COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 62, No. 2, July, pp. 47–58, 1995

Tracking Level Sets by Level Sets: A Method for Solving the Shape from Shading Problem

RON KIMMEL

Technion, Electrical Engineering Department, Technion-City, Haifa 32000, Israel

AND

ALFRED M. BRUCKSTEIN

Technion, Computer Science Department, Technion-City, Haifa 32000, Israel

Received September 16, 1993; accepted May 25, 1994

A new algorithm for recovering depth to a *Lambertian* $\mathscr{C}^1$ smooth object given its gray-level image under uniform illumination from the viewing direction is presented. To recover depth, an almost arbitrarily initialized surface is numerically propagated on a rectangular grid, so that a level set of this surface tracks the height contours of the depth function. The image shading controls the propagation of the surface. When the light direction is tilted with respect to the viewing direction the problem is solved by tracking the projection of equal-height contours defined with respect to the light source direction. This projection approach provides a solution that overcomes ambiguity problems encountered in previous work, while the level set approach of implementing the contour propagation overcomes numerical problems and some of the topology problems of the evolving contours. © 1995 Academic Press, Inc.

1. INTRODUCTION

The problem of reconstructing three-dimensional objects from their shaded images has been extensively investigated by computer vision researchers over the past two decades, starting from the classic "shape from shading" work of Horn [5]. Until recently, iterative schemes to solve this problem were derived by regularization versions of the direct problem formulation; see [6] for a survey on these methods. New iterative methods were recently suggested by Oliensis and Dupuis [14], and by Rouy and Tourin [20] and Osher and Rudin [15] based on viscosity solutions to partial differential equations. They represented the shape from shading problem as a first-order *Hamilton–Jacobi* equation. Viscosity solutions to the Hamilton–Jacobi equation guarantee uniqueness (up to +/− ambiguity of the shape from shading problem) and existence. The viscosity solutions are related to the smoothness of the Hamiltonian under consideration.

A direct approach to solving the shape from shading problem based on the characteristic strip expansion method was provided by Horn in his original work. A different direct approach, using equal-height contour propagation, was suggested by Bruckstein [1, 2]. An accurate numerical scheme for implementing this method based on a level set propagation algorithm devised by Osher and Sethian [18] was then developed in [8, 10] and was also compared with viscosity solutions and other traditional methods in [11]. In [10], the numerical properties of the level set-based approximation were explored, like the effects of noise and its performance on real images. This novel level set-based formulation for the equal-height contour propagation method for the shape from shading problem turned out to be a general method for solving Dirichlet problems for Hamilton–Jacobi equations; see Osher [17].

In this paper we first summarize the level set-based numerical approach to the equal-height contour propagation method for solving the shape from shading problem. As mentioned above it is based on some nice results on numerical approximation of front propagation, published by Osher and Sethian in [18]. Then we derive the contour propagation equation for the case of general light source direction, where the contours are "equal-height" with respect to the light source direction instead of the height axis; see Fig. 1. This new formulation solves some ambiguity problems that were encountered in our previous work [8, 10, 11]. We note that light source coordinates were used in [12] to improve the shape from shading method of [7, 19] and later in [3] as a natural coordinate system with respect to the singular points (the brightness points in the image).

In the next section we give a formal definition of the shape from shading problem under Lambertian reflectance. In Section 3, the equal-height contour approach to the shape from shading problem is summarized and the

1077-3142/95 $12.00

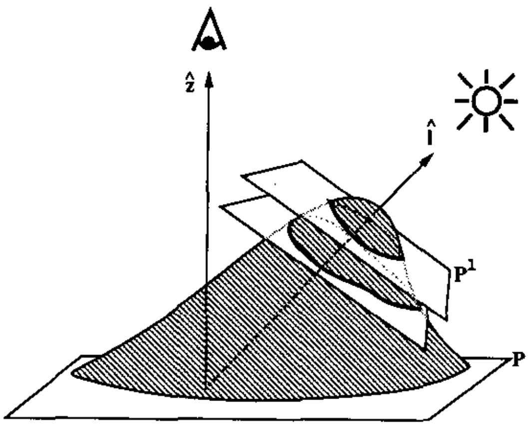

FIG. 1. The case where the viewer and the light source are at different directions ($\hat{l} \neq \hat{z}$) is solved by propagating curves which are "equal-height" contours with respect to the given light source direction ($\hat{l}$).

new formulation to the general light source direction is presented. Section 4 presents the level set-based algorithm for tracking the equal height contours. Section 5 presents results of applying the new algorithm to some synthetic as well as real shading images.

2. PROBLEM FORMULATION AND THE REFLECTANCE MAP

Suppose we are given a continuous function of two variables, $z(x, y)$, which describes the surface of an object. The shaded image of that surface is defined as a brightness distribution $E(x, y)$. The brightness values are defined by the properties of the surface, by its orientation at every point, and by the illumination. The brightness $E(x, y)$ is determined via a so-called shading rule or *reflectance map*, which characterizes the surface properties and provides an explicit connection between the image and the surface orientation. The shape from shading problem is to recover the depth function $z(x, y)$, from the image $E(x, y)$.

Let us first specify the surface orientation using the components of the surface gradient $p = \partial z/\partial x$ and $q = \partial z/\partial y$. The surface unit normal can then be written as

$$\hat{N} = \frac{1}{\sqrt{1 + p^2 + q^2}}(-p, -q, 1)^{\mathrm{T}}.$$

In case of a surface with so-called Lambertian, or diffuse reflection properties and uniform illumination, $E(x, y)$ is proportional to the cosine of the angle between the surface normal $\hat{N}$ and the direction to the light source $\hat{l}$ (see Fig. 2).

In the general case we define the light source direction as

$$\hat{l} = \frac{1}{\sqrt{1 + p_l^2 + q_l^2}}(-p_l, -q_l, 1)^{\mathrm{T}}. \tag{1}$$

The dependence between brightness and surface orientation (the reflectance map) can be written as a function mapping the surface normal direction $\hat{N}$ to the brightness image $E(x, y)$ as

$$E(x, y) = \text{Function of } (\hat{N}) = R(p(x, y), q(x, y)). \tag{2}$$

This is the image irradiance equation. For example in the Lambertian case we have

$$E(x, y) = R(p, q) = \hat{l} \cdot \hat{N} = \frac{1 + p_l p + q_l q}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_l^2 + q_l^2}}.$$

In this case the normal direction lies on an ambiguity cone whose main axis is directed toward the light source; see Fig. 3b. There are some constant factors multiplying the R function due to the surface albedo, the beam luminance, etc., in the above equation. It is, however, possible to factor them out by simply rescaling the image gray levels.

For the simple case where $\hat{l} = (0, 0, 1)^{\mathrm{T}}$ we have

$$R(p, q) = \frac{1}{\sqrt{1 + p^2 + q^2}},$$

and the ambiguity of the surface normal direction is an upward directed cone; see Fig. 3a.

Equation (2) is a nonlinear partial differential equation that has to be satisfied by the surface $z(x, y)$. Therefore, solving the shape from shading problem amounts to solving a nonlinear partial differential equation. Clearly

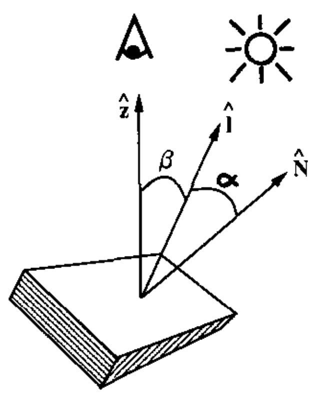

FIG. 2. On a patch of a surface, the brightness under Lambertian shading rule is given by the cosine of the angle between the surface normal and the light source direction, $E = \cos \alpha = \hat{N} \cdot \hat{l}$.

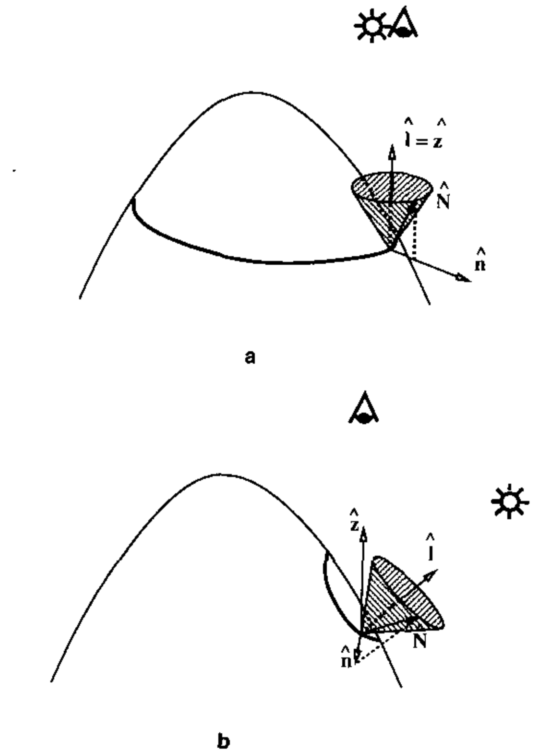

FIG. 3. (a) When the light source is from above ($\hat{l} = \hat{z}$) the ambiguity cone is directed upward. (b) For other light source directions the ambiguity cone is tilted toward the light source.

boundary conditions are necessary. Any shape from shading scheme reconstructs a function $z(x, y)$ by implicitly or explicitly recovering its normal $\hat{N}(x, y)$ everywhere. The surface normal at each point is represented by two numbers, and the only constraint we have so far is the image's irradiance equation (2). The two variables representing the surface normal direction at each point can only be computed by using more than one equation. The "art" of recovering a shape from its shaded image requires introducing boundary conditions and local constraints that follow from reasonable assumptions concerning the relation of each point on the surface to its neighborhood.

3. SHAPE FROM SHADING VIA EQUAL-HEIGHT CONTOURS

3.1. *Light Source from Above*

Let us first briefly review Bruckstein's shape from shading approach (see [1, 2]). An equal-height contour or a *level curve* is a continuous curve in the (x, y) plane on which the function $z(x, y)$ is constant. Defining $\{x(s), y(s)\}$, $s \in [0, S]$, as the parametric representation of the contour, we have

$$\frac{\partial z}{\partial s} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial s} = 0$$

or[1]

$$p x_s + q y_s = 0.$$

The unit normal to the equal-height contour on the (x, y) plane is given by

$$\hat{n}(s) = \frac{1}{\sqrt{x_s^2(s) + y_s^2(s)}}(y_s(s), -x_s(s)).$$

Clearly $\hat{n}$ is in the direction of the projection of $\hat{N}$ on the image plane. When considering the case of propagation from small circles around the singular points the unit normal $\hat{n}$ is directed outward, i.e., pointing outside the region bounded by the curve. Consider the case where $\hat{l} = \hat{z}$ (that is, $\hat{l} = (0, 0, 1)$) and define dz as the height we climb while progressing a distance $\mathcal{D}$ in the normal direction $\hat{n}$ in the (x, y) plane. From elementary geometry we have

$$\mathcal{D} = dz \cot \alpha,$$

where α is the surface orientation angle, the angle between the surface normal $\hat{N}$ and the light source direction $\hat{l}$ ($\cos \alpha = \hat{l} \cdot \hat{N}$). Under the simple Lambertian shading rule where $R(p, q) = \cos \alpha = 1/\sqrt{1 + p^2 + q^2}$, we have

$$\mathcal{D} = dz \cot \alpha = dz \frac{1}{\sqrt{p^2 + q^2}} = dz \frac{E}{\sqrt{1 - E^2}}.$$

If, from the first contour, we uniformly climb dz, we get to the next equal-height contour via

$$\{x(s, dz), y(s, dz)\} = \{x(s, 0), y(s, 0)\} + \mathcal{D}(s) \cdot \hat{n}.$$

This yields the propagation of the equal-height contours as a nonlinear initial value *PDE* problem. Given $\{x(s, 0), y(s, 0)\}$ the evolution equations are [1, 2]

$$\begin{cases} x_t(s, t) = F(x, y)\dfrac{y_s}{\sqrt{x_s^2 + y_s^2}}, \\ y_t(s, t) = F(x, y)\dfrac{-x_s}{\sqrt{x_s^2 + y_s^2}} \end{cases} \tag{3}$$

where $t \equiv z$ and

$$F(x(s, t), y(s, t)) = \frac{E(x(s, t), y(s, t))}{\sqrt{1 - E^2(x(s, t), y(s, t))}}. \tag{4}$$

[1] The subscripts s, t, x, y, refer to partial derivatives, (e.g., $x_t \equiv \partial x/\partial t$), and the subscripts (and superscripts) l, n refer to the light source direction and the planar normal, respectively.

Note that z is replaced by t so that the equal height contour propagation in height is now referred to as a curve evolving in "time."

Define $C(0) = \{x\ (s, 0), y(s, 0)\}$ as the smooth (and, in some cases, closed) initial equal-height curve, and $C(t)$ as the one-parameter family of curves generated by moving $C(0)$ along its normal vector field with a speed given by F. Here, F is a given scalar function of the brightness E. Using this notation, Eq. (3) can be written as a planar curve evolution equation $(\partial/\partial t)C(t) = F(x, y) \cdot \hat{n}$. Sethian [21] called the direct implementation of such propagation models "Lagrangian" evolution equations because the physical coordinate system moves with the propagataing front.

3.2. *General Light Source Direction*

When the light source direction is $\hat{l}$ (as defined in (1)), the brightness map under the Lambertian shading rule is $E = \hat{l} \cdot \hat{N}$. In this case the surface normal is on a tilted ambiguity cone as described in Fig. 3b. In [8, 10] it was discussed how to propagate an equal-height contour in this situation. It was noted that there are possible ambiguities in recovering the normal direction, and these had to be solved by adding smoothness constraints along the equal-height contour or along the surface. In the rest of this section we present a different kind of curve propagation that overcomes these ambiguities and frees us from the need to consider additional constraints while reconstructing the shape.

Consider the plane P^l defined by the light source direction $\hat{l}$; see Fig. 1. The equal-height contours with respect to this plane are given by the previous propagation rule

$$C_t^l = F^l(x^l, y^l)\, \mathbf{t} \times \hat{l},$$

where $\mathbf{t}$ is the tangent vector to C^l, and F^l is the velocity which is a function of the image as it would have been seen from the light source direction. This evolution may be projected to the (x, y) plane by first projecting the velocity and then taking only the normal component of the projected velocity v_n, since in the case of planar curve propagation the tangent component of the velocity affects only the internal parameterization of the curve while the trace of the curve is determined only by the normal component (see [4] for justification). The projected evolution is then given by

$$C_t = \tilde{F} v_n \hat{n},$$

where the correction factor v_n is calculated in the Appendix (see also [9] for a similar approach to solve the shortest path problem on surfaces). The fact that the contours lie on parallel planes, constant heights with respect to $\hat{l}$, enables us to predict the "deformation" in time of $\tilde{F}$. Define $\cos \beta = \hat{z} \cdot \hat{l}$ (where $\hat{z} \equiv (0, 0, 1)$), then

$$\tilde{F}(x, y, t) = F\left(x - tp_l/\sqrt{1 + p_l^2 + q_l^2},\; y - tq_l/\sqrt{1 + p_l^2 + q_l^2}\right),$$

which means that the image shifts in time along the direction defined by $(-p_l, -q_l)/\sqrt{p_l^2 + q_l^2}$ with a constant velocity of $\sin \beta$; see Fig. 4a.

This formulation forces us to interpolate the image intensity between the pixels as the image moves in time. Another problem is the fact that we calculate the heights with respect to P^l so that in the last stage we should project the heights from P to P^l, or stretch the elevation array given on a uniform grid in P with respect to the light source direction. This stretching results in a nonuniform elevation array.

Another manipulation on the evolution equation frees us from the need to take care of these problems. Instead of shifting the image forward it is possible to include a

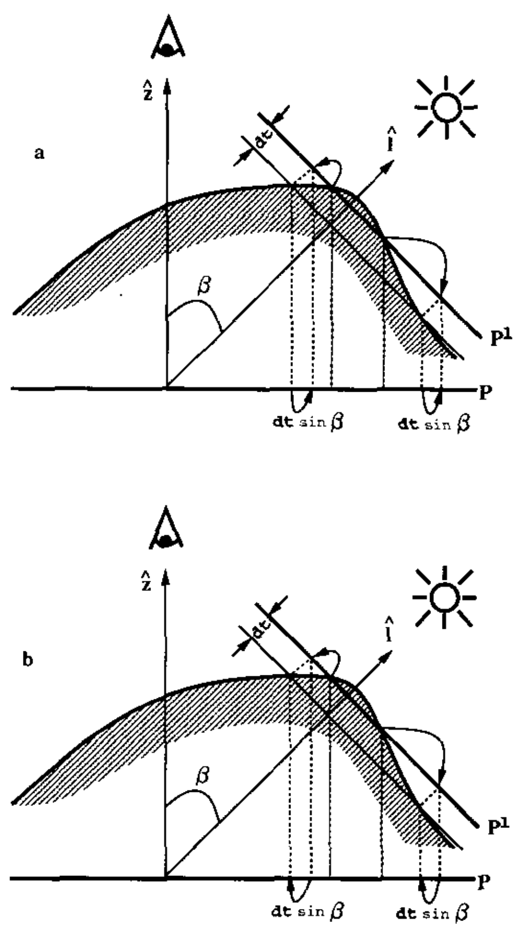

FIG. 4. (a) When propagating an equal-height contour with respect to $\hat{l}$ on P^l, the underlying image which specifies the propagation velocity shifts in time with a constant velocity of $\sin \beta$. (b) The constant velocity of the shifting image may be converted to a constant shifting of the evolving curve.

backward propagation constant velocity term in the curve evolution rule. This way we can keep the image in its place and transfer the constant motion to the evolving curve; see Fig. 4b. It also frees us from the need to project the heights to the P^l plane, and the height at each grid point may be calculated without any interpolation. The new evolution rule is given by

$$C_t = F(x, y)v_n\hat{n} - \sin\beta \frac{(-p_l, -q_l)}{\sqrt{1 + p_l^2 + q_l^2}}, \tag{5}$$

and again, taking only the normal component of the second term (the tangential components have no effect on traces of the evolving curves), we obtain

$$C_t = \left(F(x, y)v_n + \frac{(p_l n_1 + q_l n_2)}{\sqrt{1 + p_l^2 + q_l^2}}\right)\hat{n}, \tag{6}$$

where $\hat{n} = (n_1, n_2)$. By using the v_n obtained in the Appendix, we define

$$G(x, y, \hat{n}) = \frac{1}{\sqrt{1 + p_l^2 + q_l^2}} \times [F(x, y)\sqrt{n_1^2(1 + q_l^2) + n_2^2(1 + p_l^2) - n_1 n_2 2 p_l q_l} + (p_l n_1 + q_l n_2)]. \tag{7}$$

Then, Eq. (6) may be written as

$$C_t = G(x, y, \hat{n})\hat{n}, \tag{8}$$

which is again a Lagrangian formulation of the new evolution rule. Note that Eq. (3) is a special case of Eq. (8).

Direct numerical implementations of Lagrangian formulations suffer from instability and topological problems due to the fact that local representations of propagating fronts are followed; see [18, 21]. Control algorithms are needed where topological changes occur. If, for example, we start with two separate closed contours that grow up to a merging point from which they continue to grow as a single contour, it is necessary to handle this merging process by an external control procedure. To avoid the various problems that occur in this approach, a novel and miraculous "Eulerian formulation," described below, was developed by Osher and Sethian [18].

4. SOLUTION VIA THE EULERIAN FORMULATION

The Eulerian scheme is a recursive procedure that propagates height contours while inherently implementing a so-called *entropy condition*. This entropy condition is necessary to select the "physically correct" propagation [13] among all possible weak solutions.

Introduce a function $\phi(x, y, t)$ initialized so that $\phi(x, y, 0) = 0$ yields the curve $C(0)$. Assume that $C(0)$ is a closed curve and restrict ϕ to be negative in the interior and positive in the exterior of the level set $\phi(x, y, 0) = 0$. Furthermore ϕ has to be smooth and Lipschitz continuous.

The idea is to determine an evolution of the surface $\phi(x, y, t)$ so that the level sets $\phi(x, y, t) = 0$ provide the height contours $C(t)$ as if propagated by (8) and obeying the entropy condition. If $\phi(x, y, t) = 0$ along $C(t)$ then, by the chain rule, we have

$$\phi_t + \nabla\phi \cdot C_t(t) = 0. \tag{9}$$

The scalar velocity of each curve point in its normal direction is

$$G = C_t(t) \cdot \hat{n}(t). \tag{10}$$

In our case the velocity is given by the scalar function $G(x, y, \hat{n})$ as defined in Eq. (7), which is a function of the local image brightness and the given normal of the propagating curve. The gradient $\nabla\phi \equiv (\partial/\partial x, \partial/\partial y)\phi$ is always normal to the curve given by $\phi(x, y, t) = 0$ so that $\hat{n}(s, t) = -\nabla\phi/\|\nabla\phi\|$, the minus sign indicating the inward direction of propagation, hence

$$G = C_t \cdot \frac{-\nabla\phi}{\|\nabla\phi\|}. \tag{11}$$

Substituting this into (9) yields

$$\phi_t - G\|\nabla\phi\| = 0, \tag{12}$$

or explicitly

$$\phi_t = \frac{1}{\sqrt{1 + p_l^2 + q_l^2}} \times [F(x, y)\sqrt{\phi_x^2(1 + q_l^2) + \phi_y^2(1 + p_l^2) - \phi_x\phi_y 2 p_l q_l} - (p_l\phi_x + q_l\phi_y)]. \tag{13}$$

Sethian called this approach based on propagating the ϕ function *Eulerian*, since the coordinates here are the natural physical coordinates (x, y). Therefore, if we have a surface ϕ propagating according to (13) with the level set $\phi(x, y, 0) = 0$ coinciding with $C(0)$, then $\phi(x, y, t) = 0$ will produce $C(t)$ propagated according to (8) and solve topological problems. In order to derive a numerical scheme for the surface propagation equation which obeys the natural entropy condition, we modify results obtained in [18] (see also [22]) based on Hamilton–Jacobi equations, weak solutions, and conservation laws.

Define $\phi_{ij}^n \equiv \phi(i\Delta x, j\Delta y, n\Delta t)$. We use a simple forward derivative approximation in time

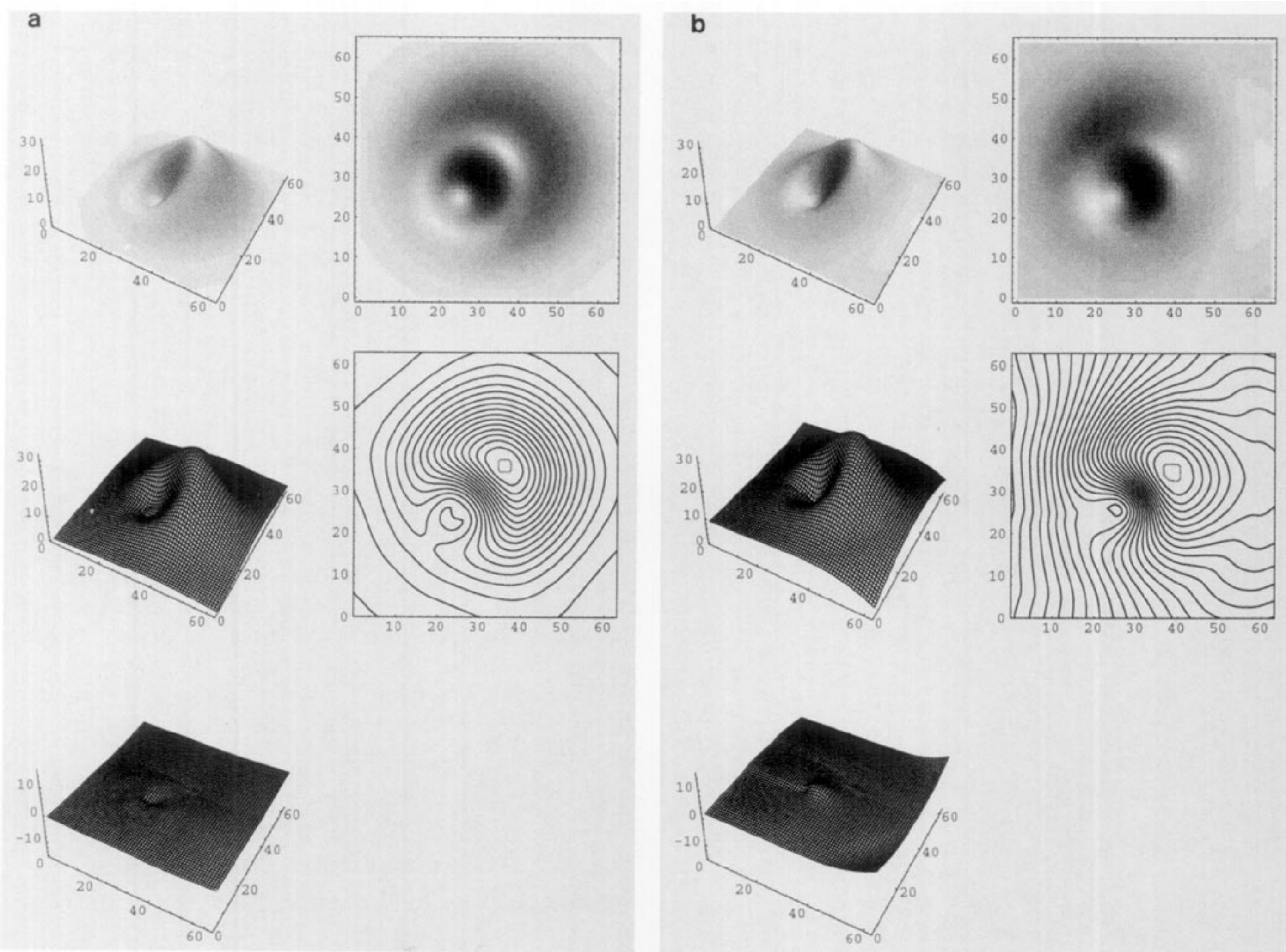

FIG. 5. Reconstruction of synthetic images given on a 64 × 64 grid. (a) The light source direction is from above ($\hat{l} = \hat{z}$). (b) A tilted light source direction $l = (-0.364, 0, 1)$ ($\beta = 20°$). (c) The light source direction is from above ($\hat{l} = \hat{z}$). (d) A tilted light source direction $l = (-0.364, 0, 1)$ ($\beta = 20°$). (e) The light source direction is from above ($\hat{l} = \hat{z}$). (f) A tilted light source direction $l = (-0.364, 0, 1)$ ($\beta = 20°$). (g) The light source direction is from above ($\hat{l} = \hat{z}$).

$$\phi_t|_{t=n\Delta t} \approx \frac{\phi^{n+1} - \phi^n}{\Delta t},$$

and slope-limiter approximations [13] for the spatial derivatives. The squared terms are approximated by

$$\phi_x^2|_{x=i\Delta x, y=j\Delta y} \approx \max(D_+^x\phi_{i,j}, -D_-^x\phi_{i,j}, 0)^2,$$

and the same for ϕ_y^2. Here $D_-^x\phi_i^n = (\phi_i^n - \phi_{i-1}^n)/\Delta x$, $D_+\phi_i^n = (\phi_{i+1}^n - \phi_i^n)/\Delta x$, and the same for the y derivatives. The minmod [13] function is used to approximate ϕ_x and ϕ_y,

$$\phi_x|_{x=i\Delta x, y=j\Delta y} \approx \text{minmod}(D_-^x\phi_{i,j}, D_+^x\phi_{i,j}),$$

and the same for ϕ_y, where

$$\text{minmod}(a, b) = \begin{cases} \text{sign}(a)\min(|a|, |b|) & \text{if } ab > 0 \\ 0 & \text{otherwise.} \end{cases}$$

All the above approximations yield a so-called first-order numerical scheme that approximates Eq. (13). This scheme automatically enforces the entropy condition and frees one from the need to take care of topological changes. In fact this formulation deals with the topology of all up-going (or down-going) surfaces with respect to the light source direction without any external control or outside interference.

The algorithm also deals with the shock formation in the propagating contours which indicates sharp corners

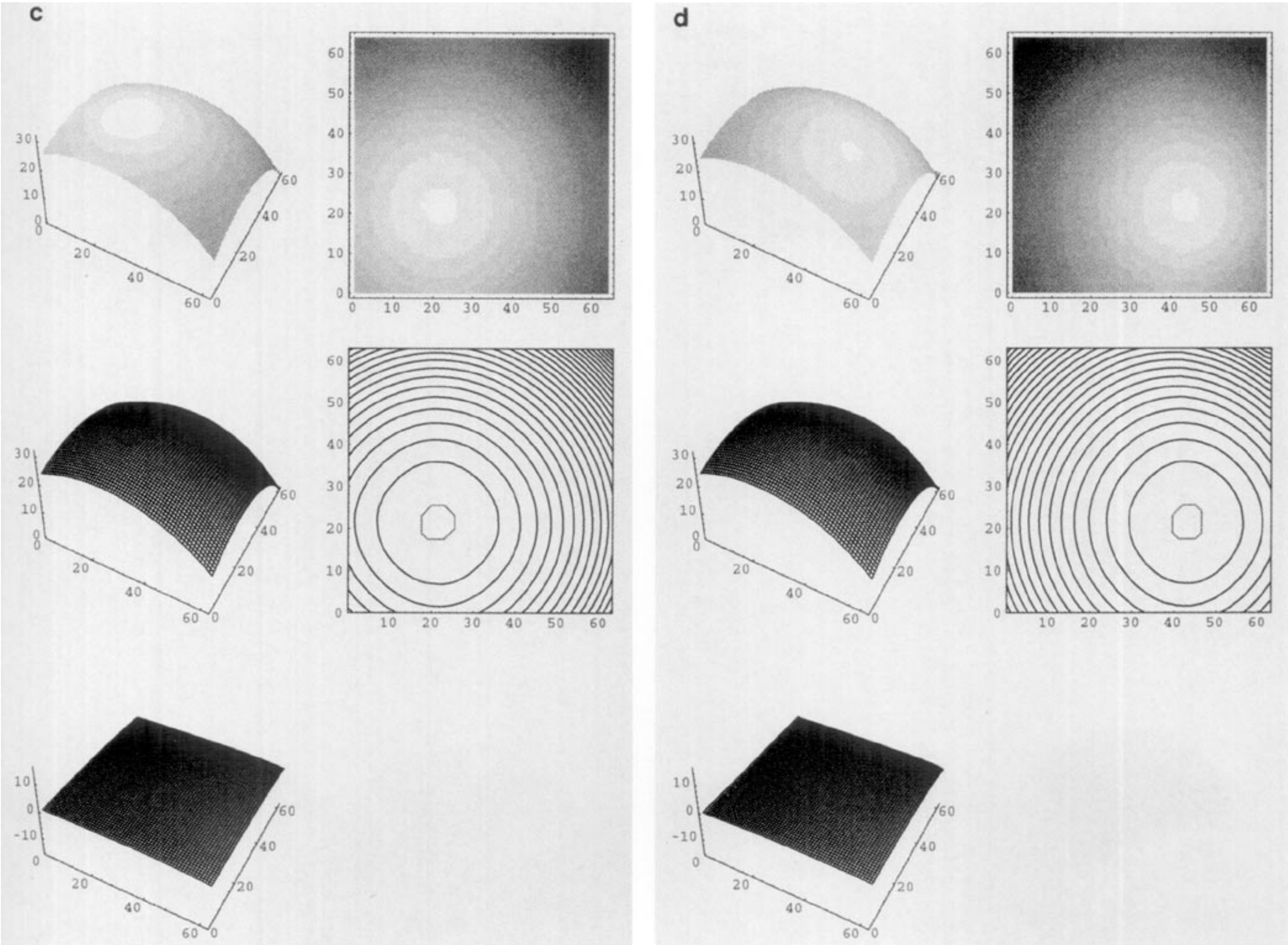

FIG. 5–*Continued*

in the reconstructed surfaces within the numerical flow. One of the great advantages of the Eulerian formulation is that the coordinate system of the propagated ϕ function is fixed, thereby avoiding the stability problems of the Lagrangian formulation. More sophisticated higher order schemes are presented in [16, 18].

In our problem the velocity G is position- and normal-dependent. In order to avoid numerical problems in which $F \rightarrow \infty$, we truncate the brightness function to the maximum values of $E_{\max} < 1$, which yields $F_{\max} = E_{\max}/\sqrt{1 - E_{\max}^2}$.

4.1. *Initialization and Height Assignment*

Every ϕ function which obeys the demands described earlier provides a good initialization. We present a simple way to initiate the ϕ function, obeying smoothness and continuity so that $\phi(x, y, 0) = 0$ gives the initial contour. Let us define the initial height contour by first thresholding the gray levels in the picture and separating all the "singular areas" (the singular areas is the set of points in the image for which $E(x, y) \approx 1$, at such points the normal is known to be $\hat{l}$ and therefore may be used as boundary conditions or candidates to start the propagation). The "singular" points are the saddle, minima, and maxima points with respect to the light source direction. Then use the gray-level function to initiate the ϕ function in a simple way. For example, if the singular areas are defined by $\{(x, y): E(x, y) > T\}$ (where $T \in (0, 1)$ is the selected threshold), then the first level contour can be approximated as the level set $E(x, y) = T$. In this case we can take $\phi(x, y, 0) = E(x, y) - T$ near the selected singular areas as the required initialization, making direct use of the continuity of the gray levels in the picture, without any extra calculations. This is the initialization method used in our examples. Observe that the set $E(x, y) = T$

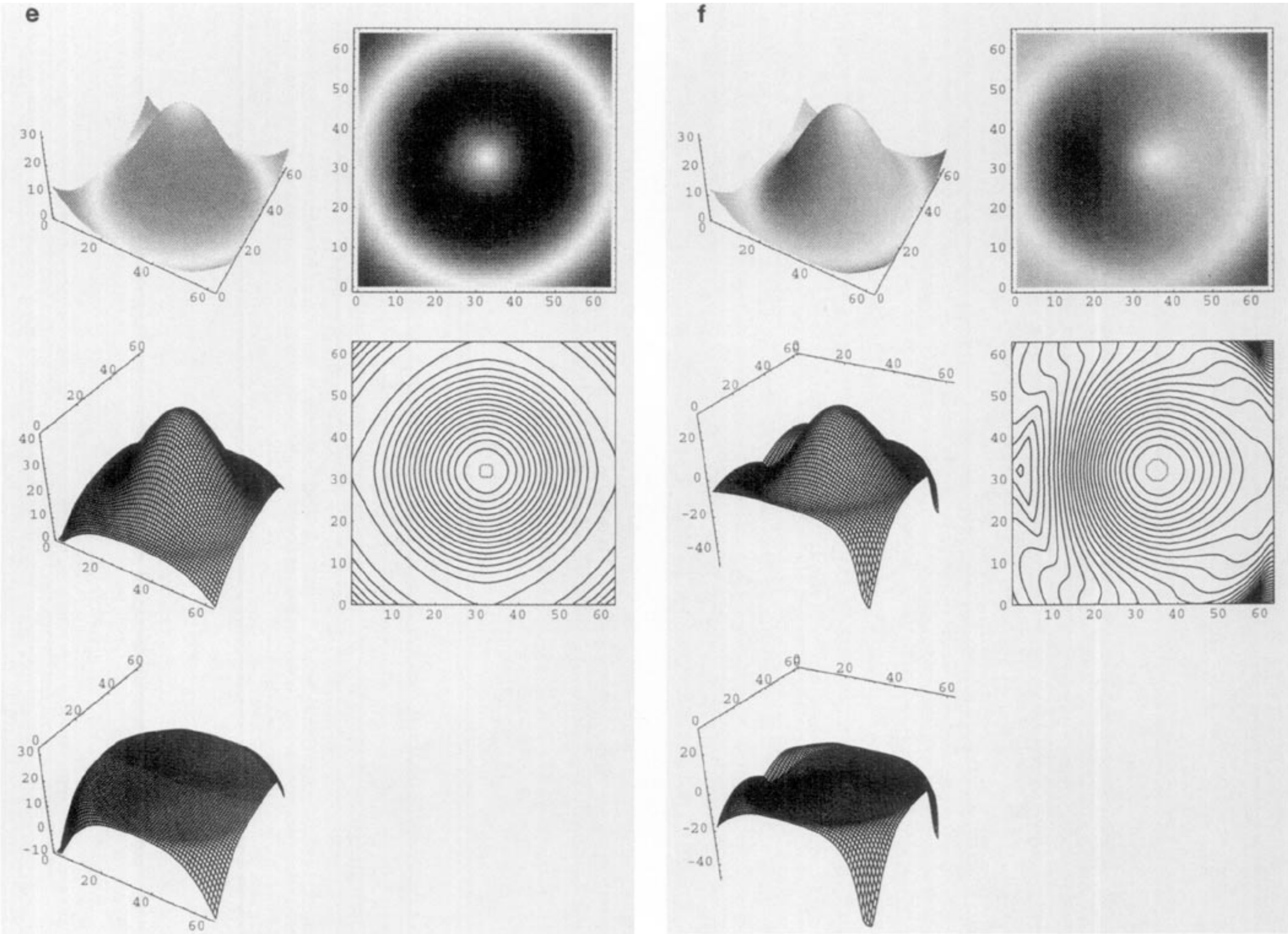

FIG. 5—*Continued*

is not a set of discrete points, but an implicit representation of a curve given by the zero level set of $\phi(x, y, 0)$.

After initialization has been completed the ϕ is propagated according to the above described algorithm. Our goal is to find the elevation at each grid point while the ϕ function is propagated on the grid. A way to achieve accurate results using a simple linear interpolation is as follows:

At every iteration step, for each grid point, check
If $(\phi^n_{i,j} \cdot \phi^{n-1}_{i,j} < 0)$
then $$height_{i,j} = -\left[\Delta t\left(n - \frac{\phi^n_{i,j}}{\phi^n_{i,j} - \phi^{n-1}_{i,j}}\right)\right] \sqrt{1 + p_l^2 + q_l^2} + i\Delta x p_l + j\Delta y q_l.$$

Using the above procedure each grid point gets its height at the "time" when the ϕ function's zero level passes through it.

5. EXAMPLES AND RESULTS

We demonstrate the performance of the proposed algorithm by applying it to several synthetic and real shaded images. The synthetic images were generated for surfaces assumed to be Lambertian, the size of these images being quite small (64 × 64 pixels). The real images were taken by a simple CCD camera and scaled to 128 × 128 pixels. The initialization is achieved by using gray-level thresholding to specify initial singular areas.

The reconstruction of the synthetic models is displayed in Fig. 5 where the upper right is the given shading image, the upper left surface is the original model, the middle left surface is the reconstructed surface, and the middle right is a description of the curve evolution starting from the gray-level contour (dotted line). An error surface for each experiment, computed by subtracting the reconstructed surface from the original one, is displayed on the

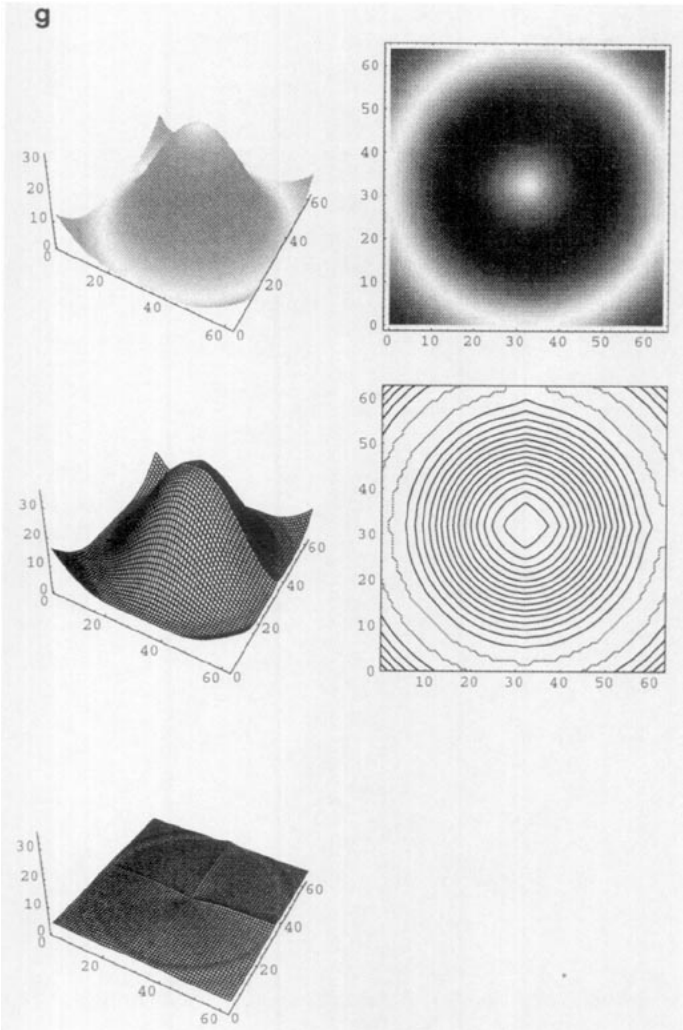

FIG. 5–*Continued*

lower left. For each surface model two light direction were applied; the first is the vertical $\hat{l} = (0, 0, 1)$ (that is, $\beta = 0°$), and the second is a tilted light source direction of $l = (-0.364, 0, 1)$ (that is, $\beta = 20°$). The propagation is performed from the singular area outward, outside the singular region surrounded by the initial contour. This causes the singular areas to appear flat in the reconstructed surfaces.

Figures 5a and 5b show two examples of reconstruction for a "volcano" surface, starting from a small curve around the singular area at the top of the mountain, shown as a dotted line. In the equal-height contours picture of the reconstructed surface one can observe the way topological problems like the saddle on the lowest left corner are inherently solved through this "down-going" process. Note however that starting at the base of the volcano would require special treatment to proceed beyond the saddle point; such issues are discussed in [8, 10]. Figures 5c and 5d, demonstrate the reconstruction of part of a sphere surface. Figures 5e and 5f show the topological mistake that occurs when trying to reconstaruct a "Mexican hat"-like object. This topological error may be avoided if one chooses the singular strip surrounding the middle hill as a starting area; see Fig. 5g. In this case there are two initial contours segmenting the singular selected area from the rest of the image; the first is propagating outward and the second inward yielding the desired reconstruction.

The next two examples show the reconstruction of a nose and face from real images. Figure 6a presents a low resolution, low quality image of a nose taken by a CCD camera and the reconstructed surface displayed from three different viewpoints. The first contour was selected to be around the singular area at the tip of the nose. The second example is an attempt to apply the algorithm to a facial image containing texture, background non-Lambertian face, etc. Though the reconstruction is obviously not perfect, it is possible to learn about the 3D shape of the face from the reconstructed surface. Observe the way the eyebrow is reconstructed to be concave instead of convex due to topological errors that cannot be avoided. In our next paper we will address such problems by adding a new kind of smoothness assumption that will help resolve the topology ambiguity.

6. CONCLUDING REMARKS

We have described a method for recovering the shape of a smooth object from its shaded image by an equal-height contour propagation method and gave a new formulation to the general light source direction problem. A numerically stable implementation in which topological problems in the propagated height contours are often inherently avoided was presented. In this method shocks, cusps, and other singularities formed in the contours are also readily dealt with in an efficient numerical scheme originally developed to solve wavefront propagation.

APPENDIX: PLANAR NORMAL VELOCITY CALCULATION

Given the simple propagation rule on the plane defined by the light source direction P^l

$$C_t^l = \hat{\mathbf{t}} \times \hat{l},$$

we want to project this evolving curve to the (x, y) plane P and take only the normal component of the velocity vector of the curve in P; see Fig. 7.

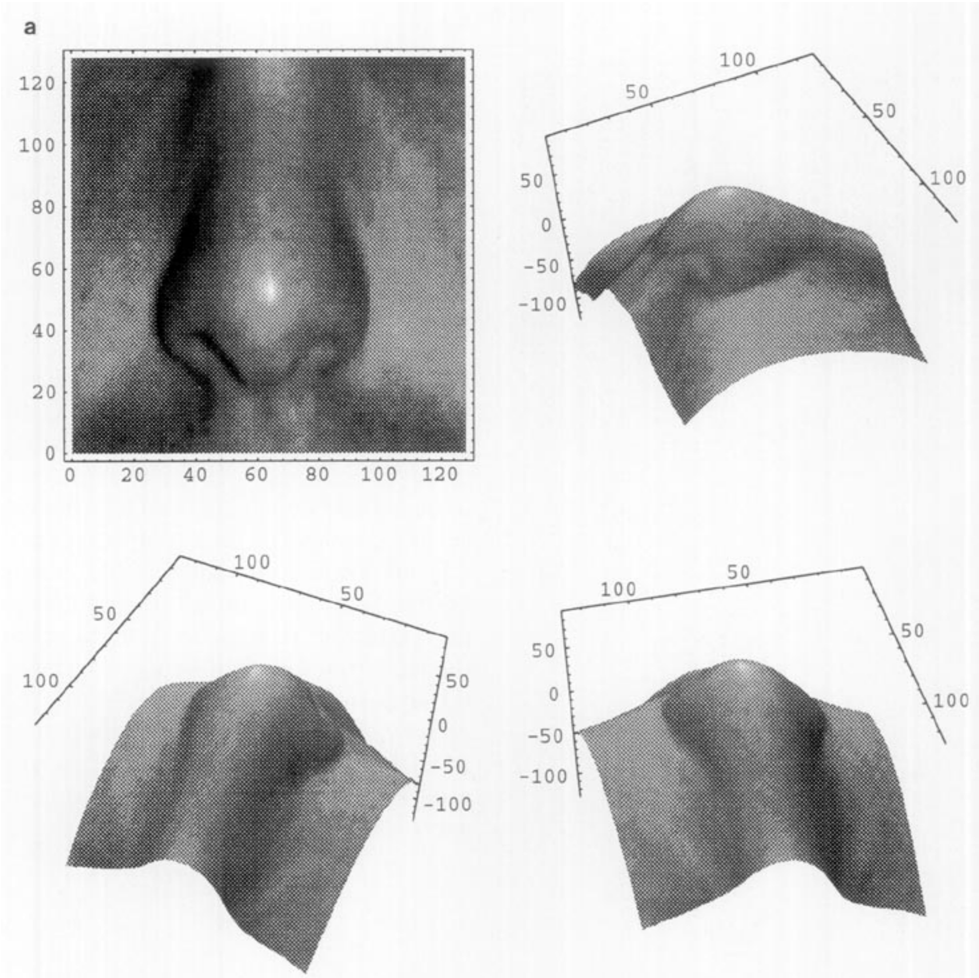

FIG. 6. Reconstaruction of real images given on a 128 × 128 grid. (a) A "real" nose and the reconstructed surface from three different viewpoints. The process starts at the tip of the nose. (b) "Real" face and the reconstructed surface from three different viewpoints. The process starts at the tip of the nose.

The above equation may be written as

$$(x_t, y_t, z_t) = \frac{(x_u, y_u, z_u)}{\sqrt{x_u^2 + y_u^2 + z_u^2}} \times \frac{(-p_l, -q_l, 1)}{\sqrt{1 + p_l^2 + q_l^2}}$$
$$= (w_1, w_2, w_3),$$

where u is the parameter of the curve $C^l(u, t) = (x(u, t), y(u, t), z(u, t))$.

The projection of this curve evolution on the (x, y) plane may be written using $C_t = v_n \hat{n}$, or

$$(x_t, y_t) = v_n \frac{(-y_u, x_u)}{\sqrt{x_u^2 + y_u^2}}.$$

We are looking for the velocity v_n, which is the projection of (w_1, w_2) on the planar normal $\hat{n}$. This may be written as

$$v_n = \langle (w_1, w_2), \hat{n} \rangle = \frac{-w_1 y_u + w_2 x_u}{\sqrt{x_u^2 + y_u^2}}.$$

By calculating the vector product and eliminating (due to the projection) the $\hat{z}$ component, we get

$$(w_1, w_2) = \frac{(y_u + q_l z_u, -x_u - p_l z_u)}{\sqrt{x_u^2 + y_u^2 + z_u^2}\sqrt{1 + p_l^2 + q_l^2}}.$$

Now we are ready to calculate v_n,

$$v_n = \frac{(y_u + q_l z_u, -x_u - p_l z_u)}{\sqrt{1 + p_l^2 + q_l^2}\sqrt{x_u^2 + y_u^2 + z_u^2}} \cdot \frac{(-y_u, x_u)}{\sqrt{x_u^2 + y_u^2}}.$$

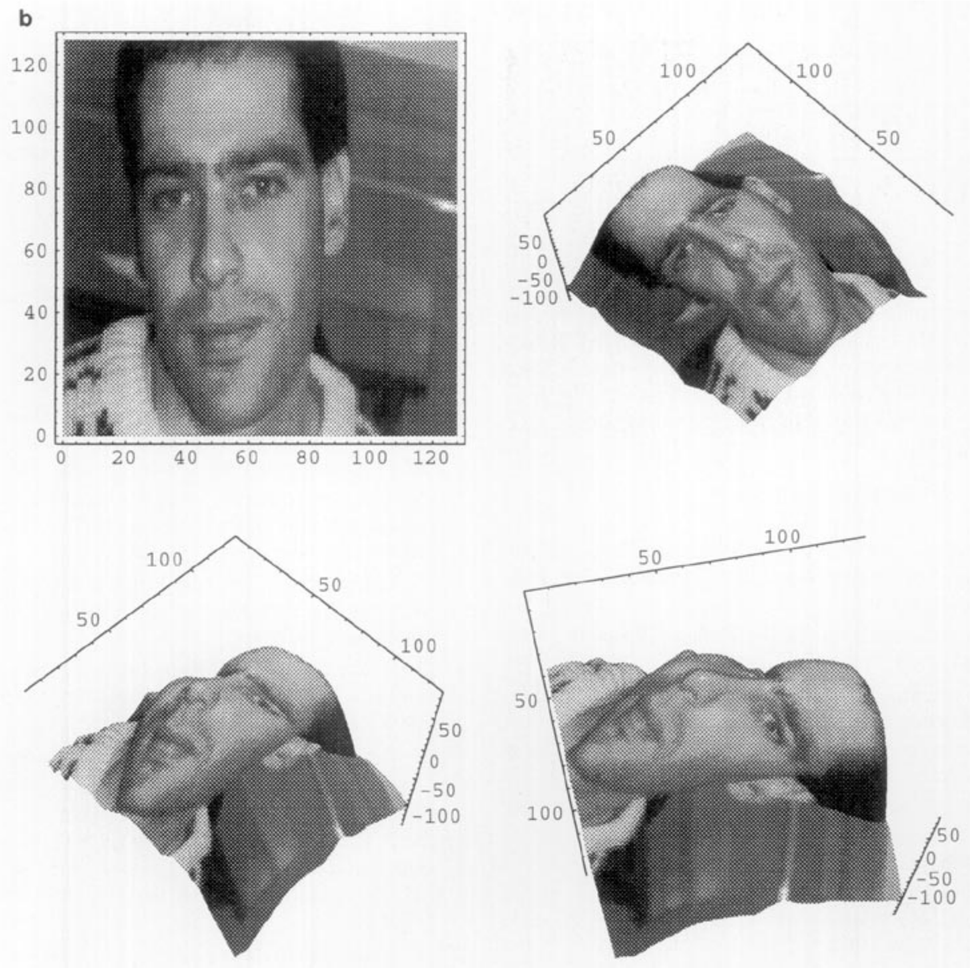

FIG. 6—*Continued*

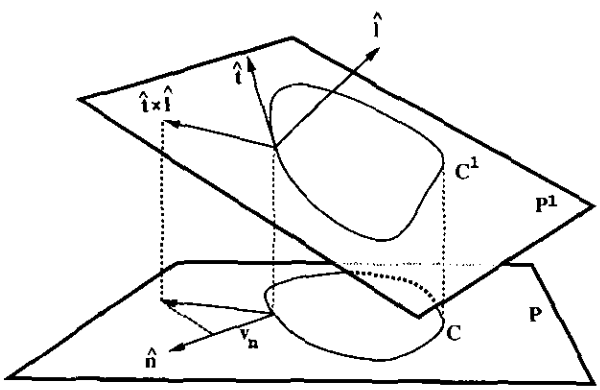

FIG. 7. C^l is evolving with a constant unit velocity along its normals directions, and C is the projection of C^l to P. The propagation rule that determines the evolution of C is specified by a velocity v_n along its normals. v_n is achieved by first projecting the C^l normals to P and then taking only the planar components of that projection.

Using the chain rule $z_u = z_x x_u + z_y y_u = p_l x_u + q_l y_u$, we get

$$v_n = \sqrt{\frac{x_u^2(1 + p_l^2) + y_u^2(1 + q_l^2) + 2p_l q_l x_u y_u}{(1 + p_l^2 + q_l^2)(x_u^2 + y_u^2)}}.$$

Writing the planar normal as its components $\hat{n} = (n_1, n_2) = (-y_u, x_u)/\sqrt{x_u^2 + y_u^2}$, we conclude with

$$v_n = \frac{1}{\sqrt{1 + p_l^2 + q_l^2}} \sqrt{n_1^2(1 + q_l^2) + n_2^2(1 + p_l^2) - n_1 n_2 2 p_l q_l}. \tag{14}$$

We have noted in Section 4 that the planar unit normal to any level set of a given continuous function ϕ can be written as $\hat{n} = -\nabla\phi/\|\nabla\phi\|$. Using this notation in Eq. (14) we obtain

$$v_n = \frac{1}{\sqrt{\phi_x^2 + \phi_y^2}\sqrt{1 + p_l^2 + q_l^2}} \times \sqrt{\phi_x^2(1 + q_l^2) + \phi_y^2(1 + p_l^2) - \phi_x\phi_y 2p_lq_l}.$$

ACKNOWLEDGMENTS

We thank Professors James Sethian, Stanley Osher, Moshe Israeli, and Allen Tannenbaum, Dr. Guillermo Sapiro, and Professor J. Rubinstein for their help in introducing us to the world of numerical analysis and moving surfaces. We also thank the anonymous referees for their thorough and deep review that helped us to improve this paper.

REFERENCES

1. A. M. Bruckstein, On shape from shading: Some theoretical considerations, Electrical Engineering Report 524, Technion-Israel Institute of Technology, June 1985.
2. A. M. Bruckstein, On shape from shading, *Comput. Vision Graphics Image Process.* **44,** 1988, 139–154.
3. P. Dupuis and J. Oliensis, An optimal control formulation and related numerical methods for a problem in shape reconstruction, Department of Computer Science COINS TR93-03, University of Massachusetts at Amherst, 1993.
4. C. L. Epstein and M. Gage, The curve shortening flow, in *Wave Motion: Theory, Modeling, and Computation* (A. Chorin and A. Majda, Eds.), Springer-Verlag, New York, 1987.
5. B. K. P. Horn, Obtaining shape from shading information, In *The Psychology of Computer Vision* (P. H. Winston, Ed.), pp. 115–155, McGraw–Hill, New York, 1987.
6. B. K. P. Horn, Height and gradient from shading, *Int. J. Comput. Vision* **5,** 1990, 37–75.
7. B. K. P. Horn and M. J. Brooks (Eds.), *Shape from Shading,* MIT Press, Cambridge, MA, 1989.
8. R. Kimmel, Shape from shading via level sets, M.Sc. thesis. Technion-Israel Institute of Technology, June 1992. [In Hebrew].
9. R. Kimmel, A. Amir, and A. M. Bruckstein, Finding shortest paths on graph surfaces, Center for Intelligent Systems Report CIS 9301, Technion-Israel Institute of Technology, Israel, January 1993.
10. R. Kimmel and A. M. Bruckstein, Shape from shading via level sets, Center for Intelligent Systems Report CIS 9209, Technion-Israel Institute of Technology, Israel, June 1992.
11. R. Kimmel, K. Siddiqi, B. B. Kimia, and A. M. Bruckstein, Shape from shading: Level set propagation and viscosity solutions, *Int. J. Comput. Vision,* in press.
12. C. H. Lee and A. Rosenfeld, Improved methods of estimating shape from shading using the light source coordinate system, *Artif. Intell.* **26**(1), 1985, 125–143.
13. R. J. LeVeque, *Numerical Methods for Conservation Laws,* Lectures in Mathematics, Birkhauser-Verlag, Basel, 1992.
14. J. Oliensis and P. Dupuis, Direct method for reconstructing shape from shading, in *Proceedings SPIE Conference 1570 on Geometric Methods in Computer Vision, 1991,* pp. 116–128.
15. S. Osher and L. Rudin, Rapid convergence of approximate solutions to shape-from-shading problem, Technical report, Cognitech Inc., 1993, unpublished.
16. S. Osher and C. W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, *SIAM J. Numer. Anal.* **28**(4), 1991, 907–922.
17. S. J. Osher, A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations, *SIAM J. Math. Anal.* **24**(5), 1993, 1145–1152.
18. S. J. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* **79,** 1988, 12–49.
19. A. P. Pentland, Local shading analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(2), 1984, 170–187. Also in *Shape from Shading* (B. K. P. Horn and M. J. Brooks), MIT Press, Cambridge, MA, 1989.
20. E. Rouy and A. Tourin, A viscosity solutions approach to shape-from-shading, *SIAM J. Numer. Anal.* **29**(3), 1992, 867–884.
21. J. A. Sethian, A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed, *J. Different. Geom.* **33,** 1989, 131–161.
22. J. A. Sethian and J. Strain, Crystal growth and dendritic solidification, *J. Comput. Phys.* **98,** 1992, 231–253.