Shape from Shading: Level Set Propagation and Viscosity Solutions

RON KIMMEL* Electrical Engineering Dept. Technion, Haifa 32000, Israel

KALEEM SIDDIQI AND BENJAMIN B. KIMIA[†] Laboratory for Man/Machine Systems, Brown University, USA

ALFRED M. BRUCKSTEIN[‡] Computer Science Dept. Technion, Haifa 32000, Israel

Received 1992

Abstract. We present a new implementation of an algorithm aimed at recovering a 3D shape from its 2D gray-level picture. In order to reconstruct the shape of the object, an almost arbitrarily initialized 3D function is propagated on a rectangular grid, so that a level set of this function tracks the height contours of the shape. The method imports techniques from differential geometry, fluid dynamics, and numerical analysis and provides an accurate shape from shading algorithm. The method solves some topological problems and gracefully handles cases of non-smooth surfaces that give rise to shocks in the propagating contours. Real and synthetic images of 3D profiles were submitted to the algorithm and the reconstructed surfaces are presented, demonstrating the effectiveness of the proposed method.

1 Introduction

Computer vision researchers have invented several methods of reconstructing three dimensional objects from their two dimensional images, see e.g. (Brooks and Horn 1989; Bruckstein 1988; Frankot and Chellappa 1988; Horn 1975, 1990; Horn and Brooks 1986, 1989; Ikeuchi and Horn 1981; Pentland 1984, 1990). Other interesting and fruitful research concerns the numerical approximation of wave-front propagation in fluid dynamics, crystal growth, etc. Some nice results in the numerical approximation of front propagation were recently published by Osher and Sethian (1988). In this paper we modify a numerical method of front propagation suggested in this context, to build a scheme for reconstructing a shape from its shaded image. The resulting algorithm combines equal-height contour propagation as suggested by Bruckstein (1988) with well behaved (i.e. stable and locally accurate) numerical methods proposed by Osher and Sethian for

*R. Kimmel's work was partly supported by the Ollendorf Fund. [†]B.B. Kimia's work was supported by the Sun Microsystems Academic Equipment Grant. wave-front evolution. We show that some topological problems are readily solved in this formulation of the problem. Furthermore smoothness of the surface is not a necessary condition, in the sense that object corners that would lead to shocks in propagating contours are gracefully dealt with by the numerical scheme.

2 Problem Formulation and the Reflectance Map

Suppose we are given a continuous function of two variables, z(x, y) describing the surface of an object. The shaded image of that surface is defined as a brightness distribution E(x, y), the brightness values depending on properties of the surface, its orientation at (x, y) and on illumination. The brightness E(x, y) is determined via a so-called shading rule or *reflectance map*, characterizing the surface properties and providing 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 surface orientation using the components of the surface gradient $p = \partial z/\partial x$ and $q = \partial z/\partial y$. The surface normal at each point is perpendicular to the plane determined by the vectors $(1, 0, p)^T$

siddiqi@lems.brown.edu, kimia@lems.brown.edu

freddy@cs.technion.ac.il

ron@tx.technion.ac.il

[‡]A.M. Bruckstein's work was supported in part by the Fund for the Promotion of Research at the Technion.

and $(0, 1, q)^T$, therefore its direction is their vector product

$$N = \begin{vmatrix} \hat{x} & \hat{y} & \hat{z} \\ 1 & 0 & p \\ 0 & 1 & q \end{vmatrix} = (-p, -q, 1)^T$$

hence the unit normal can then be written as

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

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).

For simplicity let us first assume that the illumination is uniform and that it illuminates the surface from above, i.e., from $z = +\infty$, or $\hat{l} = (0, 0, 1)^T$. In the general case we define the light source direction as

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

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

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

This is the image irradiance equation.

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

where ρ and λ are proportionality factors that can be neglected ($\rho\lambda = 1$) by rescaling the image irradiance. (ρ known as the *albedo*, is the ratio of the total reflected light flux to the total incident light flux, is here assumed to be constant, and λ is the strength of the illumination.) In this case the normal direction lies on an ambiguity cone whose main axis is directed towards the light source (see Fig. 1b).

For the simple case where $\hat{l} = (0, 0, 1)^T$ 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. 1a).

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 boundary conditions are necessary.

Given the image E(x, y) it is, in general, impossible to unambiguously recover the height profile z(x, y). As an immediate example of the ambiguity simply consider the function -z(x, y), which, under a Lambertian shading rule, maps into the same image as z(x, y). Some further information on the function z(x, y) is



Fig. 1. a. When the light source is from above $(\hat{l} = \hat{Z})$ the ambiguity cone is directed upwards. b. For other light source directions the ambiguity cone is tilted towards the light source creating many possible projections of the surface normal on the (x, y)-plane.



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 $E = \cos \alpha = \hat{N} \cdot \hat{l}$.

therefore needed. This is usually given as a smoothness constraint on the surface (e.g., C^1 or C^k continuity), and exact or approximate values of z(x, y) together with the corresponding surface orientation, at either a discrete set of points $\{(x_i, y_i)\}$, or on a continuous curve on the (x, y)-plane (boundary conditions). The given boundary conditions and smoothness assumptions are not always enough to resolve ambiguities, and it is in fact difficult to determine, in general situations, sufficient conditions for a unique solution surface.

In summary, our task is to reconstruct a function z(x, y) by 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 irradiance Eq. (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 local constraints that follow reasonable assumptions concerning the relation of each point on the surface to its surrounding area. With the additional constraints the shape reconstruction should proceed with no difficulty.

3 Historical Review of Shape from Shading Schemes

Shape from shading schemes can be roughly divided into two main groups: the iterative (global) methods and the non-iterative (local) methods. In both cases assumptions are made about the surface. These assumptions relate points on the surface to their surrounding neighborhood, and are used for producing a second constraint at each point.

3.1 Iterative Methods

We first briefly review the development of the variational approaches. In the sequel, the minimization problems from which the numerical schemes are devised, are presented and the main advantages and drawbacks of each scheme are discussed. Iterative numerical schemes for solving these minimization problems are discussed by Horn (1990). These methods were developed over the last decade (Ikeuchi and Horn 1981; Brooks and Horn 1985; Horn and Brooks 1986; Horn 1990). The basic idea behind the iterative schemes produced by a variational approach is the search for surfaces z(x, y) that minimize the brightness error

$$B \equiv E(x, y) - R(p, q)$$

on the picture. Direct minimization of

$$\iint B^2 \, dx \, dy$$

is meaningless since it yields infinite choices for $\{p(x, y), q(x, y)\}$. To come up with a viable numerical iterative scheme, conditions are needed to select one from the infinite number of solutions. One can define a "departure from smoothness measure"

$$S \equiv p_x^2 + p_y^2 + q_x^2 + q_y^2$$

and a local "integrability deficiency"

$$I_1 \equiv (z_x - p)^2 + (z_y - q)^2$$

or

$$I_2 \equiv (p_y - q_x)^2$$

These definitions can be used to quantify the additional assumptions used to produce meaningful minimization processes.

A direct formulation of the shape from shading problem, with a smoothness condition on the surface, is the following minimization problem (see Courant and Hilbert 1953; 1962)

$$\iint (S + \lambda B) \, dx \, dy \to \min$$

This problem provides the Euler equations

$$\begin{cases} \nabla^2 p + \lambda R_p = 0\\ \nabla^2 q + \lambda R_q = 0 \end{cases}$$

and by eliminating the Lagrange multiplier λ one has to solve

$$\begin{cases} R_q \nabla^2 p = R_p \nabla^2 q \\ E(x, y) = R(p, q) \end{cases}$$

but, unfortunately no convergent iterative scheme has been found (see Ikeuchi and Horn 1981). Brooks (1985) proposed a regularization of the original problem by requiring

$$\iint (B^2 + \lambda S) \, dx \, dy \to \min$$

This optimization problem does yield an iterative scheme, but the "true" surfaces solving the original shape from shading problem are not necessarily fixed points of this scheme. The algorithm may even "walk away" from the correct solution because it prefers to minimize the smoothness error (the constraint in the regularized formulation) while compromising on a small error in the brightness error functional. Recovering the height from the gradient which is obtained by the above scheme can readily be done by integration. This raises the question of integrability. Minimizing the functional

$$\iint I_1 \, dx \, dy \to \min$$

results in the Poison equation $\nabla^2 z = p_x + q_y$, and yields an iterative scheme for updating z on the grid.

Another way of dealing with integrability is by considering

$$\iint \left(B^2 + \lambda I_2\right) dx \, dy \to \min$$

see (Horn and Brooks 1986), which is an ordinary calculus problem. The resulting scheme avoids the excessive smoothing, but was found to be less stable. In (Horn 1990), Horn proposes to consider

$$\iint \left(B^2 + \mu I_1\right) dx \, dy \to \min$$

and

$$\iint I_1\,dx\,dy\to\min$$

to get a scheme which does not "walk away" from the solution. Improvement of the schemes' performance is achieved by using a local linearization of the reflectance map (Horn 1990). When (p, q) are "close enough" to (p_0, q_0) the reflectance map linear approximation is given by

$$R(p,q) = R(p_0,q_0) + (p - p_0)R_p(p_0,q_0) + (q - q_0)R_q(p_0,q_0) + \cdots \approx R(p_0,q_0) + (p - p_0)R_p(p_0,q_0) + (q - q_0)R_q(p_0,q_0)$$

Incorporating a departure from a smoothness penalty term provides the following minimization problem

$$\iint (B^2 + \lambda S + \mu I_1) \, dx \, dy \to \min$$

The calculus of variations solution yields Euler equations that can be approximated to generate an iterative scheme. The resulting scheme contains a "departure from smoothness" penalty term which diminished when the generated solution seems to be close to the real one and therefore stability is achieved while preventing "walking away" from the true solution.

Frankot and Chellappa (1988) used projection on a set of integrable functions in order to force the integrability condition at each phase of the iterative process. Ascher and Carter (1993) used a multigrid method to speed up the iterative reconstruction process.

In all the above iterative schemes there is no guarantee of convergence to the proper solution and the schemes can get stuck in local minima. There is also a trade-off between stability (the smoothing condition) and "walking away" from the solution when given as an initial condition to the scheme.

3.2 Non-Iterative Methods

As mentioned earlier the problem of shape from shading is equivalent to finding the values of two variables representing the surface normal direction \hat{N} at each point of the shading picture. We have only one equation at each point, $E(x, y) = R(\hat{N})$. In order to find the exact surface normal direction, another equation is needed. Pentland, in (1984), adds the assumption that each point lies on a sphere $z(x, y) = \sqrt{R^2 - x^2 - y^2}$. Motivated by the finding that human eye seems to be able to sense brightness variations up to the second derivative, he then shows that using the six values $E, E_x, E_y, E_{xx}, E_{yy}$ and E_{xy} the normal direction can be found, together with the local curvature $\frac{1}{R}$, the light source direction \hat{l} and the brightness factor $(\lambda\rho)$ typical of Lambertian illumination $E = \lambda\rho(\hat{N}\cdot\hat{l})$. The assumption that each point lies on a sphere is of course usually not true. For example if $E_{xx} \cdot E_{yy} < 0$ the point may be a saddle point. Pentland therefore consider five types of local surfaces: concave sphere, convex sphere, plane, cylinder and a saddle. Under these assumptions one can even try to estimate the illumination direction (\hat{l}) using the distribution of brightness derivatives as a function of the image direction.

In Pentland (1990) deals with the problem of the linear reflectance map $R(p,q) = k_1 + k_2p + k_3q$, and shows that using the radial Fourier transform of the image, the surface can be reconstructed. Standard filters like Weiner filtering are used to remove noise and non linear components. Approximating the reflectance map linearly is valid only for surfaces with slants of small degrees, in the Lambertian case. In other cases, the linear assumption does not hold anymore.

3.3 The Characteristic Strip Expansion Method

The image irradiance equation describing the image is a non-linear equation, to be more specific, it is a first order nonlinear partial differential equation. The characteristic strip expansion method is a general procedure of solving Cauchy type boundary value problems associated to nonlinear partial differential equations. Assume that the surface is smooth and that second derivatives exist everywhere. Suppose we know the height $z(x_0, y_0)$ and the orientation $\{p(x_0, y_0), q(x_0, y_0)\}$ at a given point. Then the height profile and the surface orientation can be determined along a well-defined curve in the (x, y)-plane known as a "characteristic strip" (see Horn 1975; John 1982). The height and orientation at the given point (x_0, y_0) being known, we wish to extend the solution by stepping a small step δx in the x direction and δy in the y direction. Then the change in height is given by

$$\delta z = p \delta x + q \delta y$$

While exploring the surface, we need to keep track of p and q as well as x, y and z. The changes in p and q are given by

$$\begin{cases} \delta p = z_{xx} \delta x + z_{xy} \delta y \\ \delta q = z_{yx} \delta x + z_{yy} \delta y \end{cases}$$

Note that according to the smoothness assumption $z_{xy} = z_{yx}$. It seems that we need to keep track of the

second partial derivatives of the height function which depend on the third partial derivatives, then the third derivatives updates depend on the forth derivatives and so on. This infinite chain is broken when the image irradiance equation (E(x, y) = R(p, q)) is added to the game. Differentiating the image irradiance equation with respect to x and y, leads to

$$\begin{cases} E_x = z_{xx}R_p + z_{xy}R_q\\ E_y = z_{yx}R_p + z_{yy}R_q \end{cases}$$

We are free to choose any $(\delta x, \delta y)$ we want, and choosing

$$\begin{cases} \delta x = R_p \delta s\\ \delta y = R_q \delta s \end{cases}$$
(3)

yields

$$\begin{cases} \delta p = E_x \delta s \\ \delta q = E_y \delta s \end{cases}$$

Hence the following set of five ordinary differential equations

$$\begin{cases} dx = R_p ds \\ dy = R_q ds \\ dz = (pR_p + qR_q) ds \\ dp = E_x ds \\ dq = E_y ds \end{cases}$$
(4)

trace a curve on the surface z(x, y), s being the parameter determining the flow along the curve known as a "characteristic strip". This result is the basis of Horn's classical shape from shading method (Horn 1975). He proposed to look at the brightness map E(x, y) and start height recovery around singular points; there the image E(x, y) attains the maximum value of 1, i.e., where p = q = 0 (under the Lambertian shading rule). From the neighborhood of these points, one can propagate characteristics outward, simultaneously, and use certain neighborhood rules in the propagation such as not allowing crossovers of adjacent strips and interpolating new characteristic strips when neighboring strips separate too far. Note that when p = q = 0 we have a start-up problem for the algorithm, since (3) will not pull the strips away from the singular points. Hence we must add further assumptions about the behavior of z(x, y) about each starting point (i.e., to classify singular points as a local maxima or minima; of course, problems arise at saddle points). Implicitly we have to assume the knowledge of the initial slopes (p, q) on a small loop around the singularities.

Direct numerical implementations of characteristic strip expansions are very sensitive to brightness errors. These affect the direction of the growing characteristics and can therefore lead to theoretically impossible crossover of characteristics, (the gradient at a crossing point would have two directions!). Horn suggested propagation control algorithms, like preserving the relation between two neighboring characteristics by enforcing the brightness condition E(x, y) = R(p, q), and $z_t = px_t + qy_t$ along curves defined by fronts of developing characteristics.

Let us try to characterize the accumulation of error in a simple discrete numerical model of the characteristic strip expansion process, under a Lambertian shading rule. The numerical scheme which approximates an analytic model should be as stable and as accurate as possible. We have observed that the characteristic strips expansion method is very sensitive and control algorithms must be used in order to supervise the whole process (preventing crossovers etc.). The accuracy of the results can be assessed through the following simple scheme, representing the behavior on a characteristic.

Taking a forward approximation of the "time" derivatives, (time is s in this case) we get the following discrete numerical approximation scheme of Eq. (4):

$$\begin{cases} x^{n+1} = x^n + R_p^n \Delta s \\ y^{n+1} = y^n + R_q^n \Delta s \\ z^{n+1} = z^n + (p^n R_p^n + q^n R_q^n) \Delta s \\ p^{n+1} = p^n + E_x^n \Delta s \\ q^{n+1} = q^n + E_y^n \Delta s \end{cases}$$
(5)

Here *n* stands for the parameter at time $s = n\Delta s$. Assume that E_x^n and E_y^n are the exact derivatives, and that the direction between point $P^n \equiv (x^n, y^n)$ and the next point in the generated solution $P^{n+1} \equiv (x^{n+1}, y^{n+1})$ is fixed. (We make these assumptions in order to prevent other errors, such as those caused by the interpolation of the brightness values between the given pixels and those caused by a change in the characteristic direction, from affecting the error analysis given here.)

The line connecting P^n and P^{n+1} can be described parametrically as

$$L^{n}(t) = (P^{n+1} - P^{n})t + P^{n}$$
 where $t \in [0, 1]$

Define the height error $Z_{\text{err}}^{n+1} \equiv \hat{z}^{n+1} - z^{n+1}$, where \hat{z} is the approximation and z is the exact change of the height corresponding to the change of the brightness

between P^n and P^{n+1} .

$$Z_{err}^{n+1} = \hat{z}^{n+1} - \left\{ z^n + \Delta s \int_{L^n(t)} (pR_p + qR_q) dt \right\}$$

= $z^n + (p^n R_p^n + q^n R_q^n) \Delta s - z^n$
 $- \Delta s \int_{L^n(t)} (p(t)R_p(t) + q(t)R_q(t)) dt$
= $\Delta s \left\{ (p^n R_p^n + q^n R_q^n) - \int_{L^n(t)} (p(t)R_p(t) + q(t)R_q(t)) dt \right\}$

In the simple case where $R(p,q) = \frac{1}{\sqrt{1+p^2+q^2}}$, the height error is

$$Z_{\rm err}^{n+1} = \Delta s \left\{ (E^n)^3 - E^n - \int_{L^n(t)} (E^3(t) - E(t)) \, dt \right\}$$

Summing the errors along the characteristic gives

$$Z_{\text{Total Err}}^{n} \equiv \sum_{i=0}^{n-1} Z_{\text{err}}^{i}$$
$$= \Delta s \sum_{i=0}^{n-1} \left\{ (E^{i})^{3} - E^{i} - \int_{L^{i}(t)} (E^{3}(t) - E(t)) dt \right\}$$

It seems that as $\Delta s \rightarrow 0$ the total error $Z_{\text{Total Err}}^n \rightarrow 0$, but in the above we have neglected the errors caused by the approximation of E, E_x and E_y between the grid points. Taking this into consideration and letting $\Delta s \rightarrow 0$ would cause even bigger errors in the growing characteristics.

A general analysis of accuracy and stability for these kind of models is quite difficult to perform. Experiments show that crossovers of characteristics are common, indicating the inherent instability of such methods. Control algorithms on the growing curves can be found in (Horn 1975), however new sources of error are introduced when interpolating new characteristics.

3.4 Shape from Shading via Equal-Height Contours

Let us first summarize the main results from (Bruckstein 1988): 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

$$px_s + qy_s = 0$$

This reflects that there is no change in height along the equal height contour. 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. 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 basic geometry we have

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

where α is the surface orientation angle. Under the simple Lambertian shading rule where $R(p, q) = \cos \alpha = \frac{1}{\sqrt{1+p^2+q^2}}$, 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 *P.D.E.* problem. Given $\{x(s, 0), y(s, 0)\}$ the evolution equations are

$$\begin{cases} x_t(s,t) = F(x,y) \frac{y_s}{\sqrt{x_s^2 + y_s^2}} \\ y_t(s,t) = F(x,y) \frac{-x_s}{\sqrt{x_s^2 + y_s^2}} \end{cases}$$
(6)

where $t \equiv z$ and

$$F(x(s,t), y(s,t)) = \frac{E(x(s,t), y(s,t))}{\sqrt{1 - E^2(x(s,t), y(s,t))}}$$
(7)

Define $C(0) \equiv X(s, 0) = \{x(s, 0), y(s, 0)\}$ as the smooth (and, in some cases, closed) initial curve, and C(t) = X(s, t) as the one-parameter family of curves generated by moving C(0) along its normal vector field with speed F. Here, F is a given scalar function of the brightness E. Using this notation, Eq. (6) can be written as a planar curve evolution equation $\frac{\partial}{\partial t}C(t) = C(t)$

 $F(x, y) \cdot \hat{n}$. Sethian (1989) called such propagation models a "Lagrangian" evolution equations because the physical coordinate system moves with the propagating front.

In the sequel we consider the accuracy and stability problems of a simple numerical approximation of Eq. (6). Following an analysis of the problem as was done by Sethian (1989), a simple difference approximation is considered. Divide the parametrization interval [0, S] into M equal intervals of size Δs and t into equal intervals of length Δt . Define a marker point as $P_i^n \equiv \{x_i^n, y_i^n\} = \{x(i\Delta s, n\Delta t), y(i\Delta s, n\Delta t)\}$. A numerical algorithm should produce new values P_i^{n+1} from previous positions. Approximate parameter derivatives using a central difference approximation and time derivatives using a forward difference approximation as follows

$$\frac{d(P_i^n)}{ds} \approx \frac{P_{i+1}^n - P_{i-1}^n}{2\Delta s}$$
$$\frac{d(P_i^n)}{dt} \approx \frac{P_i^{n+1} - P_i^n}{\Delta t}$$

Substitution of these approximations into Eq. (6) gives

$$P_{i}^{n+1} = P_{i}^{n} + \Delta t F(x_{i}^{n}, y_{i}^{n}) \\ \times \frac{\{y_{i+1}^{n} - y_{i-1}^{n}, -(x_{i+1}^{n} - x_{i-1}^{n})\}}{\sqrt{(y_{i+1}^{n} - y_{i-1}^{n})^{2} + (x_{i+1}^{n} - x_{i-1}^{n})^{2}}}$$
(8)

The discretization interval Δs has been eliminated. Consequently, as the P's come together, quotients in the right hand side of (8) approach zero over zero, a very sensitive calculation. The lack of stability which characterizes such numerical algorithms results from a feedback cycle, where small errors in the approximation of the new P's, cause local variations in the derivatives, producing errors in the direction of propagation of each P, in turn yielding errors in the approximation of the new P's. Therefore after few iterations small variations will grow, and the solution can even become unbounded. There are some ways to control the stability of such algorithms. For example, it is possible to reparametrize the wave front at every iteration step. and to redistribute the P's according to arclength. The reparametrization adds a smoothing term to the speed function, and is difficult to analyze. Control algorithms are also 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

114 Kimmel et al.

to handle this merging process by an external control procedure.

The reason that Lagrangian formulation suffers from these stability and topological problems is due to the fact that it follows a local representation of the propagating front.

In computer vision problems we are usually working on pictures which are samples (pixels) of the real information on a given grid. The values between the grid points have to be estimated (by interpolation) when working with such schemes.

In order to approximate the height error caused by the change in brightness between two marker points on successive contours, we first define

$$P_i^{n+1} \equiv P_i^n + \left\{ \Delta x_i^n, \, \Delta y_i^n \right\}$$

Using this definition and the approximation (8) we have

$$\{\Delta x^n, \Delta y^n\} = \Delta t F(x^n, y^n) \frac{\{y_s, -x_s\}}{\sqrt{x_s^2 + y_s^2}}$$

Assume that there is no change in direction of propagation between P^n and P^{n+1} . The distance between two marker points is given by

$$\Delta r^n = \sqrt{(\Delta x^n)^2 + (\Delta y^n)^2} = \Delta t F(x^n, y^n)$$

Integration of the change in height along the straight line L^n between P^n and P^{n+1} gives the exact change in height neglected by the approximation. Define

$$dr^{n} = \sqrt{(dx^{n})^{2} + (dy^{n})^{2}}$$
$$= \sqrt{(d\tau\Delta x^{n})^{2} + (d\tau\Delta y^{n})^{2}} = d\tau\Delta r^{n}$$

then along $L^{n}(\tau)$ the difference in the height is given by

$$dt = \frac{dr^n}{F(x, y)} = \frac{\Delta r^n}{F(x, y)} d\tau \qquad \text{for every } \tau \in [0, 1]$$

Define $F^n \equiv F(x^n, y^n)$ and $F^n(\tau) \equiv F(x^n + \Delta x^n \tau, y^n + \Delta y^n \tau)$. The height error can then be written as

$$Z_{\text{err}}^{n+1} \equiv \hat{z}^{n+1} - z^{n+1}$$
$$= \Delta t - \int_{P^n}^{P^{n+1}} dt$$
$$= \Delta t - \int_0^1 \frac{\Delta r^n}{F^n(\tau)} d\tau$$
$$= \Delta t \left(1 - F^n \int_0^1 \frac{1}{F^n(\tau)} d\tau \right)$$

or as a function of the image brightness

$$Z_{\rm err}^{n+1} = \Delta t \left(1 - \frac{E^n}{\sqrt{1 - (E^n)^2}} \int_0^1 \frac{\sqrt{1 - (E^n(\tau))^2}}{E^n(\tau)} \, d\tau \right)$$

The total error according to the above assumptions is given by

$$Z_{\text{Total Err}}^{n} \equiv \sum_{i=0}^{n-1} Z_{\text{err}}^{i}$$
$$= \Delta t \sum_{i=0}^{n-1} \left\{ 1 - \frac{E^{i}}{\sqrt{1 - (E^{i})^{2}}} \right.$$
$$\times \int_{L^{i}(\tau)} \frac{\sqrt{1 - (E^{i}(\tau))^{2}}}{E^{i}(\tau)} d\tau \right\}$$

As for the characteristic strips expansion, it seems that when $\Delta t \rightarrow 0$ the height error $Z_{\text{err}}^n \rightarrow 0$. But because of the same arguments, when $\Delta t \rightarrow 0$, the errors caused by the estimation of *E* between the grid points badly affect the result.

4 A New Shape from Shading Algorithm based on Level-Sets

We here propose to recover a shape from its shaded image via a very ingenious algorithm that was invented in fluid dynamics for solving evolution equations of the type (6). This algorithm translates the curve evolution into a 3D-surface evolution, so that curves changing according to (6) are zero (or level) sets of evolving surface (see Kimmel and Bruckstein 1992; Kimmel 1992). As the 3D surface evolves it inherently handles curve shocks by implementing a physically motivated "entropy condition" together with the Huygens principle of the front propagation. The algorithm that produces the desired results works on an image defined on a grid and is based on a recently discovered efficient numerical implementation of surface evolution equations.

4.1 Huygens Principle and the Entropy Condition

According to the Huygens principle (Sethian 1985), the solution of the curve propagation according to $\frac{\partial}{\partial t}\mathbf{X}(s,t) = F\hat{n}$ given $\mathbf{X}(s,0)$ at time dt, $\mathbf{X}(s,dt)$, corresponds to the envelope generated by the set of all disks of radii Fdt centered on the initial curve $\mathbf{X}_0(s)$. Problems occur in the curve evolution when the characteristics (i.e. the normals of the fronts) of the propagating curve collide or cross and hence the curvature becomes singular. In order to obtain the solution according to Huygens' principle after a singularity has developed, an "entropy condition" should be enforced on the propagating curve. One can regard the curve as the wavefront of a propagating prairie fire separating two areas—the shape interior which is not burnt yet and the burnt exterior area. The flame propagates in the direction of the curve normals (the ignition curves). If two ignition curves collide at some time t^* neither one should have any effect on the propagating curve at t > t^* . The principle: "What was burnt until t can not burn beyond t" (Sethian 1985), is the natural "entropy condition" of this and many other curve evolution processes.

So far we have seen that the direct approach of propagating the curve according to the "Lagrangian" formulation is both numerically unstable and suffers from topological problems (see Osher and Sethian 1988; Sethian 1989).

To avoid the various problems that occur in this approach, like the need for reparametrization in order to keep numerical stability and to solve topological problems of self intersections by an external control procedure, the "Eulerian formulation," described below, was developed.

Another algorithm which approximates the Lagrangian evolution, solves topological problems and obeys the entropy condition is a volume of fluid type of algorithm presented by Chorin (1980, 1985). In this technique the algorithm tracks the motion of the interior region instead of the boundary of the propagating front. The interior is discretized by employing a grid on the domain and assigning each cell a "volume fraction" corresponding to the amount of interior "fluid" currently located at that cell. Considering the gray-level of each picture cell as the amount of its initial volume fraction may serve as a shape from shading version of this method. Unfortunately, such representation of the boundary causes some difficulties in calculating the normal direction which leads to inaccuracy in the solution.

4.2 Solution via the Eulerian Formulation

The Eulerian scheme is a recursive procedure which propagates the height contour while inherently implementing the entropy condition. Introduce a function $\phi(x, y, t)$ initialized so that $\phi(x, y, 0) = 0$ yields the curve $\mathbf{X}(s, 0)$. Assume that $\mathbf{X}(s, 0)$ 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 $\mathbf{X}(s, t)$ as if propagated by (6) and obeying the entropy condition. If $\phi(x, y, t) = 0$ along $\mathbf{X}(s, t)$ then by the chain rule we have

$$\frac{\partial}{\partial t}\phi(x, y, t) + \frac{\partial}{\partial x}\phi(x(s, t), y(s, t), t) \cdot x_t + \frac{\partial}{\partial y}\phi(x(s, t), y(s, t), t) \cdot y_t = 0$$

or

$$\phi_t + \nabla \phi \cdot \tilde{\mathbf{X}}_t(s, t) = 0 \tag{9}$$

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

$$v = \vec{\mathbf{X}}_t(s, t) \cdot \hat{n}(s, t)$$
(10)

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

$$v = \vec{\mathbf{X}}_t \cdot \hat{n} = \vec{\mathbf{X}}_t \cdot \frac{-\nabla \phi}{\|\nabla \phi\|} = F.$$
(11)

Substituting this into (9) yields

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

Sethian called this approach Eulerian, since the coordinates here are the natural physical coordinates (x, y). Therefore, if we have a surface ϕ propagating according to (12) with the level set $\phi(x, y, 0) = 0$ coinciding with $\mathbf{X}(s, 0)$, then $\phi(x, y, t) = 0$ will produce $\mathbf{X}(s, t)$ propagated according to (6) and solving the topological problems due to shocks. In order to drive a numerical scheme for the surface propagation equation which obeys the "entropy condition" we follow (Osher and Sethian 1988), and show the connection to Hamilton Jacobi methods, weak solutions and conservation laws.

Consider the one dimensional equation

$$\phi_t - \|\nabla \phi\| = \phi(x, t)_t - \sqrt{\phi_x^2} = 0$$
 (13)

If we define $u \equiv \phi_x$, and $H[u] = -\sqrt{u^2}$, differentiation of the above with respect to x will result in a so called Hamilton Jacobi equation, in a conservation law form

$$u_t + [H[u]]_x = 0 (14)$$

The weak solution of the above equation is defined as u(x, t) that satisfies

$$\frac{d}{dt} \int_{a}^{b} u(x,t) \, dx = H[u(a,t)] - H[u(b,t)] \quad (15)$$

To devise a numerical scheme define $u_i^n = u(i\Delta x, n\Delta t)$. A differential scheme of three points is said to be in conservation form if there is a "flow" function $g(u_1, u_2)$ so that

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{g(u_i^n, u_{i+1}^n) - g(u_{i-1}^n, u_i^n)}{\Delta x}$$
(16)

where g(u, u) = H(u) is the consistency condition. A scheme is said to be monotone if $u_i^{n+1} = T(u_{i-1}^n, u_i^n, u_{i+1}^n)$ is an increasing monotone function in its three variables. It is a basic result in numerical analysis that a scheme which is monotone and can be represented in a conservation form automatically obeys the entropy condition (Sod 1985).

Some schemes based on this idea, like the Lax-Friedrichs and Godonov's schemes, are presented Osher and Sethian (1988). The simplest flow function that can be used for implementation is the so-called HJflow, where for $H(u) = f(u^2)$ the numerical flow can be given by using in (16) the function

$$g_{HJ}(u_i^n, u_{i+1}^n) = f((\min(u_i^n, 0))^2 + ((\max(u_{i+1}^n, 0))^2)$$
(17)

and the appropriate (weak) entropy solution of ϕ can be written, by integrating Eq. (16) with respect to x, as

$$\phi_i^{n+1} = \phi_i^n - \Delta t \cdot g\left(D_-\phi_i^n, D_+\phi_i^n\right)$$
(18)

where $D_-\phi_i^n = \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x}$ and $D_+\phi_i^n = \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x}$. This is a so called first order scheme. More so-

This is a so called first order scheme. More sophisticated higher order schemes are presented in Osher and Sethian (1988) and Osher and Shu (1991). When adding complicated F velocities (which can be place and time dependent) the nice stability and accuracy properties of such schemes still hold (see Sethian and Strain 1992). The above scheme is readily extended to more than one dimension, for example for the case $H(u, v) = f(u^2, v^2)$ (in our case $u = \phi_x$, $v = \phi_y$)

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t \cdot g \left(D_-^x \phi_{ij}^n, D_+^x \phi_{ij}^n; D_-^y \phi_{ij}^n, D_+^y \phi_{ij}^n \right)$$
(19)

Here

$$g_{HJ} = f\left(\left(\min\left(D_{-}^{x}\phi_{ij}^{n},0\right)\right)^{2} + \left(\max\left(D_{+}^{x}\phi_{ij}^{n},0\right)\right)^{2}; \\ \left(\min\left(D_{-}^{y}\phi_{ij}^{n},0\right)\right)^{2} + \left(\max\left(D_{+}^{y}\phi_{ij}^{n},0\right)\right)^{2}\right)$$
(20)

The result is the following algorithm

- Choose a function $\phi(x, y, 0)$ such that
 - $\phi(x, y, 0) = 0$ provides the initial curve **X**(s, 0).
- $-\phi(x, y, 0) < 0$ in the interior of the initial curve.
- $\phi(x, y, 0) > 0$ in the exterior of the initial curve.

 $-\phi(x, y, 0)$ is Lipschitz continuous.

 Propagate φ on an x, y-grid of desired spatial resolution according to

$$\phi_t - F \|\nabla \phi\| = 0$$

using a conservation form numerical scheme.

• Draw an equal height contour every $\frac{\Delta H}{\Delta t}$ time steps, by finding the contour (level set) $\phi(x, y, k\Delta H) = 0$ which is $\mathbf{X}_{k\Delta H}(s)$. The result is a weak solution of (6), obeying the entropy condition.

This algorithm automatically enforces the entropy condition, and frees one from the need to take care of topological changes (see Fig. 3).

In fact this formulation deals with the topology of all up going (or down going) surfaces without any external control or outside interference.



Fig. 3. When ϕ propagates in time, the function may stay continuous while the height contours form two separate close curves which are not connected anymore.

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

The Eulerian formulation was introduced by Osher and Sethian in order to deal with constant or curvature dependent velocities. In our problem the velocity Fis position dependent, a function of the image brightness. From the velocity definition (7) it is obvious that as $E \rightarrow 1$, at the singular points, the velocity $F \rightarrow \infty$. In order to avoid numerical problems we restrict the brightness function to get the maximum values of $E_{\text{max}} < 1$ as follows

$$\tilde{E} = \begin{cases} E & 0 \le E \le E_{\max} \\ E_{\max} & E_{\max} < E \le 1 \end{cases}$$

which yields $F_{\text{max}} = \frac{E_{\text{max}}}{\sqrt{1-E_{\text{max}}^2}}$. This restriction is also necessary in order to specify Δt for which the numerical flow still obeys the monotonicity demand and the CFL (Courant Friedrichs Lewy) condition.

4.3 Initialization

Every ϕ function which obeys the demands described earlier provides a good initialization. We present several ways to initiate the ϕ function, obeying smoothness, continuity and $\phi(x, y, 0) = 0$ gives the initial contour. Given $X_0(s)$, it is possible to produce the following initialization

$$\phi(x, y, 0) = \begin{cases} +d((x, y), \mathbf{X}(s, 0)) \\ (x, y) \in \text{exterior of } \mathbf{X}(s, 0) \\ -d((x, y), \mathbf{X}(s, 0)) \\ (x, y) \in \text{interior of } \mathbf{X}(s, 0) \\ 0 \\ (x, y) \in \mathbf{X}(s, 0) \end{cases}$$
(21)

where $d(\cdot, \cdot)$ is the (minimal) Euclidean distance of the point from the initial contour. Alternatively, by limiting the values of ϕ to [-C, +C], we can have

$$\phi(x, y, 0) = \begin{cases} \min[+d((x, y), \mathbf{X}(s, 0)), C] \\ (x, y) \in \text{exterior of } \mathbf{X}(s, 0) \\ \max[-d((x, y), \mathbf{X}(s, 0)), -C] \\ (x, y) \in \text{interior of } \mathbf{X}(s, 0) \\ 0 \quad (x, y) \in \mathbf{X}(s, 0) \end{cases}$$
(22)

Here C is an arbitrary constant. If we choose $h = \Delta x = \Delta y = C = 1$ then the values of the $\phi(x, y, 0)$ function on the grid will vary in the interval [-1, 1]. The values of the open interval (-1, 1) will only be given to grid points at a distance less then the mesh size from the curve. This initialization process is quite simple.

When dealing with rotationaly symmetric reflectance maps it is possible to define the initial height contour by first thresholding the gray-levels in the picture and separating all the "singular" areas. Then one can use the gray-level function to initiate the ϕ function in a simple way. For example if the gray levels of the shape $\in (0, G]$ (where $G \in (0, 1)$ is the selected threshold), the singular areas $\in (G, 1]$ (possibly white), then the first level contour can be approximated as the level set of gray level G. In this case we can take $\phi(x, y, 0) = E(x, y) - G$ 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 later examples.

4.4 Height Assignment

After initialization has been completed the ϕ is propagated according to the above described algorithm. Our main goal is finding the height of 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:

Every iteration step, for each grid point, check

If
$$\left(\phi_{i,j}^{n} \cdot \phi_{i,j}^{n-1} < 0\right)$$

then height_{*i*,*j*} = $\Delta t \left(n - \frac{\phi_{i,j}^{n}}{\phi_{i,j}^{n} - \phi_{i,j}^{n-1}}\right)$.

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

4.5 The Contour Finder

If height contours of the reconstructed shape are needed, a simple contour finder for $\mathbf{X}(s, L)$ can be generated following (Sethian and Strain 1992) in the following manner: for each grid point (i, j), use a cell definition as follows $\mathcal{N}_{ij} = \{\phi_{i,j}, \phi_{i+1,j}, \phi_{i+1,j+1}, \phi_{i,j+1}\}$. Now, if $\max[\mathcal{N}_{ij}] < 0$ or $\min[\mathcal{N}_{ij}] >$ 0 then the contour $\mathbf{X}(s, L)$ does not pass through the cell. Otherwise find the entrance and exit points

118 Kimmel et al.



Fig. 4. Reconstruction of a "volcano" mountain, see text. The gray curve in d. is the initial contour.

of $\phi = 0$ by linear interpolation; this provides a line segment of $\mathbf{X}(s, L)$ belonging to the contour. The line segments need neither to be ordered nor directed in the same direction in order to display the desired contour (see Fig. 5), however using additional information like the knowledge of the interior, one can produce any desired representation of the curve like polygonal, cubic or any other polynomial representation.

4.6 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 earlier (see Fig. 1b). However, it is possible to find the normal direction recalling that the surface normal projection on the (x, y)plane is in the direction of the contour normal \hat{n} . In other words, the surface normal is the intersection of the ambiguity cone and the plane defined by \hat{n} and \hat{Z} (where $\hat{Z} \equiv [0, 0, 1]^T$). We get the following pair of equations

$$\begin{cases} E = \hat{l} \cdot \hat{N} \\ (\hat{n} \times \hat{Z}) \cdot \hat{N} = 0 \end{cases}$$
(23)

The contour normal direction is given by $\hat{n} = -\frac{\nabla \phi}{\|\nabla \phi\|}$. When propagating the function ϕ it is important to estimate the exact velocity of F near the current





Fig. 5. Contour Finder finds a line segment in \mathcal{N}_{ii} .

contour. The velocity F is calculated by using the estimated projection of \hat{N} on the time changing \hat{n} . In this case, Eq. (23) produces two possible solutions for each grid point. The choice between the two is quite simple when one of them is negative (the projection on the \hat{n} or on the \hat{Z} axis is negative). In this case we must choose the positive solution. In the other case, when the two possible solutions are positive, the selection process is done by using the continuity of surface normals, that is, by using the unambiguous normals along the the curve to eliminate the wrong solutions. The velocity is "time" dependent in the general case of light source direction, because of the fact that the planar normals directions change as the 3D surface propagates, and it can be calculated every few iterations, (F(x, y, t)) is the surface normals' projection on \hat{Z} .)

4.7 About Topological Problems

There are several topological problems that can not be solved by the algorithm introduced earlier. Bruckstein proposed in Bruckstein (1988) to rely on Maxwells' (1870) and Cayleys' (1859) results on the possible behavior of equal height contours of smooth surfaces. We shall discuss two methods for handling complex topological problems, or at least understanding the difficulties in solving such problems. To use these methods we use the possibility of detecting and characterizing saddle points while propagating the contours (see also Oliensis 1990, 1991).

For the first method, let us imagine an object in an empty container. Filling water into the container will generate height contours on the object. If there is a hole or a deep crater in the object, it would not be filled with water even when the water level exceeds the lowest point in the crater. Water will start flowing into the crater only after the water line has reached a certain saddle-point. After that the water will flow over the saddle and fill the crater. The waters' path from the saddle to the lowest point of the crater is a characteristic strip starting at the saddle point and ending at a local minimum. After the water level in the crater reaches the saddle point, adding more water will raise the general water level in the container. The type of saddle point described above is reached by the height contour only from one side (a simple saddle point is characterized by four sides).

An algorithmic interpretation of the processes we have just described is the following. When developing the shape from shading algorithm via height contours, if we detect that the current height contour approaches a singular point from one side only we can conclude that this point is a saddle point, stop the propagating curves and attempt to produce characteristic strips starting from the "opposite" side of the saddle point. One scharacteristic will reach the local minimum of the crater, and we can start propagating height contours inside the crater. When this propagating contour reaches the saddle point from which the characteristic strip began, it will merge with the outside contour (water level) and produce a complete equal height curve from which the process can continue. This process goes on till the whole object is reconstructed (covered with water). Note that if a saddle point is reached from two (opposite) sides the algorithm automatically takes care of the topology.

The second method has some similarities with the first one. It also uses saddle points as merging points. According to this second method, we start propagating height contours from all the assumed local minima/maxima as separate processes. A propagating contour will stop when it reaches a singular (saddle) point from only one direction, and "wait" for another curve from a different process to reach the same saddle point from the opposite direction. It could be possible to find the source from which the second propagating contour begins by some heuristic, such as considering the distance of the source (singular point) from the saddle point or by propagating a characteristic strip to the other side of the saddle as a probe searching for the minimum, (which is actually the first method). two contours will merge into one. The first contour assigns its height to the second one, and the local processes will merge and continue propagating together. This procedure should go on, until all singular points have been covered by the merging and propagating contours, thereby reconstructing the entire object. Note that the height assignment between the merging local processes is somewhat similar to a chain reaction.

The Eulerian algorithm handles all "going up" or "going down" surfaces, when the initial conditions are known. Given an initial contour, the surface may be reconstructed automatically if the propagating contour does not have to change its propagation course. This is the case in the volcano example described in the next section, assuming starting from the top. However, if the propagating contour has to start decreasing instead of increasing its height, the change of direction will cause the topological problems discussed in this section. In some further research we explored the topological problems. We showed that by adding smoothness assumptions and excluding pathological singularities, the topological ambiguities may be resolved in a simple deterministic way (Kimmel and Bruckstein 1993b). Still, the +/- ambiguity of the shape from shading problem can not be solved, and is obviously an inherent characteristic of the problem. In the next section we demonstrate the Eulerian algorithm running on real and synthetic images without considering the topological problems that might arise.

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, and the size of these images being quite small (64×64 pixels). The initialization is achieved by using gray-level thresholding to specify initial "singular areas". In all of the examples the light source direction is known to coincide with the viewer direction for simplicity. In any case the light source direction is assumed to be known. However, a more general formulation of the shape from shading via curve evolution is possible. The new formulation overcomes the need for special treatment along the contour in cases of oblique light sources (Kimmel and Bruckstein 1993c). Figure 4 shows an example of reconstruction for a "volcano" surface, starting from a small curve around the singular area at the top of the mountain. In the equal height contours picture of the reconstructed surface (Fig. 4d) 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.

Figures 6a, b and c present the errors of the original object compared to its reconstruction. The difference between the original synthetic object and its reconstruction is displayed as a density map, as an elevation array and as a graph of a slice along the main diagonal running from (1, 1) to (64, 64). The heights along the slice as well as the errors are shown in two graphs. The error analysis is performed for three different parameters: ($\Delta t = 0.01, E_{\text{max}} = 0.999$), ($\Delta t =$ 0.07, $E_{\text{max}} = 0.99$), and ($\Delta t = 0.21$, $E_{\text{max}} = 0.977$), in Figs. 6a, b and c, respectively. The gray curve on the lower left slice graph is the reconstruction while the black curve is the original height along the slice. As we increase E_{max} while truncating the gray levels, Δt should be decreased (the CFL condition must be satisfied).

Another simple example is the reconstruction of an "up-going" surface. Three mountains are the original surface (Fig. 7a) producing image of Fig. 7b. The



Fig. 6a. Errors in reconstructing the volcano model, see text. $\Delta t = 0.001$, $E_{\text{max}} = 0.999$.



Fig. 6b. Errors in reconstructing the volcano model, see text. $\Delta t = 0.07$, $E_{\text{max}} = 0.99$.



Fig. 6c. Errors in reconstructing the volcano model, see text. $\Delta t = 0.21$, $E_{\text{max}} = 0.977$.

reconstruction of the surface from the image and the reconstructed equal height contours, using the contour finder described earlier, are shown in Figs. 7c and 7d. In this example the initial curve is the border of the large singular area surrounding the three mountains.

The surfaces shown so-far are smooth, and no shocks where formed in the propagating contours. A simple example demonstrating shock formation in the propagating contours, and the way shocks are dealt with by the numerical scheme, is provided in Fig. 8d. This example is an "up-going" surface from the given initial curve at the basis of two adjoining pyramids. The shapes of the pyramids are determined by the initial contour surrounding the base of the two pyramids. Starting from a circular contour surrounding a constant gray level area and propagating inwards would have resulted in a cone shape. The apex as well as the edges of the pyramids exist somewhere between the image pixels and are therefore not shown in the synthetic image. The initial contour is determined by the white (singular) area surrounding the two squares, where the gray level of the squares determines the slope of the pyramids.

Figures 9, 10 show the behavior when the image is corrupted by Gaussian noise. The reconstructed surface is affected by the noise but it can still be recognized even when the noise variance is quite large.

In Figs. 11 and 12 "Salt and Pepper" noise is added to the original picture. The algorithm overcomes such local disturbances by not allowing the grid samples of the ϕ function to get isolated negative valued points (see Figs. 11b, 12b), or by simply reconstructing the isolated black grid points as small pyramids (the "shocks" in Figs. 11c, 12c.)

Applying the algorithm to real images is demonstrated in the two following examples. The CCD camera pixel is rectangular, with aspect ratio 0.7, and this can be accounted for by the algorithm by simply defining $\Delta x = 0.7$ and $\Delta y = 1$. In Fig. 13, a reconstruction from the Socrates image, the picture of a small plastic statue (Fig. 13a) is demonstrated on a 128 × 128 pixel grid. Two views from different angles of the reconstructed surface are shown in Fig. 13c and 13d. Figure 14 is the result of the algorithm performing on a standard photo digitized on a 128 × 128 grid.

We have chosen to compare our algorithm to the viscosity solution approach that is a representative of a new generation of the shape from shading iterative algorithms. There is a close geometric relation between the two approaches: The viscosity solution uses an iterative numerical scheme to deform a surface until tan α (the surface orientation angle) coincides with $\sqrt{1-E^2}/E$, while our algorithm uses a numerical algorithm to propagate an equal height contour with local velocity of cot $\alpha = E/\sqrt{1-E^2}$. This relation and the connection to weighted distance transforms is further explored in (Kimmel and Bruckstein 1993a, c; Kimmel et al. 1994).



Fig. 7. Reconstruction of a synthetic image given on 64×64 grid, see text. The gray curve in d. is the initial contour.

6 A Comparison to the Viscosity Solutions Approach

Rouy and Tourin recover a Lambertian surface from it's shaded image by finding a viscosity solution to a Hamilton-Jacobi equation (Rouy and Tourin 1992). In this section we illustrate their approach with numerical simulations on three synthetic surfaces and compare it to the method of propagating level sets.

6.1 Hamilton-Jacobi Formulation

Recall that in the case of a Lambertian surface illuminated by a single distant overhead light source (with $\rho \lambda = 1$) the image irradiance equation reduces to

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

With $\vec{x} = (x, y)$ and $\nabla z(\vec{x}) = (p, q)$ this is a first order Hamilton-Jacobi equation of the form

$$H(\vec{x}, \nabla z(\vec{x})) = E(\vec{x})\sqrt{1 + |\nabla z(\vec{x})|^2} - 1 = 0$$
 (24)

Rouy and Tourin show that with $\Omega'\subset \Omega$ defined by

$$\Omega' = \{ \vec{x} \in \Omega / E(\vec{x}) \neq 1 \}$$



10

10

Fig. 8. Reconstruction of a synthetic image given on 64×64 grid, see text.



Fig. 9. Reconstruction after adding Gaussian noise to a 64×64 synthetic picture. a. Variance of 58. b. Variance of 318.

the equation

$$H(\vec{x}, \nabla z(\vec{x})) = 0 \quad \text{in } \Omega'$$
$$z(\vec{x}) = \phi \quad \text{on } \delta \Omega'$$
(25)

has at most one viscosity solution. This is a valuable result; if the value of the solution is known on $\delta\Omega'$, the shape is totally determined. If we set

$$n(\vec{x}) = \sqrt{\frac{1}{E(\vec{x})^2} - 1} \quad \text{on } \bar{\Omega}$$

$$|\nabla z(\vec{x})| = n(\vec{x}) \quad \text{in } \Omega'$$

 $z(\vec{x}) = \phi \quad \text{on } \delta \Omega'$ (26)

6.2 Algorithm

Rouy and Tourin show that a numerical approximation Z of equation 25 satisfies¹

$$Z_{ij} = \phi(x_i, y_j) \quad \forall (i, j) \in \delta Q';$$

$$g_{ij}(D_x^- Z_{ij}, D_x^+ Z_{ij}, D_y^- Z_{ij}, D_y^+ Z_{ij}) = 0$$

$$\forall (i, j) \in Q'.$$
(27)



Fig. 10. Reconstruction after adding Gaussian noise to a 64×64 synthetic picture. a. Variance of 58. b. Variance of 318.

They employ a fast algorithm proposed by Osher and Rudin (1993) to compute a solution of Eq. (27). In our experiments, we choose a simpler (but slower) scheme suggested by them (Rouy and Tourin 1992). We do so for ease of implementation; we are not particularly concerned with its speed. Let G be the operator defined on the space of all $\overline{Z} = (Z_{ij})_{(i,j) \in \overline{Q}}$ by

$$\begin{split} G(Z)_{ij} &= g_{ij}(D_x^- Z_{ij}, D_x^+ Z_{ij}, D_y^- Z_{ij}, D_y^+ Z_{ij}); \\ \forall (i, j) \in \mathcal{Q}' \end{split}$$

• Step $\mathbf{n} = \mathbf{0}$: Choose $Z^0 = (Z_{ij}^0)_{(i,j)\in \tilde{Q}}$ such that $Z_{ij}^0 = \phi(x_i, y_j) \ \forall (i, j) \in \delta Q' \text{ and } G(Z^0) \leq 0.$ • Step $\mathbf{n} + \mathbf{1}$: Let $Z_{ij}^{n+1} = Z_{ij}^n - \Delta t G(Z)_{ij};$ $\forall (i, j) \in Q'; \ \forall n \in \mathcal{N}.$

Results and Discussion 6.3

Figures 15, 16 and 17 show results of the viscositysolutions approach for the pyramids, three-mountains and volcano surfaces. Recall that for each shaded image, the viscosity solution applies in the region Q'where the normalized image intensity is not equal to

and let $\Delta t < \min(\Delta x, \Delta y)$.



Fig. 11. Reconstruction after adding 7% "Salt and Pepper" noise to a 64×64 synthetic picture.



Fig. 12. Reconstruction after adding 7% "Salt and Pepper" noise to a 64×64 synthetic picture.



Fig. 13. Reconstruction from a real image of Socrates plastic statue given on 128 × 128 grid.

1, and that the height of the solution surface must be specified on the boundary of this region $\delta Q'$ as an initial condition (Eq. 24). These figures depict the region of interest Q' and its boundary $\delta Q'$ in the following way: Q' is the white area and $\delta Q'$ is the boundary between the white area and the black background. We examine the performance of the viscosity-solutions approach under two circumstances: (1) when elevation information is fully specified as required by the algorithm i.e. at all the singular areas; (2) when only partial elevation information is available.

First, while the reconstruction of the *pyramids* surface is very good (Fig. 15), the reconstructed *threemountains* surface is not smooth in the hyperbolic areas between the mountain peaks (Fig. 16 row 2). Similarly, the reconstructed *volcano* surface is not smooth around the rim of the indentation and in addition has a 'jump' at the top (Fig. 17, row 2). In contrast the level-sets approach yields smooth results without artificial jumps or discontinuities, as shown in Fig. 4.

Second, the elevation of the surface at its singular areas may be known only partially. For example, when elevation information is specified at all local minima of height, the reconstruction of the *three-mountains* surface does not get worse, and the reconstruction of the *volcano* surface (Fig. 17, row 3) actually improves slightly; the 'jump' at its top has disappeared. As another example, when elevation information is specified at all local maxima of height and the solution surfaces are turned upside down, the reconstructed *three-mountains* surface is smooth in the hyperbolic areas, although accuracy is lost in the region around its base; similarly, the reconstructed *volcano* surface is smooth along the rim of its indentation, although accuracy is again lost in the region around its base.

a. Photograph Picture

b. Equal Height Contours



Fig. 14. Reconstruction from a photograph picture of the first author given on 128×128 grid.



Fig. 15. Results for the pyramids. Top left: the shaded image; top middle: equal height contours of the true surface; top right: 3D mesh plot of the true surface; bottom left: region (white is interior); bottom middle: equal height contours of the solution surface; bottom right: 3D mesh plot of the solution surface.



Fig. 16. Results for the three-mountains. Row one left: the shaded image; row one middle: equal height contours of the true surface; row one right: 3D mesh plot of the true surface; row two left: region one (white is interior); row two middle: equal height contours of solution surface one; row two right: 3D mesh plot of solution surface one; row three left: region two (white is interior); row three middle: equal height contours of solution surface one; row three left: region two (white is interior); row three middle: equal height contours of solution surface two; row three right: 3D mesh plot of solution surface two. Row four left: region three (white is interior); row four middle: equal height contours of solution surface three; row four right: 3D mesh plot of solution surface three inverted.



Fig. 17. Results for the volcano. Row one left: the shaded image; row one middle: equal height contours of the true surface; row one right: 3D mesh plot of the true surface; row two left: region one (white is interior); row two middle: equal height contours of solution surface one; row two right: 3D mesh plot of solution surface one; row three left: region two (white is interior); row three middle: equal height contours of solution surface two; row three right: 3D mesh plot of solution surface two. Row four left: region three (white is interior); row four middle: equal height contours of solution surface three; row four right: 3D mesh plot of solution surface three; row four right: 3D mesh plot of solution surface three inverted.

In conclusion, Rouy and Tourin prove a valuable uniqueness theorem and design an algorithm for recovering a Lambertian surface from its shaded image. In many cases their approach yields results that are comparable to those of the level-sets approach, in particular when only local minima or local maxima of height determine the initial condition. However, the use of elevation information at all singular areas can lead to artificial 'jumps' and discontinuities in the reconstructions. In contrast the level-sets approach requires fewer initial conditions, a single equal height contour, and in addition the reconstructions are free of artificial 'jumps' and discontinuities.

7 Concluding Remarks

We have described a method for recovering the shape of an object from its shaded image by an equal height contour propagation method. Topological problems in the propagated height contours are often inherently avoided in this method. An efficient and numerically stable implementation was presented. In this method shocks, cusps and other singularities formed in the contours are also readily dealt with in an efficient numerical scheme. The algorithm works on the pixel grid. It is easy to implement the algorithm in parallel using each mesh point as a small calculating device which communicates with its four close neighbors. In each iteration we need to calculate the values of $\phi(x, y, t)$ in those grid points close to the current contour and the rest of the grid points serve as sign holders. This can be exploited to reduce calculation effort.

In summary we propose to import to the computer vision field some recent advances in numerical methods for fluid dynamics. We have shown that wavefront propagation methods in fluid dynamics also provide a nice approach to the problem of shape from shading.

Acknowledgments

We thank Prof. James Sethian for providing us his reports on numerical methods of front evolution. We thank Prof. Moshe Israeli, Prof. Allen Tannenbaum and Prof. J. Rubinstein for their help in introducing us to the world of numerical analysis and moving surfaces.

Notes

1. Ω is the rectangular domain of \mathcal{R}^2 under consideration. Z_{ij} is the value of the numerical approximation of the solution at $(x_i, y_j) = (i\Delta x, j\Delta y)$, with mesh sizes $\Delta x, \Delta y > 0$. The index sets $Q', \delta Q'$ and \overline{Q} are defined by

$$Q' = \{(i, j) \in \mathcal{N}^2 / (x_i, y_j) \in \Omega'\}$$

$$\delta Q' = \{(i, j) \in \mathcal{N}^2 / (x_i, y_j) \in \delta \Omega'\}$$

$$\bar{Q} = \{(i, j) \in \mathcal{N}^2 / (x_i, y_j) \in \bar{\Omega}\}$$

 N_{ij} is the value of *n* at (x_i, y_j) . For all approximations *Z*, for all $(i, j) \in Q'$

$$D_x^+ Z_{ij} = \frac{Z_{i+1j} - Z_{ij}}{\Delta x}$$
$$D_y^+ Z_{ij} = \frac{Z_{ij+1} - Z_{ij}}{\Delta y}$$
$$D_x^- Z_{ij} = \frac{Z_{ij} - Z_{i-1j}}{\Delta x}$$
$$D_y^- Z_{ij} = \frac{Z_{ij} - Z_{ij-1}}{\Delta y}$$

g is a vector of functions from \mathcal{R}^4 to \mathcal{R} defined by

$$g_{ij}(a, b, c, d) = \sqrt{\max(a^+, b^-)^2 + \max(c^+, d^-)^2} - N_{ij}; \forall (i, j) \in Q'; \forall (a, b, c, d) \in \mathcal{R}^4$$

References

- Ascher, U.M. and Carter, P.M. 1993. A Multigrid Method for Shape from Shading. SIAM J. Numer. Anal. 30(1):102–115.
- Brooks, M.J. and Horn, B.K.P. 1985. Shape and Source from Shading. Proc. Intern. Joint Conf. Artif. Intell., Los Angeles, pp. 932– 936. Also in Horn and Brooks, 1989.
- Bruckstein, A.M. 1988. On Shape from Shading. Comput. Vision Graphics Image Process, 44:139–154. First appeared On Shape from Shading: Some Theoretical Considerations. EE Report No. 524, Technion-Israel Institute of Technology, 1985.
- Cayley, A. 1859. On Contour and Slope Lines. *Philos. Mag.*, pp. 264–268.
- Chopp, D.L. 1991. Computing Minimal Surfaces via Level Set Curvature Flow. Lawrence Berkeley Lab. and Dep. of Math., Uni. of CA. Berkeley, LBL-30685, Ph.D Thesis.
- Chopp, D.L. and Sethian, J.A. 1991. Flow Under Mean Curvature: Singularity Formation and Minimal Surfaces. *Center for pure applied math.* Univ. of CA. Berkeley, PAM-541.
- Chorin, A.J. 1980. Flame Advection and Propagation Algorithms. J. of Comput. Phys., 35:1–11.
- Chorin, A.J. 1985. Curvature and Solidification. J. of Comput. Phys., 58:472–490.
- Courant, R. and Hilbert, D. 1953. Methods of Mathematical Physics, Vol. I. Wiley: New York.
- Courant, R. and Hilbert D. 1962. Methods of Mathematical Physics, Vol. II. Wiley: New York.
- Frankot, R.T. and Chellappa, R. 1988. A Method of Enforcing Integrability in Shape from Shading Algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4(10):439–451. Also in Horn and Brooks, 1989.

- Horn, B.K.P. 1975. Obtaining Shape from Shading Information. In P.H. Winston (ed.), *The Psychology of Computer Vision*, McGraw Hill: New York, pp. 115–155. Also in Horn and Brooks, 1989.
- Horn, B.K.P. 1986. Robot Vision. MIT Press: Cambridge, MA; and McGraw-Hill: New York.
- Horn, B.K.P. 1990. Height and Gradient from Shading. Int. J. of Comput. Vision, 5:37–75.
- Horn, B.K.P. and Brooks, M.J. 1986. The Variational Approach to Shape from Shading. *Comput. Vision, Graph. Image Process*, 2(33):174–208. Also in Horn and Brooks, 1989.
- Horn, B.K.P. and Brooks, M.J. (eds.) 1989. Shape from Shading. MIT Press: Cambridge, MA.
- Ikeuchi, K. and Horn, B.K.P. 1981. Numerical Shape from Shading and Occluding Boundaries. *Artificial Intelligence*, 17(1-3): 141-184. Also in Horn and Brooks, 1989.
- John, F. 1982. Partial Differential Equations. In *Applied Mathematical Sciences*, 4th ed., Springer Verlag, New York.
- Kimia, B.B. 1990. Toward a Computational Theory of Shape. Ph.D. Thesis, Department of Electrical Engineering, McGill University, Montreal.
- Kimia, B.B. and Siddiqi K. 1994. Geometric Heat Equation and Non-linear Diffusion of Shapes and Images. In Proc. of CVPR.
- Kimia, B.B., Tannenbaum, A., and Zucker, S.W. 1990. Toward a Computational Theory of Shape: An Overview. In *Proceedings* of the First European Conference on Computer Vision, Springer Verlag, Antibes, France.
- Kimia, B.B., Tannenbaum, A., and Zucker, S.W. 1992. On the Evolution of Curves via a Function of Curvature, (I): The Classical Case. J. of Math. Analysis and Applications, 163(2):000–000.
- Kimia, B.B., Tannenbaum, A., and Zucker, S.W. (to appear). Shapes, Shocks, and Deformations, I: The Components of Shape and the Reaction-Diffusion Space. *International Journal of Computer Vi*sion.
- Kimmel, R. 1992. Shape from Shading via Level Sets. M.Sc. Thesis (in Hebrew), Technion-Israel Institute of Technology.
- Kimmel, R. and Bruckstein, A.M. 1992. Shape from Shading via Level Sets. CIS Report #9209, CS Department, Technion.
- Kimmel, R. and Bruckstein, A.M. 1993a. Distance Maps and Weighted Distance Transforms. *Proceedings SPIE—Geometric Methods in Computer Vision II*, San Diego, California, 2031: 259–268.
- Kimmel, R. and Bruckstein, A.M. 1993b. Global Shape from Shading. CIS Report #9327, CS Department, Technion, to appear in CVIU.
- Kimmel, R. and Bruckstein, A.M. 1993c. Tracking Level Sets by Level Sets: A Method for Solving the Shape from Shading

Problem. CIS Report #9319, CS Department, Technion. CVIU, 62(1):47–58, July 1995.

- Kimmel, R., Kiryati, N., and Bruckstein, A.M. 1994. Distance Maps and Weighted Distance Transforms. To appear in J. of Mathematical Imaging and Vision, Special Issue on Topology and Geometry in Computer Vision.
- Maxwell, J.C. 1870. On Hills and Dales. *Philos. Mag.* (Maxwell's Collected Works), pp. 233–240.
- Mitchel, A.R. and Griffiths, D.F. 1985. The Finite Difference Method in Partial Differential Equations. Wiley: New York.
- Oliensis, J. 1990. Uniqueness in Shape from Shading. Int. J. of Comput. Vision, 6(2):75–104.
- Oliensis, J. 1991. Shape from Shading as Partially Well-Constrained Problem. CVGIP: Image Understanding, 54(2):163–183.
- Osher, S. and Rudin., L. 1993. Rapid convergence of approximate solutions to shape-from-shading problem, *Cognitech Inc.*, Unpublished.
- Osher, S.J. and Sethian, J.A. 1988. Fronts Propagating with Curvature Dependent Speed: algorithms based on Hamilton-Jacobi formulations. J. of Comp. Phys, 79:12–49.
- Osher, S.J. and Shu, C.W. 1991. High-Order Essentially Nonoscillatory Schemes for Hamilton-Jacobi Equations. SIAM J. Numer. Anal., 28(4):907–022.
- Pentland, A.P. 1989. Local Shading Analysis. *IEEE Trans. on PAMI*, 6(2):170–187. Also in Horn and Brooks, 1989.
- Pentland, A.P. 1990. Linear Shape from Shading. Int. J. of Comput. Vision, 4:153–162.
- Rouy, E. and Tourin, A. 1992. A viscosity solutions approach to shape-from-shading. SIAM. J. Numer. Anal., 29(3): 867–884.
- Sethian, J.A. 1985. Curvature and the Evolution of Fronts. Commun. in Math. Phys., 101:487–499.
- Sethian, J.A. 1987. Numerical Methods for Propagating Fronts. In Variational methods for free surface interfaces, P. Concus and R. Finn (eds.), Springer-Verlag, New York.
- Sethian, J.A. 1989. A Review of Recent Numerical Algorithms for Hypersurfaces Moving with Curvature Dependent Speed. J. of Diff. Geom., 33:131–161.
- Sethian, J.A. and Strain, J. 1992. Crystal Growth and Dendritic Solidification. submitted for publication: J. of Computational Physics, 98:231–253.
- Sod, G.A. 1985. Numerical Methods in Fluid Dynamics, Cambridge Univ. Press.
- Zhu, J. and Sethian, J.A. 1991. Projection Methods Coupled to Level Set Interface Techniques. Lawrence Berkeley Lab. and Dep. of Math., Univ. of CA. Berkeley, LBL-30183.