Direct Method For Reconstructing Shape From Shading

John Oliensis
Paul Dupuis

COINS TR92-32

May, 1992

Direct Method For Reconstructing Shape From Shading[1]

John Oliensis†
Department of Computer Science
University of Massachusetts at Amherst
Amherst, Massachusetts 01003
and
Paul Dupuis‡
Box F, Division of Applied Mathematics
Brown University
Providence, Rhode Island 02912

Abstract

A new approach to shape from shading is described, based on a connection with a calculus of variations/optimal control problem. An explicit representation is given for the surface corresponding to a shaded image; uniqueness of the surface (under suitable conditions) is an immediate consequence. The approach leads naturally to an algorithm for shape reconstruction that is simple, fast, provably convergent, and, in many cases, provably convergent to the correct solution. In contrast with standard variational algorithms, it does not require regularization. Given a continuous image, the algorithm can be proven to converge to the continuous surface solution as the image sampling frequency is taken to infinity. Experimental results are presented for synthetic and real images, for general lighting direction.

[1] This work was supported by the National Science Foundation under grants IRI-9113690, CDA-8922572†and NSF-DMS-9115762‡, and by a grant from DARPA, via TACOM, contract number DAAE07-91-C-R035†.

1. Introduction

Shape from shading has traditionally been considered an ill–posed problem, with potentially infinitely many different surfaces corresponding to a shaded image. Therefore, most algorithms for reconstructing shape have incorporated *regularization* techniques to guarantee recovery of a unique, 'physically reasonable' surface solution.

More recently, it was suggested that shape from shading need not be ill–posed when the image contains singular points, i.e., maximally bright image points [1,5,18,19,16,15]. This was shown for the case of illumination from—or symmetric around—the camera direction in [16]. In addition, a general shaded image was shown to *uniquely* determine shape under the assumed lighting conditions [16]. Singular points provided the essential constraints.

Singular points continue to give strong constraints on the surface solutions for illumination from a general direction [15]. Thus, shape from shading should not be assumed ill–posed in general, and regularization should be used with caution. Also, the image of the occluding boundary gives no useful constraint on surface reconstruction [15]. Singular points, therefore, provide the primary constraints.

Nevertheless, shape–from–shading algorithms in the past have not taken full advantage of the strong constraints due to singular points. Algorithms based on the method of characteristic strips [4] have used these constraints explicitly, but in an approximate way. These algorithms have usually been applied to rather simple images, and are nonrobust in the presence of noise.

Most recent algorithms for recovering shape from shading have been based on the variational approach (e.g., [7,6,5]). These algorithms have had significant successes on complex images, but do not explicitly use the singular point constraints. This is seen experimentally in the fact that these algorithms do better on images with many singular points than on images with just one (see below and also [11]); yet for such simple images, the sole singular point is known to directly and uniquely constrain the surface reconstruction [1,19].

In this paper, an algorithm is presented that takes full advantage of the singular point constraints. It is simple, fast, provably convergent, and, in many cases, provably convergent to the correct solution. In particular, if the surface is known to be *unimodal* at a singular point in the image (i.e., locally concave or convex at this point), then the algorithm provably reconstructs the correct surface in a region around the singular point. The algorithm is robust against noise and, unlike previous algorithms, does not employ regularization. There is no problem with false minima, in contrast to the standard variational approach. Finally, this approach is capable of dealing with some orientation discontinuities—images for which the intensity function is only piecewise continuous.

The algorithm is based on establishing the equivalence of shape from shading to a calculus of variations/optimal control problem. For the general case with illumination from an arbitrary direction, the optimal control problem can be extended to a differential game. In general, a variety of optimal control/differential game formulations is possible [20,2].

This equivalence facilitates the theoretical analysis of shape from shading, and makes the algorithm highly adaptable. It also gives intuition about the convergence performance of the algorithm. Below, we present a simple uniqueness proof for shape from shading which generalizes from the local uniqueness results of Bruss [1] and Saxberg [19]. This is possible because, in the optimal control representation, an expression for the surface corresponding to a shaded image can be exhibited explicitly. Some of the results presented in this paper have also been derived by E. Rouy and A. Tourin [20].

2. Shape from Shading as a Problem of Optimal Control: Heuristic Derivation

The imaged surface is assumed to be Lambertian, and viewed from above along the $-\hat{z}$ direction. It is represented in the explicit form $z(x,y)$, where $z : \mathbf{R}^2 \to \mathbf{R}$ is the *height function* to be reconstructed. We consider first the simpler case of illumination along the viewing direction $-\hat{z}$ (vertical light). The case of illumination from a general direction is discussed later.

Under these conditions, the image irradiance equation is:

$$I(x,y) = \frac{1}{(1 + |\nabla z(x,y)|^2)^{1/2}}. \tag{2.1}$$

It is convenient to rewrite this in the eikonal form:

$$|\nabla z(x,y)|^2 = \frac{1}{(I(x,y))^2} - 1 \equiv V(x,y), \tag{2.2}$$

where $I(x,y) \in (0,\ 1]$, $V(x,y) \in [0,\ \infty)$. This type of equation arises frequently in the dynamical programming approach to problems of optimal control. In this section, the connection of shape–

from–shading to an optimal control problem is derived heuristically using dynamic programming, in the simplified situation where the image contains a single singular point. Note that this derivation does not generalize directly to the multiple singular point case. A rigorous argument which does apply to this more general situation is presented below in Section 5.

We show that the height function z has a representation as the solution to a calculus of variations/optimal control problem, and that this representation gives a solution of eq. 2.2. The optimal control representation is more specific than the partial differential eq. 2.2, which in general has additional (even classical) solutions. Other possible representations for the height function z could also be considered [20, 2]; the one described below is chosen for its algebraic simplicity. In Section 4, we discuss a modification of this calculus of variations problem by the inclusion of a terminal cost, as is necessary for the multiple singular point case.

Consider the following control problem: a 'particle' initially located at $(x_0,\ y_0)$ moves in the image plane in response to control parameters u, v, according to:

$$\dot{x} = u, \qquad \dot{y} = v, \qquad x(0) = x_0, \qquad y(0) = y_0. \tag{2.3}$$

The control parameters are to be chosen to minimize a cost function for the particle's trajectory $(x(s),\ y(s))$:

$$\begin{aligned} &U(x_0, y_0, T) = \inf\{ \\ &\tfrac{1}{2}\textstyle\int_0^T ds\,(u(s)^2 + v(s)^2 + V(x(s), y(s)))\}. \end{aligned} \tag{2.4}$$

In this equation, the minimal cost has been defined as a function of the trajectory's starting point $(x_0,\ y_0)$. The infimization is over all piecewise continuous functions $u(\cdot)$, $v(\cdot)$ on $[0,\ T]$. Let

$$U(x, y) = \lim_{T\to\infty} U(x, y, T).$$

$U(\cdot)$ will turn out, in the unimodal case, to be the surface $z(x, y)$ up to a translation. A *unimodal* surface is one with a single local maximum or local minimum. To show this formally, we assume that $U(\cdot)$ is a differentiable function of the starting point, and formally demonstrate using a dynamical programming argument that it satisfies eq. 2.2.

Let δT be a small time increment. Then the principle of optimality implies

$$\begin{aligned} & U(x_0, y_0, T) = \inf_{(u,v)}\{ \\ & \quad U(x(\delta T), y(\delta T), T - \delta T) \\ & \quad + \tfrac{1}{2}\int_0^{\delta T} ds\,(u(s)^2 + v(s)^2 + V(x(s), y(s)))\}. \end{aligned} \tag{2.5}$$

The explicit infimization is now over the part of the trajectory with $s \in [0,\ \delta T]$; the infimization over the rest of the trajectory is included in the cost function $U(\cdot, T - \delta T)$. Since δT is small, U can be expanded to first order in this quantity, which gives (for $\delta T \to 0$):

$$\begin{aligned} & \tfrac{\partial U}{\partial T}(x, y, T) = \inf_{(u(0),v(0))}\{ \\ & \quad \tfrac{1}{2}(u^2(0) + v^2(0) + V(x, y)) \\ & \quad + \tfrac{\partial U}{\partial x}(x, y, T)u(0) + \tfrac{\partial U}{\partial y}(x, y, T)v(0)\}. \end{aligned} \tag{2.6}$$

Performing the minimization over $u(0)$ and $v(0)$ yields:

$$u(0) = -\frac{\partial U}{\partial x}(x, y, T), \qquad v(0) = -\frac{\partial U}{\partial y}(x, y, T), \tag{2.7}$$

and

$$\begin{aligned} & \tfrac{\partial U}{\partial T}(x, y, T) = \\ & \tfrac{1}{2}\left[V(x, y) - \left(\tfrac{\partial U}{\partial x}(x, y, T)\right)^2 - \left(\tfrac{\partial U}{\partial y}(x, y, T)\right)^2\right]. \end{aligned} \tag{2.8}$$

Suppose that the image region under consideration is a small neighborhood of a singular point, at which $I = 1$ and $V = 0$. A minimal cost trajectory clearly moves toward regions of smaller V, and will converge to the singular point at which the incremental cost is zero. As the trajectory converges to this point, the total cost along the trajectory converges to a *finite* value. Therefore, the integration limit T in eq. 2.4 can be taken to infinity, and $U(x, y)$ is well defined. Since the time derivative vanishes, $U(x, y)$ satisfies:

$$\left(\frac{\partial U(x, y)}{\partial x}\right)^2 + \left(\frac{\partial U(x, y)}{\partial y}\right)^2 = V(x, y). \tag{2.9}$$

Since eq. 2.9 is just the image irradiance equation, eq. 2.2, it suggests that $U(\cdot)$ *can be identified with* $z(x, y)$. Also, u and v can be identified with $-p$ and $-q$, respectively, from eq. 2.7, where $(p,\ q) \equiv \nabla z$. Thus the minimal cost trajectories are curves of steepest descent, and are just the *characteristic strips* [4,15].

Note also that $U \geq 0$, and that $U = 0$ only at the singular point. Thus, the solution to the image irradiance equation that is locally concave at the singular point has been automatically selected by this formulation; the solution that is locally convex at the singular point is just its negative. The function U is unique, since it is equal to the infimum of the cost in the optimal control problem, which must be unique. Since an infimum of the cost always exists, the function U always exists. It must be continuous, but need not be differentiable.

This formulation also gives a way of computing U. Clearly, $U(x, y, 0) = 0$ for all $(x,\ y)$, while $U(x, y, T)$ is monotonically increasing in time: extending a trajectory cannot result in a reduced cost. Therefore, by solving eq. 2.8 iteratively in time, with initial condition $U(\cdot, T) = 0$ at $T = 0$, a sequence of functions $U(x, y, T)$ is obtained which at every point converges monotonically upward to $z(x, y)$ as $T \rightarrow \infty$. Because this convergence is pointwise monotonic, it is clearly stable.

For the actual implementation, an iterative procedure is used that is justified by its exact relation to a discretized control problem [8]. The continuous image plane is replaced by an image discretized into pixels, and the trajectory described by eq. 2.3 is approximated by a Markov process. This is described in detail in the next section. It can be shown that this gives a discrete approximation U^h to U, which converges to the continuous U as the spatial grid size h approaches zero [9,13]. However, a naive discretization of eq. 2.8 does *not* necessarily give a stably convergent algorithm.

3. Algorithm Description

A more detailed description of the algorithm and its derivation is now presented. We consider a control problem defined on the discrete grid of pixels and chosen to approximate the continuous calculus of variations problem described above. h is the pixel spacing. For the discrete case, a 'particle' trajectory is a sequence of discrete jumps between grid sites—a poor approximation to a continuous trajectory. In order to better approximate a continuous trajectory on a discrete grid, an element of randomness is introduced.

The control problem is as follows: a 'particle', with initial image plane location $\phi_0 \equiv (i_0, j_0)h$, jumps between neighboring pixel sites in response to control parameters $\mathbf{C} \equiv (u(k),\ v(k))$, where k indicates the time step. (A 4–neighborhood is assumed.) The jumps are probabilistic, but it is required that on average

$$\langle \phi(k+1) - \phi(k) \rangle = \Delta t \mathbf{C}(k), \tag{3.10}$$

in analogy with eq. 2.3. Here Δt is the time increment from time step k to step $k+1$, and $< >$ denotes the noise average. Let $\eta(k)$ be the random vector representing the jump at time k: $\eta(k) \equiv \phi(k+1) - \phi(k)$. The jump probabilities are assigned as follows. When $u = v = 0$, $\mathbf{P}\,(\eta = \mathbf{0}) = 1$, with all other probabilities zero. In this case, Δt is arbitrarily chosen to be 1. Otherwise,

$$\begin{aligned} &\mathbf{P}\,(\eta(k) = h(\mathrm{sgn}(u), 0) \mid C(k) = (u,v), (C(i), \eta(i)), i < k) \\ &\qquad = \Delta t|u|/h = |u|/(|u| + |v|) \\ &\mathbf{P}\,(\eta(k) = h(0, \mathrm{sgn}(v)) \mid C(k) = (u,v), (C(i), \eta(i)), i < k) \\ &\qquad = \Delta t|v|/h = |v|/(|u| + |v|), \end{aligned} \tag{3.11}$$

again with all other probabilities zero. Eq. 3.11 implies that the particle jumps by one lattice site at each iteration (for a nonzero control), and therefore moves on the lattice with maximum speed, causing the algorithm to converge quickly. This has been achieved by taking Δt to depend explicitly on the controls as $\Delta t = h/(|u| + |v|)$. It is clear that eq. 3.11 implies eq. 3.10.

The analog to the calculus of variations problem is as follows: choose the control parameters to minimize the *expected* cost for the discrete trajectories:

$$\begin{aligned} &U^h(\phi_0, K) = \inf\{ \\ &\left\langle \tfrac{1}{2} \textstyle\sum_{k=0}^{K} \Delta t\,(u(k), v(k))\,(u(k)^2 + v(k)^2 + V(\phi(k))) \right\rangle\}, \end{aligned} \tag{3.12}$$

where the infimization is over all nonanticapative control sequences $\{(u(k), v(k)), k = 0, \ldots K\}$ (i.e., controls which do not depend on the future history of the particle) [8]. It can be shown the value function U^h for this discrete control problem converges to the continuous value function as the grid spacing is taken to zero [9, 13].

A dynamical programming equation can be derived for this control problem as in the previous section:

$$\begin{aligned} &U^h(\phi_0, K) = \inf_{(u(0),v(0))}\{ \\ &\tfrac{1}{2}\Delta t(u(0), v(0))\,(u(0)^2 + v(0)^2 + V(\phi_0)) \\ &+\left\langle U^h(\phi_0 + \eta, K-1)\right\rangle\}. \end{aligned} \tag{3.13}$$

The expectation in this equation is easily calculated from eq. 3.11; for nonzero controls it is

$$\frac{|u|U^h\,(\phi_0 + h(\mathrm{sgn}(u), 0)) + |v|U^h\,(\phi_0 + h(0, \mathrm{sgn}(v)))}{|u| + |v|}.$$

Performing the minimization in eq. 3.13 is slightly complicated since the cases with **C** in different quadrants must be treated separately. Eventually, the following simple algorithm is obtained. Define

$$U^h_{N1} = \mathrm{Min}(U^h(\phi \pm (1,0)h),$$

$$U^h_{N2} = \mathrm{Min}(U^h(\phi \pm (0,1)h),$$

and let $D_N \equiv U^h_{N2} - U^h_{N1}$. The update equation is

$$\hat{U}^h(\phi, K+1) = \begin{cases} \frac{1}{2}\left((2h^2V - D_N^2)^{1/2} + U^h_{N2} + U^h_{N1}\right) \\ \qquad \text{if } \ h^2V(\phi) > D_N^2 \\ \\ h|V|^{1/2}(\phi) + \mathrm{Min}_i(U^h_{Ni}) \\ \qquad \text{otherwise.} \end{cases} \tag{3.14}$$

The lower case corresponds to the minimum in eq. 3.13 being realized on one of the axes in the u–v plane (with the origin excluded); the upper corresponds to an off–axis minimum. As in the previous section, the initial value for $U^h(\phi, \cdot)$ can be taken as 0. Since the expected cost for an optimal trajectory cannot decrease with time, an iterative solution $U^h(\phi, K)$ to the above equation increases monotonically at every point. This algorithm is more efficient than the one previously reported in [14] because the time increment Δt is adjusted optimally as a function of the controls. In fact, one can show in the case of a single singular point that the iterative scheme described above is a contraction, and thus any initial condition can be used. We note, however, that in general it is better to use a large initial condition. Such an initial condition is necessary in the setup of the next section, which deals with the case of multiple singular points.

To avoid indeterminacy, it is necessary to impose the boundary condition that no trajectory exits the image, as is easily done [14]; the significance of this is discussed below. Then, assuming that there are singular points in the image where $V = 0$, as $k \to \infty$ all optimal trajectories must converge to the singular points. Thus $U^h(\phi, K)$ converges monotonically as $K \to \infty$ to a solution $z = U^h(\phi)$; in fact, convergence occurs in a finite number of iterations [13]. This solution satisfies a discretized version of the shape–from–shading equation [14], and is always nonnegative, since the summand in eq. 3.12 is. Also, $z = 0$ at a singular point: a trajectory beginning at a singular point achieves minimal cost by remaining there, since $V = 0$ at the point. Thus, z attains a local minimum at a singular point. Also, as in eq. 2.7, the expected optimal trajectories are approximate curves of steepest descent [14].

The algorithm described in this section is appropriate for *unimodal* images—images containing just one singular point where the height has either a local minimum or maximum. For these images, the iterative solution of eq. 3.14 will correctly reconstruct the original surface at all points where this surface is theoretically determined [15]—that is, at all points connected by a steepest descent curve on the original surface to the singular point. Such points "learn" their height from the singular point. In contrast, at other image points the surface reconstruction can be ambiguous [15]. These ambiguous points lie on steepest descent curves that exit the image rather than terminating at the singular point. Imposing the boundary condition as above that no trajectory exits the image only affects the surface reconstruction at these ambiguous points. Our algorithm does not necessarily reproduce the original surface at ambiguous points. A modified algorithm appropriate for the multimodal case (many singular points, or even singular sets) is described in the next section.

4. Modifications of the Algorithm

An important modification introduces a *terminal cost* term into the cost function. This gives an algorithm capable of dealing with multimodal images. Including this term, the minimal cost is (compare eq. 3.12):

$$U^h(\phi_0, K) = \inf_{(u,v)} \Big\langle g(\phi(K)) + \tfrac{1}{2} \times \\ \textstyle\sum_{k=0}^{K-1} \Delta t\,(u(k), v(k))\,(u(k)^2 + v(k)^2 + V(\phi(k)))\Big\rangle . \tag{4.15}$$

The terminal cost, $g(\phi(K))$, introduces a penalty term for a trajectory stopping at the position $\phi(K)$. It causes an optimal trajectory to not remain in regions of high terminal cost, and converge instead to points of low terminal cost. This can dramatically improve the convergence speed of the algorithm even in the case of a single singular point. A high terminal cost is necessary for the multimodal setup with many singular points, where it can be used to distinguish between singular points of different type (concave, convex, saddle). In the final surface solution, only a concave–type singular point should be the terminus for optimal trajectories, since these are descending curves. By placing a high terminal cost at other singular points, trajectories can be prohibited from terminating at these points. Then the surface solution will only be "learned" from the concave singular points. Also, if the heights of the concave singular points are known, e.g., using stereo, then this can be specified in the algorithm by setting the terminal costs at these points equal to their heights. Since the singular points are distinctive, it is likely that their heights, and the local nature of the surface, can be determined easily from stereo.

The algorithm can also be adapted easily to the case in which the heights are known at the maximum singular points. We are investigating the possibility of using preliminary, incorrect reconstructions as the basis for determining the correct relative heights of two local minima whose "domains of attraction" touch.

The dynamical programming equation corresponding to the cost in eq. 4.15 is exactly the same as eq. 3.14, as is easily seen. The algorithm differs only in the initial condition for U^h: clearly, U^h should be set initially to $g(\phi)$, not 0 as before. Thus, from the optimal control viewpoint, the choice of initial values for U^h in the algorithm has a concrete and intuitive interpretation.

In another modification, the minimizing controller is allowed to terminate the motion of the particle at any time, and pay the terminal cost corresponding to its stopping point. Since the controller prefers to halt the motion rather that permit an increased cost, the cost cannot increase over time. Thus, this results in an algorithm that converges monotonically. The dynamical programming equation is similar to the previous one: the updated value for $U^h(\phi)$ should now be taken as

$$\mathrm{Min}\left(\hat{U}^h(\phi), g(\phi)\right),$$

where $g(\phi)$ is the terminal cost, and $\hat{U}^h(\phi)$ is as in eq. 3.14. The extra minimization in this updating equation accounts for the possibility that the particle stops after zero iterations, with the entire cost given by the terminal cost.

The algorithm described above is of the Jacobi type, with the surface updated everywhere in parallel at each iteration. It can be shown that the algorithm also converges if implemented via Gauss–Seidel, with updated surface estimates used as soon as they are available [13]. Our experiments show that this produces a significant speedup.

5. Proof of Equivalence

In this section we will assume the situation of ***vertical light***, as described in Section 2. For this case (and under suitable assumptions) the height function has a representation in terms of an associated calculus of variations problem. For the general case of ***oblique light*** there is a representation in terms of a differential game [13].

The data available for the determination of the function $z(\cdot)$ is encoded in the intensity function $I(x,y)$ determined by eq. 2.1. I is well defined at all points (x,y) where $z(\cdot)$ is differentiable. We will always assume that the function $I(\cdot)$ is defined on a bounded open set of the form $G = \cap_{i=1}^{N} G_i$, $N < \infty$, where each G_i has a C^1 boundary ∂G_i. Let $\hat{n}_i(x,y)$ denote the inward normal to G_i at $(x,y) \in \partial G_i$. First consider the following situation.

Assumption 5.1 *1. $z(\cdot)$ is C^1 on $\overline{G}$.*

2. There is exactly one point $(\bar{x},\bar{y})$ such that $\nabla z(\bar{x},\bar{y}) = 0$.

3. $(\bar{x},\bar{y})$ is a local minimum.

4. $\nabla z(x,y) \cdot \hat{n}_i(x,y) < 0$ whenever $(x,y) \in \partial G \cap \partial G_i$.

(4) implies that the steepest descent direction is always inward on the boundary. We next define a calculus of variations problem. Fix $(x,y) \in \overline{G}$, and set

$$U(x,y) = \inf \int_0^{\tau} L(\phi(s), \dot{\phi}(s))ds. \tag{5.16}$$

Here $\tau = \inf\{t : \phi(t) = (\bar{x},\bar{y})\}$, and the infimum is over all piecewise continuously differentiable paths $\phi : [0,\infty) \to \overline{G}$ that satisfy $\phi(0) = (x,y)$. The variational integrand $L(\cdot)$ is given by

$$\begin{aligned} L((x,y),(u,v)) &= \frac{1}{2}\left(u^2+v^2\right) + \frac{1}{2}\left(\frac{1}{I(x,y)^2} - 1\right) \\ &= \frac{1}{2}\left(u^2+v^2\right) + \frac{1}{2}|\nabla z(x,y)|^2. \end{aligned}$$

We follow the usual convention of defining $\inf \emptyset = +\infty$. Thus if $\phi(t) \neq (\bar{x},\bar{y})$ for all t, then $\tau = +\infty$.

Theorem 5.2 *Under the conditions of Assumption 5.1 we have*

$$z(x,y) - z(\bar{x},\bar{y}) = U(x,y).$$

Proof. Let $\phi(\cdot)$ be any piecewise continuously differentiable path that starts as (x,y). For all $\varepsilon \geq 0$ define

$$\tau^\varepsilon = \inf\{t : |\phi(t) - (\bar{x},\bar{y})| \leq \varepsilon\}.$$

Fix $\delta > 0$ and choose $\varepsilon > 0$ such that

$$z(x,y) \leq z(\bar{x},\bar{y}) + \delta$$

for $|(x,y)-(\bar{x},\bar{y})| \leq \varepsilon$.

To prove $z(x,y) - z(\bar{x},\bar{y}) \leq U(x,y)$, we consider two cases. First assume $\tau^\varepsilon = +\infty$. By Assumption 5.1 there exists $c > 0$ such that

$$L((x,y),(u,v)) \geq c$$

for all (x,y) satisfying $|(x,y)-(\bar{x},\bar{y})| \geq \varepsilon$ and all $(u,v) \in \mathbb{R}^2$. Thus, in such a case

$$\int_0^\tau L(\phi(s),\dot{\phi}(s))ds = \int_0^{\tau^\varepsilon} L(\phi(s),\dot{\phi}(s))ds = +\infty.$$

Next assume $\tau^\varepsilon < \infty$. By the chain rule,

$$\frac{d}{dt}\left[-z(\phi(t))\right] = -\nabla z(\phi(t)) \cdot \dot{\phi}(t) \leq \frac{1}{2}|\dot{\phi}(t)|^2 + \frac{1}{2}|\nabla z(\phi(t))|^2$$

almost surely in t. Therefore

$$\begin{aligned} z(x,y) - z(\tilde{x},\tilde{y}) &\leq z(x,y) - z(\phi(\tau^{\epsilon})) + \delta \\ &= -[z(\phi(\tau^{\epsilon})) - z(\phi(0))] + \delta \\ &= \int_0^{\tau^{\epsilon}} -\nabla z(\phi(t)) \cdot \dot{\phi}(t) dt + \delta \\ &\leq \int_0^{\tau^{\epsilon}} L(\phi(t), \dot{\phi}(t)) dt + \delta \\ &\leq \int_0^{\tau} L(\phi(t), \dot{\phi}(t)) dt + \delta. \end{aligned}$$

Sending $\delta \to 0$ we obtain $z(x,y) - z(\tilde{x},\tilde{y}) \leq U(x,y)$.

To prove $z(x,y) - z(\tilde{x},\tilde{y}) \geq U(x,y)$, let $\phi(\cdot)$ be a solution (note that there may not be uniqueness since z is only assumed C^1) to the equation

$$\dot{\phi}(t) = -\nabla z(\phi(t)), \ \phi(0) = (x,y).$$

By Assumption 5.1 $\phi(\cdot)$ never touches ∂G for $t > 0$, and therefore the solution is well defined for all $t > 0$. Let $\tau = \inf\{t : \phi(t) = (\tilde{x},\tilde{y})\}$ and let $a \wedge b$ denote the smaller of a and b. For any $t < \infty$, we have

$$\begin{aligned} z(x,y) - z(\tilde{x},\tilde{y}) &\geq z(\phi(0)) - z(\phi(t \wedge \tau)) \\ &= -[z(\phi(t \wedge \tau)) - z(\phi(0))] \\ &= \int_0^{t\wedge\tau} -\nabla z(\phi(s)) \cdot \dot{\phi}(s) ds \\ &= \int_0^{t\wedge\tau} |\nabla z(\phi(s))|^2 ds \\ &= \int_0^{t\wedge\tau} L(\phi(s), \dot{\phi}(s)) ds. \end{aligned}$$

Sending $t \to \infty$ we conclude that

$$z(x,y) - z(\tilde{x},\tilde{y}) \geq \int_0^{\tau} L(\phi(s), \dot{\phi}(s)) ds \geq U(x,y).$$

■

The solution to the calculus of variations problem uniquely identifies the height function up to an overall translation in z. This ambiguity can be removed by specifying $z(\bar{x},\bar{y})$.

We next consider a more general situation involving more than one stationary point. Let M be the set of local minima of $z(\cdot)$.

Assumption 5.3 *1. $z(\cdot)$ is C^1 on $\overline{G}$.*

2. The value of $z(\cdot)$ is known on M.

3. $\nabla z(x,y)\cdot\hat{n}_i(x,y) < 0$ whenever $(x,y)\in\partial G\cap\partial G_i$.

Define the terminal cost function

$$g(x,y) = \begin{cases} z(x,y) & \text{if } (x,y)\in M, \\ +\infty & \text{otherwise.} \end{cases}$$

Consider the calculus of variations problem

$$U(x,y) = \inf\left[\int_0^\tau L(\phi(s),\dot{\phi}(s))ds + g(\phi(\tau))\right]. \tag{5.17}$$

Here the infimum is over all $\tau<\infty$ and absolutely continuous paths $\phi:[0,\tau]\to\overline{G}$ that satisfy $\phi(0)=(x,y)$. Unlike the case of a single stationary point, it is necessary that a terminal cost be included in order to guarantee that trajectories do not get "stuck" at stationary points that are not local minima.

We have the following result for this case.

Theorem 5.4 *Under the conditions of Assumption 5.3 we have*

$$z(x,y) = U(x,y).$$

Remarks on the proof. The proof is very similar to that of Theorem 5.2 and will only be sketched. Consider any path $\phi(\cdot)$ which starts at (x,y) and for which the cost

$$\int_0^\tau L(\phi(s),\dot{\phi}(s))ds + g(\phi(\tau)) \tag{5.18}$$

is finite. Boundedness of the cost implies $\phi(\tau)\in M$. Suppose $\phi(\tau)=(\tilde{x},\tilde{y})$. The proof of Theorem 5.2 then shows that $z(x,y)-z(\tilde{x},\tilde{y})\le\int_0^\tau L(\phi(s),\dot{\phi}(s))ds$. Together with the definition of $g(\cdot)$ this implies $z(x,y)\le U(x,y)$.

Next consider the reverse inequality. As in the proof of Theorem 5.2, we would like to construct a particular path $\phi(\cdot)$ that starts at (x,y) so that the cost (5.18) is arbitrarily close to $z(x,y)$. We first note that by a perturbation argument [15,12,17] we can assume that there are at most finitely many points such that $\nabla z(x,y) = 0$. It can be shown that there exists a dense subset D of $\overline{G}$ with the property that whenever the path $\phi(\cdot)$ satisfies

$$\dot{\phi}(t) = -\nabla z(\phi(t)), \phi(0) \in D,$$

then $\phi(t)$ converges to a local minimum $(\tilde{x},\tilde{y})$ of $z(\cdot)$ as $t \to \infty$. Using the argument of Theorem 5.2 and the fact that $z(\phi(t))$ is nondecreasing we conclude $z(x,y) \geq U(x,y)$ for $(x,y) \in D$. By continuity of both $z(\cdot)$ and $U(\cdot)$ (which is easy to prove) we have $z(x,y) \geq U(x,y)$ for $(x,y) \in \overline{G}$. ■

Previous uniqueness proofs [1,19,16] assumed that $z(x,y)$ was at least C^2; here z is only assumed C^1. A fortiori, no conditions are placed on the second derivatives of the intensity; in particular, the singular points are not required to be "good" or "nondegenerate" [19,16].

6. Illumination from a General Direction

For a Lambertian surface, the image irradiance equation for the intensity is:

$$I(x,y) = \hat{L} \cdot \frac{(-z_x, -z_y, 1)}{(1 + z_x^2 + z_y^2)^{1/2}},$$

where $\hat{L}$ is a unit vector giving the light source direction, and z_x, z_y are partial derivatives of the height. For simplicity and w.l.o.g., we take the x–component of $\hat{L}$ to be zero. After some algebra, this equation may be rewritten as:

$$I^2 z_x^2 + J z_y^2 + 2L_z L_y z_y + (I^2 - L_z^2) = 0,$$

with $J(x,y) \equiv I^2(x,y) - L_y^2$.

Define a new variable $\xi \equiv (x,y,z) \cdot \hat{L}$ measuring the "height" along the light direction rather than the viewer direction $\hat{z}$. This is done so that the local cost at singular points will be zero, like V previously, causing optimal trajectories to terminate at these points. Then

$$\xi_x = L_z z_x, \qquad \xi_y = L_y + L_z z_y.$$

Substituting in the previous equation yields

$$I^2\xi_x^2 + J\xi_y^2 + 2(1-I^2)L_y\xi_y - (1-I^2) = 0.$$

J, the coefficient of ξ_y^2, is positive in an image region B that includes the singular points. When $I^2 = L_y^2$, the angle between the surface normal and $\hat{L}$ is large enough that it may correspond to a point on the occluding boundary.

In the image region B, we consider a control problem analogous to that of Section 2: a 'particle' initially located at $(x,\ y)$ is controlled using the parameters u, v:

$$\dot{x} = I^2(x,y)u, \qquad \dot{y} = J(x,y)v - (1 - I^2(x,y))L_y.$$

u, v are chosen to infimize a cost function for the particle's trajectory

$$\begin{aligned} &U(x,y,T) = \inf_{(u,v)} \\ &\tfrac{1}{2}\textstyle\int_0^T ds(I^2(x,y)u^2 + J(x,y)v^2 + (1 - I^2(x,y))). \end{aligned} \tag{6.19}$$

As before the integrand is nonnegative throughout the region B, and the local cost $1-I^2$ vanishes at singular points. Eq. 2.4 for the vertical–light case $L_y = 0$ can be recovered by dividing eq. 6.19 by I^2.

This control problem is essentially equivalent to the one previously considered, and results similar to those of the previous sections are easily obtainable. In particular, by a Schwarz inequality argument as in Section 5,

$$\begin{aligned} \frac{d}{dt}\left[-\xi(\phi(t))\right] &= -\nabla\xi(\phi(t))\cdot\dot{\phi}(t) \\ &= -\xi_x I^2 u - \xi_y\left(Jv - (1-I^2)L_y\right) \\ &\le \frac{1}{2}(I^2u^2 + Jv^2 + \\ &\quad I^2\xi_x^2 + J\xi_y^2 + 2\xi_y(1-I^2)L_y) \\ &= \frac{1}{2}\left(I^2u^2 + Jv^2 + 1 - I^2\right), \end{aligned}$$

which is just the integrand of eq. 6.19. This gives the necessary generalization for the rigorous proof of equivalence. Similarly, an algorithm can be defined in the same way as before, and will recover thè correct solution near concave (or convex) singular points.

In the image region where $I^2 - L_y^2 < 0$, the optimal control representation of the problem no longer suffices. Instead, there is a representation in terms of a *differential game* (see e.g. [3]). However, it is a particularly simple one, in which the opposing controllers effectively direct the 'particle' motion in orthogonal directions, and where the cost also splits into a sum of terms depending on the different control parameters. Thus, the Isaacs condition and the existence of a "value" follow.

The 'particle' dynamics for the differential game are:

$$\dot{x} = I^2 u, \qquad \dot{y} = J\left(\theta(J)v_1 + \theta(-J)v_2\right) - (1 - I^2)L_y,$$

where

$$\theta(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases}$$

When $I^2 - L_y^2 < 0$, a restriction on the control direction can be imposed:

$$L_y \hat{C}_y \leq -\text{sgn}(L_y)\sqrt{L_y^2 - I^2},$$

where $\hat{C}$ is the unit vector in the $(\dot{x}, \dot{y})$ direction. This follows from the requirement that the surface is visible, i.e. that the height function $z(x, y)$ is a graph.

The player associated with u and v_1 seeks to minimize the value function of the game, while the v_2 player seeks to maximize it. The value that opposing players attempt to control is

$$U(\phi, t) \equiv \tfrac{1}{2} \times$$
$$\int_0^T ds(I^2 u^2 + J(\theta(J)v_1^2 + \theta(-J)v_2^2) + (1 - I^2)).$$

A precise description of the differential game is somewhat technical (see e.g. [3]). Here we simply note that the properly defined value gives the height function (under suitable conditions), and that an algorithm on a discrete grid for approximating this value function can be derived that is similar to the vertical–light algorithm.

The algorithm is as follows: define $V \equiv (1 - I^2)/J$, generalizing from the vertical light case. Also, define

$$\xi_{N2}^h \equiv \begin{cases} \text{Min}(h^{-1}\xi(\phi \pm h(0,1)) \pm V L_y) & \text{if } J > 0 \\ \\ h^{-1}\xi(\phi - \text{sgn}(L_y)h(0,1)) - V|L_y| & \text{if } J < 0, \end{cases}$$

and

$$\xi^h_{N1} = \mathrm{Min}(h^{-1}\xi(\phi \pm h(1,0))), \qquad D_N \equiv \xi^h_{N2} - \xi^h_{N1},$$

where D_N is again the natural generalization from the vertical light case. Finally, define

$$S = \frac{I|J|^{1/2}}{I^2+J}\left|VL_z^2(I^2+J)/J - D_N^2\right|^{1/2}.$$

For $J > 0$, the update equation is:

$$\hat{\xi}(\phi) = h\begin{cases} \xi^h_{N1} + |V|^{1/2}L_z \\ \quad \text{if } D_N > |V|^{1/2}L_z \\ \\ S + \frac{I^2\xi^h_{N1}+J\xi^h_{N2}}{I^2+J} \\ \quad \text{if } -|V|^{1/2}L_z(I/J^{1/2}) \leq D_N \leq |V|^{1/2}L_z \\ \\ \xi^h_{N2} + |V|^{1/2}L_z(I/J^{1/2}) \\ \quad \text{if } D_N < -|V|^{1/2}L_z(I/J^{1/2}). \end{cases}$$

For $J < 0$, the update equation is:

$$\hat{\xi}(\phi) = h\begin{cases} \xi^h_{N2} - |V|^{1/2}L_z(I/|J|^{1/2}) \\ \quad \text{if } D_N < |V|^{1/2}L_z(I/|J|^{1/2}) \\ \\ S + \frac{I^2\xi^h_{N1}+J\xi^h_{N2}}{I^2+J} \\ \quad \text{otherwise.} \end{cases}$$

This algorithm clearly reduces to the previous one for vertical light when $L_y = 0$, $J = I^2$, $L_z = 1$. Although it is apparently singular at $J = 0$ or $I^2 + J = 0$, the correct update can be shown to depend continuously on these quantities. If necessary, to avoid artificial numerical instabilities while maintaining the algorithm's speed, the original image intensity can be perturbed slightly to avoid such zeros. For greater speed per iteration, most of the intensity–dependent terms in the above algorithm can be precomputed. Then, at each lattice site, the algorithm requires in each iteration at most a single square root operation, and five multiplications.

This algorithm, as in the vertical light case, can be modified to give monotonic convergence. Instead of $\hat{\xi}(\phi)$, take the updated value for ξ as

$$\mathrm{Min}\left(\hat{\xi}(\phi), g(\phi)\right),$$

where g is the terminal cost, i.e. the (large) initial value for $\xi(\phi)$. As before, this extra minimization corresponds in the differential game to allowing the minimizing player to halt the particle trajectory at any time. The minimizer will stop the motion rather than pay an increased cost for a longer trajectory, and therefore additional iterations cannot increase the value of $\xi(\phi)$. This algorithm also converges if implemented via Gauss–Seidel [13].

7. Experiments

Figure 1 displays a 32 by 32 surface parabolic surface which is assumed to be imaged from above. The image has one singular point. Assuming vertical light, the image intensity was first computed using the discretization of the derivative implicit in eq. 3.14 [14]. With this choice, the original surface is a fixed point of the algorithm and should be reconstructed exactly. Using Jacobi updates, the algorithm converged to the correct solution to within, on average, one part in 10^7 after 63 iterations. In general, the convergence time is expected to be on the order of the maximum length of an optimal trajectory. Since from eq. 3.11 an optimal trajectory jumps one lattice site per iteration, when the number of iterations becomes greater than the maximum trajectory length, then all image points are able to "learn" their heights from the singular points. For the given surface, the maximal trajectory length is on the order of 32, since trajectories starting at the image corners must zigzag to the singular point at the center of the image.

Convergence using Gauss–Seidel updating was faster: it was obtained after just 4 iterations. Gauss–Seidel was performed by changing the direction of the pass over the image after each iteration [2].

For an image obtained by analytically differentiating the displayed surface, the average and maximal errors were .8 and 1.6 (the latter obtained at the image boundary), compared with a range for the surface height of 25. The algorithm has also been applied to a noisy image of this surface; the result is a noisy approximation of the surface. The surface was also reconstructed assuming oblique lighting.

For comparison, Figs. 2, 3 display the result of applying our implementation of Horn's algorithm [5] to a similar surface. The intensity is computed differently than before, using the discrete forward derivatives appropriate for this algorithm. Even after 3072 iterations, the algorithm has not converged to the correct solution. We have also implemented the variational algorithms of [10] and [21], and applied them to this surface with similar results. As also noted by [11], standard variational algorithms often give a wrong, saddle–shaped surface for such simple images containing one singular point.

Figure 4 shows a more complicated 128 by 128 surface. As for Figure 1, the intensity was first computed assuming the discretization of eq. 3.14. The algorithm this time incorporated a terminal cost—an initial value for U—which was large everywhere but at the concave singular points. At these points, U was initialized to the known height values. For vertical light, the algorithm converged to a perfect reconstruction of the original surface in 100 iterations. As expected, the convergence time is on the order of the longest optimal trajectory. Using Gauss–Seidel, convergence was achieved in 10 iterations. When the intensity was derived analytically, the algorithm again converged in 10 iterations using Gauss–Seidel, with an average error of 1.7 compared to a surface range of 51 (Figure 5). Because the surface does not obey the boundary condition that it is decreasing in from the boundary (Section 3), the reconstruction is incorrect in places at the boundary, though it is good in the interior. This is clear in Figure 6, which displays the difference between the reconstruction and the original surface. This surface was also reconstructed assuming oblique light at an angle of 17.5° to the vertical. For an intensity derived as for eq. 3.14, convergence to within one part in 10^{-7} was obtained within 120 iterations. Using Gauss–Seidel, convergence was obtained in 11 iterations. Reconstruction for the analytically–derived intensity function was obtained in 14 iterations, with an average error of 2.2. As previously, the reconstruction was good in the interior but incorrect along one boundary (Figure 7).

Figure 8 shows the result for vertical light of applying the algorithm without the terminal cost. The algorithm reconstructs a surface that is locally concave at all singular points; it is correct in the neighborhood of those singular points where the surface is in fact locally concave. Note the sharp orientation discontinuities at the boundaries between the regions associated with different singular points.

Finally, our algorithm has been applied to the real 200×200 image shown in Figure 9, which was provided to us by Yvan Leclerc of SRI. The light is from above at $(0, .488, .873)$. For the reconstruction, just one singular point was used, located on the tip of the nose, although the image actually contains several. This has the effect of planing down the surface bumps associated with the other singular points. Figure 10 shows the reconstruction obtained using Gauss–Seidel after 6 iterations, illuminated from the same direction as the original. Figure 11 shows this reconstruction illuminated from below. Convergence has essentially been achieved over the face. This reconstruction took about 9 seconds of CPU time on a DEC 5000 workstation. Standard variational algorithms typically require thousands of iterations [5]. Finally, Figure 12 shows the surface reconstruction.

For comparison, Figures 13, 14 display the reconstruction obtained by the authors of [10] using a more standard variational method, developed for the purpose of including stereo information. Stereo information was used as an initial condition for this reconstruction.

Acknowledgments

We thank Yvan Leclerc for providing the image and reconstruction displayed in Figures 9, 13, 14, and for enjoyable conversations. We also thank Martin Bichsel for sending us [2].

References

1. A. R. Bruss, "The Eikonal Equation: Some Results Applicable to Computer Vision," *Journal of Math. Phys.* 23(5): 890-896, May 1982.

2. M. Bichsel, "A Simple Algorithm for Shape from Shading," MIT Media Lab Technical Report No. 172, October 1991.

3. R. J. Elliott, *Viscosity Solutions and Optimal Control*, Longman Scientific and Technical, New York, NY: 1987.

4. B.K.P. Horn, "Obtaining Shape from Shading Information," in *The Psychology of Computer Vision*, P. H. Winston (ed.), McGraw Hill: New York, 1975, pp. 115-155.

5. B.K.P. Horn, "Height and Gradient From Shading," *International Journal of Computer Vision* Vol. 5 No. 1, pp. 37–75, 1990.

6. B. K. P. Horn and M. J. Brooks, "The Variational Approach to Shape from Shading," *Computer Vision, Graphics, and Image Processing*, Vol. 33, pp. 174–208,1986.

7. K. Ikeuchi and B. K. P. Horn, "Numerical Shape from Shading and Occluding Boundaries," *Artificial Intelligence*, Vol. 17, Nos. 1–3, pp. 141–184, August 1981.

8. H. J. Kushner, "Numerical methods for stochastic control problems in continuous time,", *SIAM J. Control and Optimization*, Vol. 28, pp. 999–1048, 1990.

9. H. J. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Springer–Verlag: New York, 1992.

10. Y. G. Leclerc and A. F. Bobick, "The Direct Computation of Height from Shading," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Lahaina, Maui, Hawaii, 1991, pp. 552-558.

11. Y. G. Leclerc and A. F. Bobick, *personal communication.*

12. J. Milnor, *Morse Theory*, Annals of Mathematics Studies 51. Princeton University: New Jersey, 1970.

13. Paul Dupuis and J. Oliensis, "Direct method for reconstructing shape from shading," in preparation.

14. J. Oliensis and Paul Dupuis, "Direct method for reconstructing shape from shading," *Proc. SPIE Conf. 1570 on Geometric Methods in Computer Vision*, San Diego, California, July 1991, pp. 116-128.

15. J. Oliensis, "Shape from Shading as a Partially Well–Constrained Problem," *Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol. 54, No. 2, pp. 163-183, 1991.

16. J. Oliensis, "Uniqueness in Shape From Shading," *The International Journal of Computer Vision*, Vol 6, No. 2, pp. 75-104, 1991.

17. J. Palis and W. de Melo, *Geometric Theory of Dynamical Systems.* Springer-Verlag: NY, 1982.

18. B. V. H. Saxberg, "An Application of Dynamical Systems Theory to Shape From Shading," in *Proc. DARPA Image Understanding Workshop*, Palo Alto, CA, May 1989, pp. 1089-1104.

19. B. V. H. Saxberg, "A Modern Differential Geometric Approach to Shape from Shading," MIT Artificial Intelligence Laboratory, TR 1117, 1989.

20. E. Rouy, A. Tourin, "A Viscosity Solutions Approach To Shape–From–Shading," June 1991, to appear in *SIAM J. on Numerical Analysis.*

21. R. Szeliski, "Fast Shape from Shading," *Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol. 53, pp. 129-153, 1991.

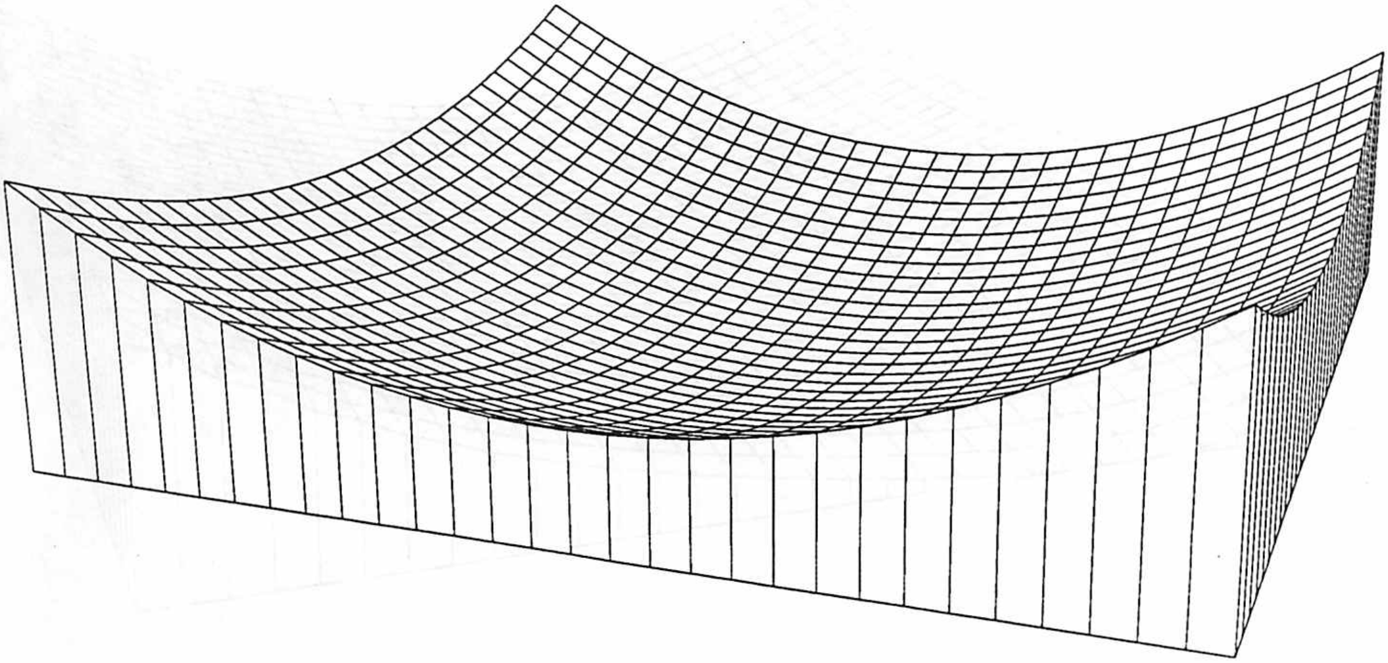

Fig. 1 Parabolic surface.

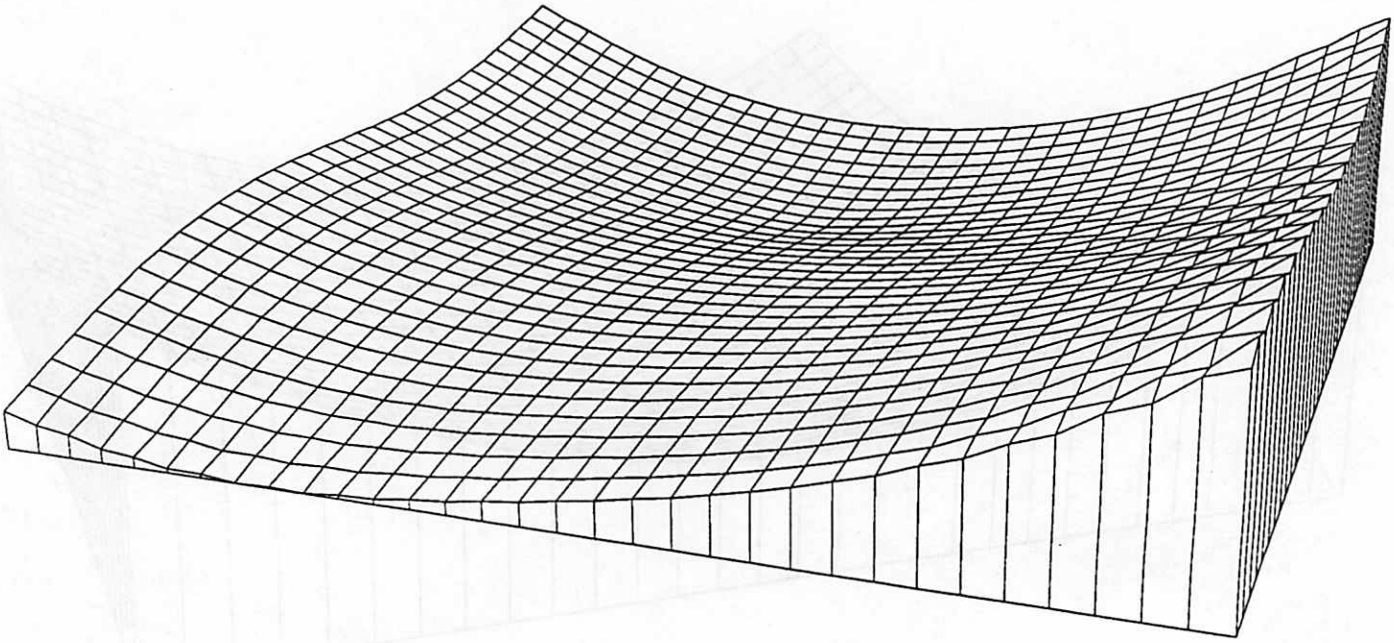

Fig. 2 Horn's algorithm: 128 iterations.

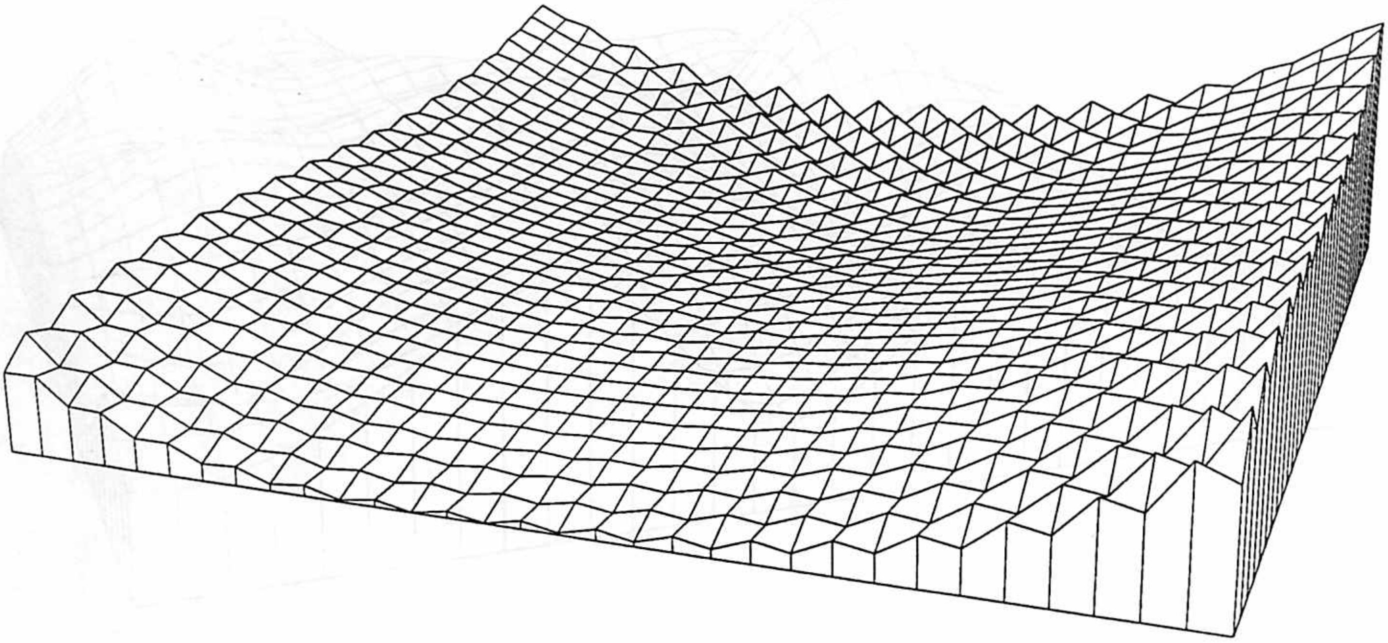

Fig. 3 Horn's algorithm: 3072 iterations.

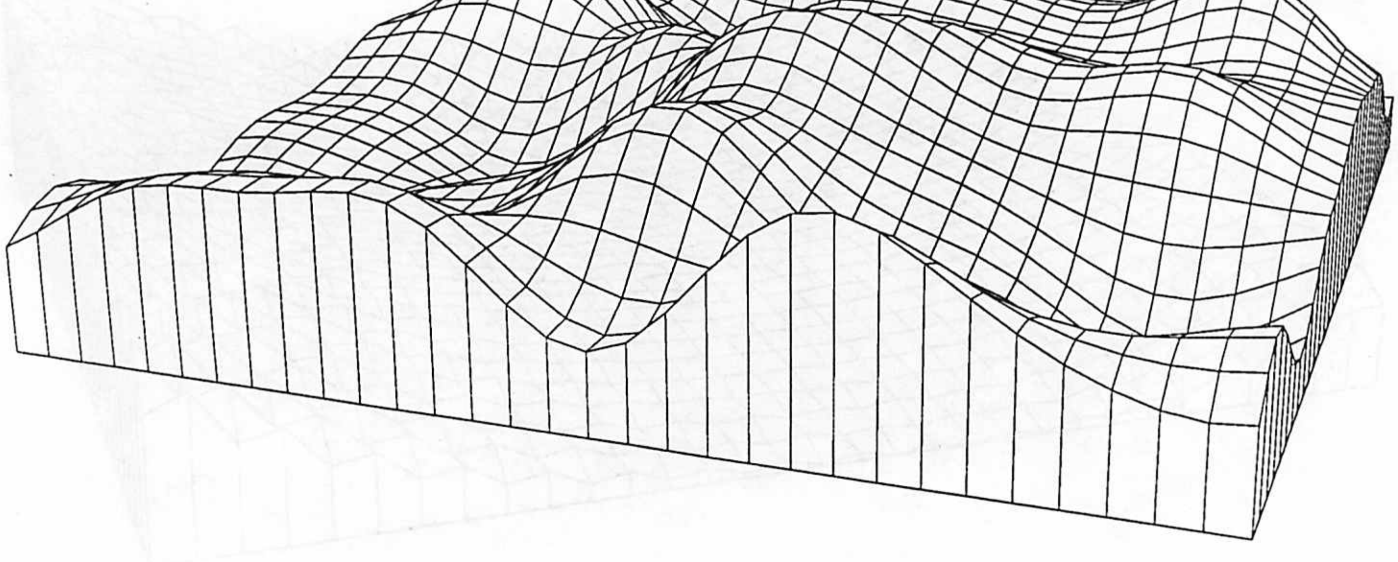

Fig. 4 Complex surface.

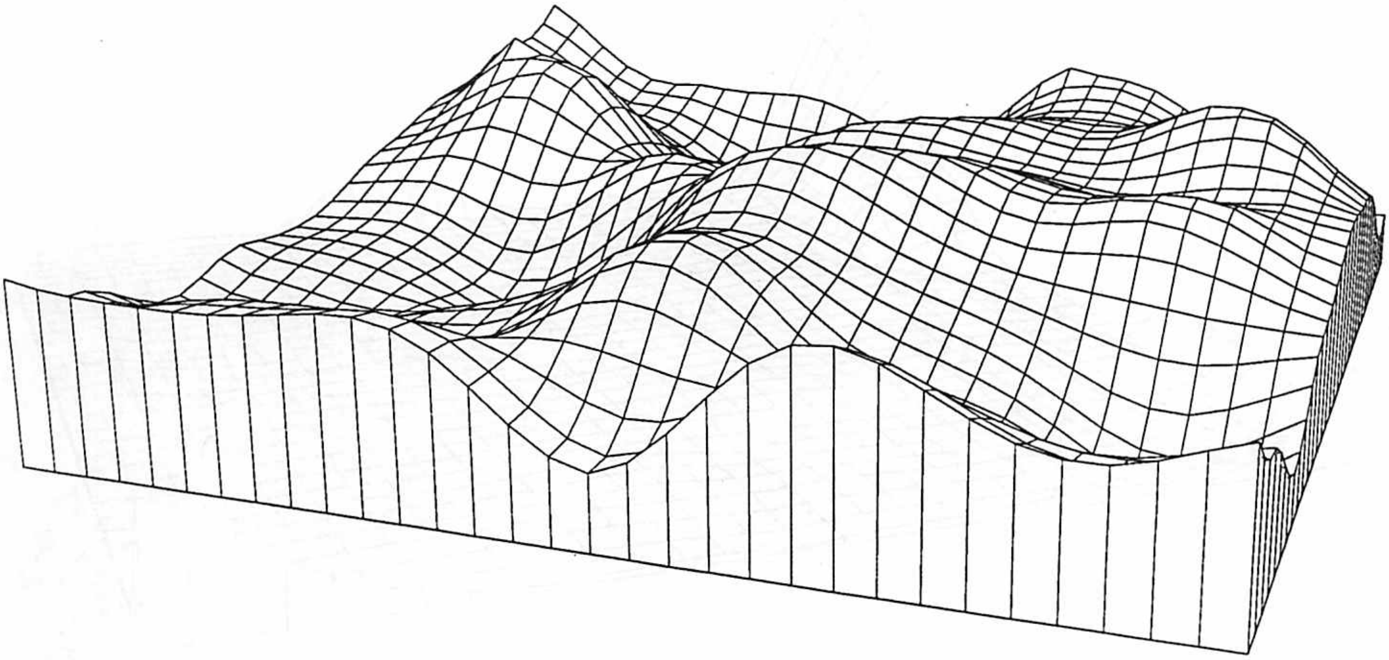

Fig. 5 Reconstruction.

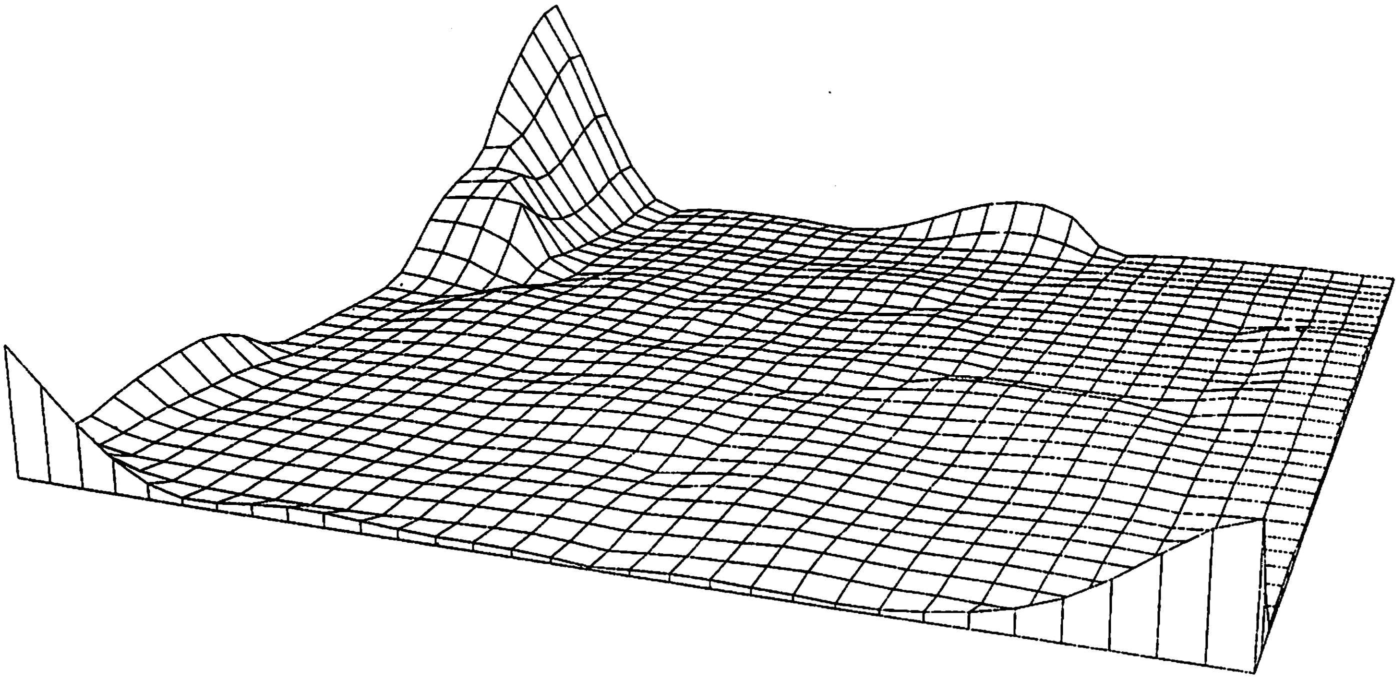

Fig. 6 Difference Surface.

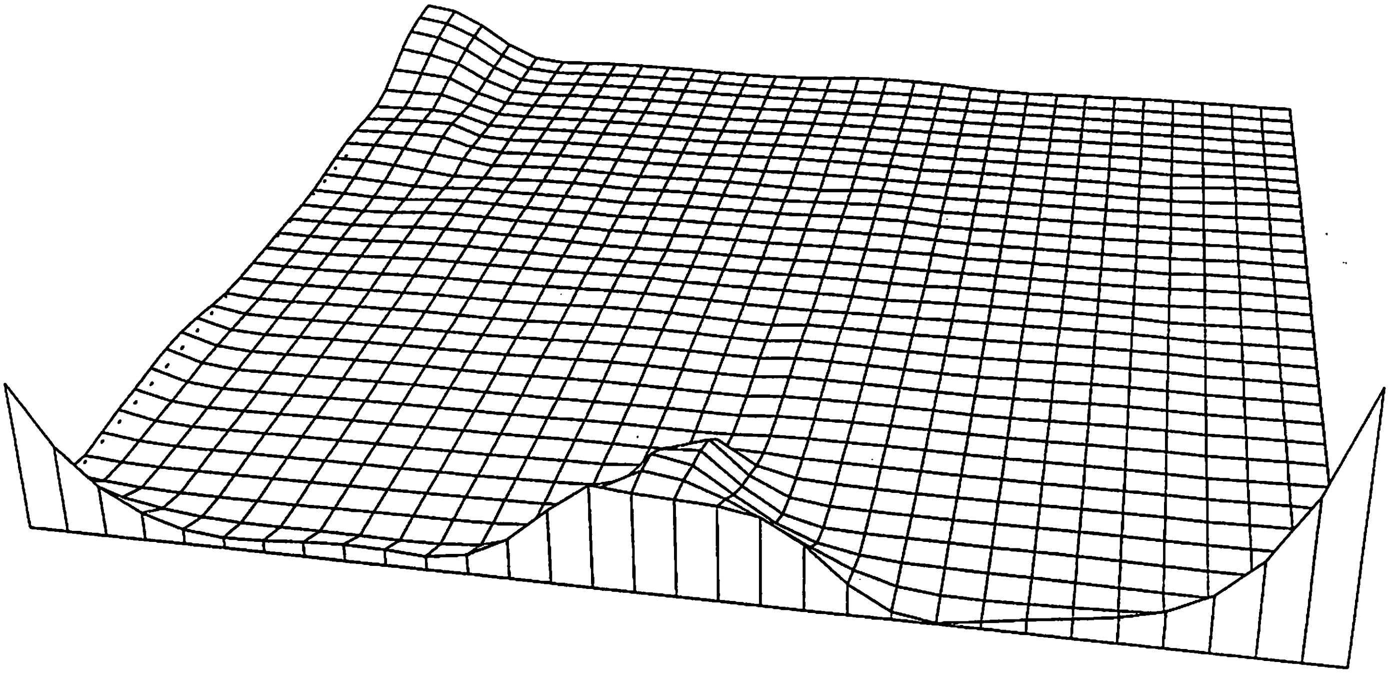

Fig. 7 Difference Surface.

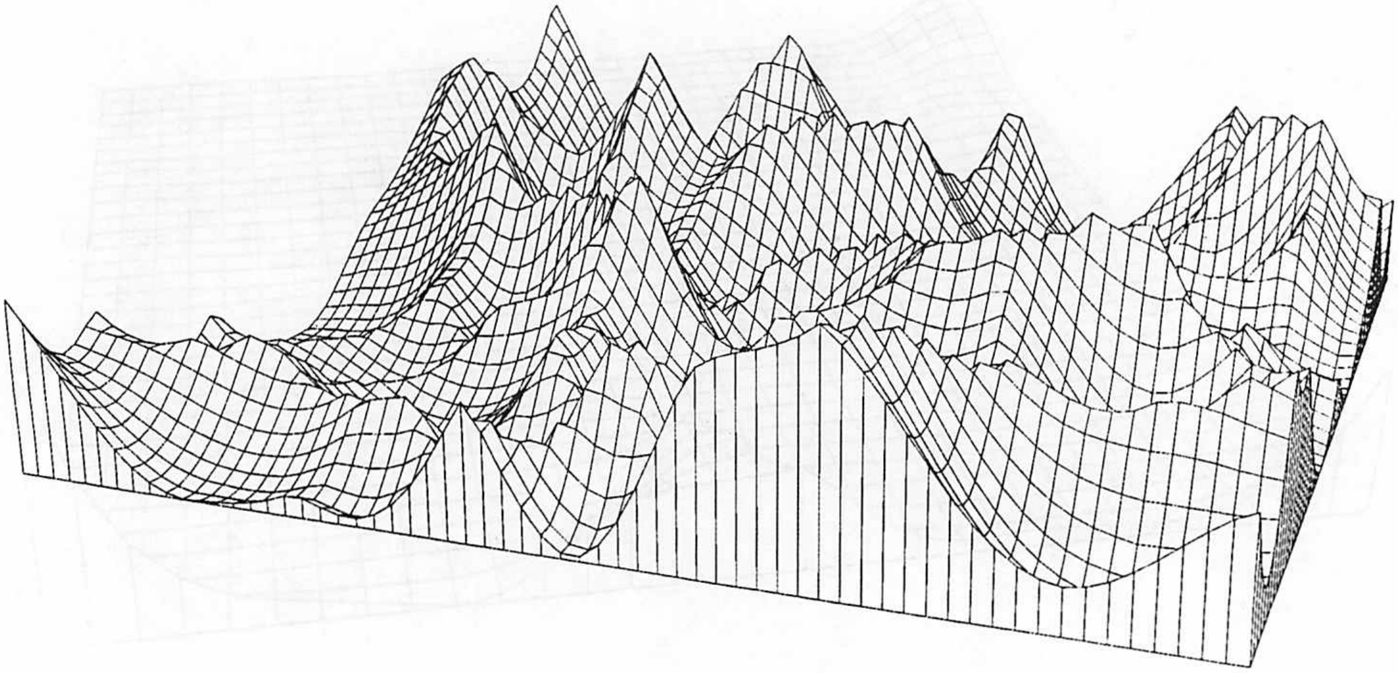

Fig. 8 Result with no terminal cost.

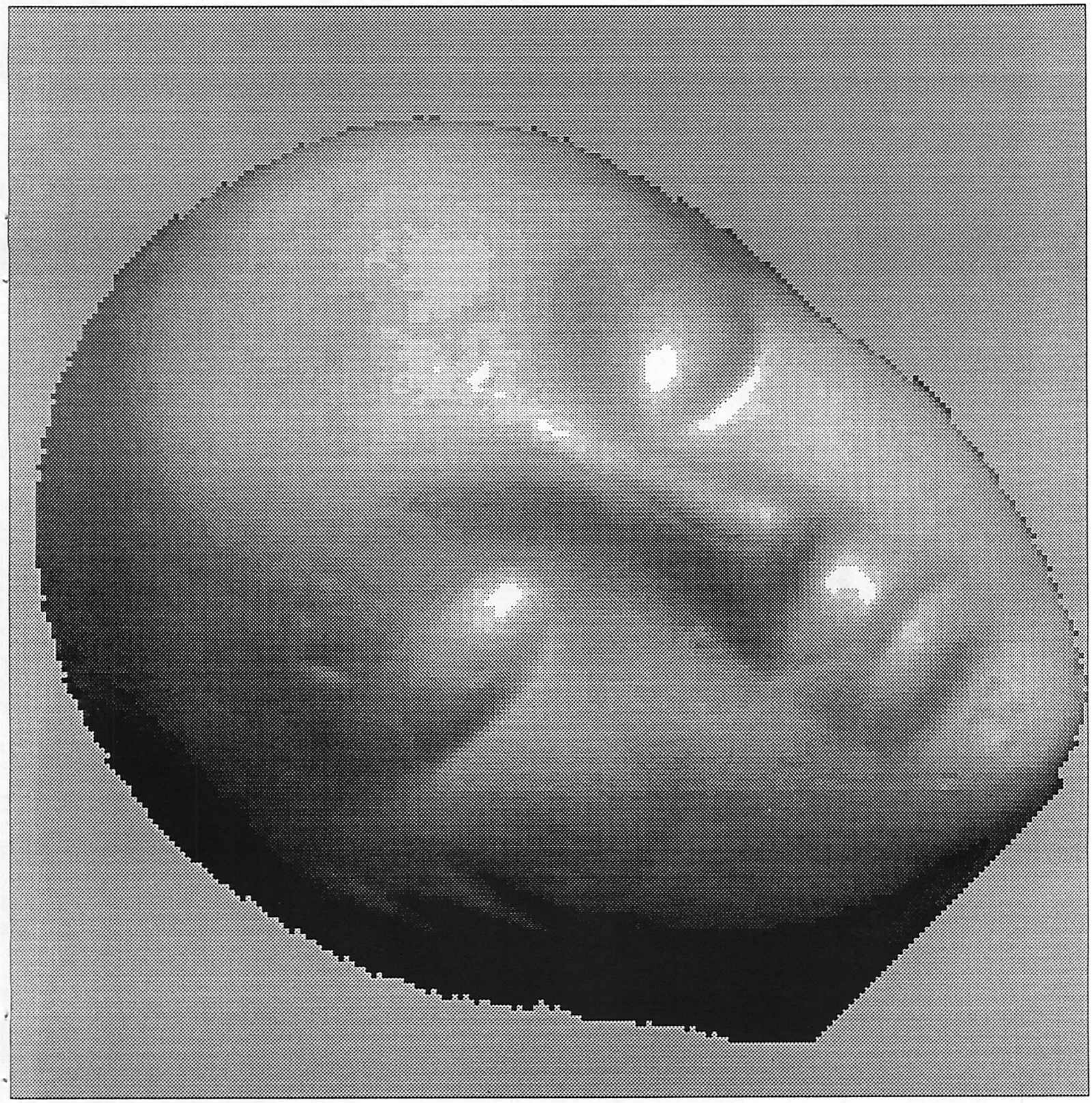

Fig. 9 Mannequin image.

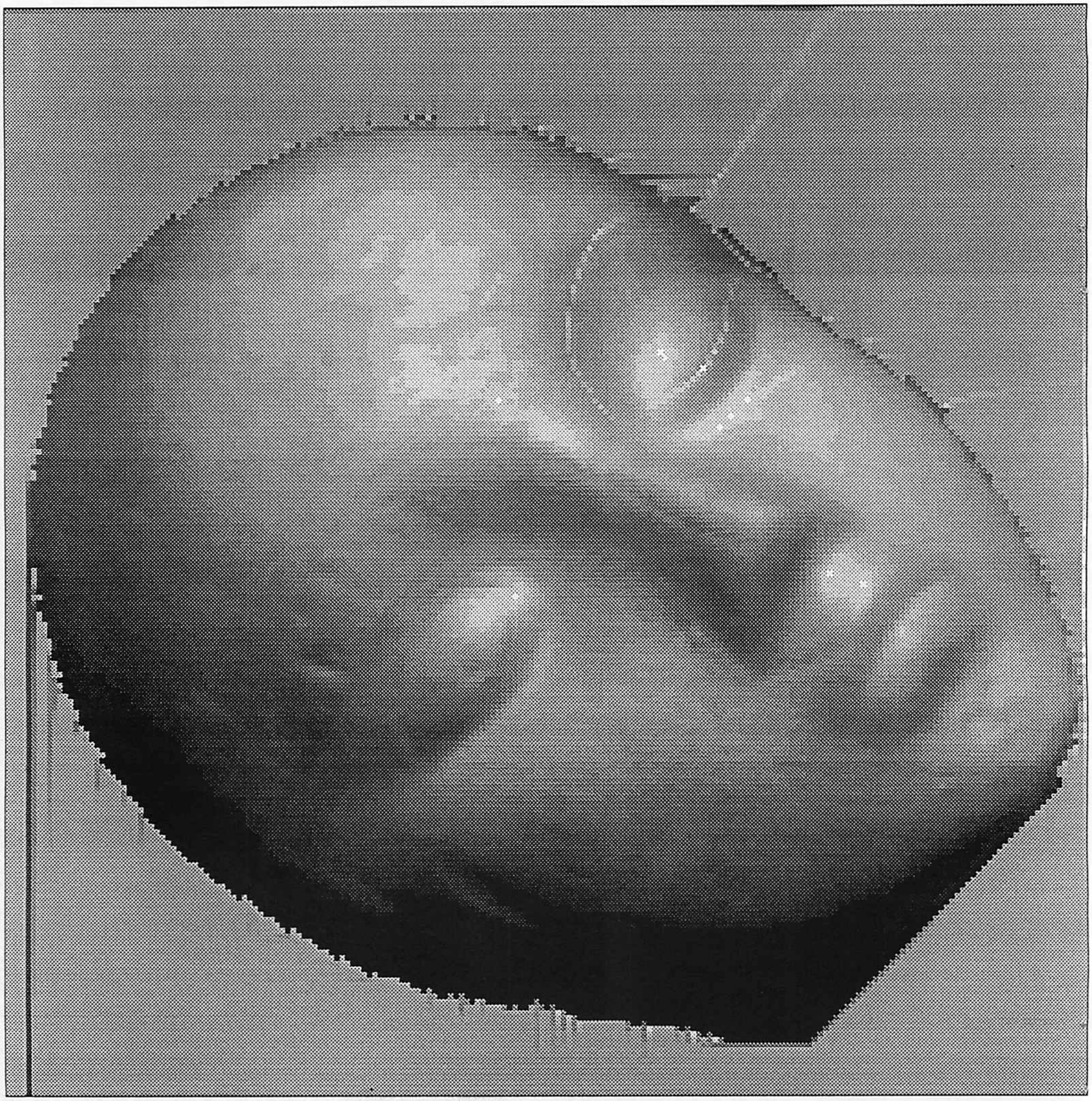

Fig. 10 Reconstruction lighted from above.

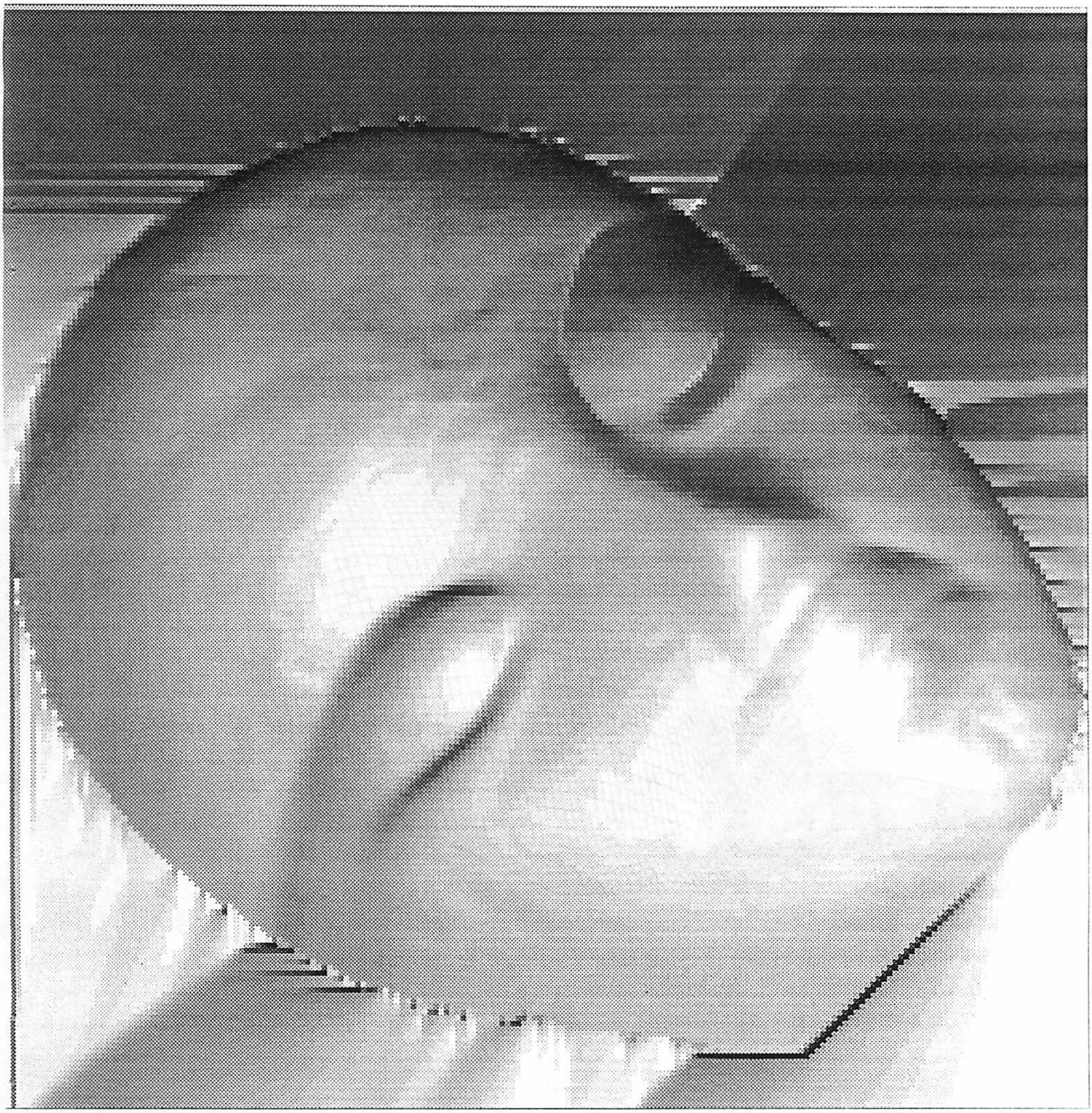

Fig. 11 Reconstruction lighted from below.

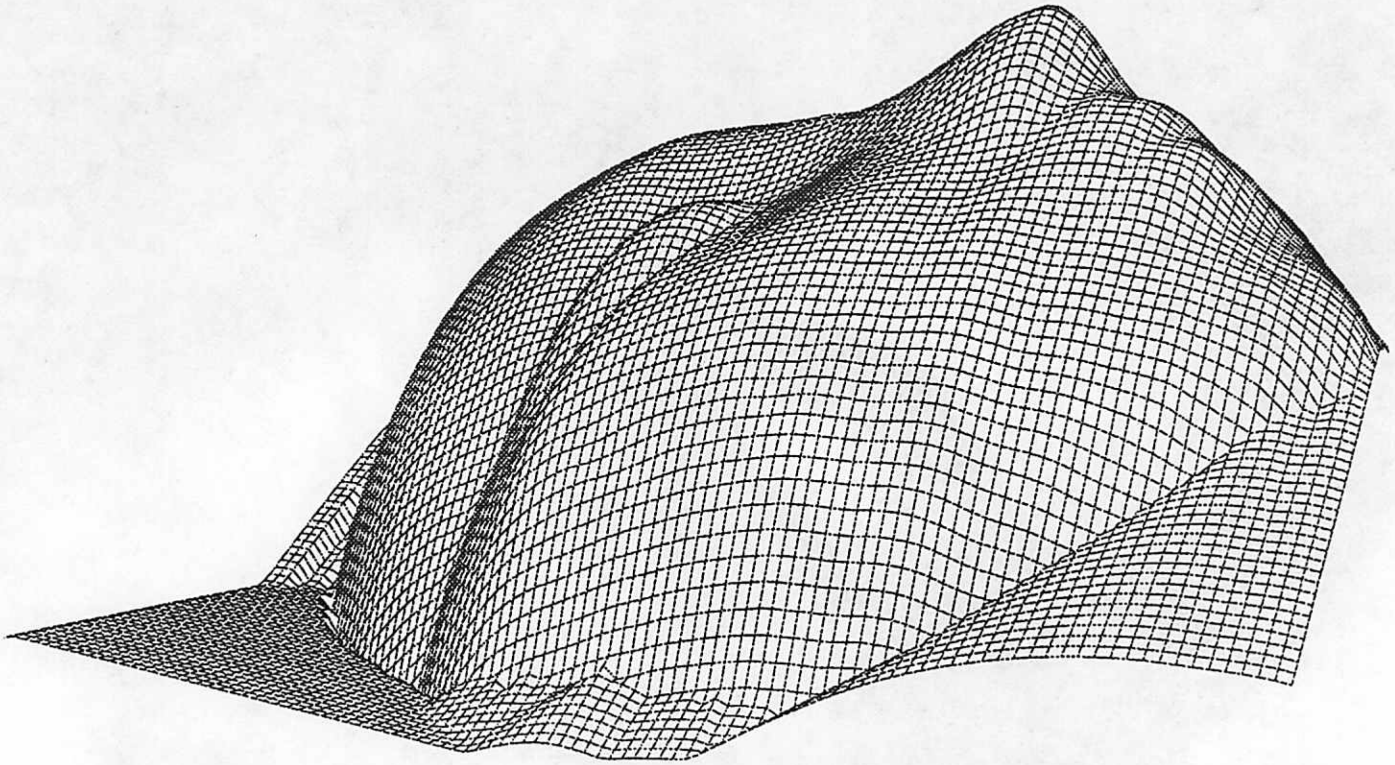

Fig. 12 Surface reconstruction.

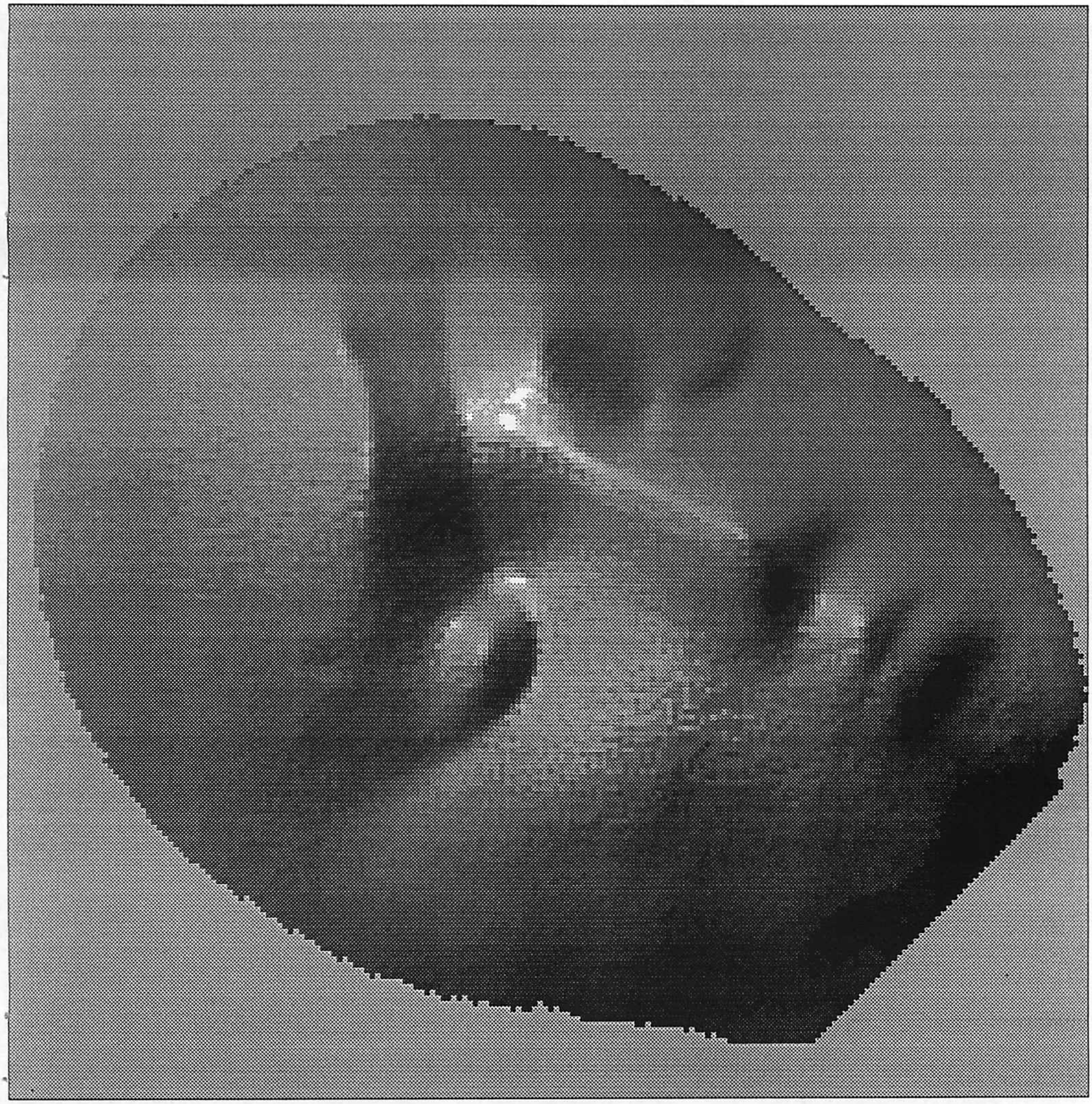

Fig. 13 Reconstruction [10] lighted from above.

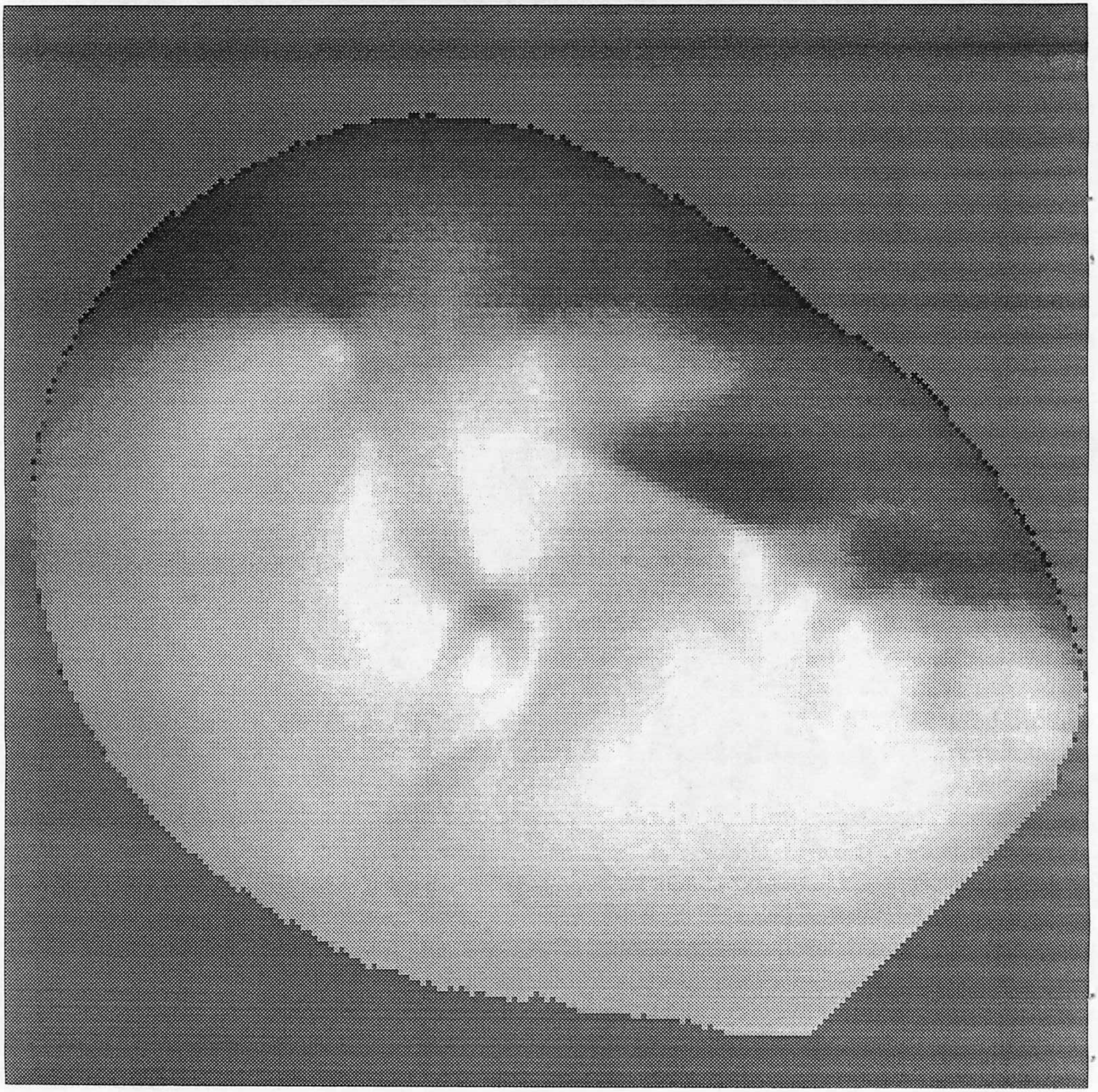

Fig. 14 Reconstruction [10] lighted from below.