USC–SIPI REPORT #228

Direct Shape Recovery from Texture Information

by

Kyoung Mu Lee and C.-C. Jay Kuo

January 1993

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

Direct Shape Recovery from Texture Information *

Kyoung Mu Lee† and C.-C. Jay Kuo †

January 4, 1993

Abstract

Most existing shape from texture algorithms extract the surface orientations of planer objects from projected texture variation in an image. In this research, we propose a new shape from texture (SFT) algorithm which recovers the surface heights of curved objects directly. By examining the perspective transformation of a local planar surface patch onto the image plane, we establish a nonlinear relationship between the measured local density of texture primitives on the image plane and the orientation and position of the surface patch in the world space. Then, by approximating a curved surface with a union of triangular surface patches and dividing an image into a set of nonoverlapping triangular domains, we can relate the densities of textural primitives in triangular domains to nodal height variables of the curved surface. This procedure leads to a nonlinear system of equations which are then solved via a successive linearization scheme. Estimation of the local textural density using dots and line segment edges of texels is also discussed. Experimental results are given to illustrate the performance of the new algorithm.

1 Introduction

Texture variation in an image provides important information about the underlying surface structure. Texture information depicted on a 3-D surface is distorted by the projection onto the 2-D image plane. The distortion depends on the relative position between the camera and a surface point and the orientation of the surface at that point. By modeling the imaging geometry appropriately, one can establish a relationship between texture distortion and the relative distance and the shape of an textured object. To recover the shape or orientation of a surface from projected texture information in an image, known as the SFT (Shape From Texture) problem, was initiated by Gibson [7] and has been studied by quite a few researchers [2], [3], [4], [5], [6], [11], [13], [14], [17], [19], [21], [23].

*This work was supported by the National Science Foundation Young Investigator Award ASC-9258396.

†The authors are with the Signal and Image Processing Institute and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, California 90089-2564. E-mail: kyoungmu@sipi.usc.edu and cckuo@sipi.usc.edu.

Conventional shape from texture algorithms differ in their assumptions on the texel identification and distribution and the projection model used. Most algorithms are based on two assumptions: the *homogeneity assumption* that the textural density is uniform on the surface so that the surface orientation can be specified by the texture gradient in an image [7], [11], [14], [21], and the *isotropy assumption* that texture elements have uniform edge distribution in all direction so that the surface orientation can be estimated from the deviation from the isotropy in an image [23], [6]. Many algorithms consider explicit representation of textures by means of identifiable texels or edge elements and determine the texture gradient using these measured textural primitives. More recently, local spatial frequency analysis of textures has been proposed and received more attention including the Fourier transform [3], autocorrelation [5], the Wigner distribution [11], the spectrogram [14] and the Gabor wavelet [21]. With the local spatial frequency analysis, textures can be readily represented without identifying texels.

The projection model plays an important role in formulating the SFT problem. It defines the relationship between surface textural primitives and imaged textural primitives. Many SFT algorithms assume the orthographic projection for its simplicity and consider only foreshortening distortion due to the surface orientation exclusively [6], [11], [21]. Since it ignores the apparent distance and position effects, it is not valid in many real situations. In contrast, the perspective projection model takes all distortion effects into account and, therefore, provides an ideal model. However, its formulation becomes very complicated. Recently, a semi-perspective model, which is an approximation of perspective projection, has been proposed and used by several researchers [2], [18]. With this model, one can approximate the texture distortion by a 2-D affine transform.

Most conventional shape from texture or pattern algorithms have been developed to recover planar surface orientations. Several algorithms are known to be applicable to curved surfaces [2], [10], [21]. However, not much work has been done to determine the absolute depth from the camera to a surface point. Even some algorithms using the semi-perspective projection (or the perspective) model which explicitly contains the distance effect assume that the distance variation is negligible and focus on the effect due to relative orientations [2], [14]. We observe that texture distortion depicted in an image is affected by the surface orientation as well as the apparent distance. They cannot be easily separated, and neither of them can be simply ignored. This can be verified by computing the variation of distortion rate of projected textural primitives of an object according to the distance from the camera to the surface. Moreover, even given the computed surface orientations, it may not be straightforward to construct the corresponding surface due to the *integrability*

problem. The problem becomes particularly serious with respect to the perspective projection model. This is because that even for regular grid points in an image, the corresponding surface points may become irregular in the world space due to the perspective law. Thus, without explicit depth information, it is extremely difficult to determine the position in the world space.

In this paper, we propose a new SFT algorithm which recovers depth information of a curved object directly. Our algorithm assumes homogeneity of the surface texture and uses the semi-perspective projection model which captures and represents all possible effects of texture distortion by a convenient solid angle approach under the assumption of local planarity of a surface. By examining the perspective transformation of a local planar surface patch onto the image plane, we establish a nonlinear relationship between the measured local density of texture primitives on the image plane and the orientation and position of the surface patch in the world space. Then, by approximating a curved surface with a union of triangular surface patches and dividing the measured textural densities in an image into a set of nonoverlapping triangular domains, we can relate the densities of textural primitives to nodal height variables of the curved surface. This procedure leads to a nonlinear system of equations which are then solved via a successive linearization scheme.

This paper is organized as follows. We discuss the perspective projection model and the triangular-element surface model in Section 2. In Section 3, we formulate the SFT problem by deriving a nonlinear relationship between the local density of projected textural primitives and the orientation and position of a surface patch with the solid angle approach. The estimation of the density of textural primitives is also discussed. The SFT algorithm is presented in Section 4. By using the triangular element surface model, we express the nonlinear relationship in terms of surface nodal height variables, which are then determined by minimizing a quadratic cost functional consisting of squares of error. We show in Section 5 that the SFT prolbem with orthographic projection is a special case of the SFT problem with perspective projection. Thus, our proposed approach applies to both cases. Experimental results on several synthetic and real images are given in Section 6.

2 Projection and Surface Models

In this section, we discuss the perspective image projection model and the triangular element surface model.

2.1 Perspective Projection Model

A projection model defines a relationship between points in the 3-D space and their corresponding pixels in a 2-D picture. An ideal camera can be modeled as a pin-hole camera consisting of a lens and a plane on which the image is formed. Consider a viewer-centered cartesian coordinate system with the origin at the center of projection and the Z-axis aligned with the optical axis as depicted in Fig. 1. The image of a point formed on the image plane can be determined by its intersection with the ray connecting the lens center and the given point. Let the x- and y-axes of the image plane be parallel to the X- and Y-axes and the *focal length*, i.e. the distance between the origin and the image plane, be f. Then, the image point $\mathbf{p} = (x, y, -f)^T$ of an object point $\mathbf{P} = (X, Y, Z)^T$ is given by

$$x = -f\frac{X}{Z}, \qquad y = -f\frac{Y}{Z}, \tag{2.1}$$

which relates the image coordinates (x, y) to the world coordinates (X, Y, Z) of a point. This is known as the perspective transformation.

2.2 Triangular Element Surface Model

A triangular element surface model has been recently proposed and applied successfully to the SFS (Shape From Shading) problem [15], [16]. With this model, we approximate a smooth non-occluded object surface with a union of small triangular patches. Since each triangular patch can be uniquely determined by its three vertices vectors, both the surface height and its orientation can be uniquely represented by the nodal vectors. Therefore, a direct relationship between the measurement of depth cues in a projected triangle and the nodal heights of a triangular surface patch can be derived through an appropriate mapping model.

Consider a small triangular surface patch S_k and its projected triangle T_k on an image plane as shown in Fig. 2. Let V_k be the index set of vertices points of T_k or S_k. Then, from Fig. 2, we have

$$V_k = \{l, m, n\}.$$

The three vertices of S_k are denoted by the vectors

$$\mathbf{P}_l = (X_l, Y_l, Z_l), \quad \mathbf{P}_m = (X_m, Y_m, Z_m), \quad \mathbf{P}_n = (X_n, Y_n, Z_n),$$

and their corresponding projected points on the image plane are

$$\mathbf{p}_l = (x_l, y_l, -f), \quad \mathbf{p}_m = (x_m, y_m, -f), \quad \mathbf{p}_n = (x_n, y_n, -f).$$

The surface normal $\mathbf{n}_k$ of the triangular surface patch S_k can be determined with its three nodal vectors $\mathbf{P}_l, \mathbf{P}_m$ and $\mathbf{P}_n$ by

$$\begin{aligned} \mathbf{n}_k &= \frac{(\mathbf{P}_n - \mathbf{P}_l) \times (\mathbf{P}_m - \mathbf{P}_l)}{|\,(\mathbf{P}_n - \mathbf{P}_l) \times (\mathbf{P}_m - \mathbf{P}_l)\,|} \\ &= \frac{(X_n - X_l, Y_n - Y_l, Z_n - Z_l) \times (X_m - X_l, Y_m - Y_l, Z_m - Z_l)}{|\,(X_n - X_l, Y_n - Y_l, Z_n - Z_l) \times (X_m - X_l, Y_m - Y_l, Z_m - Z_l)\,|}, \end{aligned} \tag{2.2}$$

where $\times$ denotes the outer-product of two vectors. Since the vertices of the projected triangle T_k on the image plane can be related to the vertices of the surface patch S_k via the perspective law (2.1), we can rewrite (2.2) as

$$\mathbf{n}_k = \frac{(\frac{1}{f}(x_l Z_l - x_n Z_n), \frac{1}{f}(y_l Z_l - y_n Z_n), Z_n - Z_l) \times (\frac{1}{f}(x_l Z_l - x_m Z_m), \frac{1}{f}(y_l Z_l - y_m Z_m), Z_m - Z_l)}{|\,(\frac{1}{f}(x_l Z_l - x_n Z_n), \frac{1}{f}(y_l Z_l - y_n Z_n), Z_n - Z_l) \times (\frac{1}{f}(x_l Z_l - x_m Z_m), \frac{1}{f}(y_l Z_l - y_m Z_m), Z_m - Z_l)\,|}.$$

From the following relationship between the surface normal $\mathbf{n}_k$ and the gradient (p_k, q_k),

$$\mathbf{n}_k = \frac{(-p_k, -q_k, 1)^T}{\sqrt{1 + p_k^2 + q_k^2}},$$

one can easily verify that

$$\begin{bmatrix} p_k \\ q_k \end{bmatrix} = \frac{1}{\gamma_k} \begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix}, \tag{2.3a}$$

where

$$\begin{bmatrix} \alpha_k \\ \beta_k \\ \gamma_k \end{bmatrix} = \begin{bmatrix} f(y_l - y_n) & f(y_m - y_l) & f(y_n - y_m) \\ f(x_l - x_n) & f(x_m - x_l) & f(x_n - x_m) \\ (x_l y_n - x_n y_l) & (x_m y_l - x_l y_m) & (x_n y_m - x_m y_n) \end{bmatrix} \begin{bmatrix} Z_l Z_n \\ Z_l Z_m \\ Z_n Z_m \end{bmatrix}. \tag{2.3b}$$

3 Problem Formulation

3.1 Semi-perspective Transformation of Textured Surface

The relationship between variation of the textural primitives measured on the image plane and the object shape is important in formulating the SFT problem. It will be derived in this section. Our derivation is based on two assumptions, namely, the local planar surface assumption and the texture homogeneity assumption. The first assumption implies that the curved surface can be well approximated by a planar surface locally whereas the second assumption implies that the textural density is approximately constant everywhere on a surface.

To establish a proper approximation of the textured surface transformation under the perspective projection, we use the solid angle concept used in radiometry [8]. The solid angle subtended by a surface patch is defined by the cone whose vertex is at the point of radiation and whose axis is the line segment going from the point of the radiation to the center of the surface. The size of the solid angle is equal to the area intercepted by the cone on a unit sphere centered at the point of radiation. The solid angle Θ subtended by a local planar surface area A, whose center is at distance d from a point of radiation and whose normal makes an angle of θ with respect to the cone axis, can be written as

$$\Theta = \frac{A\cos\theta}{d^2} \tag{3.1}$$

with the assumption of $d^2 \gg A$.

Consider a local planar approximation of a surface with nomal $\mathbf{n}_S$ at a point $\mathbf{P}$ and its projected image at a point $\mathbf{p}$ in an image plane as depicted in Fig. 3. Let us use A_S and A_I to denote the areas formed by the solid angles Θ_S and Θ_I on the object surface and the image plane, respectively. Then, from (3.1) the area A_S can be written as

$$A_S = \frac{\Theta d_S^2}{\cos\theta_S} = \frac{\Theta d_S^2}{(\mathbf{n}_S^T\mathbf{l})}, \tag{3.2}$$

where d_S is the distance the between center of projection $\mathbf{O}$ and the point $\mathbf{P}$ on the surface, $\mathbf{n}_S$ is the unit surface normal vector at point $\mathbf{P}$, and $\mathbf{l}$ is the unit vector of the direction of a ray from $\mathbf{P}$ to the origin $\mathbf{O}$ passing through the point $\mathbf{p}$, i.e.

$$\mathbf{n}_S = \frac{(-p,-q,1)^T}{(1+p^2+q^2)^{1/2}}, \tag{3.3}$$

$$\mathbf{l} = \frac{(-x,-y,f)^T}{(x^2+y^2+f^2)^{1/2}}. \tag{3.4}$$

Similarly, we have

$$A_I = \frac{\Theta d_I^2}{\cos\theta_I} = \frac{\Theta d_I^2}{(\mathbf{n}_I^T\mathbf{l})}, \tag{3.5}$$

where d_I is the distance the between center of projection $\mathbf{O}$ and the point $\mathbf{p}$ on the image plane, and

$$\mathbf{n}_I = (0,0,1)^T \tag{3.6}$$

is the unit nomal vector of an image plane. Based on (3.2) and (3.5), we can relate A_I to A_S as

$$A_I = \frac{d_I^2}{d_S^2}\frac{(\mathbf{n}_S^T\mathbf{l})}{(\mathbf{n}_I^T\mathbf{l})}A_S. \tag{3.7}$$

Substituting (3.3), (3.4) and (3.6) into (3.7), we have

$$A_I = \frac{d_I^2}{d_S^2 f} \frac{(f + xp + yq)}{(1 + p^2 + q^2)^{1/2}} A_S. \tag{3.8}$$

Using the fact that

$$\frac{d_I}{d_S} = \frac{f}{d},$$

we can rewrite (3.8) as

$$A_I = \frac{f}{d^2} \frac{(f + xp + yq)}{(1 + p^2 + q^2)^{1/2}} A_S. \tag{3.9}$$

Equation (3.9) is known as the semi-perspective projection model which includes the distance (d), foreshortening (p and q) and position (x and y) effects. Although this model was also derived by Ohta, Maenobu and Sakai [18] and Aloimonos and Swain [2] with a 2-D affine approximation of the perspective projection model, the above derivation using the solid angle is easier and provides insights into the approximation. Our derivation is similar to that given by Haralick and Shapiro in [8].

We assume that the number of textural primitives in surface area A_S is n and the number of textural primitives in the projected area A_I on the image plane is also n. Let D_I and D_S denote the textural densities over areas A_I and A_S, respectively. Then, we have

$$D_I = \frac{n}{A_I} \quad \text{and} \quad D_S = \frac{n}{A_S}.$$

Since D_S is constant over the surface by the homogeneity assumption, we can rewrite (3.9) as

$$E(x, y) = \frac{D_S}{D_I(x, y)} = R(d, x, y, p, q), \quad 0 < E(x, y) \leq 1, \tag{3.10}$$

where

$$R(d, x, y, p, q) = \frac{f}{d^2} \frac{(f + xp + yq)}{(1 + p^2 + q^2)^{1/2}}. \tag{3.11}$$

Equation (3.11) characterizes the relationship between textural intensity $E(x, y)$, which is inversely proportional to the density $D_I(x, y)$ of projected textural primitives, the distance to the surface d, the surface orientation (p, q) and the position vector $(x, y, -f)$. Since (3.10) is similar to the image irradiance equation for the shape from shading problem [9], we call (3.10) the *textural irradiance equation*, $E(x, y)$ the *textural intensity* (it is actually the textural intensity normalized by the quantity $1/D_S$ known as the *textural albedo*), and R the *textural reflectance map.*

3.2 Estimation of Textural Intensity

To solve the SFT problem, we usually assume that the textural density constant D_S is known a priori. One way to estimate D_S is to examine an image point (x, y) with $(p, q) = (0, 0)$ (i.e. the image point in the region where the textural gradient is zero) and estimate the corresponding $d(x, y)$. Then, by using (3.10) and (3.11), one obtain

$$D_S = \left(\frac{f}{d(x,y)}\right)^2 D_I(x,y), \tag{3.12}$$

where $D_I(x, y)$ is the observed textural density at (x, y) in an image.

It is clear from (3.10) that the quality of the reconstructed surface depends on the accuracy of the estimation of the textural density $D_I(x, y)$. Direct calculation of the textural density requires the texel identification and extraction process as preprocessing. Techniques have been proposed for texel identification such as the multiscale region detector [4]. However, since not all textures are so well structured to be identifiable, to extract reliable texels is in general not an easy task. Thus, indirect methods for textural density estimation have been developed by many researchers. Based on the assumption that the sum of the length of the edges is uniformly distributed on a surface, indirect methods for estimating textural density by computing and measuring density of edges i.e, the texel boundaries have been studied by researchers [1], [12], [20], [22]. It has been verified experimentally that the homogeneity assumption of edgels is valid for many natural and man-made textured surface [1]. Another way to estimate the density of local textural primitives without the requirement of identification and extraction of explicit texels or edgels is to use the local spatial frequency analysis. It expresses the spatial-frequency information of a texture as a function of location. The underlying idea of this approach is that since the texture is fundamentally a frequency phenomenon, the shift of local spatial frequencies on the perspectively projected image gives information about the shape of a local surface. A variety of work presented in literatures can be categorized to this class, including the use of the Fourier transform [3], [14], Wigner distribution [11], and Gabor wavelet transform [21].

In this work, we adopt an approach introduced by Kanatani and Chou [12] to estimate the textural intensity $E(x, y)$. Two special discrete textural patterns are particularly of our interest, i.e. dot and line segment textures. Since the detected texel boundaries behave like dot or line segment textures, the two patterns can be used to represent a large class of texture patterns. The

local textural intensity E appearing in (3.10) can be computed by

$$E(x,y) = \left(\frac{\eta \int_W t(\tilde{x},\tilde{y}) d\tilde{x} d\tilde{y}}{\int_W \rho(\tilde{x},\tilde{y}) t(\tilde{x},\tilde{y}) d\tilde{x} d\tilde{y}} \right)^{\kappa}, \tag{3.13}$$

where $(\tilde{x},\tilde{y})$ denotes the coordinates of the image plane, W is a window centered at (x,y), $\rho(x,y)$ is the corresponding textural density, $t(\tilde{x},\tilde{y})$ is an appropriate smooth test function and $\kappa = 1$ for dot textures and $\kappa = 2$ for line segment textures. The quantity η is a constant. The $1/\eta$ can be interpreted as "area per dot" or "area per unit length line segment". The choice for the test function can be simply $t(\tilde{x},\tilde{y}) = 1$ or

$$t(\tilde{x},\tilde{y}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\tilde{x}^2 + \tilde{y}^2}{2\sigma^2} \right),$$

where the value σ depends on the window size. For dot and line segment textures (edgels), we have

$$\int_W \rho(x,y) t(x,y) dx dy = \begin{cases} \sum_{P_i \in W} t(x_i, y_i) & \text{for dot textures,} \\ \sum_{L_i \subset W} \int_{L_i} t(x(l), y(l)) dl & \text{for line segment textures,} \end{cases} \tag{3.14}$$

where $P_i(x_i, y_i)$ are the positions of the dots, L_i are line segments and dl denotes the line integral along the arc of line segments.

Equation (3.13) can be derived as follows. Let (u,v) denote the coordinates of the local planar surface in the world space. There is one-to-one correspondence between $(\tilde{x},\tilde{y})$ and (u,v), and W_S is the corresponding window in the u-v plane with respect to the window W in the image plane. Let ρ_s be the texture density in the u-v domain. The texture homogeneity can be expressed as follows.

$$\int_{W_S} \rho_s(u,v) t_s(u,v) du dv = \eta \int_{W_S} t_s(u,v) du dv, \tag{3.15}$$

where η is constant and $t_s(u,v) \equiv t(\tilde{x}(u,v), \tilde{y}(u,v))$. Besides, from (3.9) and the associations $A_S = dudv$ and $A_I = d\tilde{x}d\tilde{y}$, we have

$$dudv = \frac{d\tilde{x}d\tilde{y}}{R(d,x,y,p,q)}. \tag{3.16}$$

By using the above relationship, the right-hand-side of (3.15) can be rewritten as

$$\eta \int_{W_S} t_s(u,v) du dv = \frac{\eta}{R} \int_W t(\tilde{x},\tilde{y}) d\tilde{x} d\tilde{y}. \tag{3.17}$$

Now, let us examine the left-hand-side of (3.15) for two cases. For dot textures, since the number of dots in W_S is equal to that in W, we have

$$\int_{W_S} \rho_s(u,v) t_s(u,v) du dv \equiv \sum_{P_i \in W_S} t_s(u_i, v_i) = \sum_{\tilde{P}_i \in W} t(\tilde{x}_i, \tilde{y}_i) \equiv \int_W \rho(\tilde{x},\tilde{y}) t(\tilde{x},\tilde{y}) d\tilde{x} d\tilde{y}, \tag{3.18}$$

where the last equality is based on the definition (3.14). For line segment or edge textures with the isotropic orientation assumption, since the area is enlarged by the factor $1/R$ as shown in (3.16), the individual line segment becomes longer by a factor $1/\sqrt{R}$. Thus, we have

$$\begin{aligned}\int_{W_S} \rho_s(u,v)t_s(u,v)dudv &\equiv \sum_{L_i \subset W_S} \int_{L_i} t_s(u(l),v(l))dl = \frac{1}{\sqrt{R}} \sum_{\tilde{L}_i \subset W} \int_{\tilde{L}_i} t(\tilde{x}(\tilde{l}),\tilde{y}(\tilde{l}))d\tilde{l} \\ &\equiv \frac{1}{\sqrt{R}} \int_W \rho(\tilde{x},\tilde{y})t(\tilde{x},\tilde{y})d\tilde{x}d\tilde{y}, \end{aligned} \tag{3.19}$$

where the change of variable $dl = d\tilde{l}/\sqrt{R}$ is applied and the definition (3.14) is used. By using (3.17), (3.18) and (3.19), we can rewrite (3.15) as

$$\int_W \rho(\tilde{x},\tilde{y})t(\tilde{x},\tilde{y})d\tilde{x}d\tilde{y}, = \eta R^{-1/\kappa} \int_W t(\tilde{x},\tilde{y})d\tilde{x}d\tilde{y}.$$

Thus,

$$R(d,x,y,p,q) = \left(\frac{\eta \int_W t(\tilde{x},\tilde{y})d\tilde{x}d\tilde{y}}{\int_W \rho(\tilde{x},\tilde{y})t(\tilde{x},\tilde{y})d\tilde{x}d\tilde{y}} \right)^{\kappa}.$$

Since $R(d,x,y,p,q) = E(x,y)$, we define the right-hand-side of the above equation to be the textural intensity as given in (3.13).

4 Shape From Texture Algorithm

Our objective is to determine the depth map by solving (3.10) with an estimated textural intensity map E of a textured image. To do so, we use the triangular element surface model and express the textural irradiance equation in terms of surface nodal height variables. In the following discussion, we denote triangular surface patches and their projected triangles, respectively, by S_i and T_i, $i = 1,2,\ldots,M_t$, where M_t is the total number of triangles. Similarly, nodal points on a surface and the corresponding projected ones in the image plane are denoted by $\mathbf{P}_i$ and $\mathbf{p}_i$, $i = 1,2,\ldots,M_n$, respectively, where M_n is the total number of nodal points.

The textural intensity E_k over a triangle domain T_k is determined by

$$E_k = R_k(d_k,x_k,y_k,p_k,q_k) = \frac{f}{{d_k}^2} \frac{f + x_k p_k + y_k q_k}{(1 + {p_k}^2 + {q_k}^2)^{1/2}}, \tag{4.1}$$

where

$$d_k = \frac{1}{3}(Z_l + Z_m + Z_n)$$

is the distance to the center of triangular surface patch S_k and

$$(x_k,y_k) = \left(\frac{1}{3}(x_l + x_m + x_n), \frac{1}{3}(y_l + y_m + y_n) \right)$$

is the center of T_k and (p_k, q_k) is the gradient of S_k. By substituting (2.3a) and (2.3b) into (4.1), we obtain a nonlinear reflectance map function

$$R_k(Z_l, Z_m, Z_n) = \frac{9f}{(Z_l + Z_m + Z_n)^2} \frac{f\gamma_k + x_k\alpha_k + y_k\beta_k}{(\alpha_k^2 + \beta_k^2 + \gamma_k^2)^{1/2}}, \tag{4.2}$$

which only consists of nodal heights Z_l, Z_m and Z_n as variables. To simplify the problem, we consider the linearization of the nonlinear function R_k by taking the Taylor series expansion of $R_k(Z_l, Z_m, Z_n)$ about a reference nodal height (Z_l^0, Z_m^0, Z_n^0) to the first order term,

$$\begin{aligned} R_k(Z_l, Z_m, Z_n) \approx\ & R_k(Z_l^0, Z_m^0, Z_n^0) + (Z_l - Z_l^0)\frac{\partial R_k(Z_l, Z_m, Z_n)}{\partial Z_l}\Big|_{(Z_l^0, Z_m^0, Z_n^0)} \\ +\ & (Z_m - Z_m^0)\frac{\partial R_k(Z_l, Z_m, Z_n)}{\partial Z_m}\Big|_{(Z_l^0, Z_m^0, Z_n^0)} + (Z_n - Z_n^0)\frac{\partial R_k(Z_l, Z_m, Z_n)}{\partial Z_n}\Big|_{(Z_l^0, Z_m^0, Z_n^0)}. \end{aligned} \tag{4.3}$$

By substituting the above approximation into (4.1), we can express E_k over T_k as a linear combination of the surface nodal heights Z_i, $1 \le i \le M_n$, i.e.

$$E_k \approx \sum_{i=1}^{M_n} \omega_{ki} Z_i + \xi_k, \tag{4.4}$$

where

$$\omega_{ki} = \begin{cases} \frac{\partial R_k(Z_l, Z_m, Z_n)}{\partial Z_i}\Big|_{(Z_l^0, Z_m^0, Z_n^0)}, & \text{if } i \in V_k = \{l, m, n\} \text{ of } T_k, \\ 0, & \text{otherwise,} \end{cases} \tag{4.5}$$

and

$$\xi_k = R_k(Z_l^0, Z_m^0, Z_n^0) - \sum_{i=1}^{M_n} \omega_{ki} Z_i^0. \tag{4.6}$$

To determine the surface nodal height Z_i, $1 \le i \le M_n$, from a given textural intensity map $E(x, y)$ over the entire image, we adopt a global error minimization approach by defining a cost functional as

$$\mathcal{E}_d = \sum_{k=1}^{M_t} \mathcal{E}_k = \sum_{k=1}^{M_t} (E_k - \hat{E}_k)^2, \tag{4.7}$$

where $\mathcal{E}_k$ denotes the cost term corresponding to the kth triangular domain, and E_k and $\hat{E}_k$ are the observed and estimated textural densities over the kth triangular domain, respectively. By substituting (4.4) into (4.7), we have

$$\mathcal{E}_d = \sum_{k=1}^{M_t} [E_k - (\sum_{i=1}^{M_n} \omega_{ki} Z_i + \xi_k)]^2 = \frac{1}{2}\mathbf{z}^T \mathbf{A}\mathbf{z} - \mathbf{b}^T\mathbf{z} + c, \tag{4.8}$$

where $\mathbf{z} = [Z_1, Z_2, \ldots, Z_{M_n}]^T$ and the elements $a_{i,j}$ and b_i of the stiffness matrix $\mathbf{A}$ and the load vector $\mathbf{b}$ can be determined, respectively, as

$$a_{i,j} = 2\sum_{k=1}^{M_t} \omega_{ki}\omega_{kj}, \qquad b_i = 2\sum_{k=1}^{M_t}(E_k - \xi_k)\omega_{ki} \quad 1 \le i,j \le M_n. \tag{4.9}$$

The problem of minimizing (4.8) is usually ill-posed and errors may occur due to noise. Thus, it is often to employ a regularization technique by adding a smoothness term

$$\mathcal{E}_s = \frac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z}, \tag{4.10}$$

where $\mathbf{B}$ is the *smoothness* matrix. The local nodal operator of $\mathbf{B}$ is of the following stencil form

$$\mathbf{B}: \begin{vmatrix} & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \end{vmatrix}. \tag{4.11}$$

It is the discrete version of *thin plate* model which represents small deflection bending energy of a thin plate characterizing surfaces of C^1 continuity. Thus, we have a new cost functional $\mathcal{E}$ by combining (4.8) and (4.10) with a smoothing factor λ as

$$\mathcal{E} = \mathcal{E}_d + \lambda\mathcal{E}_s = \frac{1}{2}\mathbf{z}^T\mathbf{C}\mathbf{z} - \mathbf{b}^T\mathbf{z} + c, \tag{4.12}$$

where

$$\mathbf{C} = \mathbf{A} + \lambda\mathbf{B}.$$

Then, the minimization problem is equivalent to the soluton of the linear system of equations

$$\mathbf{C}\mathbf{z} = \mathbf{b}. \tag{4.13}$$

To reduce the linear approximation error of the nonlinear function R, we use a successive linearization scheme where the surface nodal height $\mathbf{z}^{n-1}$ determined at the (n-1)th iteration was used as the reference nodal height for the linearization of the R at the nth iteration. The successive linearization scheme was developed and successfully applied in our previous work for the shape from shading problem [15], [16]. The proposed successive shape from texture algorithm is summarized below.

Algorithm: The successive shape from texture algorithm.

Initialization ($k = 0$)

Set the reference nodal values $\mathbf{z}_0$ to be $\mathbf{z}_{ref}$, and construct the coefficient matrix $\mathbf{C}^0 = \mathbf{A}^0 + \lambda\mathbf{B}$ and the load vector $\mathbf{b}^0$. Solve $\mathbf{C}^0\mathbf{z}^0 = \mathbf{b}^0$ for nodal values $\mathbf{z}^0$.

Iterations ($k = 1, 2, \cdots$)

Set the reference nodal values $\mathbf{z}_0$ to be the nodal values $\mathbf{z}^{k-1}$, and construct the corresponding coefficient matrix $\mathbf{C}^k = \mathbf{A}^k + \lambda\mathbf{B}$ and load vector $\mathbf{b}^k$. Solve $\mathbf{C}^k\mathbf{z}^k = \mathbf{b}^k$ for nodal values $\mathbf{z}^k$. If $\|\mathbf{z}^k - \mathbf{z}^{k-1}\| < \epsilon$, where ϵ is a predefined small quantity, then $\mathbf{z}^k$ is the desired solution. Otherwise, go to the next iteration.

5 Orthographic Projection Case

Since the orthographic SFT problem is a special case of the perspective SFT problem, the framework developed above also applies to images with orthographic projection. However, further simplification can be made on the problem formulation for the orthographic projection case.

Consider an orthographic projection of a local planar surface onto the image plane as shown in Fig. 4. One can easily derive the relationship between the surface area A_S and the projected area A_I as

$$A_I = (\mathbf{n}_S^T\mathbf{n}_I)A_S. \tag{5.1}$$

Proceeding in a similar manner given in Section 3, one can derive the orthographic texture irradiance equation,

$$E(x, y) = \frac{D_S}{D_I(x, y)} = R(p, q), \tag{5.2}$$

where the textural reflectance map is

$$R(p, q) = \frac{1}{(1 + p^2 + q^2)^{1/2}}. \tag{5.3}$$

One can easily see from (5.3) that the distance and the position effects play no role in the SFT formulation under orthographic projection. The textural intensity E_k over the triangle T_k can therefore be expressed as

$$E_k = R_k(p_k, q_k) = \frac{1}{(1 + p_k^2 + q_k^2)^{1/2}}, \tag{5.4}$$

where (p_k, q_k) is the gradient of the surface patch S_k.

The orthographic projection model is a good approximation of the perspective projection model when an object is far away from the viewer so that the depth variation ΔZ of the object is small

compared with the average depth Z from the viewer to the object. Thus, we have the following approximation for (2.1):

$$x \approx mX, \qquad y \approx mY, \tag{5.5}$$

where m is positive constant. By using (5.5) with $m = 1$, one can approximate the surface normal $\mathbf{n}_k$ in (2.2) by

$$\mathbf{n}_k = \frac{(x_n - x_l, y_n - y_l, Z_n - Z_l) \times (x_m - x_l, y_m - y_l, Z_m - Z_l)}{|\,(x_n - x_l, y_n - y_l, Z_n - Z_l) \times (x_m - x_l, y_m - y_l, Z_m - Z_l)\,|}, \tag{5.6}$$

Then, the gradient (p_k, q_k) become

$$p_k = \frac{\hat{\alpha}_k}{\hat{\gamma}_k}, \qquad q_k = \frac{\hat{\beta}_k}{\hat{\gamma}_k}, \tag{5.7a}$$

where

$$\begin{aligned} \hat{\alpha}_k &= (y_n - y_m)Z_l + (y_l - y_n)Z_m + (y_m - y_l)Z_n, \\ \hat{\beta}_k &= (x_m - x_n)Z_l + (x_n - x_l)Z_m + (x_l - x_m)Z_n, \\ \hat{\gamma}_k &= (x_n y_m - x_m y_n) + (x_l y_n - x_n y_l) + (x_m y_l - x_l y_m). \end{aligned} \tag{5.7b}$$

We can express the textural reflectance map as a function of variables Z_l, Z_m and Z_n by substituting (5.7a) and (5.7b) into (5.4), i.e.

$$R_k(Z_l, Z_m, Z_n) = \frac{\hat{\gamma}_k}{(\hat{\alpha}_k{}^2 + \hat{\beta}_k{}^2 + \hat{\gamma}_k{}^2)^{1/2}}. \tag{5.8}$$

By taking the linear approximation of this reflectance map about a certain reference nodal height, the same shape reconstruction procedure follows as described in Section 4.

6 Experimental Results

In this section, experimental results are given to illustrated the performance of the proposed SFT algorithm. Several synthetic and real textured images obtained via perspective projection of planar and curved textured surfaces are used as test problems. In our experiments, the dot or edge density is measured to represent the density of textural primitives, the size of the window W is chosen to be 31×31, and the test function $t(\tilde{x}, \tilde{y})$ is the Gaussian function with $\sigma = 15$ for all test images. The textural intensity map is constructed by the estimated edge density map, We perform the triangulation of a given image with a regular grid of spacing $h = 2$. For each test problem, we scale the dynamic range of the estimated textural intensity map for its visualization. Even though we use the regular grid in the image plane, the corresponding grid in the world space may become sparse

and irregular. Thus, to display a fine and smooth reconstructed surface, we perform interpolation based on the set of computed surface nodal heights.

Test Problem 1:

In Fig. 5(a), we show a test plane which is tilted 225° and slanted 45°. A 128 × 128 synthetic perspectively projected image of a portion of this plane covered with dot textures are shown in Fig. 5(b). The textural intensity is estimated by calculating the dot density with in a window and depicted in Fig. 5(c). Fig. 5(d) represents the reconstructed shape. Note that although the image is square, the reconstructed surface is no longer supported by square domain in world plane since point (X,Y) depends on the depth Z to that point. Therefore, both the actual surface corresponding to the image in 5(b) and the reconstructed surface only contain a portion of a ground truth as shown in 5(a). One can clearly see that the reconstructed surface is quite close to the original one.

Test Problem 2:

Fig. 6(a) shows another test plane which is tilted 180° and slanted 60°. Figs. 6(b)-(d) are a projected image of a portion of this plane covered with dot textures, the corresponding textural intensity and the reconstructed surface. Note again that the actual surface corresponding to the square input image as given in 6 (b) does not have a square support since the plane is slanted.

Test Problem 3:

A convex cylindrical test surface covered with the grid texture and its perspectively projected image are shown in Figs. 7(a) and (b), respectively. The estimated textural intensity is given in Fig. 7(c) and the reconstructed surface is shown in Fig. 7(d). One can see that the reconstructed surface is very close to the actual one.

Test Problem 4:

The last test problem is a real image of a textured surface. It is a bump surface covered with line segment textures as shown in Fig. 8(a). Fig. 8(b) shows the edge map of the input image. The estimated textural intensity and the reconstructed surface are shown in Figs. 8(c) and (d), respectively. The input texture image has a lot noises so does the calculated edge map. The estimated textural intensity map in 8(c) is not smooth. Even though the textural intensity estimation is not satisfying, the shape of the object can still be extracted by our algorithm.

7 Conclusion

In this research, we proposed a new iterative algorithm for direct shape recovery of curved objects from a textured image under perspective projection. By approximating a surface with a triangular element surface model, we relate textural intensity (which is inversely proportional to the textural density) directly to the nodal height variables. Then, we linearize the nonlinear textural reflectance map around a certain nodal height value so that a linear relationship between the textural intensity over a triangle and the surface nodal variables is established. The depth map is determined by minimizing a cost functional consisting of the sum of textural intensity error associated with each triangle and a smoothness constraint term parameterized by the surface nodal variables. The recovery algorithm refines the reconstructed surface iteratively by a successive linearization scheme, where the linearization is performed based on updated surface nodal heights obtained from the previous iteration. We discussed several methods to estimate the textural intensity using a projected texture image. With the textural intensity estimated by texel or edgel density measure, we demonstrated the performance of the proposed algorithm with several synthetic and real texture images. Since the quality of the results of our algorithm is affected by the accuracy of the estimated textural intensity, better results can be obtained by using a more robust textural intensity estimation scheme. This is an interesting problem for future investigation.

References

[1] J. Aloimonos, "Shape from texture," *Biological Cybernetics*, Vol. 58, pp. 345–360, 1988.

[2] J. Aloimonos and M. Swain, "Shape from patterns: regularization," *International Journal of Computer Vision*, Vol. 2, pp. 171–187, 1988.

[3] R. Bajcsy and L. Lieberman, "Texture gradient as a depth cue," *Computer Graphics and Image Processing*, Vol. 5, pp. 52–67, 1976.

[4] D. Blostein and N. Ahuja, "Shape from texture: integrating texture-element extraction and surface estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-11, pp. 485–492, September 1989.

[5] L. G. Brown and H. Shvaytser, "Surface orientation from projective foreshortening of isotropic texture autocorrelation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 6, pp. 584–588, 1990.

[6] L. S. Davis, L. Janos, and S. M. Dunn, "Efficient recovery of shape from texture," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, pp. 485–492, September 1983.

[7] J. J. Gibson, "The perception of visual surface," *American Journal of Psychology*, Vol. 63, pp. 367–384, 1950.

[8] R. M. Haralick and L. G. Shapiro, *Computer and robot vision*, Addison Wesley, 1992.

[9] B. K. P. Horn and M. J. Brooks, *Shape from Shading*, Cambridge, Massachusetts: The MIT Press, 1989.

[10] K. Ikeuchi, "Shape from regular patterns," *Artificial Intelligence*, Vol. 22, pp. 49–75, 1984.

[11] J. Y. Jau and R. T. Chin, "Shape from texture using the Wigner distribution," *Computer Graphics and Image Processing*, Vol. 52, pp. 248–263, 1990.

[12] K. Kanatani and T. Chou, "Shape from texture: General principle," *Artificial Intelligence*, Vol. 38, No. 2, pp. 1–48, 1989.

[13] J. Kender, "Shape from texture," in *International Joint Conference on Artificial Intelligence*, (Tokyo, Japan), pp. 475–480, August 1979.

[14] J. Krumm and S. A. Shafer, "Shape from periodic texture using the spectrogram," in *IEEE Conference on Computer Vision and Pattern Recognition*, (Champaign, Illinois), pp. 284–289, June 1992.

[15] K. M. Lee and C.-C. J. Kuo, "Shape from shading with a linear triangular element surface model," Tech. Rep. 172, USC, Signal and Image Processing Institute, 1991. Also to appear in *IEEE Trans. on Pattern Analysis and Machine Intelligence.*

[16] K. M. Lee and C.-C. J. Kuo, "Shape from shading with perspective projection," Tech. Rep. 216, USC, Signal and Image Processing Institute, 1992.

[17] C. Marinos and A. Blake, "Shape from texture: the homogeneity hypothesis," in *Proc. of International Cof. on Computer Vision*, pp. 350–353, 1990.

[18] Y. Ohta, K. Maenobu, and T. Sakai, "Obtaining surface orientation from texels under perspective projection," in *International Joint Conference on Artificial Intelligence*, (Vancouver, Canada), pp. 746–751, 1981.

[19] M. A. S. Patel and F. S. Cohen, "Shape from texture using Markov random field model and stereo-windows," in *IEEE Conference on Computer Vision and Pattern Recognition*, (Champaign, Illinois), pp. 290–295, June 1992.

[20] A. Rosenfeld, "A note on automatic detection of texture gradients," *IEEE Trans. Computers*, Vol. C-24, pp. 988–991, 1975.

[21] B. J. Super and A. C. Bovik, "Shape-from-texture by wavelet-based measurement of local spectral moments," in *IEEE Conference on Computer Vision and Pattern Recognition*, (Champaign, Illinois), pp. 296–300, June 1992.

[22] H. Voorhees and T. Poggio, "Detection textons and texture boundaries in natural textures," in *Proc. of the First International Cof. on Computer Vision*, pp. 250–258, 1987.

[23] A. P. Witkin, "Recovering surface shape and orientation from texture," *Artificial Intelligence*, Vol. 17, pp. 17–45, 1981.

Figure Captions

Figure 1: Perspective projection.

Figure 2: Perspective projection of a triangular surface patch.

Figure 3: Approximation of a perspective projection model using solid angle.

Figure 4: Orthographic projection model.

Figure 5: Test Problem 1: (a) ground truth of a planar surface; (b) perspectively projected image of dot textures; (c) computed textural intensity; (d) reconstructed surface.

Figure 6: Test Problem 2: (a) ground truth of a planar surface (b) perspectively projected image of dot textures; (c) computed textural intensity; (d) reconstructed surface.

Figure 7: Test Problem 3: (a) ground truth of a cylindrical surface; (b) perspectively projected image of grid textures; (c) computed textural intensity; (d) reconstructed surface.

Figure 8: Test Problem 4: (a) real textured image; (b) edge image; (c) computed textural intensity; (d) reconstructed surface.

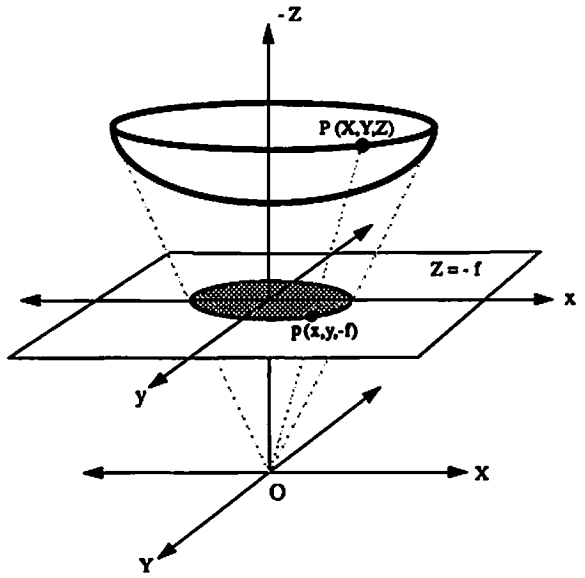


Figure1: Perspective projection.

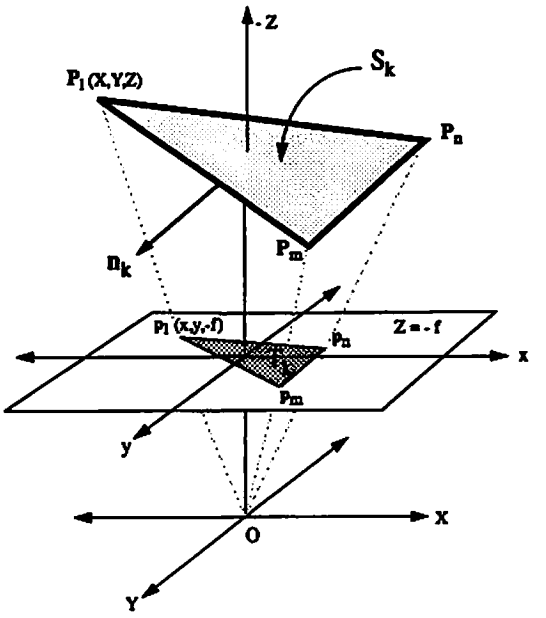


Figure 2: Perspective projection of a triangular surface patch.

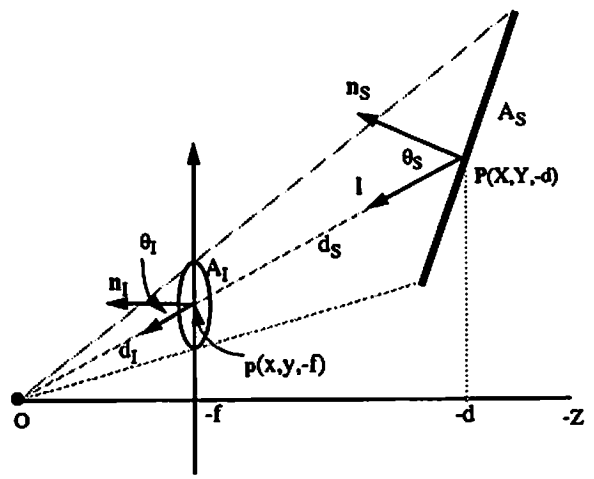


Figure 3: Approximation of a perspective projection using solid angle.

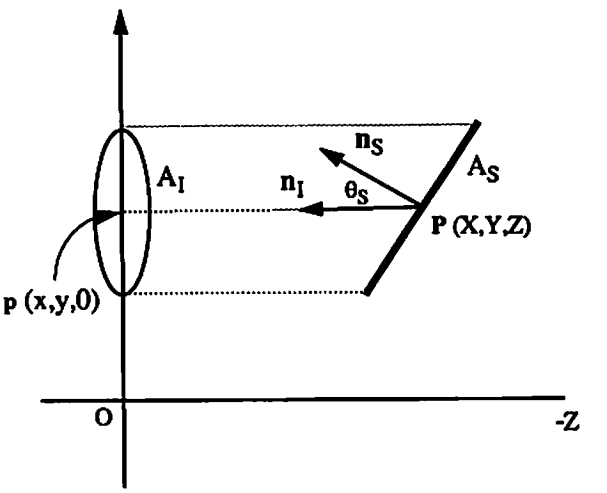


Figure 4: Orthographic projection model.

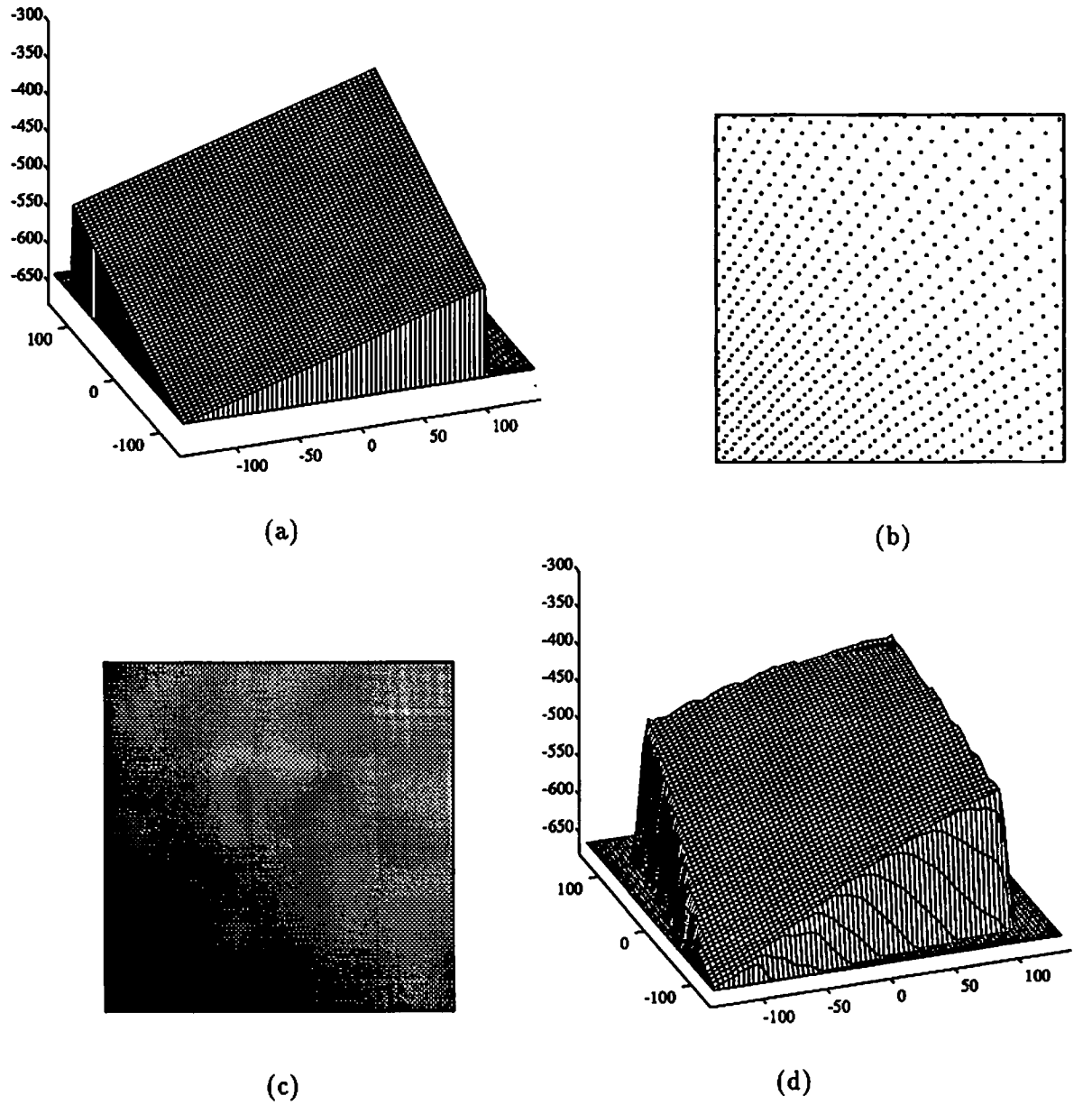


Figure 1: Test Problem 1: (a) ground truth of a planar surface; (b) perspectively projected image of dot textures; (c) computed textural intensity; (d) reconstructed surface.

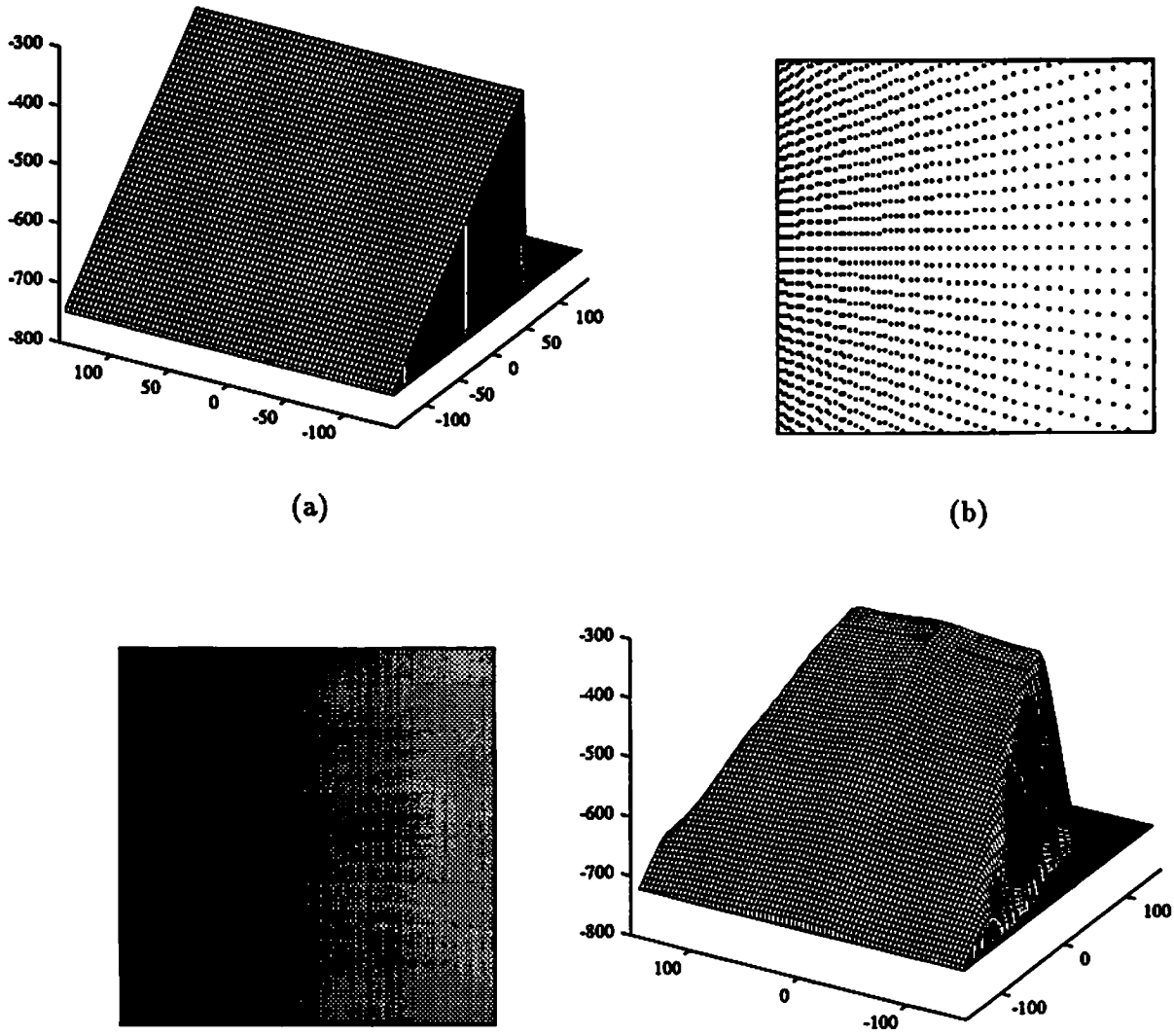


Figure 2: Test Problem 2: (a) ground truth of a planar surface; (b) perspectively projected image of dot textures; (c) computed textural intensity; (d) reconstructed surface.

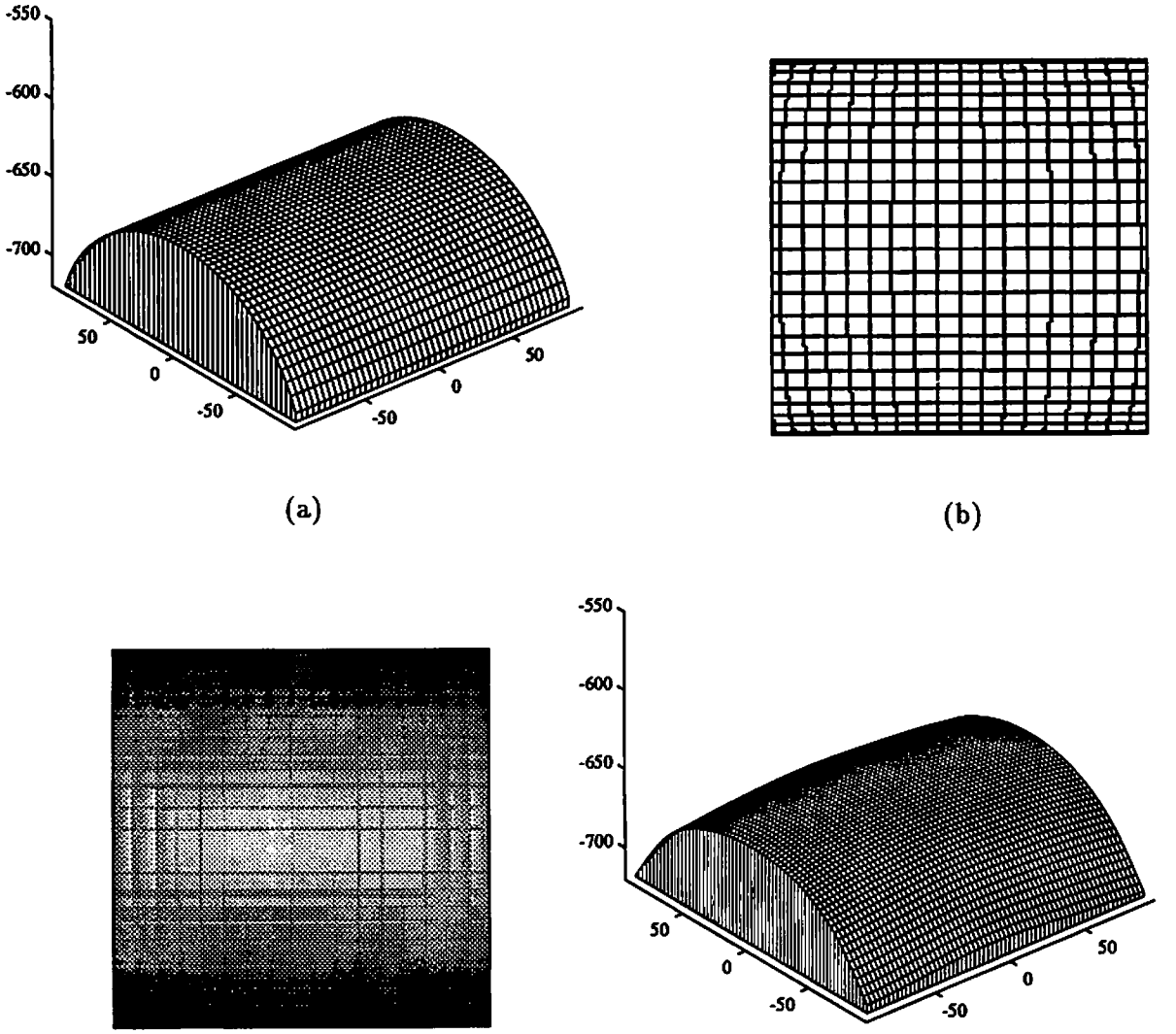
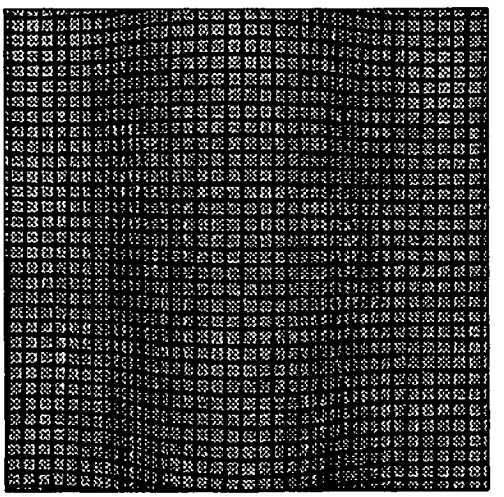
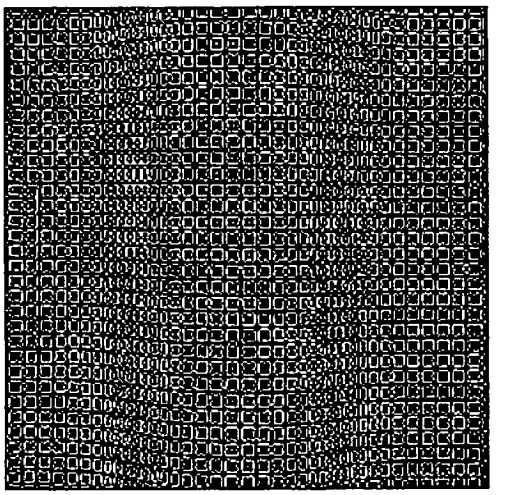


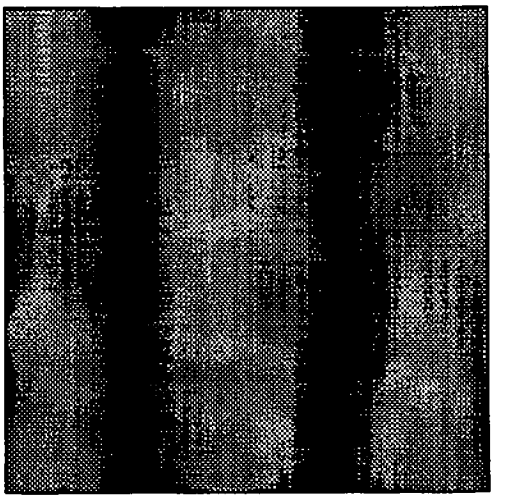
Figure 3: Test Problem 3: (a) ground truth of a cylindrical surface; (b) perspectively projected image of grid textures; (c) computed textural intensity; (d) reconstructed surface.



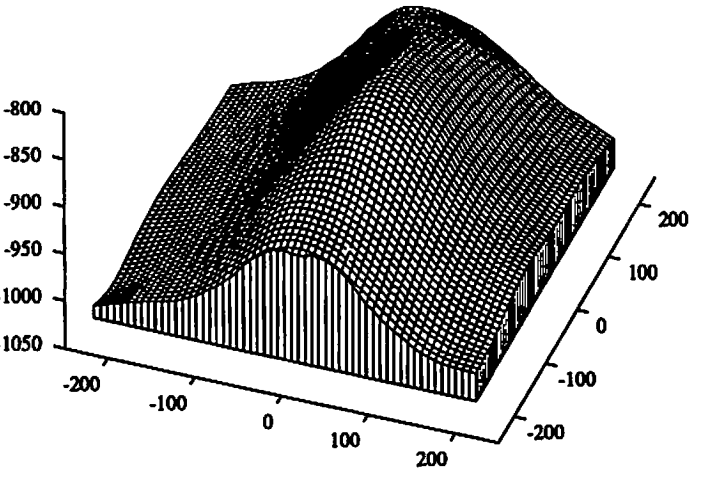
(a)



(b)



(c)



(d)

Figure 4: Test Problem 4: (a) real textured image; (b) edge image; (c) computed textural intensity; (d) reconstructed surface.