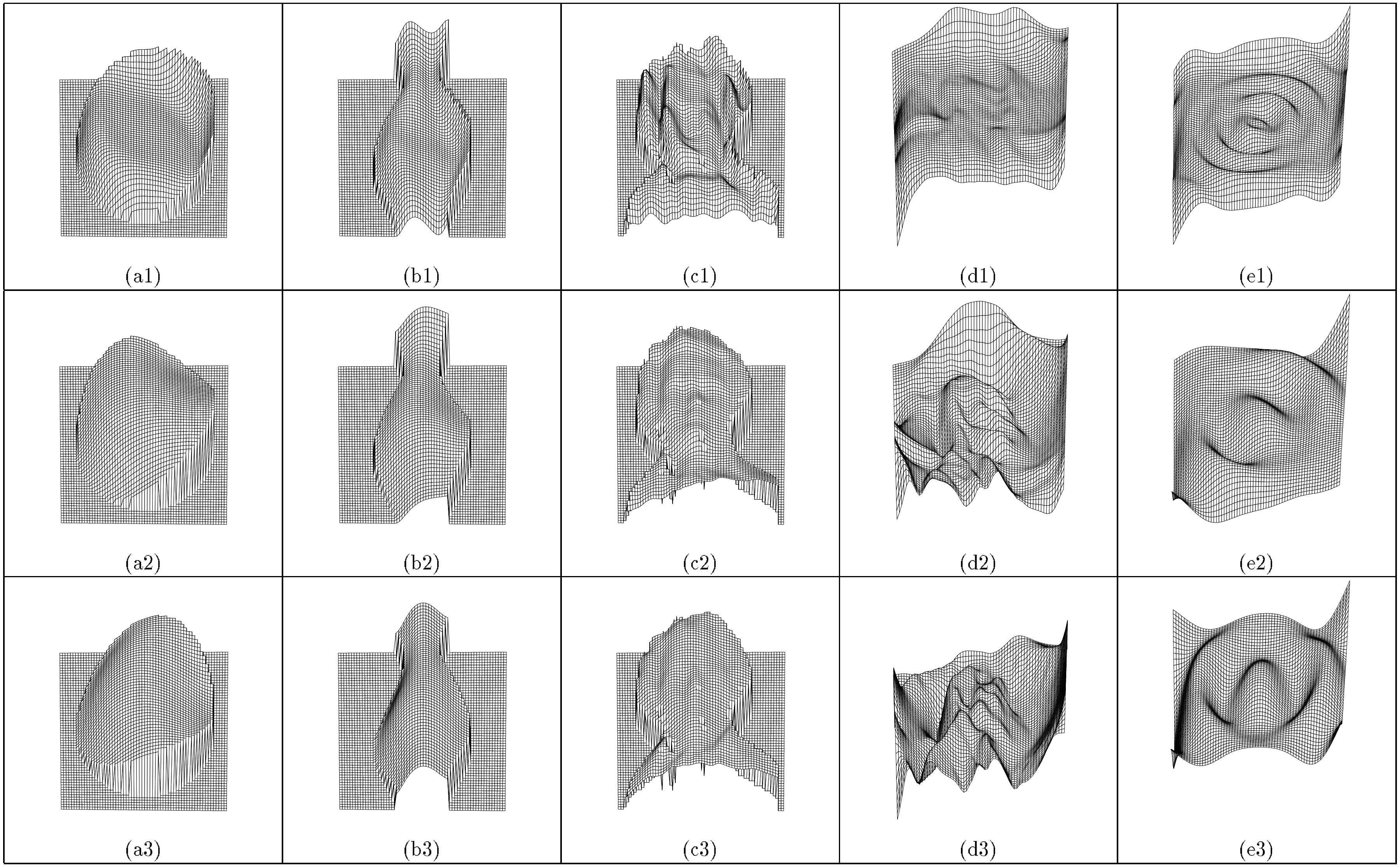


Figure 9: Results for Lee and Kuo's method on synthetic images: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The first row shows the results for test images with light source $(0, 0, 1)$. The second row shows the results for test images with light source $(5, 5, 7)$. The third row shows the results for test images with light source $(1, 0, 1)$.

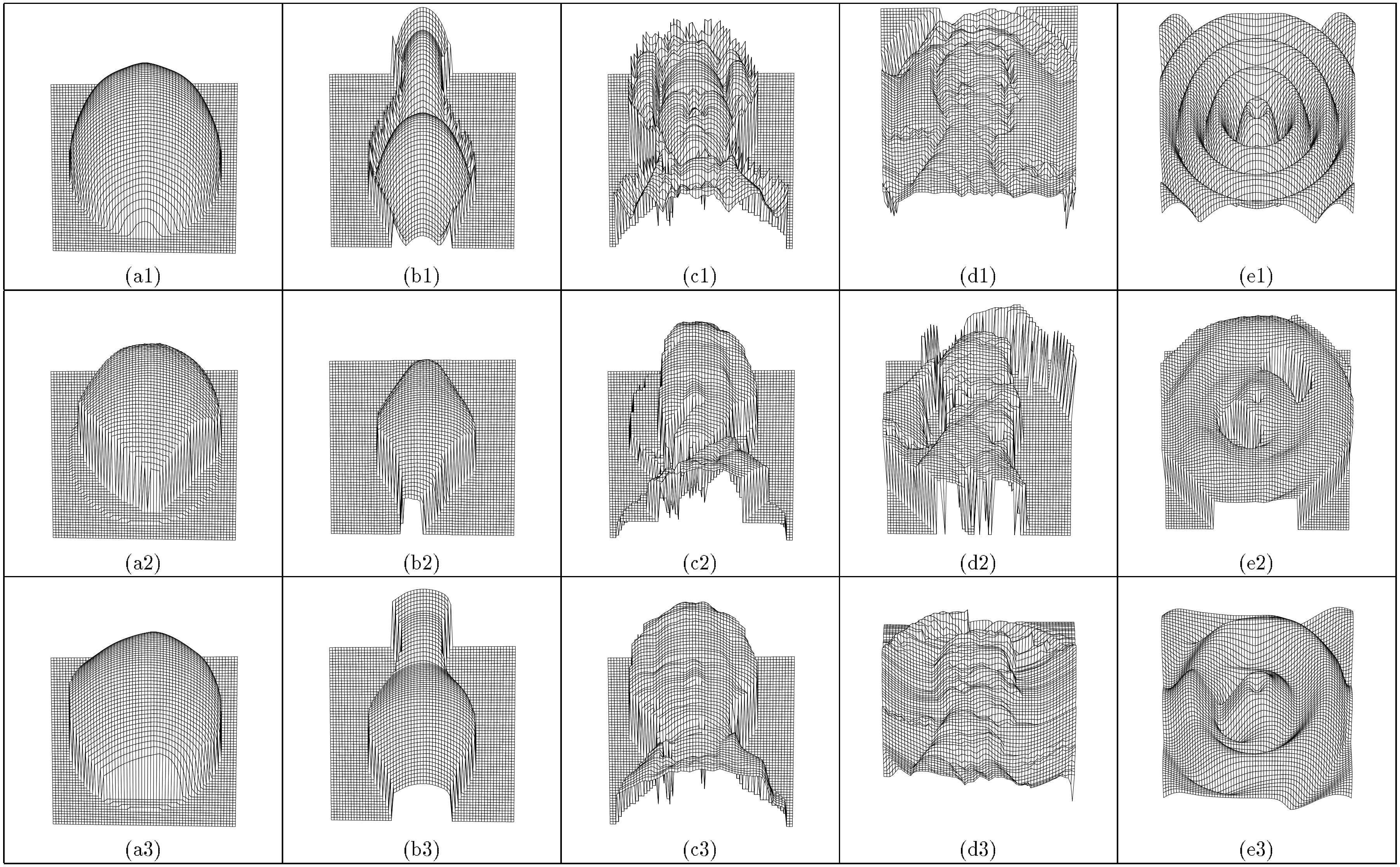
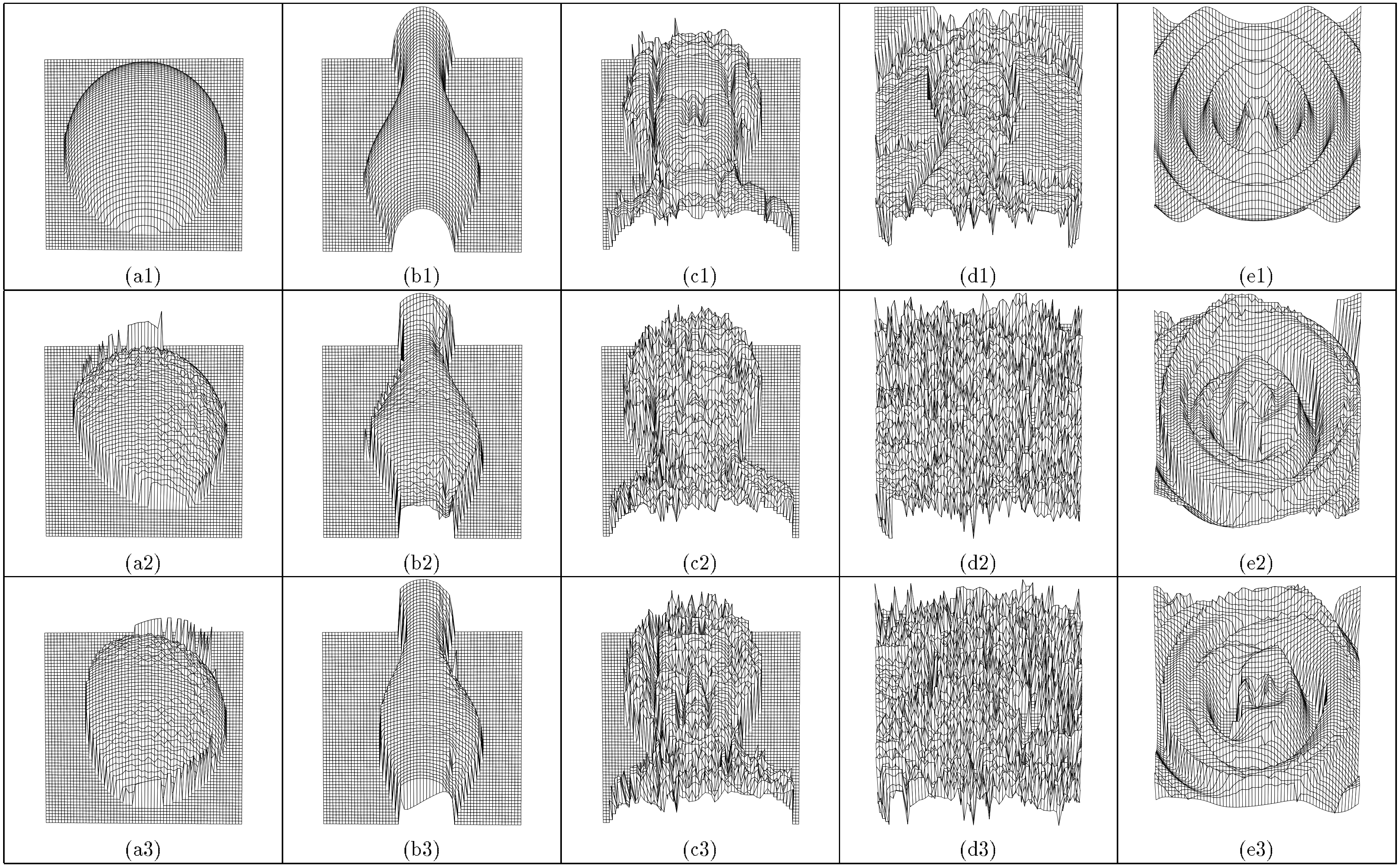

(a1) (b1) (c1) (d1) (e1)
(a2) (b2) (c2) (d2) (e2)
(a3) (b3) (c3) (d3) (e3)

Figure 10: Results for Bichsel and Pentland's method on synthetic images: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The first row shows the results for test images with light source $(0,0,1)$. The second row shows the results for test images with light source $(5,5,7)$. The third row shows the results for test images with light source $(1,0,1)$.

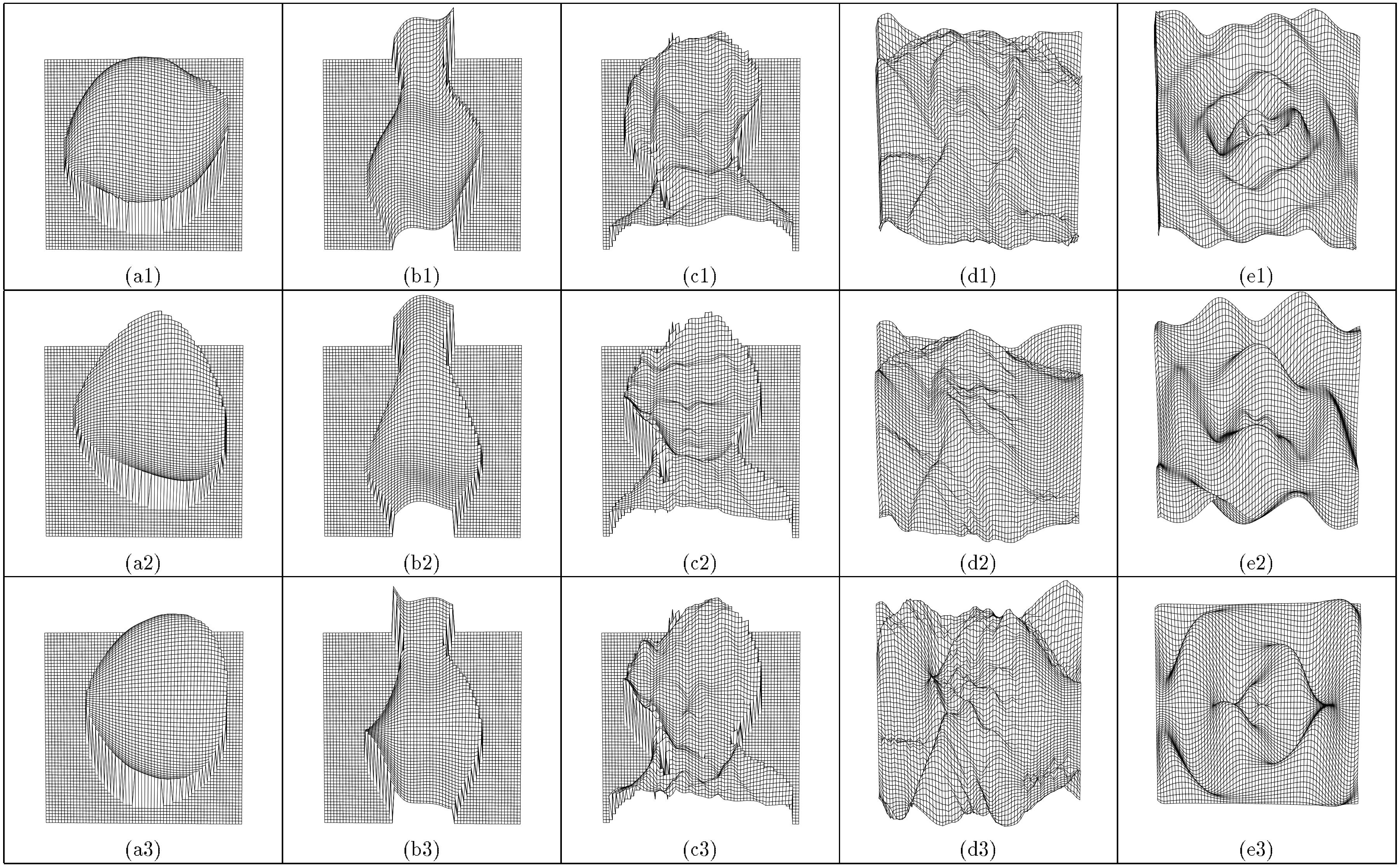


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 11: Results for Lee and Rosenfeld's method on synthetic images: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The first row shows the results for test images with light source $(0, 0, 1)$. The second row shows the results for test images with light source $(5, 5, 7)$. The third row shows the results for test images with light source $(1, 0, 1)$.


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 12: Results for Pentland's method on synthetic images: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The first row shows the results for test images with light source $(0, 0, 1)$. The second row shows the results for test images with light source $(5, 5, 7)$. The third row shows the results for test images with light source $(1, 0, 1)$.

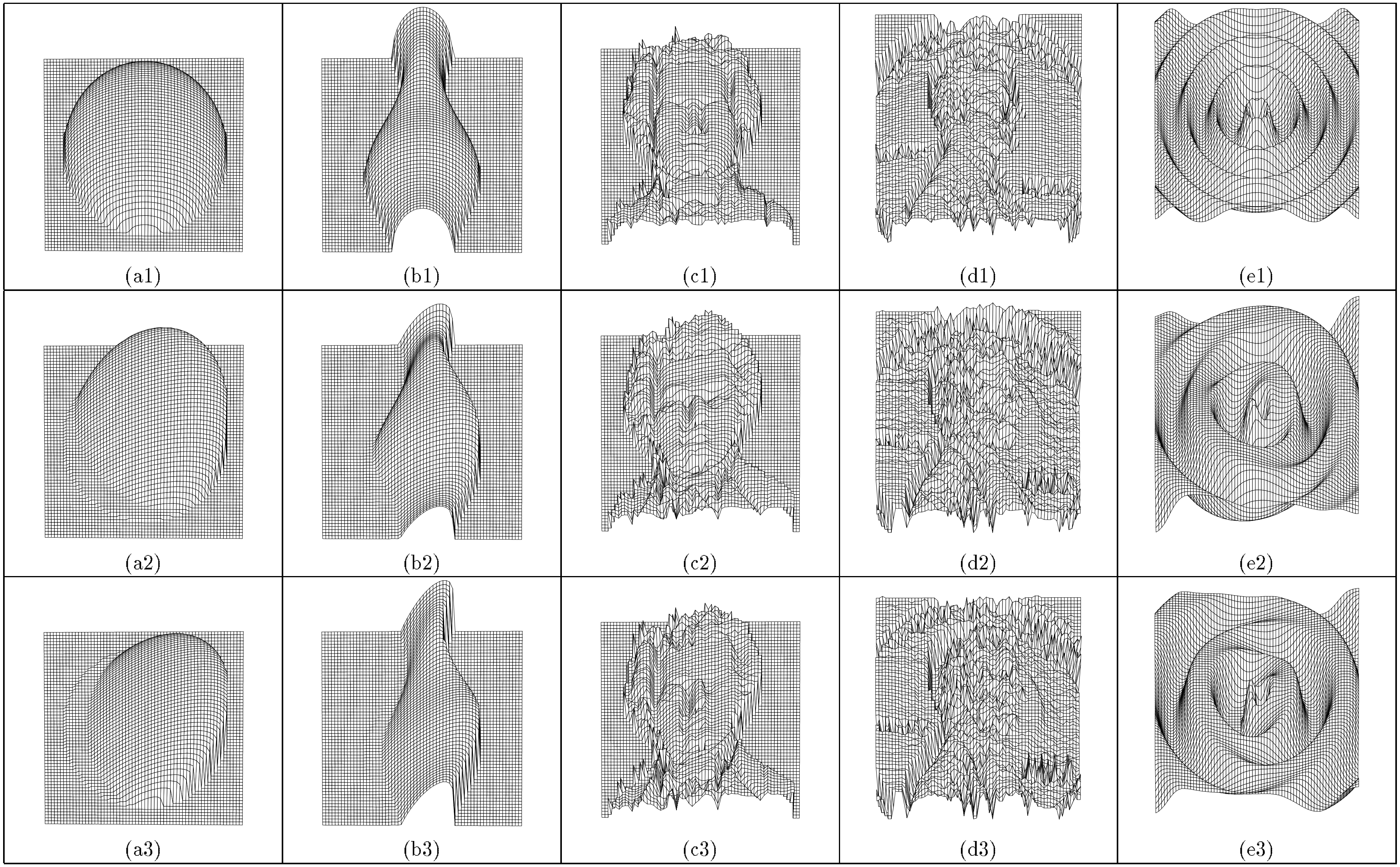
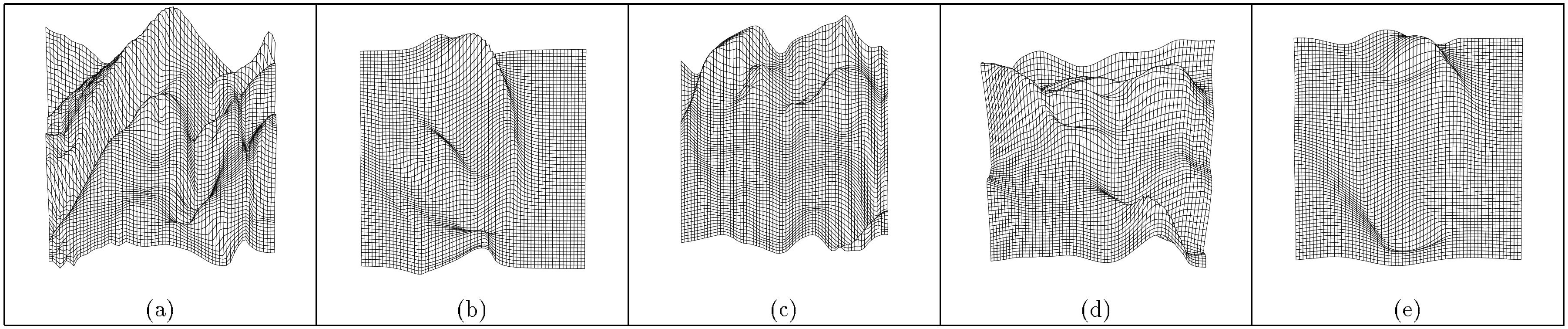
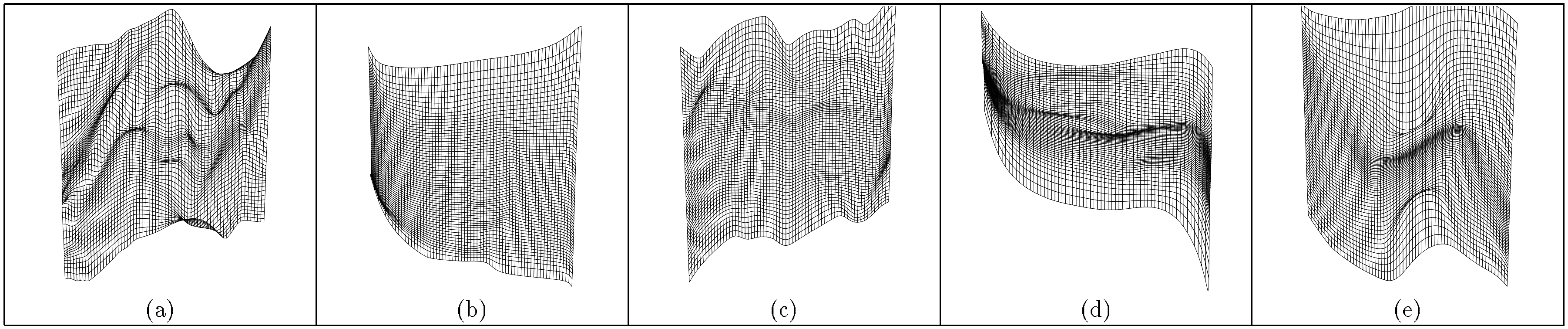


Figure 13: Results for Tsai and Shah's method on synthetic images: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The first row shows the results for test images with light source $(0, 0, 1)$. The second row shows the results for test images with light source $(5, 5, 7)$. The third row shows the results for test images with light source $(1, 0, 1)$.



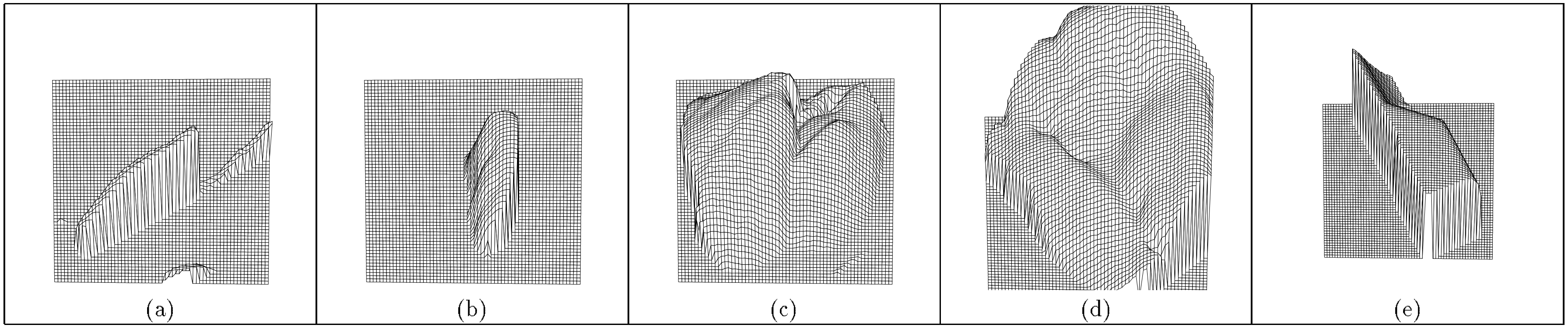
(a) (b) (c) (d) (e)

Figure 14: Results for Zheng and Chellappa's method on real images: (a) Lenna. (b) Mannequin. (c) Pepper. (d) David. (e) Vase.



(a) (b) (c) (d) (e)

Figure 15: Results for Lee and Kuo's method on real images: (a) Lenna. (b) Mannequin. (c) Pepper. (d) David. (e) Vase.



(a) (b) (c) (d) (e)

Figure 16: Results for Bichsel and Pentland's method on real images: (a) Lenna. (b) Mannequin. (c) Pepper. (d) David. (e) Vase.

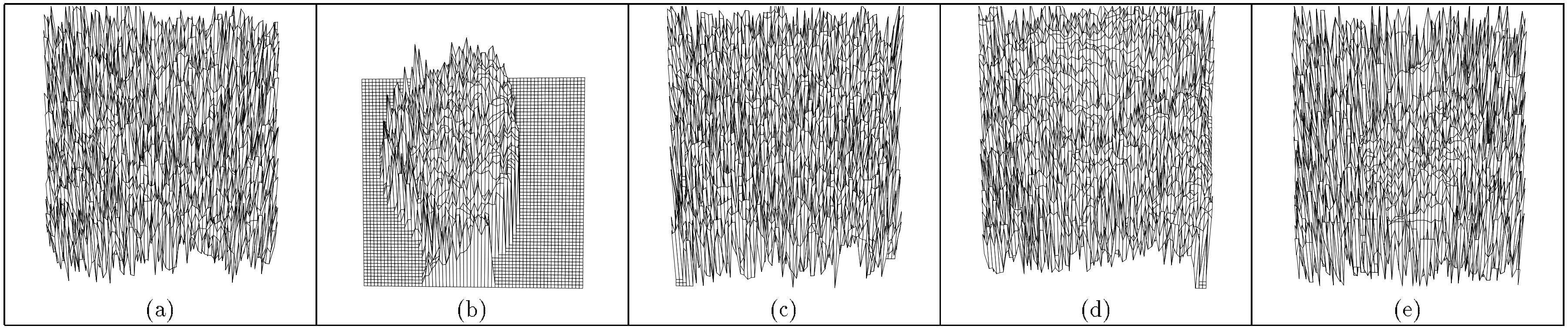

(a) (b) (c) (d) (e)

Figure 17: Results for Lee and Rosenfeld's method on real images: (a) Lenna. (b) Mannequin. (c) Pepper. (d) David. (e) Vase.

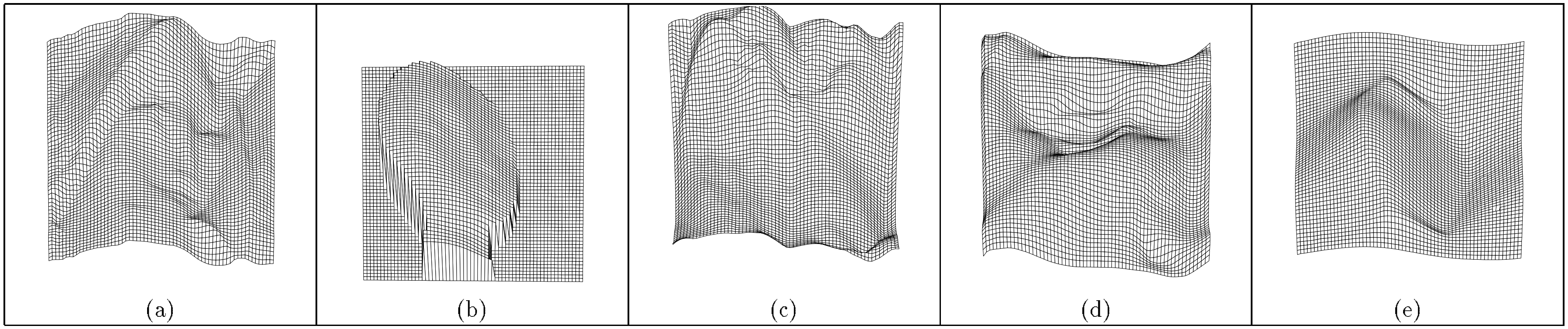

(a) (b) (c) (d) (e)

Figure 18: Results for Pentland's method on real images: (a) Lenna. (b) Mannequin. (c) Pepper. (d) David. (e) Vase.

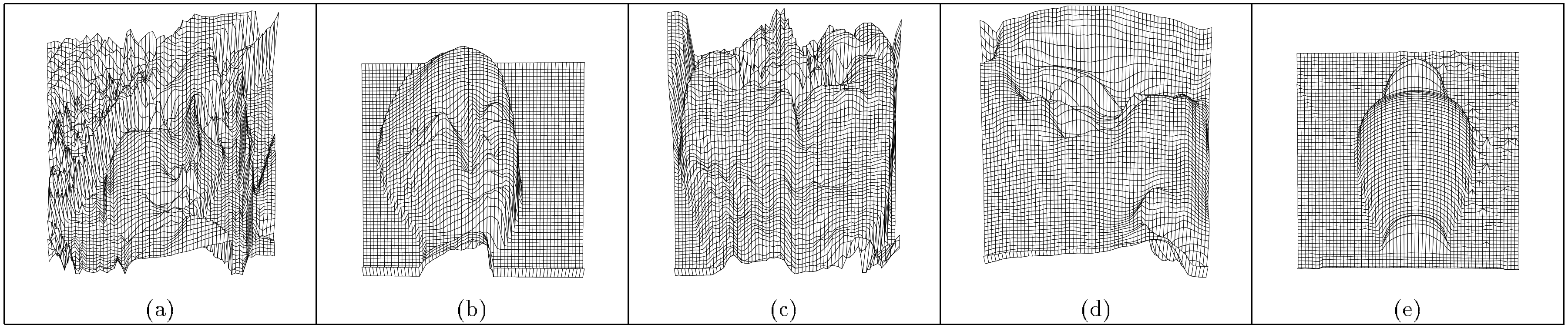

(a) (b) (c) (d) (e)

Figure 19: Results for Tsai and Shah's method on real images: (a) Lenna. (b) Mannequin. (c) Pepper. (d) David. (e) Vase.

recovered depth.

- **Difference images of the absolute depth error** (Figures 20 to 26), which provide the depth error distribution over the images to show the dependence of error on the underlying surface structure and image location. The depth error images are obtained by first calculating the absolute depth error at each point, then rescaling it using the minimum and the maximum value over the whole image. The regions which have the least depth error and the regions which have the most depth error can be easily identified from these images.

- **The histograms of the percentage depth error** (Figures 27 to 33), which show percentages of depth errors with respect to true depths and distributions of these percentages. The Y-axis of each plot represents the number of pixels. The X-axis of each plot represents the percentage depth error which is computed pixel by pixel using the following formula:

  $$\frac{|true\ depth\ -\ estimated\ depth|}{true\ depth} \times\ 100\%.$$

  There are some pixels with more than 100% error. This may happen at the points in the shadow areas, the points with convex/concave ambiguity, or at the object boundaries. All pixels which have more than 100% error are plotted as 101% error.

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.

7.1 Mean and Standard Deviation

From Table 1 (depth error), we can see that Lee and Rosenfeld's method gave the best results for the sphere; this is due to the spherical assumption used in the algorithm. However, it provides poor results for the real images. Tsai and Shah's approach produces very good results for images with light source $(0, 0, 1)$, but not for images with the light source from the side. This is due to the linearization of the reflectance function in terms of depth. When the intensity cannot globally reflect the depth information, the algorithm falls apart.

Pentland's approach also has this problem when the surface shape changes nonlinearly, as with spherical surfaces. Leclerc and Bobick's conjugate gradient approach produces the least depth error since the initial depth, used in our tests, is close to the ground truth. On the average, Bichsel and Pentland's minimum downhill approach gives good results even when the light source is from the side, and the results of Lee and Kuo's approach are close to the results of Zheng and Chellappa's approach.

The standard deviation of depth (Table 2) agrees with the average depth error in the sense that the one with smaller average error would have smaller standard deviation in most cases.

From the gradient error (table 3), we find that Pentland's approach gave the best results for most test images, except for Sphere and Sombrero images with light source direction $(0,0,1)$. This suggests that local intensity information is sufficient for a good shape estimation.

From these three error tables, we can see that there is no strict ordering for the accuracy of the algorithms, however, overall Leclerc and Bobick's is the best, since it uses good initial estimates. Lee and Kuo's places second, especially in terms of the gradient error. Zheng and Chellappa's algorithm takes third place, followed by Bichsel and Pentland's, Pentland's, Tsai and Shah's, and finally Lee and Rosenfeld's.

7.2 Difference Images

The depth error images are shown in Figures 20 to 26. From these images, we can see that Leclerc and Bobick's algorithm is still the best; most of errors occur at the boundaries of the objects. Among the remaining six algorithms which do not require accurate initial values, Lee and Rosenfeld's algorithm only has errors along half of the object boundaries for spheres with light sources $(5,5,7)$ and $(1,0,1)$. Bichsel and Pentland's algorithm also has low error distribution for these two images. The error is also very small in the center of all three vase images for Bichsel and Pentland's, and Lee and Rosenfeld's algorithms, and in the center of vase image with light source $(0,0,1)$ for Tsai and Shah's algorithm. For the Mozart images with light source $(5,5,7)$ and $(1,0,1)$, Lee and Kuo's, and Bichsel and Pentland's methods have the lowest error at the face areas. For Penny and Sombrero images, the errors for all six algorithms are equally distributed over the whole image.

7.3 Histograms

Figures 27 to 33 show the histograms of the percentage depth error. The histograms show that Leclerc and Bobick's algorithm gives the best results, since we used the truth depth with ±5% uniform random noise as the initial estimate. We can see that none of the other algorithms give good results for Penny and Sombrero images, since there are large number of pixels with more than 100% error in the histograms. For Mozart image, Lee and Kuo's algorithm gives the best results even for the image with the light source from the side. Lee and Rosenfeld's, and Bichsel and Pentland's algorithms give better results for Sphere and Vase images.

8 Timing

CPU timing (Table 4) is computed on a SUN SPARC 4. The disk I/O time is not included, and only the computational time is considered. All the synthetic images are 128 by 128, except for **Mozart**, whose size is 256 by 256. All the real images are 256 by 256, except **David**, whose size is 128 by 128. From the timing table (Table 4), we see that the three local approaches are significantly faster than the global approaches; their times depend only on the size of the input image. For the global approaches, time not only depends on the size of the input image, but also varies from scene to scene. Among the global approaches, Bichsel and Pentland's algorithm is the most efficient. Leclerc and Bobick's algorithm, without hierarchical structure,, takes the most time. Lee and Kuo's algorithm is also time consuming, since it involves multigrid iterations. Zheng and Chellappa's algorithm is reasonably fast with pyramid implementation. The order of the algorithms according to CPU time, from the slowest to the fastest, is Leclerc & Bobick's algorithm, Lee & Kuo's algorithm (in most cases, Leclerc & Bobick's algorithm is slower than Lee & Kuo's), then followed by Zheng & Chellappa's, Bichsel & Pentland's, Pentland's, Lee & Rosenfeld's, and Tsai & Shah.

Table 1: Average Z error for synthetic images.

Methods	**Images**														
	Sphere			**Vase**			**Mozart**			**Penny**			**Sombrero**		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Zheng & Chellappa	10.3	10.9	26.6	8.5	12.3	8.5	15.1	16.6	10.5	7.2	4.8	4.8	7.0	6.6	5.6
Leclerc & Bobick	2.1	3.6	3.7	1.8	3.0	*.*	1.7	4.7	5.5	2.2	*.*	*.*	1.2	*.*	*.*
Lee & Kuo	16.0	10.3	10.9	10.0	7.5	7.9	16.0	8.8	11.4	7.6	4.7	4.4	6.9	5.4	7.7
Bichsel & Pentland	0.7	9.4	5.2	10.0	8.8	7.9	20.5	17.8	7.7	12.1	8.0	8.4	13.7	11.0	6.4
Lee & Rosenfeld	0.8	3.8	4.3	8.1	8.4	11.0	18.3	17.8	17.6	11.3	8.2	7.9	11.7	8.8	8.7
Pentland	17.3	20.1	14.0	11.2	13.6	9.0	15.7	22.5	19.7	7.4	6.4	6.6	7.3	7.6	7.3
Tsai & Shah	0.1	16.4	16.4	8.3	11.8	12.7	18.5	20.1	20.0	11.0	8.5	8.6	12.6	10.2	10.1

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

Methods	**Images**														
	Sphere			**Vase**			**Mozart**			**Penny**			**Sombrero**		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Zheng & Chellappa	13.1	15.9	17.4	11.1	17.0	13.9	18.4	17.2	15.9	13.5	7.4	5.5	11.9	10.5	10.4
Leclerc & Bobick	2.4	5.0	5.1	2.9	4.1	*.*	2.1	7.1	7.7	3.1	*.*	*.*	2.4	*.*	*.*
Lee & Kuo	15.9	18.9	20.6	13.2	12.9	15.39	19.2	15.9	22.1	14.0	7.3	6.6	12.2	9.2	15.0
Bichsel & Pentland	1.2	13.4	9.0	13.8	13.6	16.9	37.4	21.9	14.6	23.4	11.7	16.6	26.6	20.3	12.5
Lee & Rosenfeld	0.4	5.8	6.6	14.6	16.4	22.3	33.0	29.8	30.3	21.2	15.2	14.8	22.5	17.0	16.9
Pentland	17.5	18.3	19.3	12.6	18.9	11.1	18.2	24.2	20.5	12.2	10.6	11.1	12.2	12.6	12.6
Tsai & Shah	0.1	20.9	21.0	15.0	16.9	19.7	33.3	30.7	30.5	20.6	15.4	15.6	24.3	18.4	18.4

S1, S2, and S3 stand for three different light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$, and '*.*' stands for unavailable data.

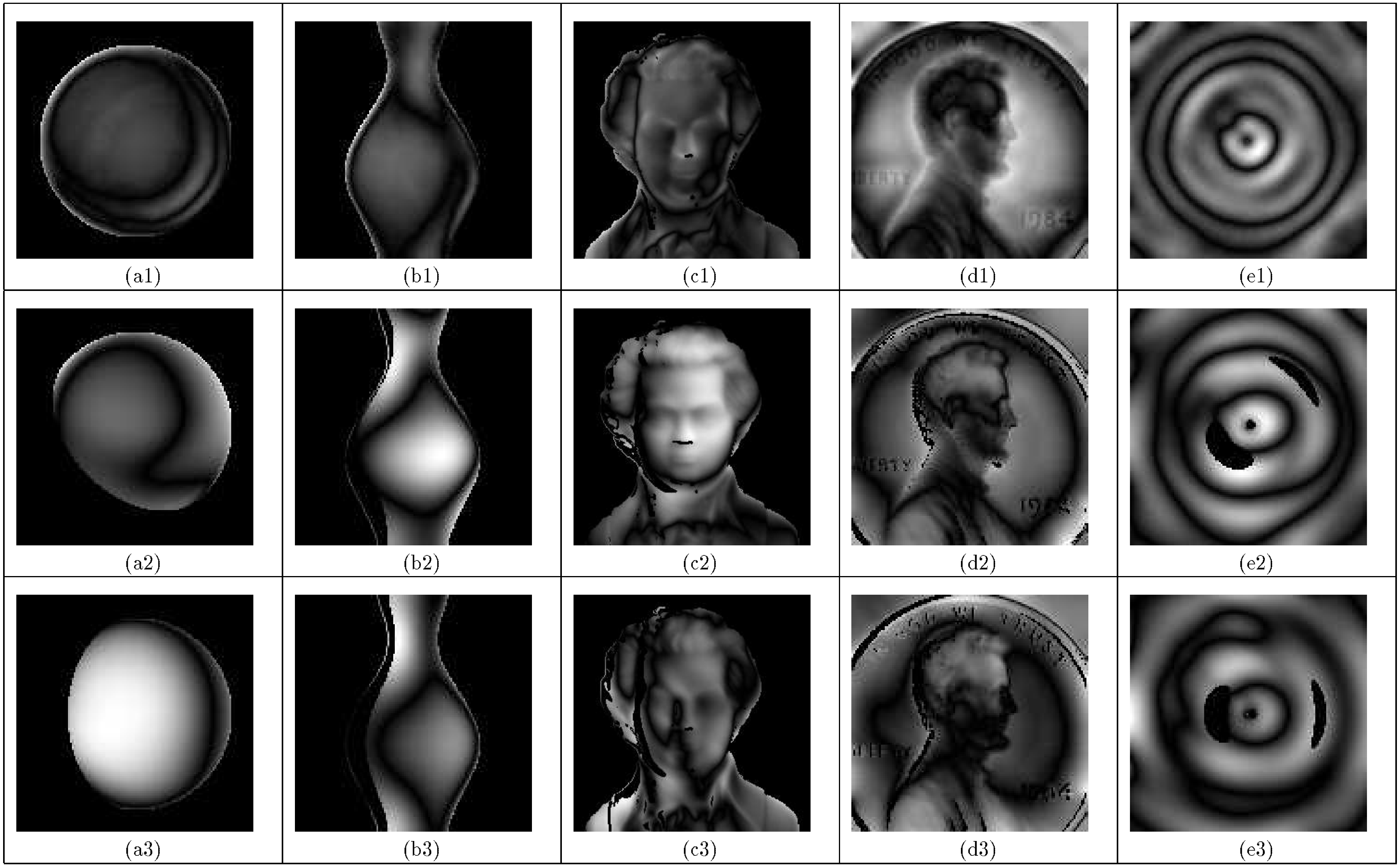
Table 3: Average p-q error for synthetic images.

Methods	Images														
	Sphere			**Vase**			**Mozart**			**Penny**			**Sombrero**		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Zheng & Chellappa	2.8	1.6	1.7	2.2	1.5	1.3	2.3	1.1	1.1	1.3	1.1	1.0	1.3	1.0	0.7
Leclerc & Bobick	0.8	4.5	4.4	1.2	3.1	*.*	0.5	8.4	9.5	1.2	*.*	*.*	0.6	*.*	*.*
Lee & Kuo	2.3	1.4	1.4	1.6	0.9	0.9	1.7	0.7	0.6	1.3	1.1	1.0	0.8	0.7	0.6
Bichsel & Pentland	0.3	5.8	2.5	2.7	4.9	1.9	3.1	8.1	1.9	1.7	4.4	1.1	1.2	3.3	0.5
Lee & Rosenfeld	0.1	6.5	6.7	1.3	3.3	2.2	6.8	13.7	12.8	4.3	8.4	7.0	1.3	2.5	2.3
Pentland	2.2	2.9	4.7	1.8	1.3	1.2	1.3	1.3	1.3	1.3	1.3	1.2	1.1	1.1	1.0
Tsai & Shah	0.1	0.9	0.9	1.4	1.4	2.6	6.7	5.5	5.6	4.2	5.2	4.8	1.2	1.5	1.5

Table 4: CPU time (in seconds) for synthetic images.

Methods	Images														
	Sphere			**Vase**			**Mozart**			**Penny**			**Sombrero**		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Zheng & Chellappa	13.1	98.54	104.78	18.1	60.3	70.8	49.6	416.40	409.5	17.8	193.3	78.1	17.1	110.6	148.6
Leclerc & Bobick	2267.8	2337.8	2310.7	2343.3	1186.4	*.*	14517.3	8477.1	17140.6	2864.2	*.*	*.*	4258.9	*.*	*.*
Lee & Kuo	188.4	1195.8	1167.8	188.4	354.9	2584.9	771.5	18278.2	20249.5	42.3	2647.0	2429.9	188.3	8951.2	6642.9
Bichsel & Pentland	6.9	6.7	6.9	7.3	6.7	6.1	27.6	27.4	26.9	7.3	6.9	6.7	7.3	6.9	6.7
Lee & Rosenfeld	1.6	1.8	1.7	1.5	1.9	1.7	6.2	7.4	6.9	1.7	2.2	1.9	1.7	2.4	1.9
Pentland	3.8	4.3	4.1	4.2	4.2	4.2	17.7	17.7	18.4	4.1	4.2	4.4	4.4	4.2	4.2
Tsai & Shah	0.8	1.3	1.2	0.9	1.2	1.2	3.8	4.8	4.8	0.7	1.2	1.2	0.8	1.2	1.2

S1, S2, and S3 stand for three different light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$. '*.*' stands for unavailable data.

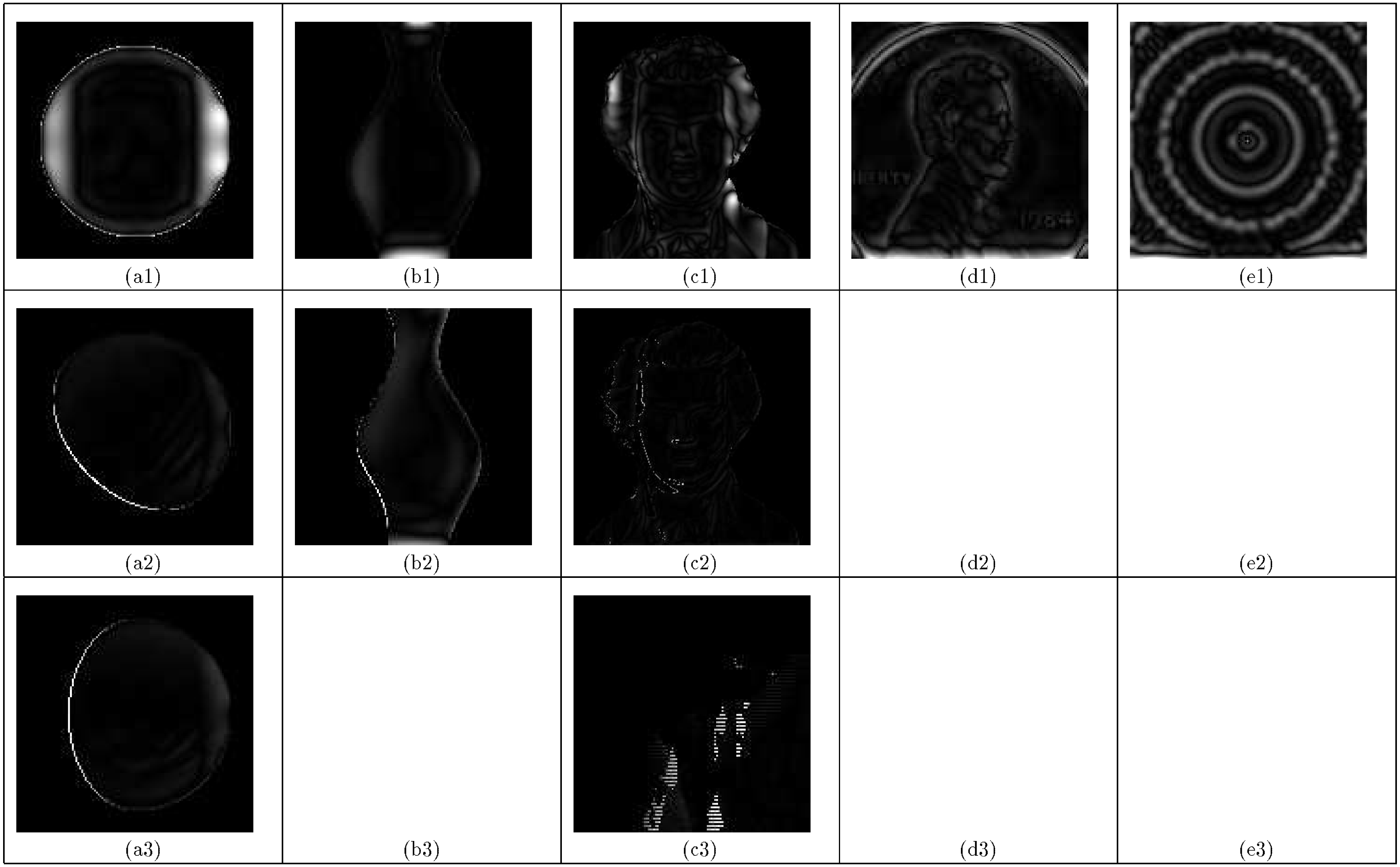


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 20: Depth error images for Zheng and Chellappa's method: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The absolute depth errors were normalized between 0-255. Dark areas represent small error, and light areas represent large error. The first row shows the errors for test images with light source $(0, 0, 1)$. The second row shows the errors for test images with light source $(5, 5, 7)$. The third row shows the errors for test images with light source $(1, 0, 1)$.

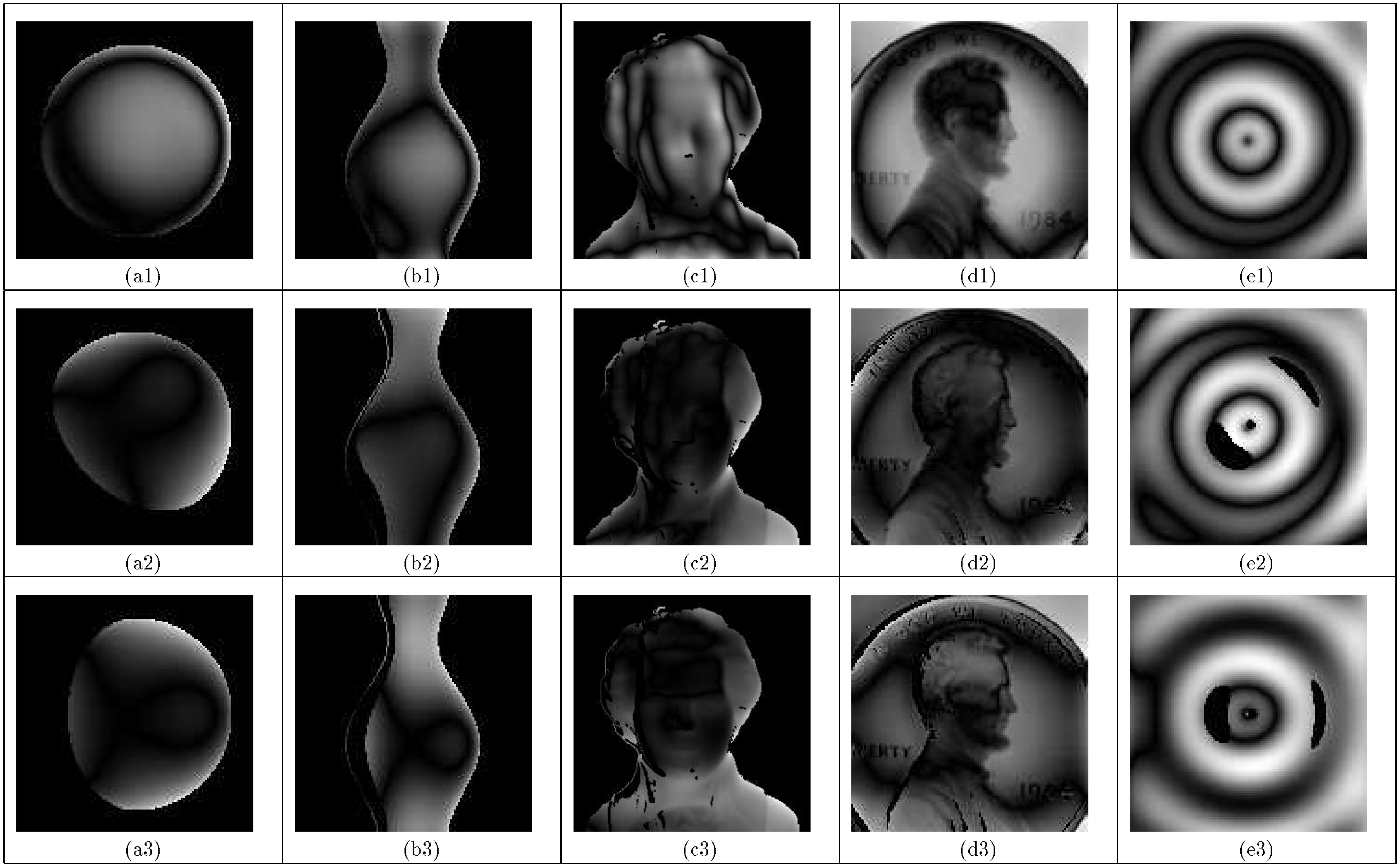


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 21: Depth error images for Leclerc and Bobick's method: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The absolute depth errors were normalized between 0-255. Dark areas represent small error, and light areas represent large error. The first row shows the errors for test images with light source $(0, 0, 1)$. The second row shows the errors for test images with light source $(5, 5, 7)$. The third row shows the errors for test images with light source $(1, 0, 1)$. Depth error images for Vase with light source $(1, 0, 1)$, (b3), Penny and Sombereo with light source $(5, 5, 7)$ and $(1, 0, 1)$, (d2, d3, e2 and e3) are not shown here, since we could not find a good initial value for the smoothness term.

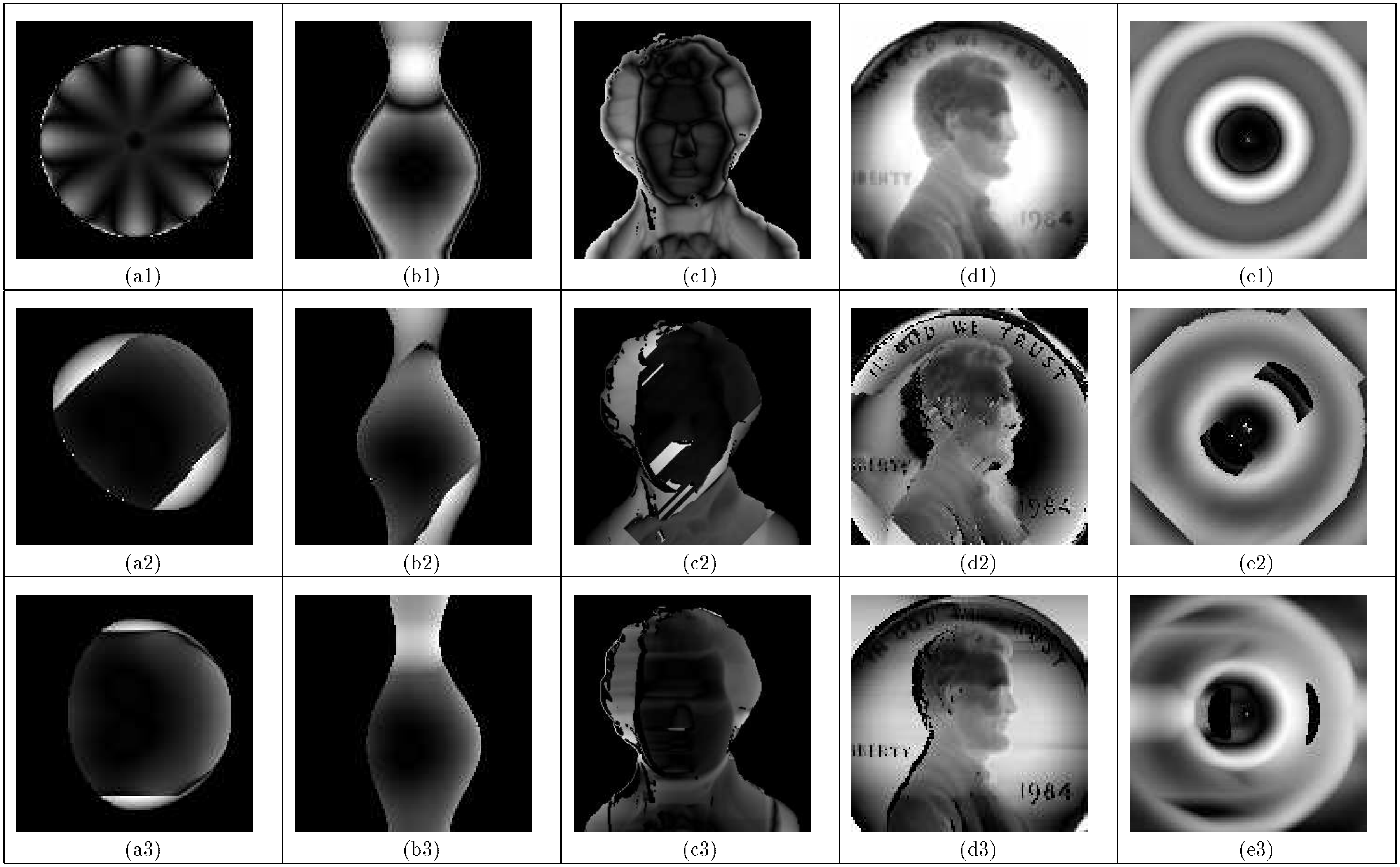


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 22: Depth error images for Lee and Kuo's method: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The absolute depth errors were normalized between 0-255. Dark areas represent small error, and light areas represent large error. The first row shows the errors for test images with light source $(0, 0, 1)$. The second row shows the errors for test images with light source $(5, 5, 7)$. The third row shows the errors for test images with light source $(1, 0, 1)$.

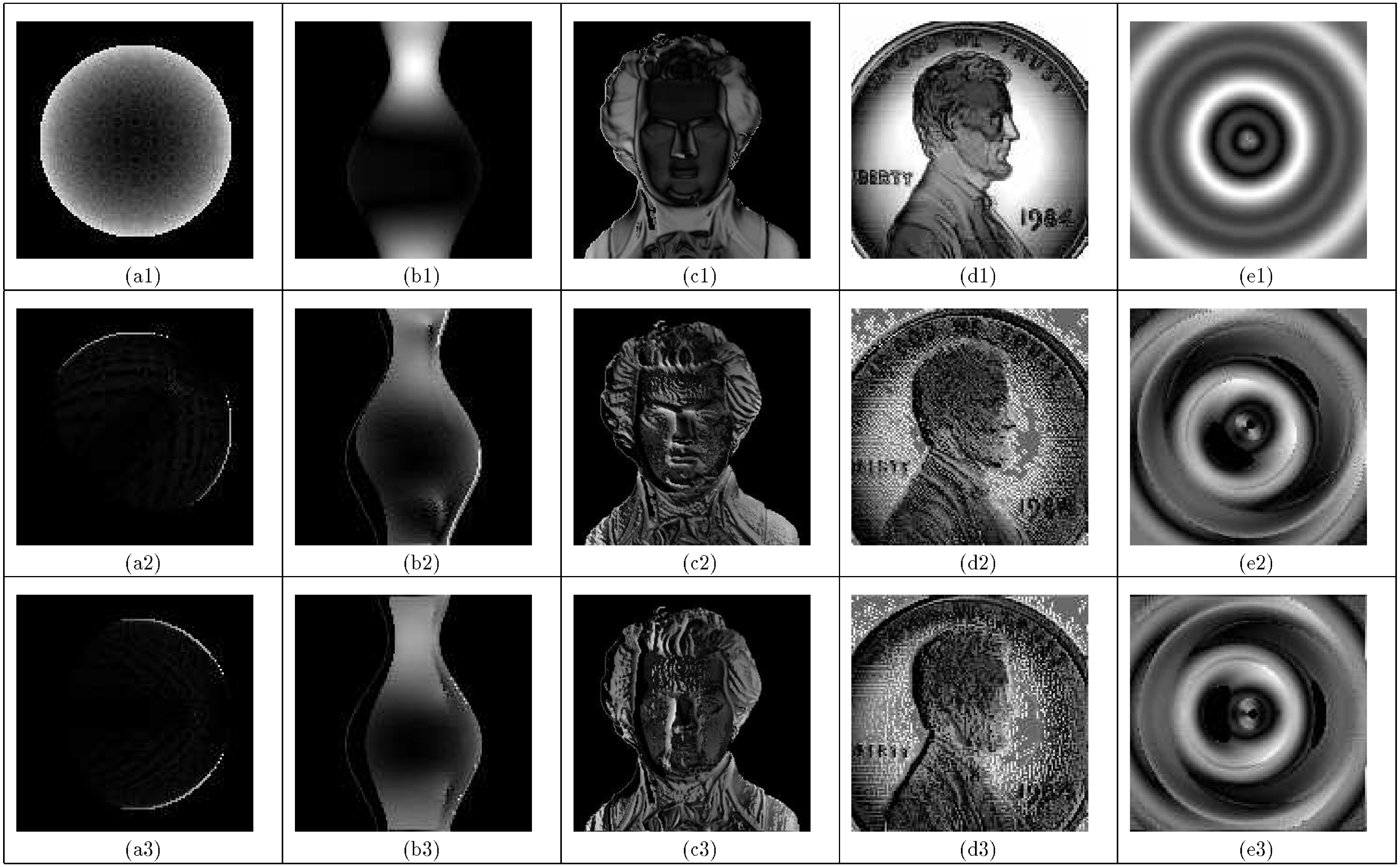


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 23: Depth error images for Bichsel and Pentland's method: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The absolute depth errors were normalized between 0-255. Dark areas represent small error, and light areas represent large error. The first row shows the errors for test images with light source $(0, 0, 1)$. The second row shows the errors for test images with light source $(5, 5, 7)$. The third row shows the errors for test images with light source $(1, 0, 1)$.

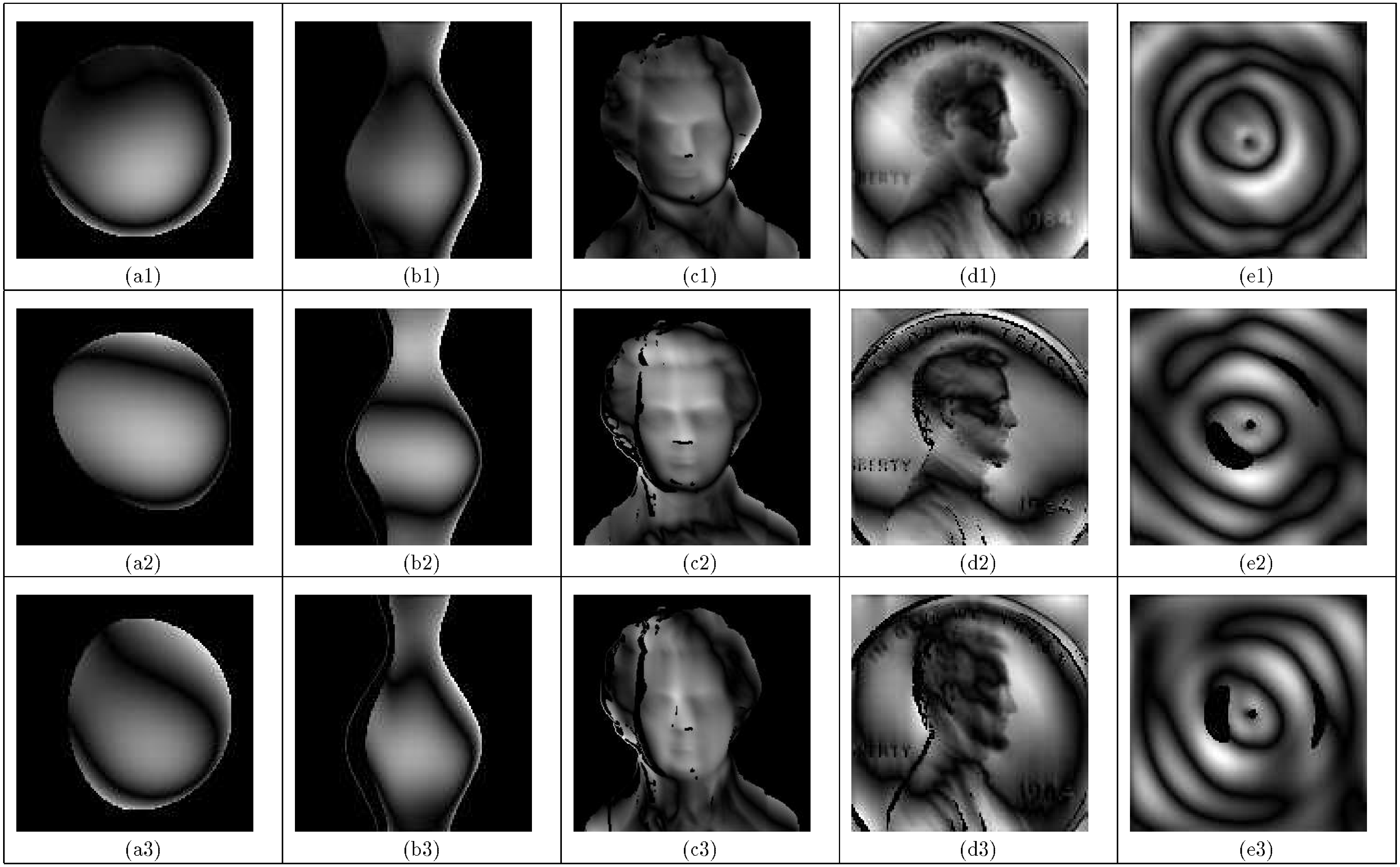


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 24: Depth error images for Lee and Rosenfeld's method: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The absolute depth errors were normalized between 0-255. Dark areas represent small error, and light areas represent large error. The first row shows the errors for test images with light source $(0, 0, 1)$. The second row shows the errors for test images with light source $(5, 5, 7)$. The third row shows the errors for test images with light source $(1, 0, 1)$.

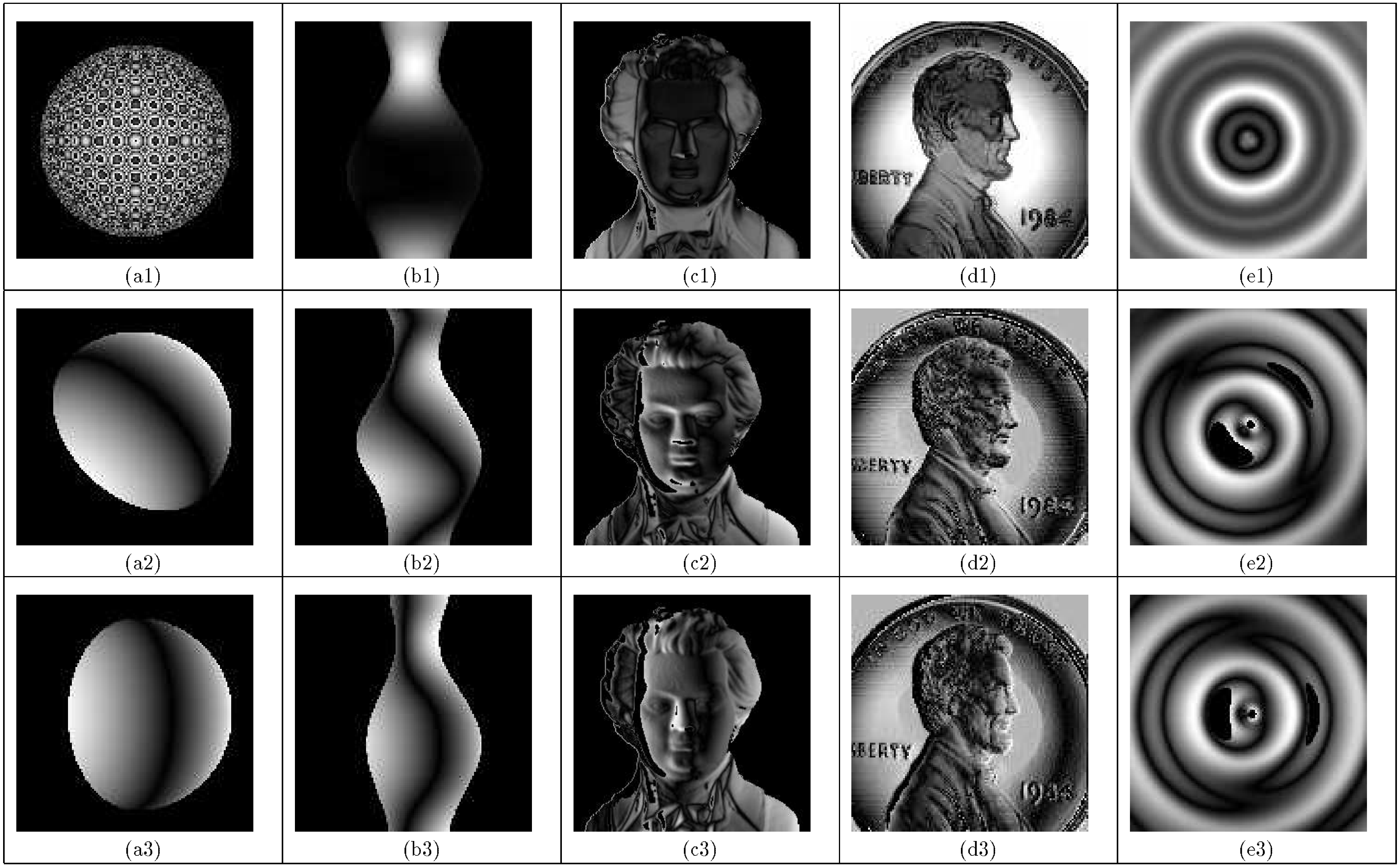


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 25: Depth error images for Pentland's method: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The absolute depth errors were normalized between 0-255. Dark areas represent small error, and light areas represent large error. The first row shows the errors for test images with light source $(0, 0, 1)$. The second row shows the errors for test images with light source $(5, 5, 7)$. The third row shows the errors for test images with light source $(1, 0, 1)$.

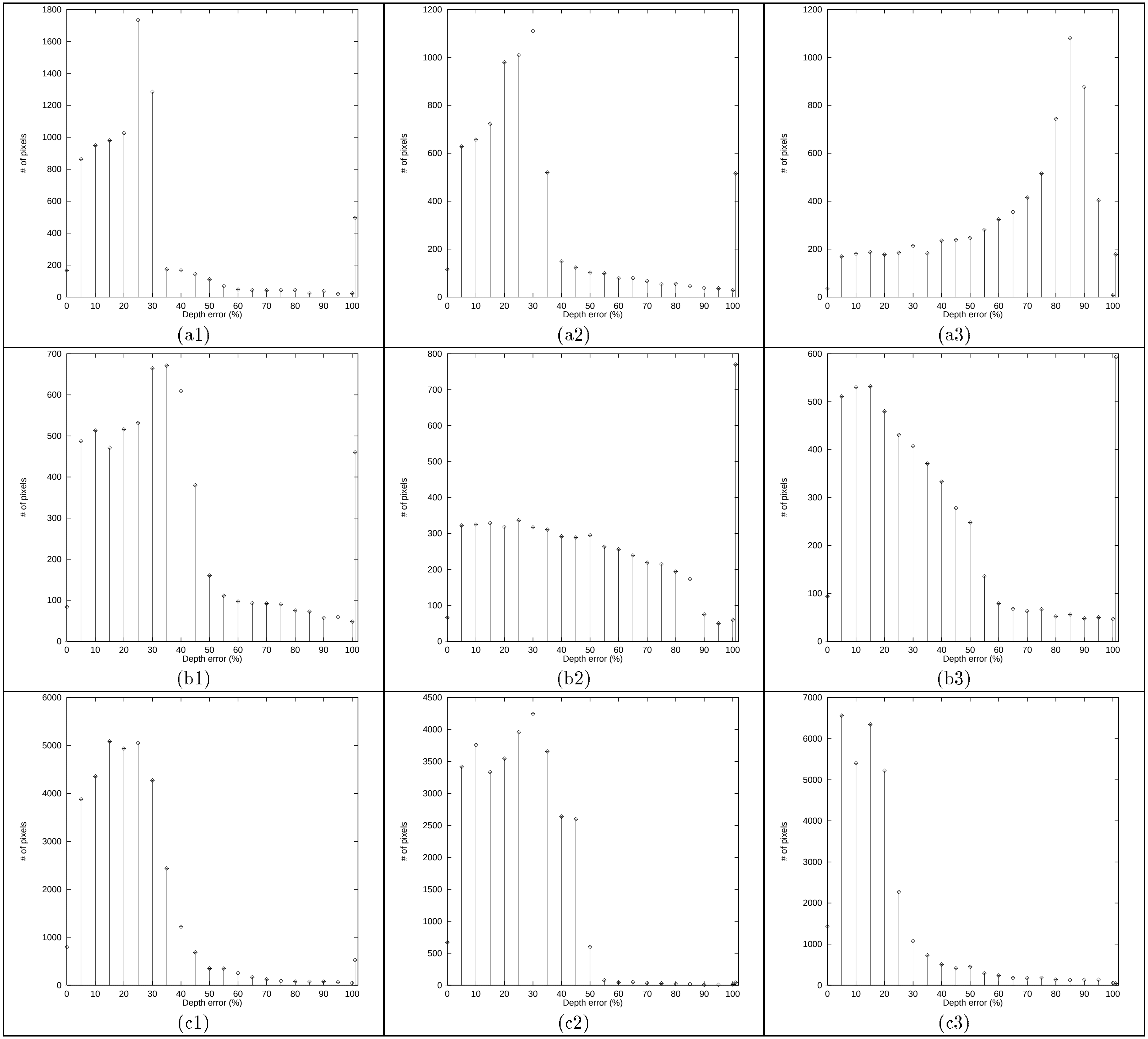


(a1) (b1) (c1) (d1) (e1)

(a2) (b2) (c2) (d2) (e2)

(a3) (b3) (c3) (d3) (e3)

Figure 26: Depth error images for Tsai and Shah's method: (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. The absolute depth errors were normalized between 0-255. Dark areas represent small error, and light areas represent large error. The first row shows the errors for test images with light source $(0, 0, 1)$. The second row shows the errors for test images with light source $(5, 5, 7)$. The third row shows the errors for test images with light source $(1, 0, 1)$.

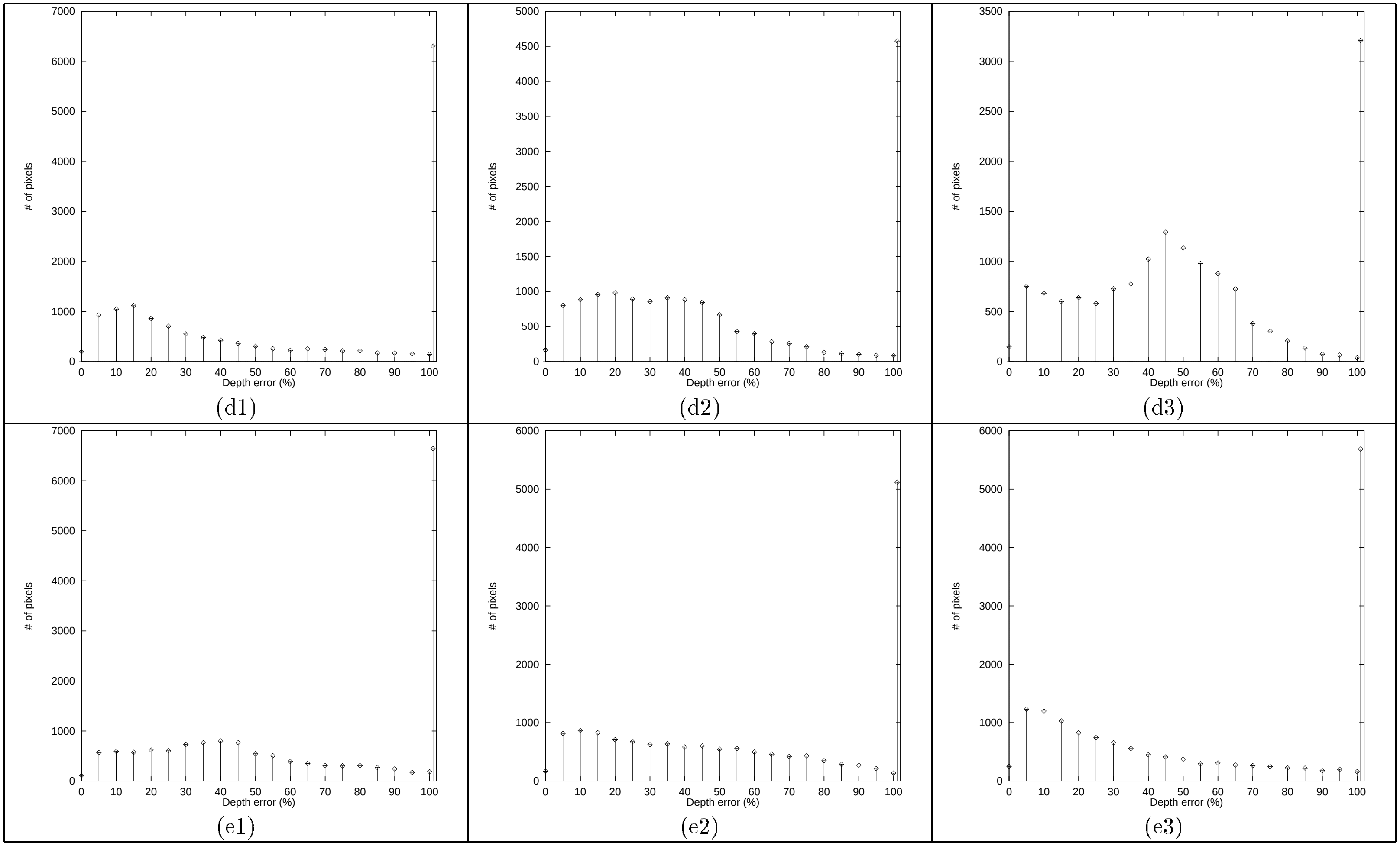


(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

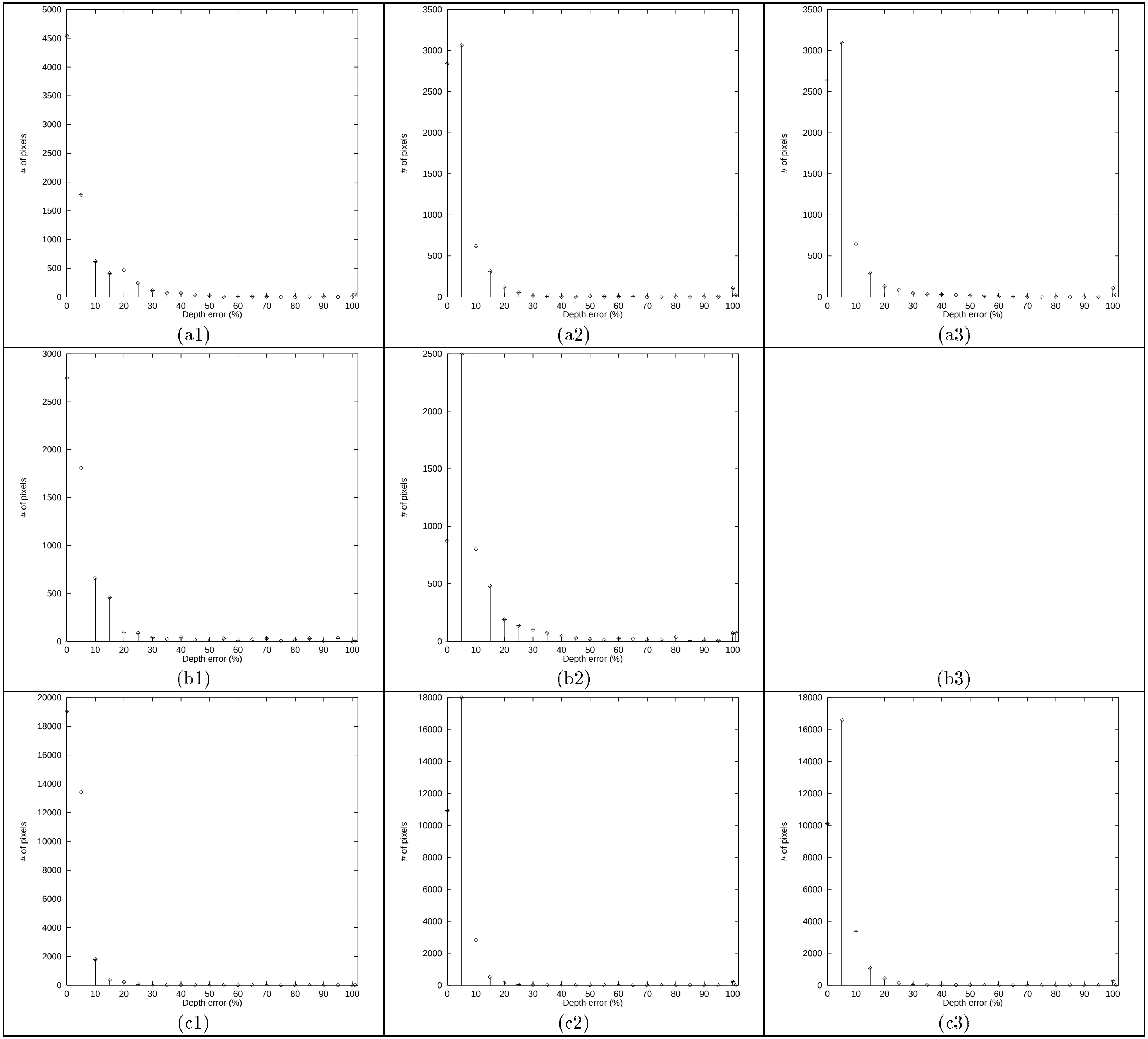
Figure 27: See caption on next page.



(d1) (d2) (d3)

(e1) (e2) (e3)

Figure 27: The histograms of depth error for Zheng and Chellappa's method. (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. Each bar in the histograms represents the summation of number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error. In each row, 1, 2, and 3 indicate respectively light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$.

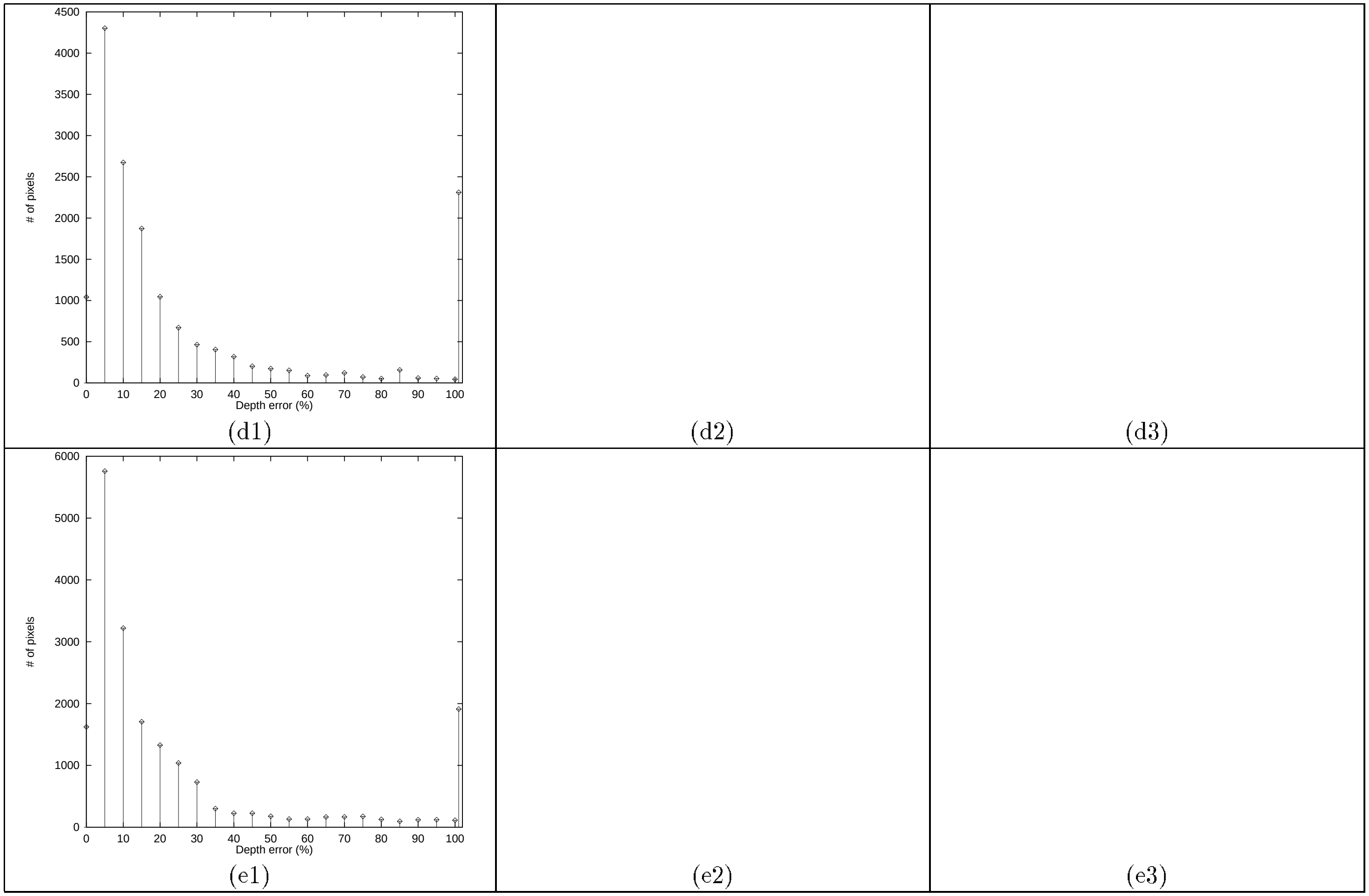


(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

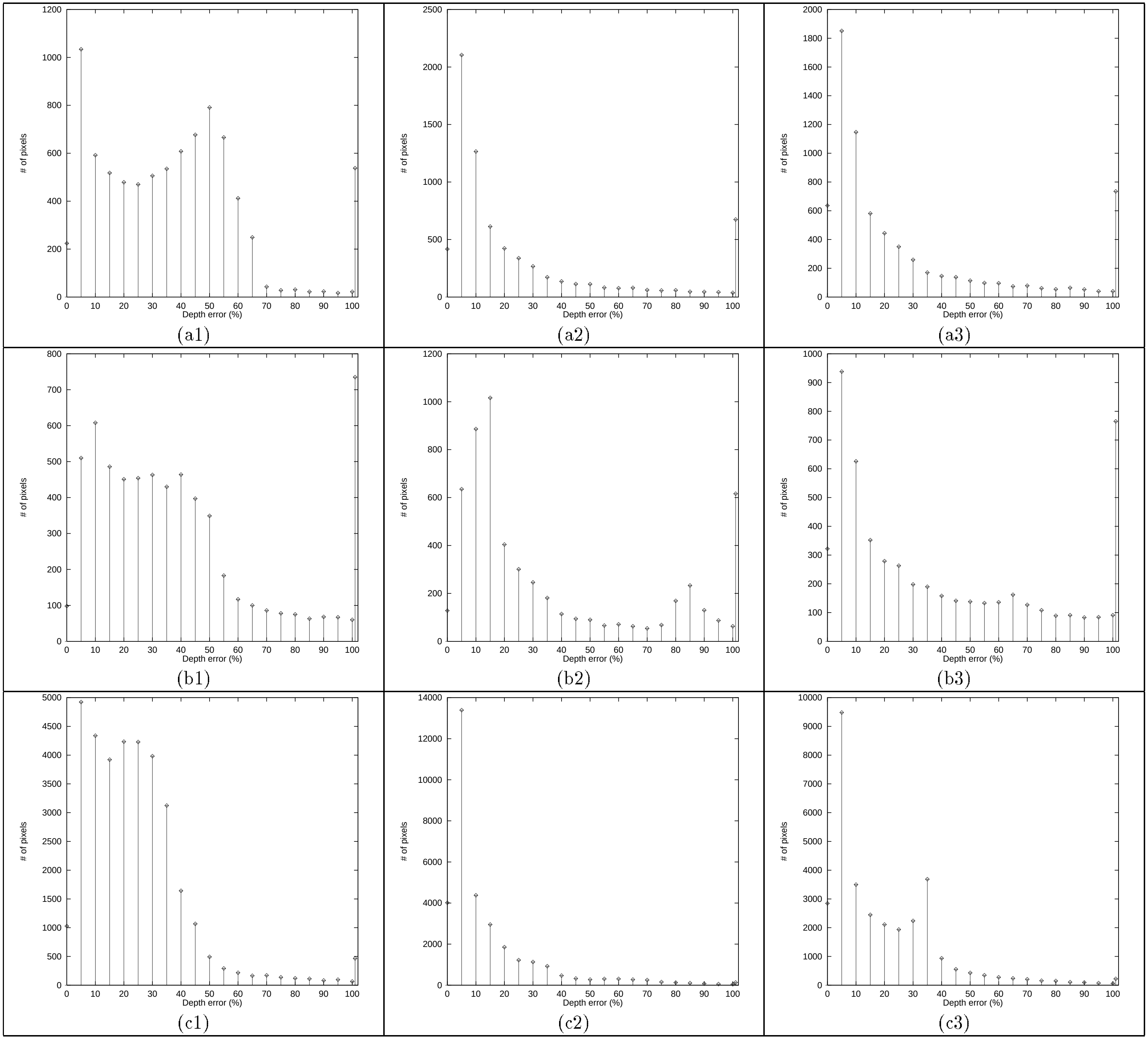
Figure 28: See caption on next page.



(d1) (d2) (d3)

(e1) (e2) (e3)

Figure 28: The histogram of percentage depth error for Leclerc and Bobick's method. (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. Each bar in the histograms represents the summation of number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error. In each row, 1, 2, and 3 indicate respectively light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$. The histogram for the Vase image with light source $(1, 0, 1)$,(b3), Penny and Sombrero images with light source $(5, 5, 7)$ and $(1, 0, 1)$,(d2, d3, e2, and e3) are not reported here, because we could not find a good initial value for the smoothness term.

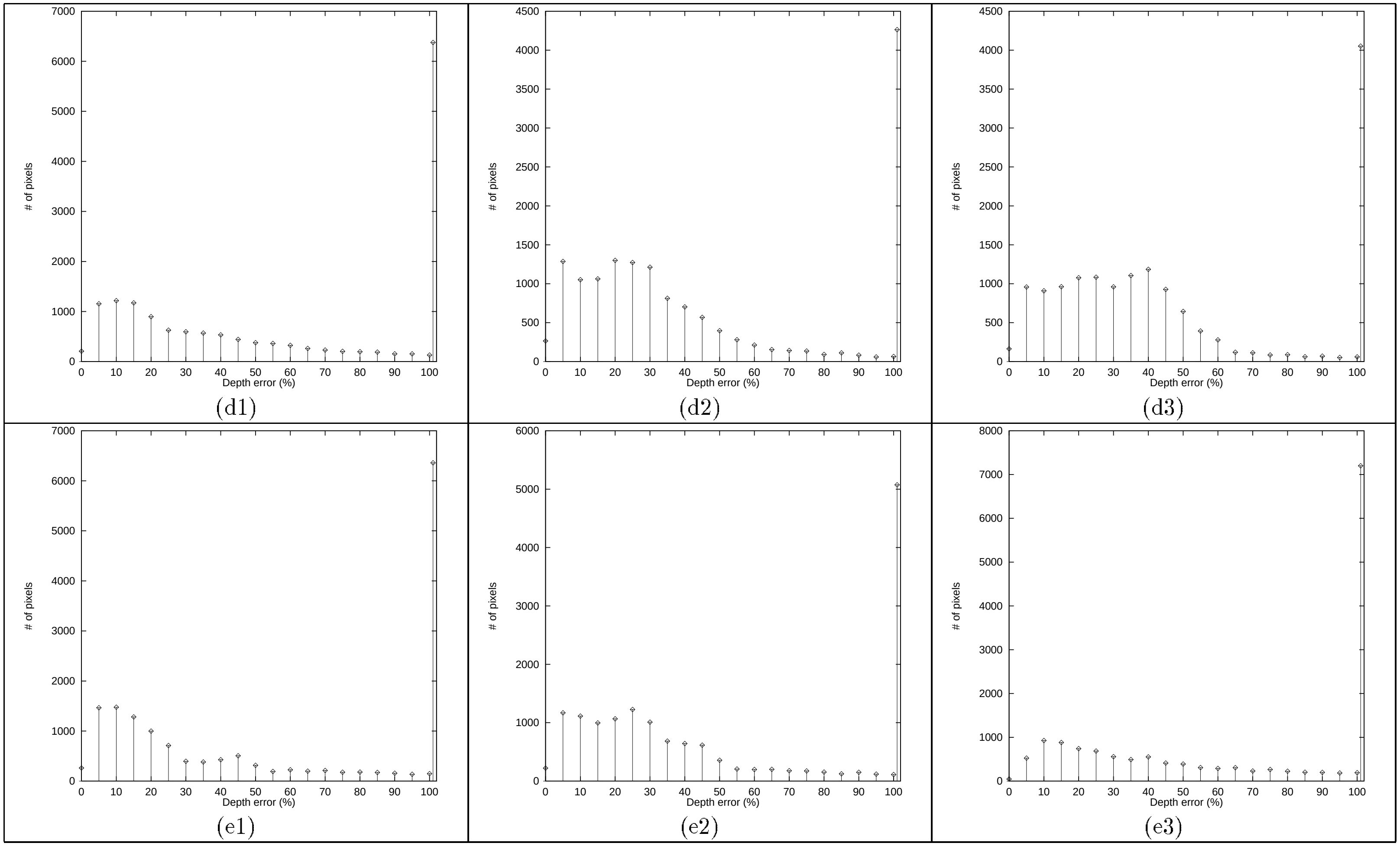


(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

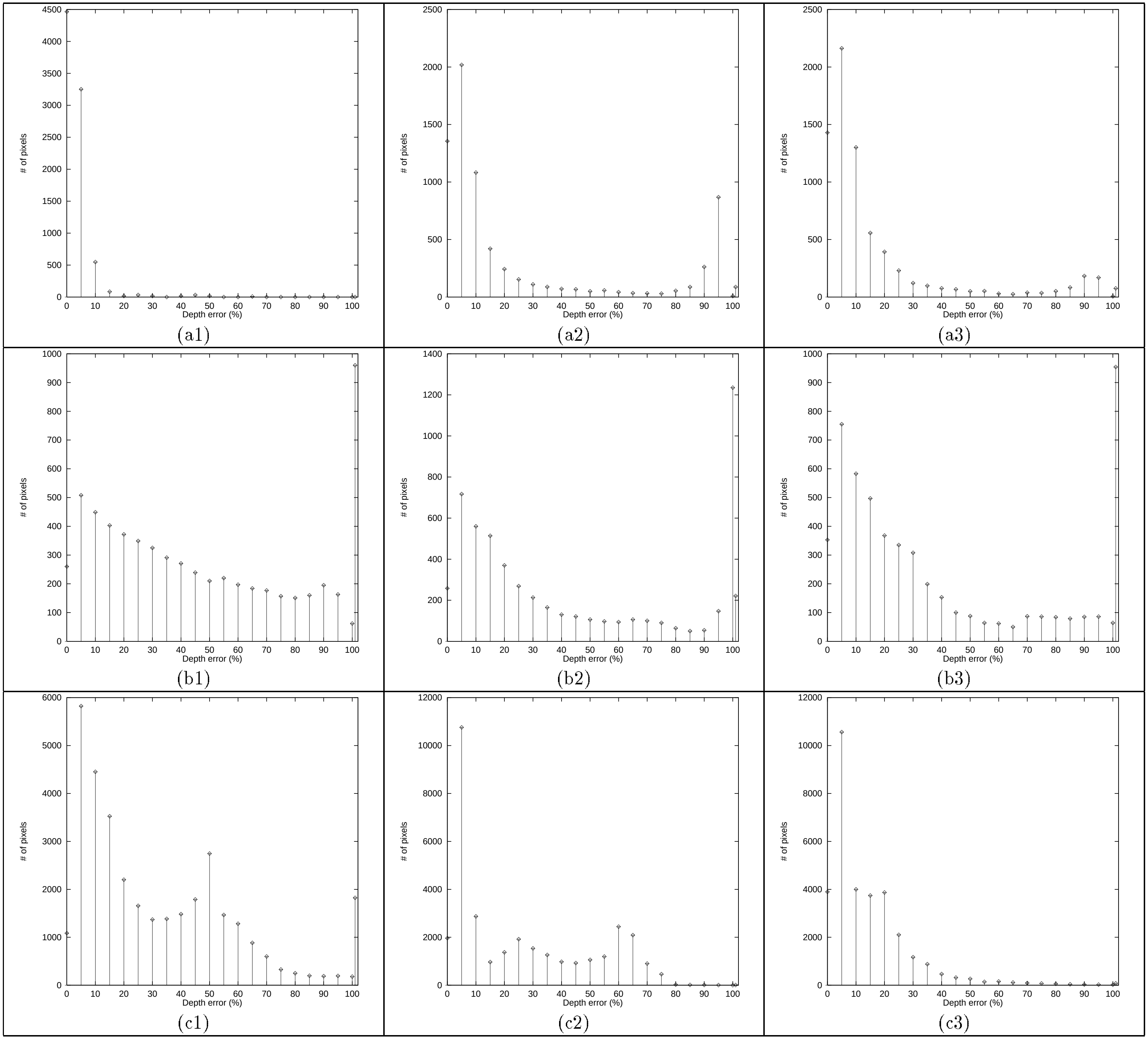
Figure 29: See caption on next page.



(d1) (d2) (d3)

(e1) (e2) (e3)

Figure 29: The histogram of percentage depth error for Lee and Kuo's method. (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. Each bar in the histograms represents the summation of number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error. In each row, 1, 2, and 3 indicate respectively light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$.



(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

Figure 30: See caption on next page.

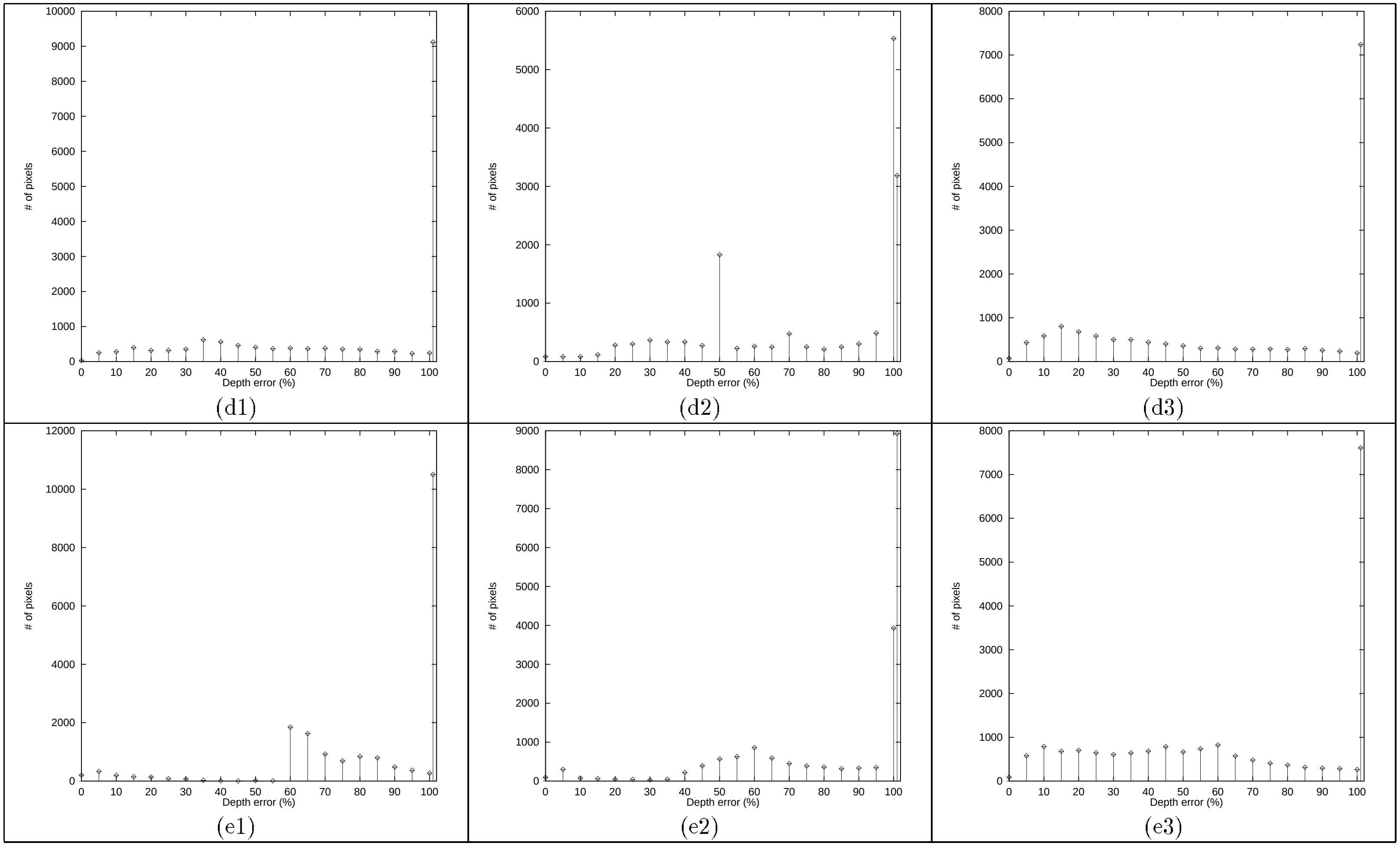
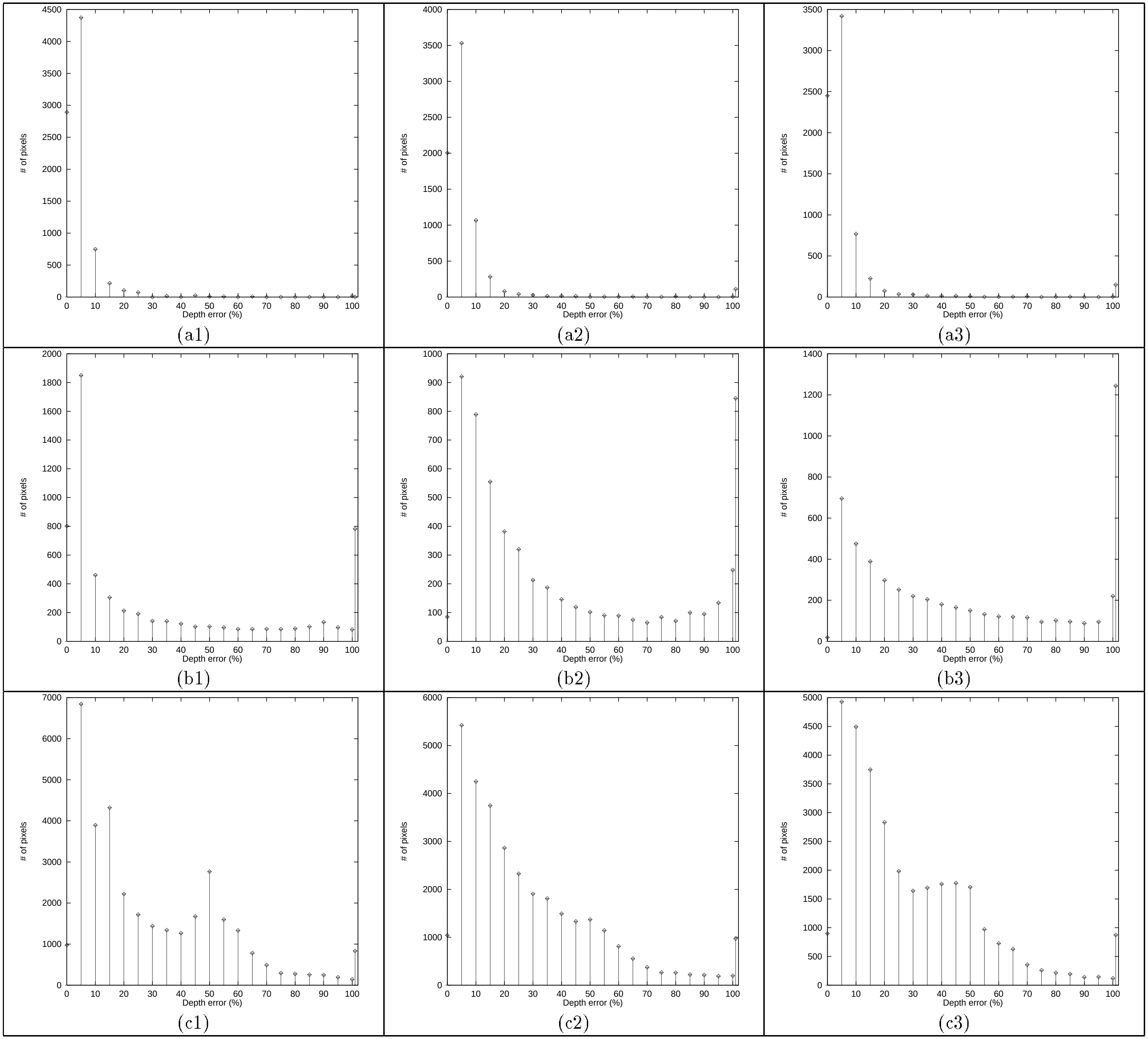


Figure 30: The histogram of percentage depth error for Bichsel and Pentland's method. (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. Each bar in the histograms represents the summation of number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error. In each row, 1, 2, and 3 indicate respectively light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$.

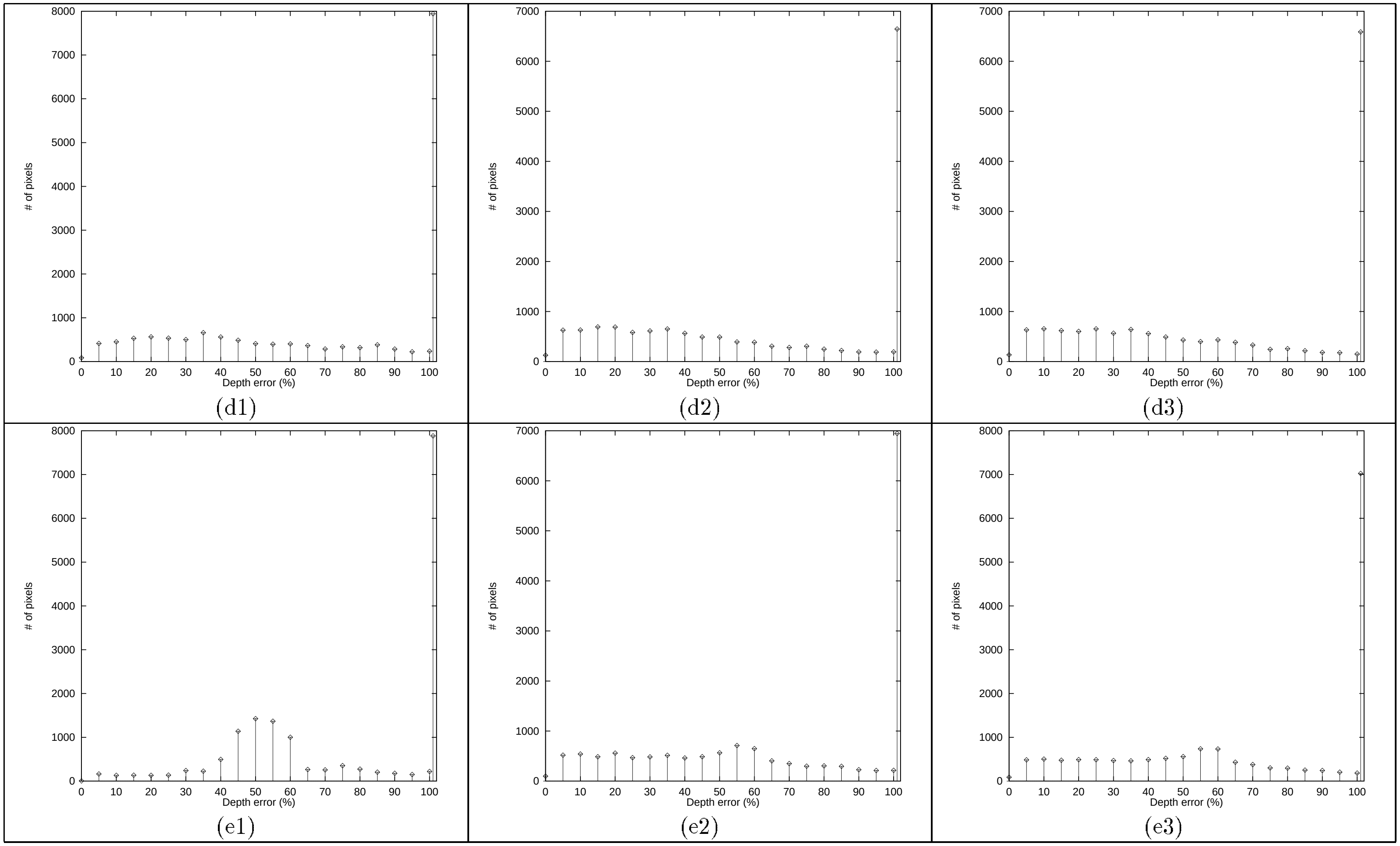


(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

Figure 31: See caption on next page.



(d1) (d2) (d3)

(e1) (e2) (e3)

Figure 31: The histogram of percentage depth error for Lee and Rosenfeld's method. (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. Each bar in the histograms represents the summation of number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error. In each row, 1, 2, and 3 indicate respectively light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$.

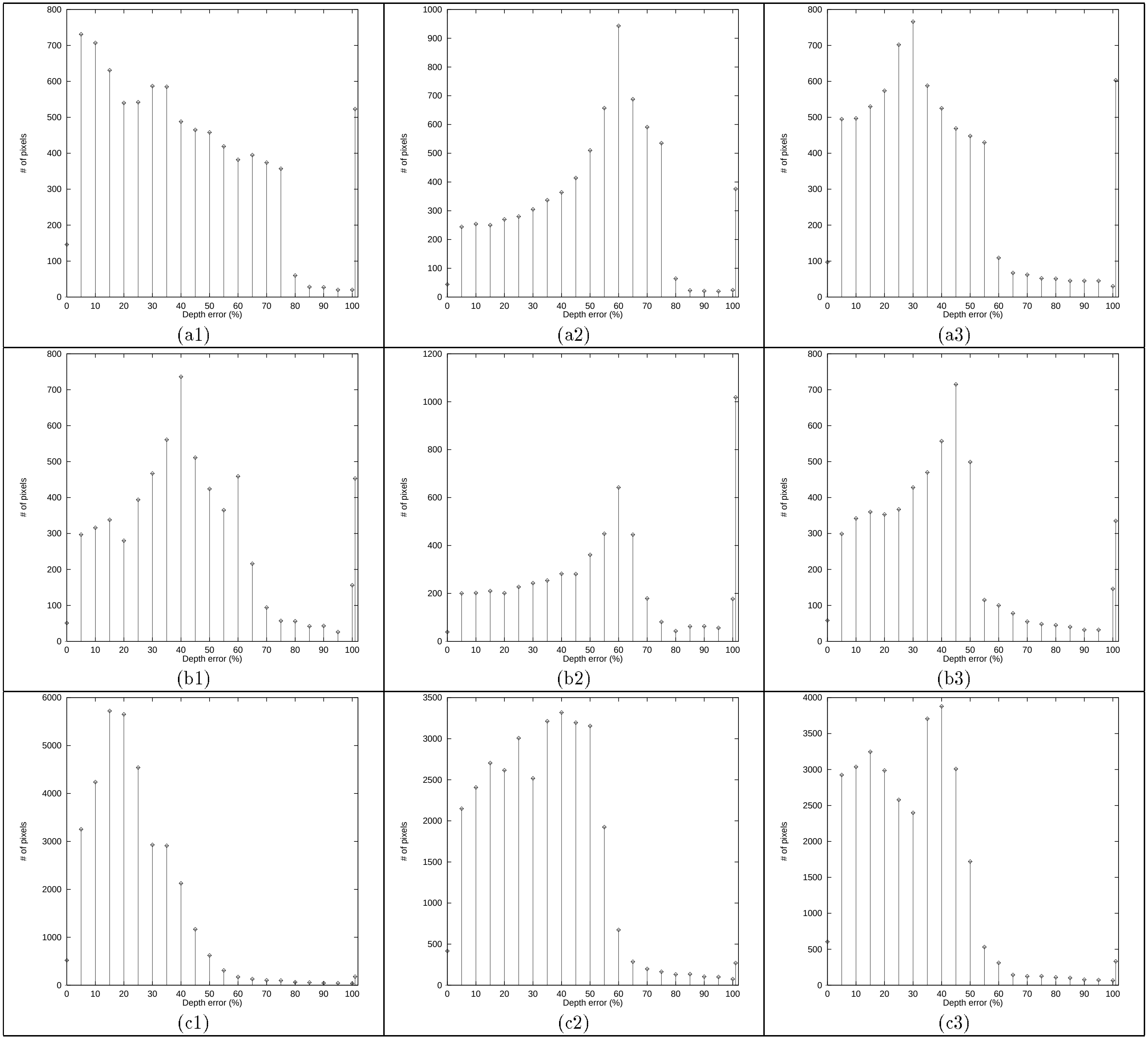
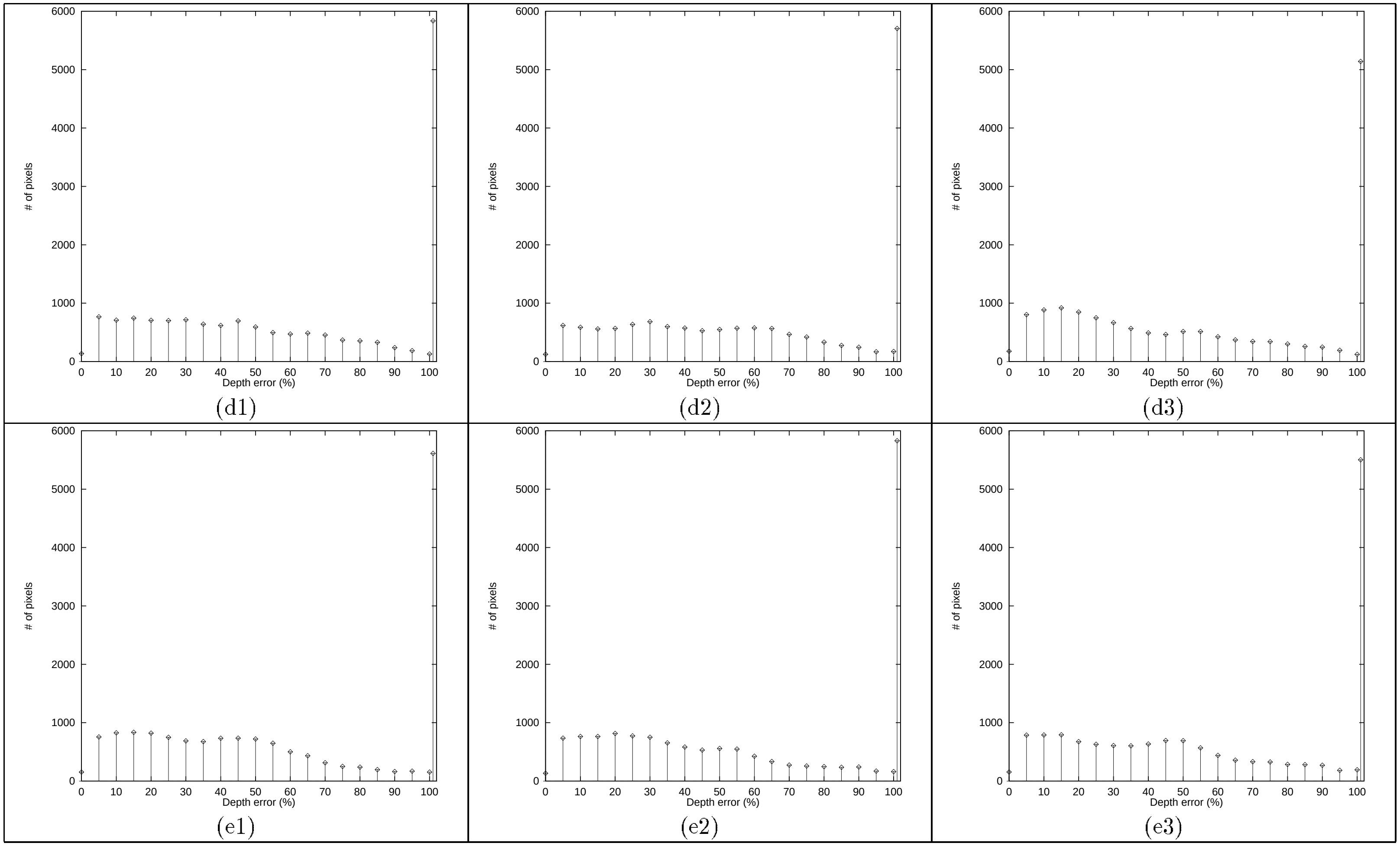


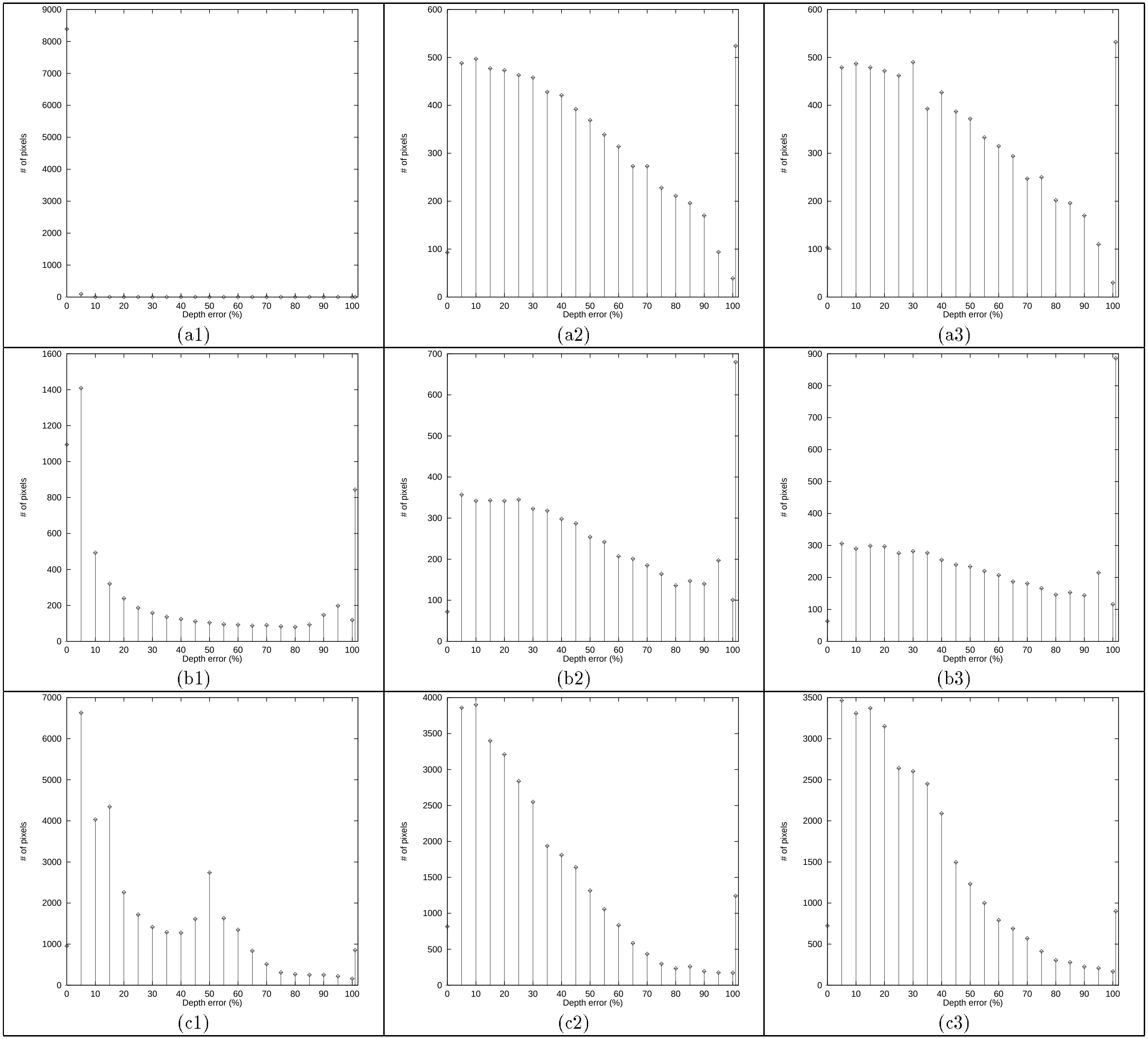
Figure 32: See caption on next page.



(d1) (d2) (d3)

(e1) (e2) (e3)

Figure 32: The histogram of percentage depth error for Pentland's method. (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. Each bar in the histograms represents the summation of number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error. In each row, 1, 2, and 3 indicate respectively light sources, $(0,0,1)$, $(5,5,7)$, and $(1,0,1)$.

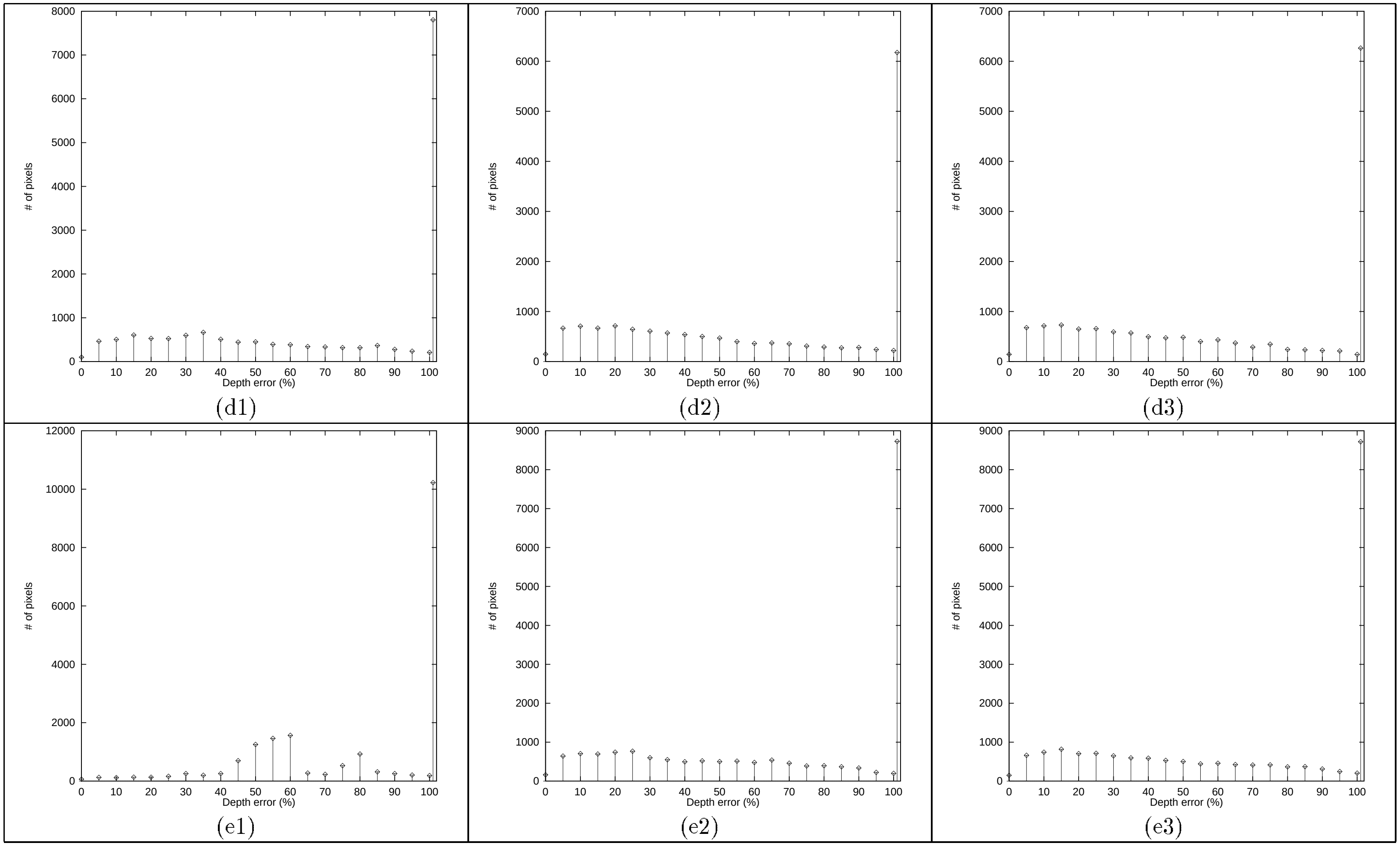


(a1) (a2) (a3)

(b1) (b2) (b3)

(c1) (c2) (c3)

Figure 33: See caption on next page.



(d1) (d2) (d3)

(e1) (e2) (e3)

Figure 33: The histogram of percentage depth error for Tsai and Shah's method. (a) Sphere. (b) Vase. (c) Mozart. (d) Penny. (e) Sombrero. Each bar in the histograms represents the summation of number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error. In each row, 1, 2, and 3 indicate respectively light sources, $(0, 0, 1)$, $(5, 5, 7)$, and $(1, 0, 1)$.

9 Convergence, Uniqueness and Existence

Unlike local approaches, which provide a closed form solution, most global approaches use iterative schemes. A few schemes, such as Ikeuchi and Horn's, Brooks and Horn's, and Zheng and Chellappa's approaches, use variational calculus to derive a non-linear iterative scheme. There is no guarantee of correct convergence for these algorithms because of the convex/concave ambiguity in SFS, the discrete formulation in these approaches, the linear approximation of the reflectance map, and the noise in real images. In some cases, an algorithm can give the correct solution by converging to the global minimum; in others, it can get stuck at a local minimum. Sometimes, an algorithm can diverge and walk away from the true solution, therefore, the initial value will greatly affect the speed of convergence and the solution to which an algorithm converges. In order to enforce proper convergence, the shape information at occluding boundaries are used as initial values in Brooks and Horn's algorithm as well as in Szeliski's algorithm.

Recently, Oliensis [23] proved that even the complete shape information at an occluding boundary does not well-determine the surface reconstruction. However, the surface orientation is uniquely determined at singular points. On the basis of Oliensis's idea, Bichsel and Pentland's algorithm starts at singular points and propagates the depth through a minimum downhill technique. The proper termination of the algorithm is guaranteed by propagating the depth away from the light source. The correctness of the solution depends on the initialization at the singular points.

Ikeuchi and Horn's, Leclerc and Bobick's, Lee and Kuo's, and Tsai and Shah's algorithms, essentially reduce the nonlinear minimization problem into a solution of a linear system. The iterative techniques for solving linear systems, such as Jacobi, Gauss-Seidel, successive over-relaxation or multigrid technique, are applied. ¿From the results of matrix algebra, the iterative scheme

$$x^{(k)} = Tx^{(k-1)} + c \qquad ,(c \neq 0) \tag{47}$$

converges to the unique solution $x = Tx + c$, if and only if the spectral radius of T (the maximum absolute value of the eigenvalues of T), $\rho(T)$, is less than 1. Since real images do not always satisfy $c \neq 0$, and $\rho(T) < 1$, the linear iterative schemes may not converge.

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 [23]. With an initial known curve, the method of characteristic strips yields a unique solution, if the first derivative of surface depth is continuous. For other cases, the uniqueness is unknown. However, Lee and Kuo [19] 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) provide a unique solution for their approach, in most cases.

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.

In the cases where there is a unique solution, existence is clear. However, there are cases where no solution exists. Horn, Szeliski and Yuille [13] presented some impossibly shaded images that could not have originated from a smooth continuous surface with uniform albedo and illumination. These cases can be detected through the examination of conditions for impossibly shaded images, or by checking for the existence of singularities in the solution to shape from shading.

10 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 analyzed a total of twelve existing algorithms and grouped them into three different categories: Global minimization techniques, global propagation techniques, and local techniques, These groupings are based on the conceptual differences among the algorithms. Eight representatives out of the twelve were implemented in order to compare their performance in terms of accuracy and time. This comparison was carried out on five synthetic surfaces, each was used to generate three synthetic images using different light source directions, and five real image, where light source directions were estimated.

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, the standard deviation of depth error, the difference images of absolute depth error, and the histogram of percentage depth error with respect to the true depth. Due to the lack of true shapes, the output for real images were only analyzed and compared visually. The conclusion drawn from the accuracy analysis is that all of the algorithms have their limitations. None of them has consistent performance for all images, since they work well for certain images, but perform poorly for others. Overall, the global minimization techniques are more robust to different scenes and noise. Among them, Leclerc and Bobick's algorithm yields very good results due to the use of good initial estimates from stereo. Lee and Kuo's algorithm produces the second best results. The global propagation techniques provide almost perfect results if the estimates for singular points are accurate. The local approaches tends to have more error for real, noisy images, especially for Lee and Rosenfeld's approach which is based on intensity derivatives and the spherical assumption.

To analyze the time, CPU time was measured for all of the algorithms on the same machine. Among all of the algorithms, Tsai and Shah's is the fastest, while Leclerc and Bobick's is the slowest. The conclusions from the timing is that the local approaches are faster than the global approaches, and the global propagation approaches are a lot faster than the global minimization approaches. The execution times for local appraoches depend only on the size of the input image. While for the global approaches, time not only depends on the size of the input image, but also varies from scene to scene.

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 (see section 2.4). 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 [17] and [31]), 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 [5]).

Multiple images can also be employed by moving either the viewer (as in [10]) or the light source (as in [38]) 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) 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.

Acknowledgements

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

References

[1] U. M. Ascher and P. M. Carter. A multigrid method for shape from shading. Technical Report 91-2, University of British Columbia, 1991.

[2] M. Bichsel and A. P. Pentland. A simple algorithm for shape from shading. In *Proceedings of Computer Vision and Pattern Recognition*, pages 459–465, 1992.

[3] 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.

[4] J. J. Clark. Active photometric stereo. In *Proceedings of Computer Vision and Pattern Recognition* , pages 29–34, 1992.

[5] J. Cryer, P. Tsai, and M. Shah. Integration of shape from x modules: Combining stereo and shading. In *Proceedings of Computer Vision and Pattern Recognition*, pages 720–721, June 1993.

[6] P. Dupuis and J. Oliensis. Direct method for reconstructing shape from shading. In *Proceedings of Computer Vision and Pattern Recognition*, pages 453–458, 1992.

[7] D. Forsyth and A. Zisserman. Mutual illumination. In *Proceedings of Computer Vision and Pattern Recognition*, pages 466–473, 1989.

[8] 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.

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

[10] J. Heel. Dynamic motion vision. *Robotics and Autonomous Systems*, 6(1), 1990.

[11] 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.

[12] B. K. P. Horn. Height and gradient from shading. *International Journal of Computer Vision*, pages 37–75, 1989.

[13] 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.

[14] 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.

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

[16] 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.

[17] Y. G. Leclerc and A. F. Bobick. The direct computation of height from shading. In *Proceedings of Computer Vision and Pattern Recognition*, pages 552–558, 1991.

[18] 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.

[19] K. M. Lee and C. C. J. Kuo. Shape from shading with a linear triangular element surface model. *Signal and Image Processing Institute, University of Southern California, Technical Report 172*, 1991.

[20] 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.

[21] S. K. Nayar, K. Ikeuchi and T. Kanade. Shape from interreflections. In *Proceedings of International Conference on Computer Vision*, pages 1–11, 1990.

[22] 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.

[23] J. Oliensis. Shape from shading as a partially well-constrained problem. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54:163–183, 1991.

[24] J. Oliensis and P. Dupuis. A global algorithm for Shape from shading. In *Proceedings of International Conference on Computer Vision*, pages 692-701, 1993.

[25] M. Oren and S. K. Nayar. Diffuse reflectance from rough surfaces. In *Proceedings of Computer Vision and Pattern Recognition*, pages 763–764, 1993.

[26] A. P. Pentland. Finding the illuminant direction. *J. Optical Society of America*, pages 448–455, 1982.

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

[28] A. Pentland. Shape information from shading: a theory about human perception. In *Proceedings of International Conference on Computer Vision*, pages 404–413, 1988.

[29] A. Pentland. Photometric motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:879–890, 1991.

[30] B. T. Phong. Illumination for computer generated pictures. *Comm. of ACM*, 18:311–317, 1975.

[31] H. Pien and J. Gauch. A variational approach to sensor fusion using registered range and intensity data. In *SPIE OE/Aerospace Sensing*, April 1993.

[32] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1990.

[33] R. Szeliski. Fast shape from shading. *Computer Vision, Graphics, Image Processing: Image Understanding*, 53:129–153, 1991.

[34] 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.

[35] P. S. Tsai and M. Shah. A simple shape from shading algorithm. In *Proceedings of Computer Vision and Pattern Recognition*, pages 734–736, 1992.

[36] 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.

[37] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.

[38] R. Zhang, P. T. Tsai, and M. Shah. Shape from photomotion. In *Proceedings of Computer Vision and Pattern Recognition*, pages 740–741, 1993.

[39] 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.