Shape from Shading via Level Sets

by

R. Kimmel and A.M. Bruckstein

CIS Report #9209
June 1992
SHAPE FROM SHADING via LEVEL SETS

Ron Kimmel
Electrical Engineering Dept.
Technion, Haifa 32000, ISRAEL.
ron@techunix.technion.ac.il

Alfred M. Bruckstein
Computer Science Dept.
Technion, Haifa 32000, ISRAEL.
freddy@cs.technion.ac.il

June 10, 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 handles gracefully 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.

*R. Kimmel's work was partly supported by the Ollendorf Fund.

1A. Bruckstein's work was supported in part by the Fund for the Promotion of Research at the Technion.
1 Introduction

Computer vision researchers invented several methods of reconstructing three dimensional objects from their two dimensional images, see e.g. [2, 3, 9, 12, 13, 15, 16, 17, 27, 26]. Another 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 in [25]. 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 [3] with well behaved (i.e. stable and locally accurate) numerical methods proposed by Osher and Sethian for 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 its 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 \( p = \frac{\partial z}{\partial x} \) and \( q = \frac{\partial z}{\partial y} \). The surface normal at each point is perpendicular to the plane determined by the vectors \( (1, 0, p)^T \) and \( (0, 1, q)^T \), therefore its direction is their vector product

\[
N = \begin{vmatrix}
\hat{z} & \hat{y} & \hat{z} \\
1 & 0 & p \\
0 & 1 & q
\end{vmatrix} = (-p, -q, 1)^T.
\]
hence the unit normal can then be written as

\[
\hat{N} = \frac{N}{\|N\|} = \frac{1}{\sqrt{1 + p^2 + q^2}}(\begin{pmatrix} -p \\
-q \\
1 \end{pmatrix})^T
\]

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 \(\hat{i}\) (see Figure 1).

![Diagram](image)

Figure 1: 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 \(E = \cos \alpha = \hat{N} \cdot \hat{i}\).

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 \(\hat{i} = (0, 0, 1)^T\). In the general case we define the light source direction as

\[
\hat{i} = \frac{1}{\sqrt{1 + p_i^2 + q_i^2}}(\begin{pmatrix} -p_i \\
-q_i \\
1 \end{pmatrix})^T
\]

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 \(E(x, y)\)

\[
E(x, y) = \text{Function of}((\hat{N}) = R(p(x, y), q(x, y)) )
\]
This is the image irradiance equation. For example in the simple Lambertian case

\[
E(z, y) = \rho \, R(p, q) = \rho \, \lambda \cos \alpha \\
= \rho \lambda \, \hat{i} \cdot \hat{N} = \rho \lambda \, \frac{1 + p^2 + q^2}{\sqrt{1 + p^2 + q^2}} \frac{1}{\sqrt{1 + p^2 + q^2}}
\]

where \(\rho\) and \(\lambda\) are proportionality factors that can be neglected \((\rho \lambda = 1)\) by rescaling the image irradiance. \((\rho\) 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 \(\lambda\) 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 Figure 2.b).

Figure 2: a. When the light source is from above \((\hat{i} = \hat{z})\) 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 \((z, y)\)-plane.

For the simple case where \(\hat{i} = (0, 0, 1)\) 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 Figure 2.a).
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 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 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 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 [13]. These methods were developed over the last decade [17, 2, 15, 13]. 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

\[
\int \int B^2 dx dy
\]

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 [10, 11])

\[
\int \int (S + \lambda B) dx dy \rightarrow \min
\]
This problem provides the Euler equations

\[
\begin{align*}
\nabla^2 p + \lambda R_p &= 0 \\
\nabla^2 q + \lambda R_q &= 0
\end{align*}
\]

and by eliminating the Lagrange multiplier \( \lambda \) one has to solve

\[
\begin{align*}
R_q \nabla^2 p &= R_p \nabla^2 q \\
E(x, y) &= R(p, q)
\end{align*}
\]

but, unfortunately no convergent iterative scheme has been found (see [17]). Brooks [2] proposed a regularization of the original problem by requiring

\[
\int \int (B^2 + \lambda S) dzdy \rightarrow \min
\]

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

\[
\int \int I_1 dzdy \rightarrow \min
\]

results in the Poisson 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

\[
\int \int (B^2 + \lambda I_2) dzdy \rightarrow \min
\]

see [15], which is an ordinary calculus problem. The resulting scheme avoids the excessive smoothing, but was found to be less stable. In [13], Horn proposes to consider

\[
\int \int (B^2 + \mu I_1) dzdy \rightarrow \min
\]
assumption that each point lies on a sphere of unit radius. Moreover, in the
exact surface normal direction, another equation is needed. Pertinent, in [26], adds the
picture. We have only one equation at each point, \( \mathbf{N}(x) \cdot \mathbf{H} = 0 \). In order to find the

As mentioned earlier, the problem of shape from shading is equivalent to finding the values

of two variables representing the surface normal direction \( \mathbf{N}(x) \) at each point of the shading

A. Non-Iterative Methods

Initial condition to the scheme

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

I used a multigrid method to speed up the iterative reconstruction process.

Frances and Chehab [6] used projection on a set of integrable functions in order to
force the integrability condition at each phase of the iterative process. In order to

The calculus of variations solution yields Euler equations that can be approximated to

\[
\min \quad \mathcal{F}(x) \int (\gamma \nabla + \delta \mathbf{H}) \int
\]

The calculus of variations solution yields Euler equations that can be approximated to

The calculus of variations solution yields Euler equations that can be approximated to

\[
(0 \theta' \theta' \theta') + (0 \theta' \theta' \theta') (0 \theta - d) + (0 \theta' \theta' \theta') \approx
\]

\[
\cdots + (0 \theta' \theta' \theta') + (0 \theta' \theta' \theta') (0 \theta - d) + (0 \theta' \theta' \theta') = (b' d)(\mathbf{H})
\]

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

\[
\min \quad \mathcal{F}(x) \int (\gamma \nabla + \delta \mathbf{H}) \int
\]

and
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 $E_z, E_y, E_{zz}, E_{yy}$ and $E_{zy}$ 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 $E_{zz} \cdot E_{yy} < 0$ 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 [27] Pentland deals with the problem of the linear reflectance map $R(p, q) = k_1 + k_2 p + k_3 q$, 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 $(z, y)$-plane known as a "characteristic strip" (see [12, 18]). 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 $\delta z$ in the $z$ direction and $\delta y$ 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 $z, y$ and $x$. The changes in $p$ and $q$ are given by

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

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 ($E(z, y) = R(p, q)$) is added to the game. Differentiating the image irradiance equation with respect to $z$ and $y$, leads to

$$\begin{align*}
E_z &= z_{xx} R_p + z_{xy} R_q \\
E_y &= z_{yx} R_p + z_{yy} R_q
\end{align*}$$

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

$$\begin{align*}
\delta z &= R_p \delta s \\
\delta y &= R_q \delta s
\end{align*}$$

yields

$$\begin{align*}
\delta p &= E_z \delta s \\
\delta q &= E_y \delta s
\end{align*}$$

Hence the following set of five ordinary differential equations

$$\begin{align*}
dz &= R_p ds \\
dy &= R_q ds \\
dz &= (pR_p + qR_q) ds \\
dp &= E_z ds \\
dq &= E_y ds
\end{align*}$$

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 [12]. He proposed to look at the brightness map $E(z, y)$ and start height recovery around singular points; there the image $E(z, y)$ 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 \( z_t = p \varepsilon_t + q \gamma_t \) 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 equation (4):

\[
\begin{align*}
    z^{n+1} &= z^n + R^n_{p} \Delta s \\
    y^{n+1} &= y^n + R^n_{q} \Delta s \\
    z^{n+1} &= z^n + (p^n R^n_{p} + q^n R^n_{q}) \Delta s \\
    p^{n+1} &= p^n + E^n_x \Delta s \\
    q^{n+1} &= q^n + E^n_y \Delta s
\end{align*}
\]

Here \( n \) stands for the the parameter at time \( t = n \Delta t \). Assume that \( E^n_x \) and \( E^n_y \) are the exact derivatives, and that the direction between point \( P^n \equiv (z^n, y^n) \) and the next point in the generated solution \( P^{n+1} \equiv (z^{n+1}, y^{n+1}) \) 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

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

Define the height error \( Z_{err}^{n+1} \equiv \hat{z}^{n+1} - z^{n+1} \), 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
\( P^{n+1} \).

\[
Z_{err}^{n+1} = z^{n+1} - z^n + \Delta s \int_{L^n(t)} (pR_p + qR_q)dt
\]

\[
= z^n + (p^nR_p^n + q^nR_q^n)\Delta s - z^n - \Delta s \int_{L^n(t)} (p(t)R_p(t) + q(t)R_q(t))dt
\]

\[
= \Delta s \{ (p^nR_p^n + q^nR_q^n) - \int_{L^n(t)} (p(t)R_p(t) + q(t)R_q(t))dt \}
\]

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

\[
Z_{err}^{n+1} = \Delta s \{ (E^n)^3 - E^n - \int_{L^n(t)} (E^3(t) - E(t))dt \}
\]

Summing the errors along the characteristic gives

\[
Z_{TotalErr}^n = \sum_{i=0}^{n-1} Z_{err}^i
\]

\[
= \Delta s \sum_{i=0}^{n-1} \{ (E^i)^3 - E^i - \int_{L^i(t)} (E^3(t) - E(t))dt \}
\]

It seems that as \( \Delta s \to 0 \) the total error \( Z_{TotalErr}^n \to 0 \), 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 \( \Delta s \to 0 \) 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 [12], 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 [3]: an equal-height contour or a level curve is a continuous curve in the \((z, y)\)-plane on which the function \(z(z, y)\) is constant. Defining \(\{z(s), y(s)\}, s \in [0, S]\), as the parametric representation of the contour, we have

\[
\frac{\partial z}{\partial s} = \frac{\partial z}{\partial s} \frac{\partial z}{\partial s} + \frac{\partial z}{\partial y} \frac{\partial y}{\partial s} = 0
\]

or

\[
p z_s + q y_s = 0
\]

This reflects that there is no change in height along the equal height contour. The unit normal to the equal height contour on the \((z, y)\)-plane is given by

\[
\hat{n}(s) = \frac{1}{\sqrt{z^2_s(s) + y^2_s(s)}} \{y_s(s), -z_s(s)\}
\]

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 \(D\) in the normal direction \(\hat{n}\) in the \((z, y)\)-plane.

From basic geometry we have

\[
D = dz \cot \alpha
\]

where \(\alpha\) 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

\[
D = dz \cot \alpha = dz \frac{1}{\sqrt{p^2 + q^2}} = dz \frac{E}{\sqrt{1 - E^2}}
\]

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

\[
\{z(s, dz), y(s, dz)\} = \{z(s, 0), y(s, 0)\} + D(s) \cdot \hat{n}
\]

This yields the propagation of the equal height contours as a nonlinear initial value P.D.E. problem. Given \(\{z(s, 0), y(s, 0)\}\) the evolution equations are

\[
\begin{cases}
    z_t(s, t) = F(z, y) \frac{y_s}{\sqrt{z^2_s + y^2}} \\
    y_t(s, t) = F(z, y) \frac{-z_s}{\sqrt{z^2_s + y^2}}
\end{cases}
\]

(6)
where \( t \equiv z \) and

\[
F(z(s,t), y(s,t)) = \frac{E(z(s,t), y(s,t))}{\sqrt{1 - E^2(z(s,t), y(s,t))}}
\]  

(7)

Define \( C(0) = X(s,0) = \{z(s,0), y(s,0)\} \) 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 given as a scalar function of the brightness \( E \). Using this notation, equation (6) can be written as a planar curve evolution equation

\[
\frac{\partial}{\partial t} C(t) = F(z, y) \cdot \hat{n}
\]

Sethian [30] 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 equation (6). Following an analysis of the problem as was done by Sethian in [30], a simple difference approximation is considered. Divide the parametrization interval \([0, S]\) into \( M \) equal intervals of size \( \Delta s \) and \( t \) into equal intervals of length \( \Delta t \). Define a marker point as \( P^n_i = \{z^n_i, y^n_i\} = \{z(i \Delta s, n \Delta t), y(i \Delta s, n \Delta t)\} \). A numerical algorithm should produce new values \( P^{n+1}_i \) from previous positions. Approximate parameter derivatives using a central difference approximation and time derivatives using a forward difference approximation as follows

\[
\frac{d(P^n_i)}{ds} \approx \frac{P^n_{i+1} - P^n_{i-1}}{2 \Delta s}
\]

\[
\frac{d(P^n_i)}{dt} \approx \frac{P^{n+1}_i - P^n_i}{\Delta t}
\]

Substitution of these approximations into equation (6) gives

\[
P^{n+1}_i = P^n_i + \Delta t F(z^n_i, y^n_i) \frac{\{y^n_{i+1} - y^n_{i-1}, -(z^n_{i+1} - z^n_{i-1})\}}{\sqrt{(y^n_{i+1} - y^n_{i-1})^2 + (z^n_{i+1} - z^n_{i-1})^2}}
\]

(8)

The discretization interval \( \Delta s \) 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 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

$$P_{i+1}^n = P_i^n + \{\Delta z_i^n, \Delta y_i^n\}$$

Using this definition and the approximation (8) we have

$$\{\Delta z_i^n, \Delta y_i^n\} = \Delta t \frac{F(z_i^n, y_i^n)(z_i^n - z_s)}{\sqrt{z_i^n + y_i^n}}$$

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

$$\Delta r^n = \sqrt{(\Delta z_i^n)^2 + (\Delta y_i^n)^2} = \Delta t \frac{F(z_i^n, y_i^n)}{\sqrt{z_i^n + y_i^n}}$$

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

$$dr^n = \sqrt{(dz^n)^2 + (dy^n)^2} = \sqrt{(dr^n \Delta z)^2 + (dr^n \Delta y)^2} = dr \Delta r^n$$

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

$$dt = \frac{dr^n}{F(z, y)} = \frac{\Delta r^n}{F(z, y)} d\tau$$

for every $\tau \in [0, 1]$. 

15
Define \( F^n \equiv F(z^n, y^n) \) and \( F^n(\tau) \equiv F(z^n + \Delta z^n \tau, y^n + \Delta y^n \tau) \). The height error can then be written as

\[
Z_{err}^{n+1} = z^{n+1} - z^{n+1}
= \Delta t - \int_{F^n}^{P^{n+1}} dt
= \Delta t - \int_0^1 \frac{\Delta x^n}{F^n(\tau)} d\tau
= \Delta t(1 - F^n \int_0^1 \frac{1}{F^n(\tau)} d\tau)
\]

or as a function of the image brightness

\[
Z_{err}^{n+1} = \Delta t(1 - \frac{E^n}{\sqrt{1 - (E^n)^2}} \int_0^1 \frac{\sqrt{1 - (E^n(\tau))^2}}{E^n(\tau)} d\tau)
\]

The total error according to the above assumptions is given by

\[
Z_{TotalErr}^n = \sum_{i=0}^{n-1} Z_{err}^i
= \Delta t \sum_{i=0}^{n-1} \left\{1 - \frac{E^i}{\sqrt{1 - (E^i)^2}} \int L(\tau) \frac{\sqrt{1 - (E^i(\tau))^2}}{E^i(\tau)} d\tau\right\}
\]

As for the characteristic strips expansion, it seems that when \( \Delta t \to 0 \) the height error \( Z_{err} \to 0 \). But because of the same arguments, when \( \Delta t \to 0 \), 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 the curve evolution into a 3D-surface evolution, so that curves changing according to (6) are zero (or level) sets of evolving surface. 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, [28], the solution of the curve propagation according to 
\[ \frac{\partial}{\partial t} X(s, t) = F \hat{n} \] given \( X(s, 0) \) at time \( dt \), \( X(s, dt) \), corresponds to the envelope generated by the set of all disks of radii \( P dt \) centered on the initial curve \( X_0(s) \). 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 \( t^* \)" [28], 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 [25, 30]).

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 in [7, 8]. 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 \( X(s, 0) \). Assume that \( X(s, 0) \) is a closed curve and restrict \( \phi \) to be negative in the interior and positive in the exterior of the level set \( \phi(x, y, 0) = 0 \). Furthermore \( \phi \) 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 \( X(s, t) \) as if propagated by (6) and obeying the entropy condition. If \( \phi(x, y, t) = 0 \) along \( X(s, t) \) then by the chain rule we have

\[
\frac{\partial}{\partial t} \phi(x, y, t) + \frac{\partial}{\partial x} \phi(x(s, t), y(s, t), t) \cdot z_t + \frac{\partial}{\partial y} \phi(x(s, t), y(s, t), t) \cdot y_t = 0
\]

or

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

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

\[
v = \mathbf{X}_t(s, t) \cdot \mathbf{n}(s, t) \tag{10}
\]

In our case the velocity is given by the scalar function \( F(x, y) \) defined by equation (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 \( \mathbf{n}(s, t) = -\frac{\nabla \phi}{||\nabla \phi||} \), the minus sign indicating the inward direction of propagation, hence

\[
v = \mathbf{X}_t \cdot \mathbf{n} = \mathbf{X}_t \cdot \frac{-\nabla \phi}{||\nabla \phi||} = F. \tag{11}
\]

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 \( \phi \) propagating according to (12) with the level set \( \phi(x, y, 0) = 0 \) coinciding with \( X(s, 0) \), then \( \phi(x, y, t) = 0 \) will produce \( X(s, t) \) 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 [25], and show the connection to Hamilton Jacobi methods, weak solutions and conservation laws.
Consider the one dimensional equation

\[ \phi_t - ||\nabla \phi|| = \phi(x, t)t - \sqrt{\phi_x^2} = 0 \]  \hspace{1cm} (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

\[ u_t + [H[u]]_x = 0 \]  \hspace{1cm} (14)

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

\[ \frac{d}{dt} \int_a^b u(x, t) dx = H[u(a, t)] - H[u(b, t)] \]  \hspace{1cm} (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

\[ \frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{g(u_i^n, u_{i+1}^n) - g(u_{i-1}^n, u_i^n)}{\Delta x} \]  \hspace{1cm} (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, [32].

Some schemes based on this idea, like the Lax–Friedrichs and Godonov’s schemes, are presented in [25]. The simplest flow function that can be used for implementation is the so-called \( HJ \) flow, where for \( H(u) = f(u^2) \) the numerical flow can be given by using in (16) the function

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

and the appropriate (weak) entropy solution of \( \phi \) can be written, by integrating equation (16) with respect to \( x \), as

\[ \phi_i^{n+1} = \phi_i^n - \Delta t \cdot g(D_- \phi_i^n, D_+ \phi_i^n) \]  \hspace{1cm} (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 sophisticated higher order schemes are presented in [25]. When adding complicated \( F \) velocities (which can be place and time
dependent) the nice stability and accuracy properties of such schemes still hold, see [31].

The above scheme is readily extended to more than one dimension, for example for the case

\[ H(u, v) = f(u^2, v^2) \]

(our case \( u = \phi_x, \ v = \phi_y \))

\[
\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t \cdot g(D_x\phi_{ij}^n, D_y\phi_{ij}^n, D_x\phi_{ij}^n, D_y\phi_{ij}^n)
\]  

(19)

Here

\[
g_{ij} = f((\min(D_x\phi_{ij}^n, 0))^2 + (\max(D_x\phi_{ij}^n, 0))^2; (\min(D_y\phi_{ij}^n, 0))^2 + (\max(D_y\phi_{ij}^n, 0))^2)
\]  

(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 \( \phi \) 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) \( \phi(x, y, k\Delta t) = 0 \) which is \( X_{k\Delta H}(s) \). 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 Figure 3). In fact this formulation deals with the topology of all up going (or down going) surfaces without any external control or outside interference.

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 \( \phi \) function is fixed, thereby avoiding the stability problems of the Lagrangian formulation.
Figure 3: When $\phi$ propagates in time, the function may stay continuous while the height contours form two separate closed curves which are not connected anymore.

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

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

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

4.3 Initialization

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

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

(21)

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

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

(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 than the mesh size from the curve. This initialization process is quite simple.

When dealing with rotationally 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 \(\phi\) 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 \(\phi\) is propagated according to the above described algorithm. Our main goal is finding the height of each grid point, while the \(\phi\) function is propagated on the grid. A way to achieve accurate results is as follows:

- Initiate each grid points' height together with a distance function which represents the minimal distance of the local \(\phi\) value from zero (\(\text{height}_{ij} = 0\) and \(D_{ij} = |\phi_{ij}(0)|\)).

- After each iteration step, for each grid point, if \(D_{ij} > |\phi_{ij}(t)|\) then update \(\text{height}_{ij} = t\), and reassign the distance flag, \(D_{ij} = |\phi_{ij}(t)|\).
Using the above procedure each grid point gets its height at the "time" when the $\phi$ 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 $X(s, L)$ can be generated following [31] in the following manner: for each grid point $(i, j)$, use a cell definition as follows $N_{ij} = \{\phi_{i,j}, \phi_{i+1,j}, \phi_{i+1,j+1}, \phi_{i,j+1}\}$. Now, if $\max[N_{ij}] < 0$ or $\min[N_{ij}] > 0$ then the contour $X(s, L)$ does not pass through the cell. Otherwise find the entrance and exit points of $\phi = 0$ by linear interpolation; this provides a line segment of $X(s, L)$ 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 Figure 4), 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.

![Figure 4: Contour Finder finds a line segment in $N_{ij}$.](image-url)
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 Figure 2.b). However, it is possible to find the normal direction recalling that the surface normal projection on the $(z, y)$-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 $\phi$ it is important to estimate the exact velocity of $F$ near the current contour. The velocity $F$ is calculated by using the estimated projection of $\hat{N}$ on the time changing $\hat{n}$. In this case, equation (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, ($F(z, y, t)$ 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 [3] to rely on Maxwell's [21] and Cayleys' [4] 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 [24, 23].)
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 characteristic 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.) The 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.

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".

Figure 5 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 (Figure 5.d) one can observe the way topological problems like the saddle on the lowest left corner are inherently solved through this "down-going" process. Not however that starting at the base of the volcano wold require special treatment to proceed beyond the saddle point.

Another simple example is the reconstruction of an "up-going" surface. Three mountains are the original surface (Figure 6.a) producing image of Figure 6.b. The reconstruction of the surface from the image and the reconstructed equal height contours, using the contour finder described earlier, are shown in Figure 6.c and 6.d. 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 this way shocks are dealt with by the numerical scheme, is provided in Figure 7.d. This example is an "up-going" surface from the given initial curve at the basis of two adjoining pyramids.

Figures 10, 11 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 Figures 12 and 13 "Salt and Pepper" noise is added to the original picture. The algorithm overcomes such local disturbances by not allowing the the grid samples of the
Figure 5: Reconstruction of a "volcano" mountain, see text. The gray curve in d. is the initial contour.
Three Mountains

a. Original Surface

b. Brightness Image

c. Reconstructed Surface

d. Equal Height Contours

Figure 6: Reconstruction of a synthetic image given on 64 x 64 grid, see text. The gray curve in d. is the initial contour
Figure 7: Reconstruction of a synthetic image given on $64 \times 64$ grid, see text.
\( \phi \) function to get isolated negative valued points (see Figures 12.b, 13.b), or by simply reconstructing the isolated black grid points as small pyramids (the "shocks" in Figures 12.c, 13.c.)

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 \( \Delta z = 0.7 \) and \( \Delta y = 1 \). In Figure 8, a reconstruction from the Socrates image, the picture of a small plastic statue (Figure 8.a,) is demonstrated on a 128 \( \times \) 128 pixel grid. Two views from different angles of the reconstructed surface are shown in Figure 8.c and 8.d. Figure 9 is the result of the algorithm performing on a standard photo digitized on a 128 \( \times \) 128 grid.

6 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.

7 Acknowledgments

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

a. Picture of the plastic statue

b. Equal Height Contours

c. Reconstructed Surface View 1

d. Reconstructed Surface View 2

Figure 8: Reconstruction from a real image of Socrates plastic statue given on $128 \times 128$ grid.
Figure 9: Reconstruction from a photograph picture of the first author given on $128 \times 128$ grid.
Figure 10: Reconstruction after adding Gaussian noise to a $64 \times 64$ synthetic picture. a. Variance of 58. b. Variance of 318.
Figure 11: Reconstruction after adding Gaussian noise to a $64 \times 64$ synthetic picture a. Variance of 58. b. Variance of 318.
Figure 12: Reconstruction after adding 7% "Salt and Pepper" noise to a $64 \times 64$ synthetic picture.
Figure 13: Reconstruction after adding 7% "Salt and Pepper" noise to a 64 x 64 synthetic picture.
References


CIS 0001  
Guy Foux, Michael Heymann, and Alfred M. Bruckstein: “Two-Dimensional Polygon Navigation Among Unknown Stationary Polygonal Obstacles”.  

CIS 0002  

CIS 0003  

CIS 0004  

CIS 0005  

CIS 0006  

CIS 0007  

CIS 0008  

CIS 0009  

CIS 0010  

ANNUAL REPORT, 1990.  

CIS 0011  
CIS 9012

CIS 9101

CIS 9102

CIS 9103

CIS 9104

CIS 9105

CIS 9106

CIS 9107

CIS 9108

CIS 9109

CIS 9110

CIS 9111
CIS 9201

CIS 9202

CIS 9202 (Revised, June 1992)

CIS 9203

CIS 9204

CIS 9205

CIS 9206

CIS 9207

CIS 9208

CIS 9209