MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

AI Memo 566

November, 1979
revised February, 1980

NUMERICAL SHAPE FROM SHADING AND OCCLUDING CONTOURS IN A SINGLE VIEW

by

Katsushi Ikeuchi

ABSTRACT

An iterative method of using occluding boundary information is proposed to compute surface slope from shading.

We use a stereographic space rather than the more commonly used gradient space in order to express occluding boundary information. Further, we use "average" smoothness constraints rather than the more obvious "closed loop" smoothness constraints. We develop alternate constraints from the definition of surface smoothness, since the closed loop constraints do not work in the stereographic space. We solve the image irradiance equation iteratively using a Gauss-Seidel method applied to the the constraints and boundary information. Numerical experiments show that the method is effective. Finally, we analyze SEM (Scanning Electron Microscope) pictures using this method. Other applications are also proposed.

KEYWORDS

Shape from shading, Shape from occluding contours, A single view
Reflectance map technique, Propagation of constraints, Gauss-Seidel method,
SEM (Scanning Electron Microscope) picture, Relaxation method

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Office of Naval Research under contract N00014-77-C-0389.

Fig. 2-A and Fig. 26 are used from "Magnification" by David Scharf under permission of the author.

0 OVERVIEW OF THIS PAPER

0.0 Main Result of This Research

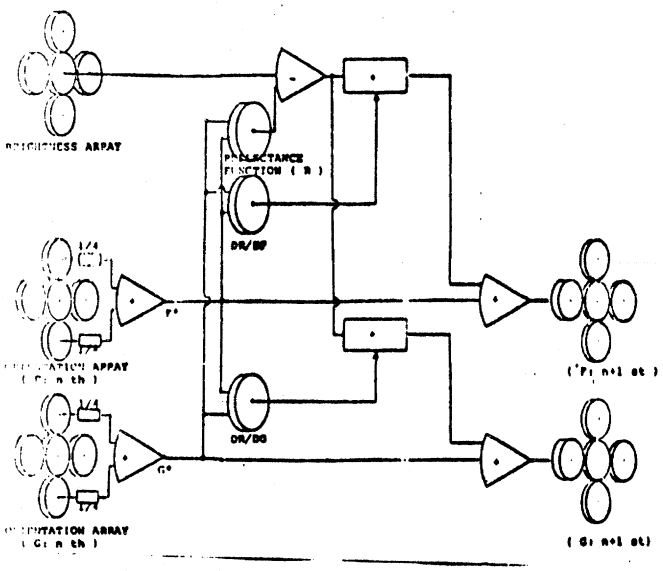


Fig. 1 diagram of our algorithm.

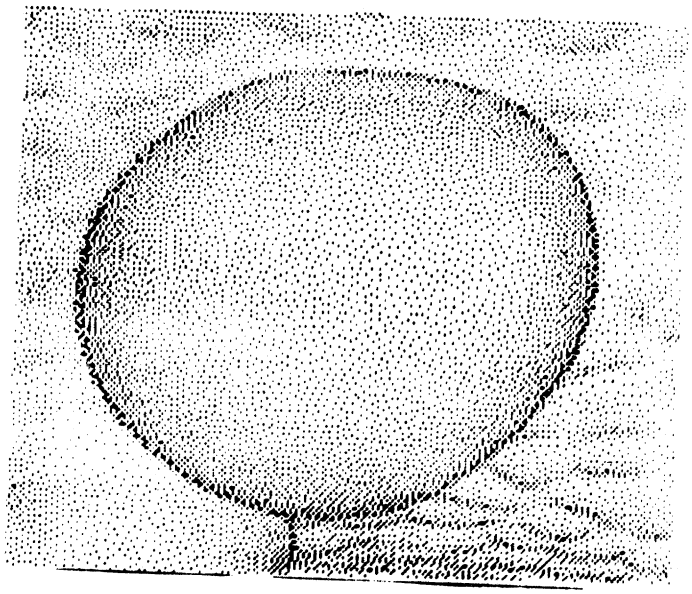


Fig. 2.A A SEM (scanning electron microscope) picture.

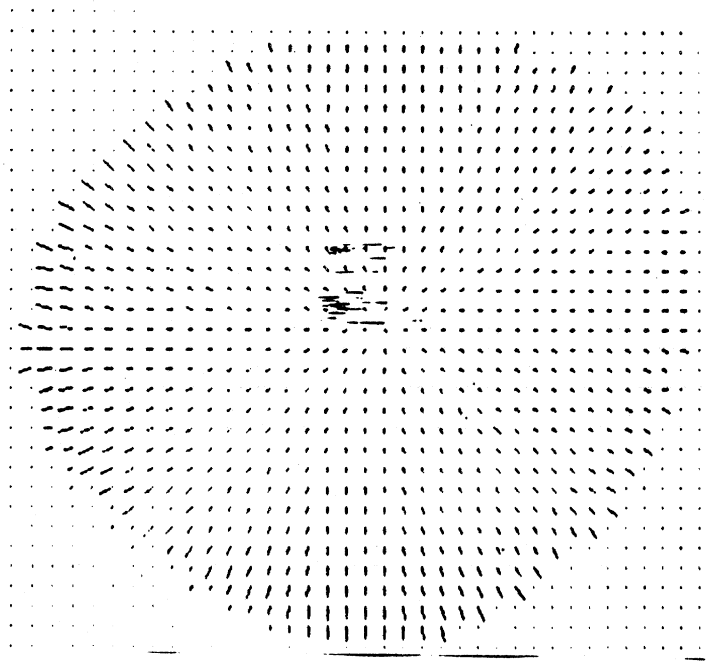


Fig. 2.B Obtained needle diagrams.

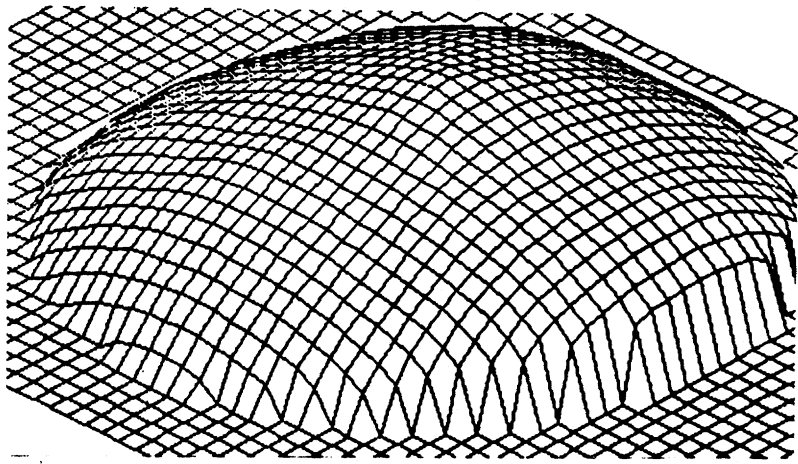


Fig. 2.C A generated surface.

0.1 Summary of the Approach

(1) Our iterative method for solving the shape from shading problem is less susceptible to noise than Horn's method.

Horn's method[1] starts at a singular point and proceeds along characteristic strips. Since the method has no constraints at the end point, there is possibility that errors accumulate along operations. On the other hand, our method try to make an equilibrium condition which satisfies two end point (occluding boundary) and intermediate point (singular point). Thus, the accumulation effect is less in our method.

(2) At an occluding boundary, the orientation of surface normals are explicit.

Namely, a surface normal must be perpendicular to an occluding line and the view line. These two facts determine the direction of the surface normal uniquely.

(3) It is necessary to replace gradient plane with a different plane in order to express the above mentioned occluding information.

Unfortunately, at an occluding boundary, the components of the gradient are infinite. Specifically, p and q are defined as the partial derivatives of the height of an object relative to a plane perpendicular to the optical axis with respect to the coordinates of the image plane. At an occluding boundary, surface normals become perpendicular to the height axis. This means that the partial derivatives become infinite.

(4) It is necessary to use "average" constraints in order to express surface smoothness.

Strat[5] and Brooks[6] presented "closed loop" constraints. Both sets of constraints based on a characteristic of the gradient plane, and depend on the exchangability of differentials and Green's theorem. This exchangability hold for onto a different project plane. It is possible to project the closed path onto the new plane; we wonder, however, if more direct constraints derive from surface smoothness.

TABLE OF CONTENTS

1 INTRODUCTION

The shape from shading problem deals with information within boundaries. We try to find surface orientation from the recorded brightness of the surface itself. This may be considered a "surface to surface problem". Horn[1] solved the image irradiance equations and demonstrated that one can decode surface orientation from surface brightness.

Horn[2],[3] generalized and simplified the image irradiance equations using the reflectance map. Horn and Sjoberg[3] recently showed how to calculate the reflectance map from NBS's BRDF (Bi-directional Reflectance Distribution Function) analytically. The reflectance map represents the relationship between surface orientation and surface brightness. The map is a function of coordinates in gradient space, which assigns a measure of brightness for a surface in the orientation specified by each point. This is called the reflectance map. If the brightness of a surface patch is known, one can determine possible points on this map, and thus, possible orientations of the surface patch.

Even though Horn's method[1] can determine surface orientations in noise-free cases, there is a possibility of accumulation of errors in noisy cases. Namely, the method tries to get surface orientation along characteristic strips. The direction of the strip at a particular point is determined from the direction of steep descent at the reflectance map at the point. If the image has a noise spot, the direction is incorrectly guided at that point. Unfortunately, the incorrect guidance will not recover and will become larger along the stages. Because (1) the method has no constraints at the end point (2) a new point depends entirely on the neighboring point on the characteristic strip. The method use a singular point as the starting point and proceeds along a characteristic strip. However, we cannot give the method constraints at the end point.

Woodham[4], Strat[5], Brooks[6], and Barrow and Tenenbaum[7] proposed cooperative algorithms for solving the shape from shading problem. Horn's method considers a neighboring point on a characteristic strip. The iterative algorithms try to reduce this restriction. These algorithms extend neighboring areas which has influence to a point. All of these algorithms use iterative techniques but differ in basic assumptions. Woodham's algorithm requires some global cues about an object. Barrow and Tenenbaum's algorithm requires that a surface be quadratic. On the other hand, Strat and Brooks proposed generalized algorithms. Unfortunately, both algorithms use a particular characteristic of the gradient space. In particular, although the Strat algorithm is general, we can not use the algorithm in practical applications because of the difficulty in obtaining boundary conditions.

Boundaries supply important information. Many other machine vision techniques first analyze boundary information, and then, analyze inside regions. For example, stereo algorithms often use discontinuities of brightness for matching elements between the two image and then compute depth. In scene analysis conditions of

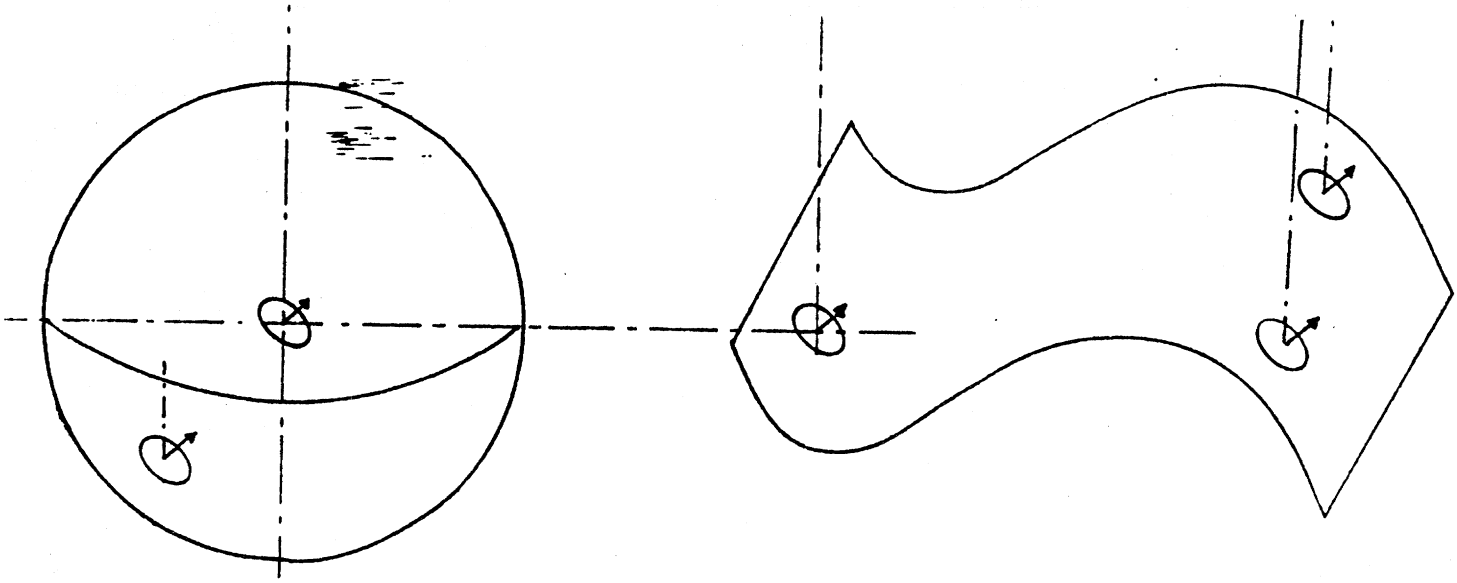
conjunctions of boundaries are considered.

This research is motivated by the observation that we can use occlusion information as boundary information. If we have some representation that can handle the boundary information, we can use this information as boundary conditions necessary for iterative techniques. All of the above mentioned iterative algorithm cannot use information from occluding boundaries. We can make an algorithm strong against noise using iterative techniques and occlusion information.

2 GAUSSIAN SPHERE AND REFLECTANCE FUNCTION

2.1 Gaussian Sphere and Apparent Brightness of a Surface Patch

We can associate surface orientations with points on the gaussian sphere. The gaussian sphere is a sphere of unit radius whose Z-axis is taken as an extended line from the axis between the north pole and south pole. Assume that we put a surface patch of an object at the center of the sphere and that the direction of the viewer is the direction from the center to the north pole. The surface patch faces some point of the sphere. For example, if a surface normal has the same direction as the south pole, the surface patch is perpendicular to the viewer. Moreover, since we assume that the line of view is always parallel (the orthographic projection) to the z-axis of the gaussian sphere, we could bring a surface patch to the center of the gaussian sphere and associate the direction of the surface patch to a point of the sphere where the surface patch faced. In this situation, surface patches of an object which share the same orientation are always associated with a particular point on the sphere. The associated point does not vary with the position of s surface patch on an object. For example, a patch which is perpendicular to the viewer can be expressed as the south pole on the gaussian sphere regardless of its position on the object. Thus, the gaussian sphere expresses all possible directions of a surface patch. If you could sit on the center, you would observe that the surface of the sphere covered all possible directions, and you could associate the direction you face with some point on the surface.



We can associate with each point on the gaussian sphere a number which indicates how bright a surface patch at the corresponding orientation would appear to a viewer. Surface brightness depends on the viewing geometry: the direction of the surface normal, and the source direction. At first we point at the direction of light source on the gaussian sphere. Then each point on the gaussian sphere expresses one particular geometric relationship. Thus, we can relate a particular surface brightness to each point of the gaussian sphere. For example, when the surface is perpendicular to a viewer, we might have , say, one unit of brightness. So we would assign this one unit of brightness to the south pole. Obviously, this diagram depends on the direction of light source. In Fig. 4, the light source locates near the viewer.

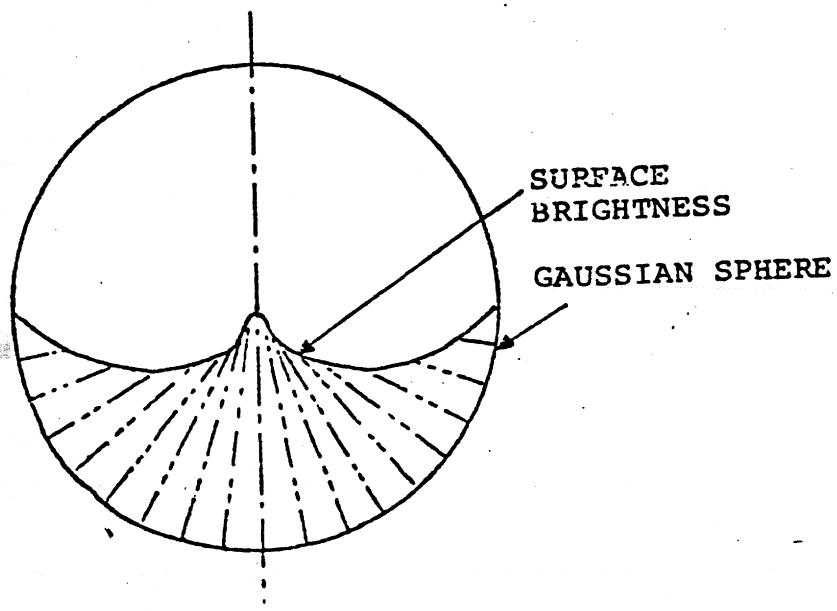


Fig. 4 Gaussian reflectance diagram.

2.2 Reflectance Maps on Various Planes

(How to Project the Gaussian Sphere to a Plane)

It will be advantages to project this gaussian sphere onto a plane for simplicity. The gaussian sphere is a three dimensional object, however, we are only interested in its surface. The surface has only two dimensions. This means that we can specify points on that surface by using two parameters. We now must determine how to cut the surface of the sphere and expand it on a plane.

All parameterization should be continuous. Since purpose is to generate surface

representations from boundary information iteratively, the projection should be continuous, meaning that the CENTRAL VALUE THEOREM should hold everywhere. A point C between A and B on the gaussian sphere should be projected as F(C) between F(A) and F(B). If the theorem did not hold, we could not interpolate a middle point between two boundary points. Roughly speaking, it means that we could not get surface orientations in an inner area from the boundary information.

Horn's method is very convenient. Horn[2] used (p,q), where p,q are the partial derivatives of z with respect to x,y, respectively. Every point on the surface of the sphere is projected to a particular point on the gradient plane, with respect to the center of the gaussian sphere.

(1) $p = \partial z/\partial x \qquad q = \partial z/\partial y$

As each point on the gradient plane expresses a particular geometrical relationship, we can assign a certain brightness to each of these points. Horn called the assigned diagram the reflectance map. Now the diagram is expressed by using two coordinates p and q. The point which used to be the south pole is now the origin of the plane. The equator is projected to an infinite point. Horn's projection is very convenient; (1) each coordinate corresponds to the first derivative of a surface height function, (2) we can easily generate surface height, (3) an integral along a closed loop is always zero on this plane because of Green's theorem. Unfortunately, the gradient plane has a defect. Namely, on that plane, the equator maps to the infinite point. We cannot express occluding boundary information on this plane.

One possible solution for this blow-up problem is to use the stereographic projection. This projection puts the sphere on a plane and projects points of the sphere to the plane from the north pole instead of the center as in Horn's case. We also assign brightness value to each point on the stereographic plane. We will use (f,g) to express this plane.

(2) $$f = \partial z/\partial x/(\sqrt{(1+(\partial z/\partial x)^2+(\partial z/\partial y)^2)} + 1)$$
$$g = \partial z/\partial y/(\sqrt{(1+(\partial z/\partial x)^2+(\partial z/\partial y)^2)} + 1),$$

where $z=Z(x,y)$.

Since the surface of the gaussian sphere is two dimensional, there are many possible projections. It can be shown that a pair of azimuth-zenith angles can be used for this projection. In this case, we use the polar coordinates in the plane. The azimuth angle can be expressed by using the phase angle, and the zenith angle is the distance from the origin to a particular point. We will denote this plane as (r,ϕ).

(3) $$r = \pi \tan^{-1} \sqrt{(\partial z/\partial x)^2 + (\partial z/\partial y)^2}$$
$$\phi = \tan^{-1} ((\partial z/\partial y)/(\partial z/\partial x)).$$

This projection can be thought of as painting the gaussian sphere and rotating it on the plane in order to register each point of the sphere to the plane. Each point on the plane

expresses the geometrical relationships brightness values. Since the surface of the gaussian sphere is two dimensional, there are many other possible pairs of real numbers which can project the sphere on a plane besides of the above three projections. As we have seen, choosing the right projection can make work easier.

3 CONSTRAINTS FROM SURFACE SMOOTHNESS, EDGES, AND IMAGE INTENSITIES

In order to determine surface orientation, we must use information to constrain the possibilities. With no information, all direction are possible. We use three categories of constraints. First, a brightness value of a surface patch indicates a set of possible orientations on a reflectance map. So the image intensity equations, or the reflectance map Second a contour line indicates a surface there; that is, at an occluding boundary, we can determine surface orientations from the contour in an image. So the second category of constraints arise from contour lines. Finally, if we assume that an object has a smooth surface, neighboring points must have similar orientations. Here, surface smoothness provides the third category of constraints. By using these constraints, we will determine a surface orientation for each surface patch.

3.1 Surface Smoothness Constraints

3.1.1 Formulation

We cannot use closed the constraints proposed by Strat[5] or Brooks[6]. Because closed constraints come from a special characteristic of the gradient plane. Since the coordinates consist of partial derivatives of z with respect to x and y, each derivative of the coordinates with respect to y and x is the same one; a twice partial derivative of z with respect to x and y or y and x.

$$(4) \qquad \partial p/\partial y = \partial^2 z/\partial x \partial y = \partial q/\partial x$$

Unfortunately, this is not true in our stereographic projection or azimuth-zenith projection.

We have to invent smoothness constraints independent on projections. It is possible to project the closed loop into our stereographic plane or azimuth-zenith plane. However, if a constraint expresses the characteristics of surface smoothness, the constraint should be valid regardless of the kind of expression we may use. If a surface is smooth on an image plane, a constraint should have a similar formula *with respect to the neighboring points on the image plane* Smoothness constraints should not be dependent upon the kind of projections.

Surface smoothness requires that the surface orientation is continuous on a image plane. First, we return to the definition of surface smoothness. In order for us to call a surface smooth, the surface should be continuous on an image plane (class c-0 w.r.t height). No one can claim that two surfaces, one of which is an object and the other a background, are smooth! Then surface orientation should be continuous on an image plane (class c-1 w.r.t

height). If a surface has a crack or keen edge, even though the surface is continuous, the surface is not smooth. Is it necessary to proceed to one more stage? This would mean that the derivative of the orientation is continuous. This condition is not necessary. Imagine there is a planar surface which continues as a cylindrical surface. Surface orientation is continuous in this case. The derivative of it is zero at the planar part and non-zero at the cylindrical part. People, however, regard this surface as smooth. So the condition on derivatives of orientations on an image plane is not necessary. Thus, we obtained a necessary condition for a surface to be smooth. Namely, the surface height function should be class c-1 on an image plane.

1) surface is continuous on an image plane. (c-0 w.r.t height)
2) surface orientation is continuous on an image plane. (c-1 w.r.t height)

(In fact, the closed loop constraints require class c-2 on a height function at the exchangeability and class c-3 for Green's Theorem)

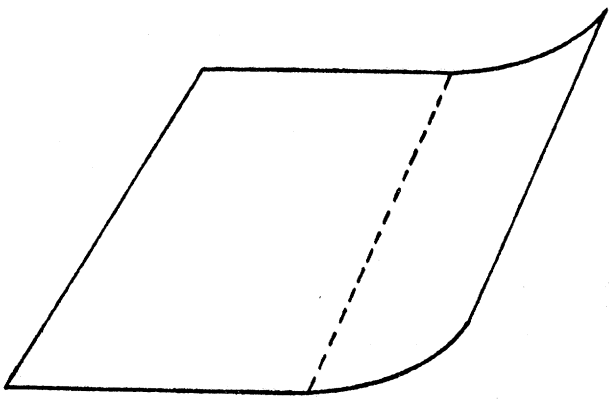


Fig. 5 A planer surface continues to a Cylindrical surface.

3.1.2 Qualitative Justification

What is the definition of continuous? *If arguments of an function approaches a particular point in the definition area, the value of the function on that arguments also approaches the function value on that particular point.* We will use this definition as a constraint.

If a function is vector valued, being continuous implies that each component of the function is continuous [9]. This allows us to consider each orientation component separately. For example, in gradient space, we can make two constraints which express

continuity of either p or q independently. They are independent, provided that the total vector valued space is continuous. In other words, the surface orientations of our object are continuous in the image plane, if and only if the constraints on p and q work at the same time.

We call Eq.6 an "average" constraint which represents the characteristic that a surface is smooth. The definition of continuity of a scalar function of two arguments is, as mentioned above,

when (x,y) *approaches* (x_0,y_0),
then the value of $F_i(x,y)$ also approaches $F_i(x_0,y_0)$.

This definition can be expressed in a more convenient form[9];

A function F_i *is continuous at* (x_0,y_0);
if given an $\varepsilon > 0$, there exits a δ such that $(x-x_0)^2+(y-y_0)^2 < \delta$

$$(5) \quad |F_i(x,y) - F_i(x_0,y_0)| < \varepsilon.$$

If we take our mesh size δ_0 to be smaller than δ, Eq. 5 guarantees that:

$$(6) \quad |F_i(x_0+\delta_0,y_0)+F_i(x_0-\delta_0,y_0)+F_i(x_0,y_0+\delta_0)+F_i(x_0,y_0-\delta_0)-4F_i(x_0,y_0)| < 4\varepsilon \text{ (for } \forall\varepsilon,\exists\delta)$$

Namely, from the definition of surface smoothness, we can obtain that, when the mesh size is small enough, the difference between the sum of the surrounding values, and four times the central value becomes zero. We call these constraints "average" constraints.

3.1.3 Quantitative Justification

One more check is necessary before applying this constraint. Namely, even if the constraint works locally, there is no guarantee that it works globally. We have obtained the result that the difference decreases with mesh size. However, we have not checked how small they are, compared to a increase in mesh number. When we apply these constraints to an image, we will make a summation of the difference over all mashes. When mesh size becomes small, the mesh number in a unit area becomes large. For our purposes, a total summation of the difference should also become small, with decreasing mesh size.

In order to simplify our discussion, we will separate an area into two regions; in one region the second derivative of F_i exists. We will call this region "region-A". In another region it does not exist provided that F_i is c-0 class (or surface height is c-1 class). We will call this region "region-∂A". An example of "region-∂A" is the region at the connecting line between a planer region and a cylindrical region in the example above in

the discussion of surface smoothness.

One additional condition is necessary for the constraints to work globally. Namely, "region-A" should occupy "almost all" of the image. In other words, if the second derivative does not exist at "almost all" of the area, the constraints do not work globally. "Almost all" means an open area which may contain finite numbers of singular points and finite numbers of singular lines. However, we cannot imagine that this situation occurs in the real world. This situation does not occur even in the micro structural situation. The total size of the area of discontinuity is infinitely small and "region-A" occupies "almost all" of the area.

In "region-A", the total summation of the constraints goes to zero when the mesh size become small, provided that we use the constraints as more than square orders. The left hand side of Eq. 6 is approximated as a product of a square of mesh size δ and a laplacian of F_i ; $\delta^2 * F_i$". On the other hand, the number of meshes in "region-A" is S/δ^2, where S is the area of "region-A". Let $F"_{max}$ be a maximum value of the laplacian in "region-A". The square of Eq. 6 is always smaller than $(F"_{max} \delta^2)^2$. Thus, the total of this value in "region-A" is smaller than $(F"_{max} \delta^2)^2 (S/\delta^2)$. Or the total summation of the square of Eq. 6 is smaller than $(F"_{max} * S)*\delta^2$, and when the mesh size δ goes to zero, the total summation also goes to zero.

The "average" constraints work globally. In "region-∂A", the left hand side of Eq. 6 is still smaller than ε, even though the laplacian does not exist. The total contribution from "region-∂A" is smaller than $\varepsilon\ \partial S$, where ∂S is total area size of "region-∂A". ∂S goes to zero with the decrease of mesh size, provided that "region-∂A" does not occupy "almost all" of the area. Since at both regions a summation of the square of the "average" constraints goes to zero with decreasing mesh size, we can use this "average" constraint globally.

3.2 Constraints from Boundary Information

In some areas of an image, we can determine surface orientations directly from contour lines. Namely, at occluding boundaries, singular points and specular points, we can determine surface orientations uniquely based on the contours of the boundaries at these points. Since we need some pre-determined surface normals in order to begin the iterations, this fact is very important.

3.2.1 Occluding Boundary

At an occluding boundary we can determine the surface normal uniquely. In order to make our discussion clear, we introduce Marr's terminology [10]. A "contour" exists on an image plane. A "contour generator" exists on the real surface. A contour generator is projected onto an image plane and becomes a contour on the plane. The following two facts are the starting points to our discussion of determining surface normal

from contour lines.

First; since we observe a contour generator there, the view line is tangent to the surface at that point. Fig. 6 shows this situation. The line of sight (s) is tangent to the contour generator at the point g. Even though a surface has a sharp edge, there is a point which is tangent to the line of sight in the real world.

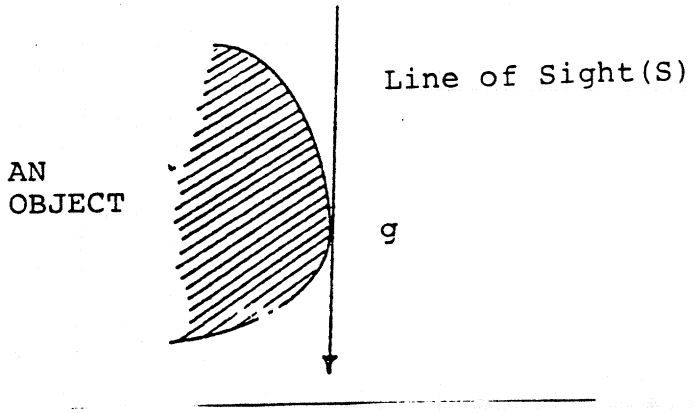


Fig. 6 A line of sight is tangent to a contour generator.

The line of sight lies at the tangent plane and is perpendicular to the surface normal at that point. Since a tangent plane should contain all the tangent lines on the tangent plane and the line of sight is a tangent line, it is obvious that the line of sight lies on the tangent plane. A surface normal is perpendicular to the tangent plane. It follows that at the point g, the surface normal is perpendicular to the line of sight.

Second; the line of sight is perpendicular to the image plane. Since we assume that we can treat the situation as an orthographic projection, all lines of view are parallel and perpendicular to the image plane. From this fact, the tangent plane is projected as a line in the image plane. This line is actually a tangent line to the contour on the image plane.

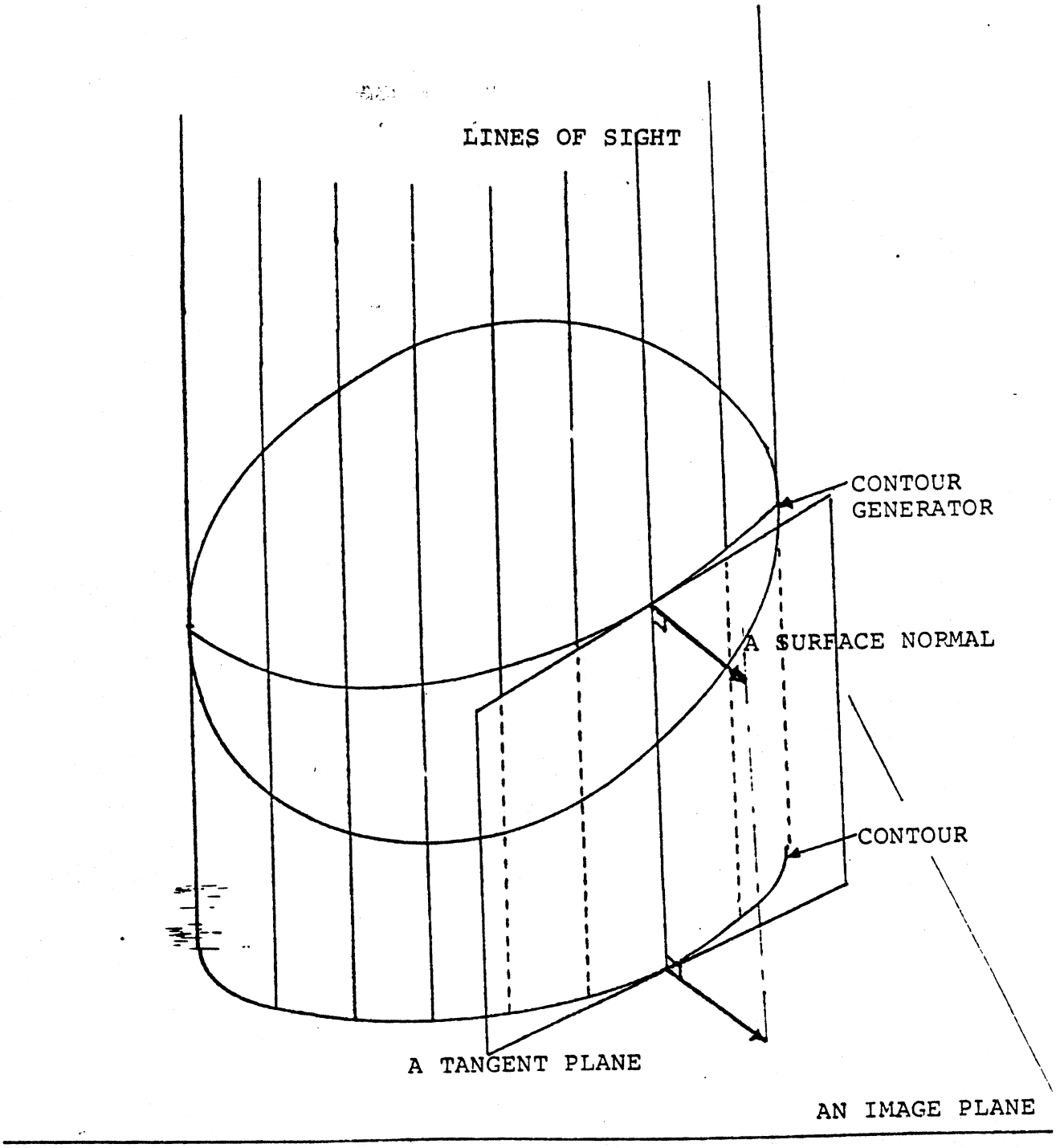


Fig. 7 A contour and a contour generator.

A normal vector to a contour generator is the same as the normal vector to the contour on the image plane, because the normal vector of the contour generator is perpendicular to the contour on the image plane. In other words, the normal vector of the contour generator is perpendicular to the tangent plane and the tangent plane contains the contour. Thus the normal vector of the contour generator is also perpendicular to the contour. Moreover, since the normal vector to the contour generator is parallel to the image plane, the normal vector to the contour generator is the same as the normal vector to the contour which exists on the image plane.

We can now obtain surface orientations on an occluding boundary. The above discussion guarantees that a normal of a contour is the same as a normal of the contour generator. Thus, if we observe an occluding contour in an image plane, we can determine the surface orientations to be the perpendicular direction to the contour line, and the vectors always lie on an image plane.

3.2.2 Self Shadow Boundary

Unfortunately at a self shadow boundary we cannot determine surface orientations as clearly as at an occluding boundary. Still, there exist some particular points on self shadow lines which give us exact information about the surface orientation. If we make some assumptions, we can get approximations for surface orientations. This is enough because the iteration method will itself determine exact values.

For at least three points, the surface normal is determined uniquely. With very few exceptions, when a self shadow line is perpendicular to the direction of the light source, the surface normal lies on the plane which consists of the light ray and the line of sight. So the direction of surface normal is perpendicular to the shadow line and is foreshortened with the angle between the rays of the light source and the line of sight. Next, a self shadow line always crosses on an occluding boundary. There are always two cross points and we can determine surface orientation there exactly.

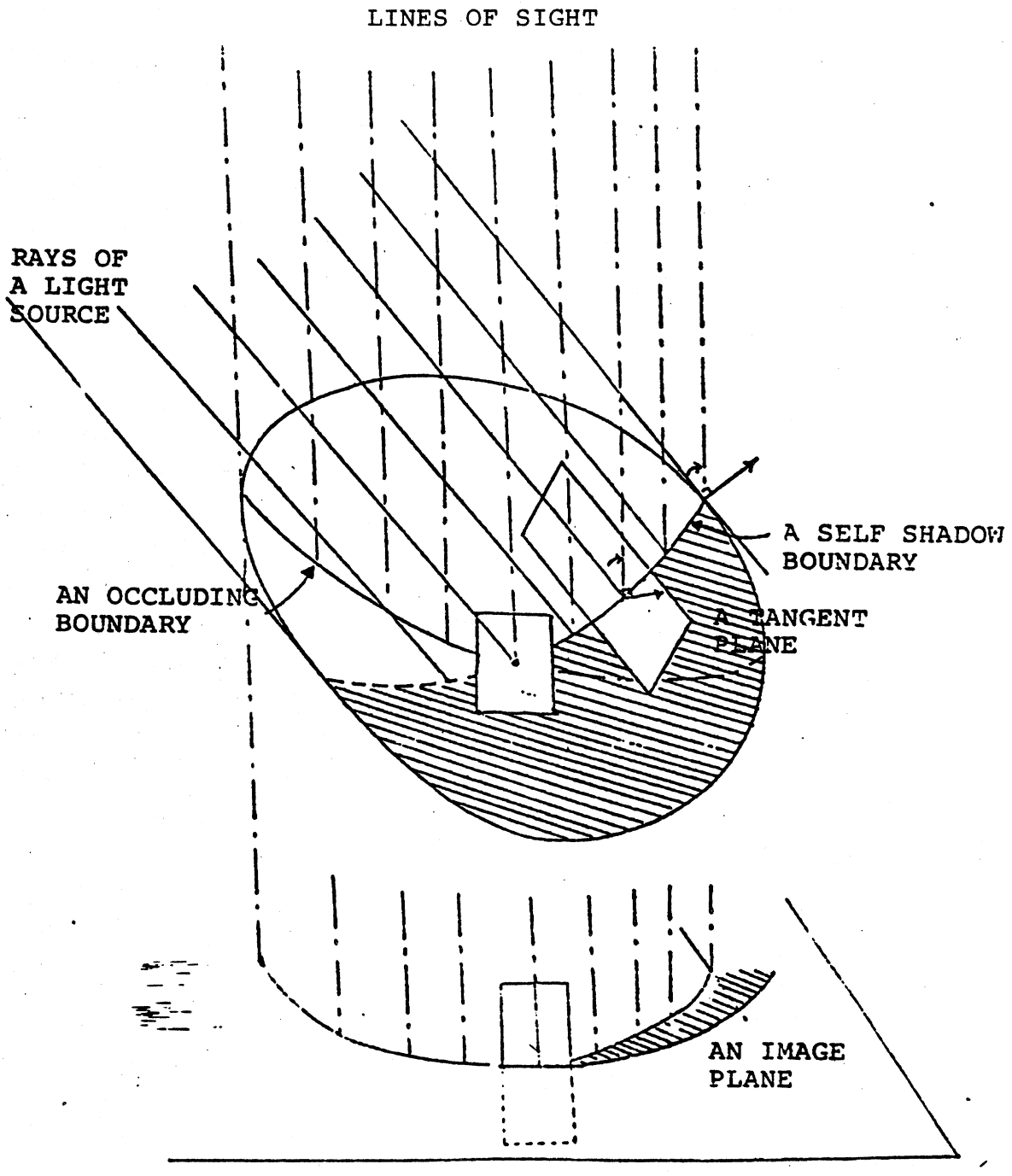


Fig. 8 A diagram on a self shadow line.

At other points, however, we cannot determine surface orientations exactly. At a self shadow boundary, light rays take the role of the line of sight in the above discussion. Namely, a surface normal is perpendicular to the light lines. But the tangent plane is not projected as a line on the image plane. Usually a tangent line to a self shadow line from a viewer is different from the tangent line from the light source.

One possibility is to assume that a self shadow line lies on a plane perpendicular to the light source. This situation occurs when an object is a sphere. Fig. 9 shows the difference between the actual surface orientation and surface orientations based on this assumption. Clearly, when an object is sphere-like, there are few errors. Also in the case of the light source near the viewer, the errors are small.

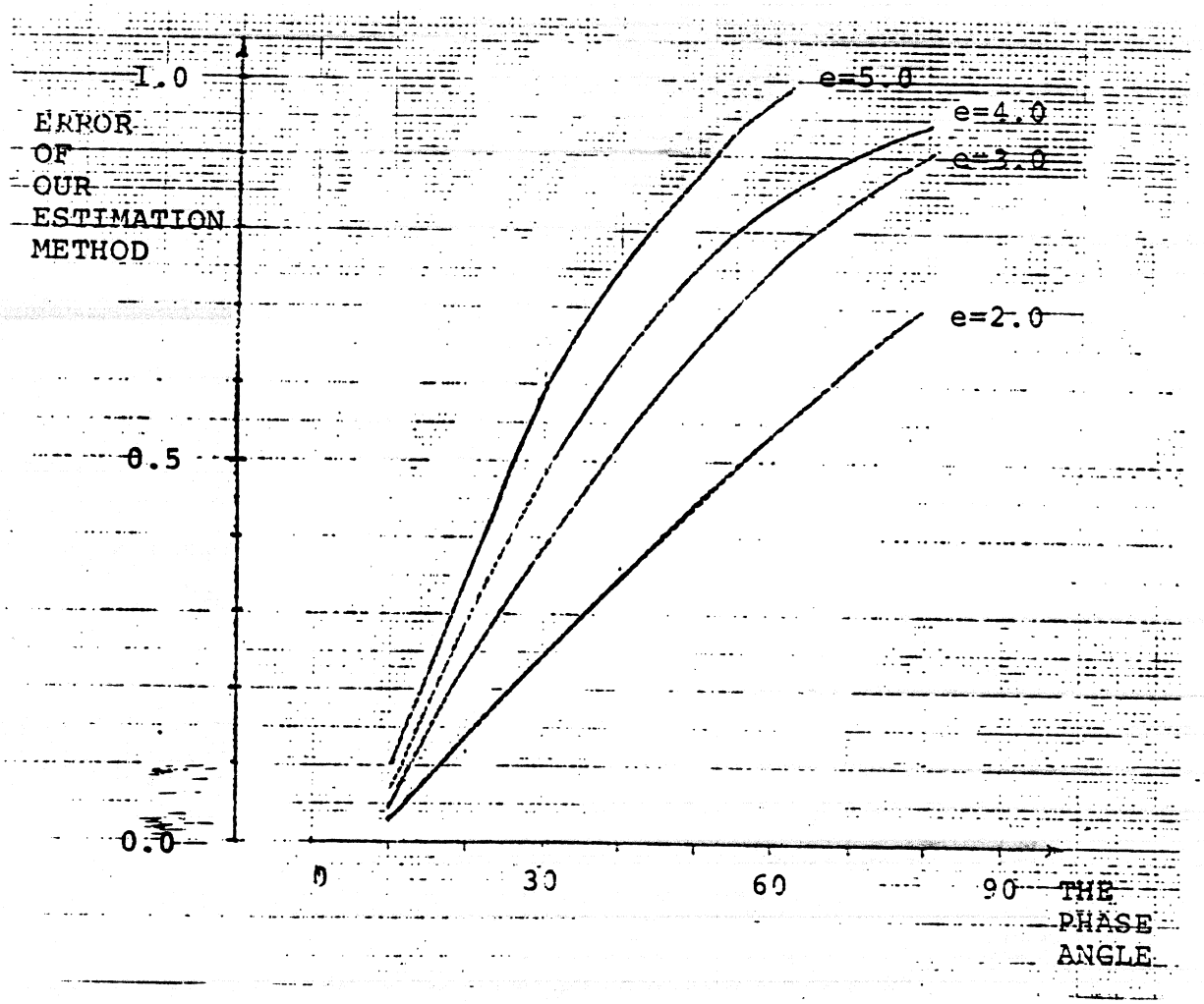


Fig. 9 Errors of our estimation method about surface orientations on a self shadow line.

Globally, there is no problem in the self shadow error. In the final stage, the iteration method finds appropriate surface orientations at this self shadow boundary based on the smoothness assumption and occluding information. These estimates on a self shadow boundary accelerate the convergence speed of the iteration.

3.2.3 Specular Point and Singular Point

At a specular point and a singular point, we can also determine surface orientations. A specular point exists about half way between the origin of the plane and the source point near the origin. More precisely, in the gradient plane

$$(7)\quad p_s = p(\sqrt{(1+p^2+q^2)}-1)/(p^2+q^2),$$
$$q_s = q(\sqrt{(1+p^2+q^2)}-1)/(p^2+q^2).$$

In the stereographic plane,

$$(8)\quad f_s = f(\sqrt{(1+f^2+g^2)}-1)/(f^2+g^2),$$
$$g_s = f(\sqrt{(1+f^2+g^2)}-1)/(f^2+g^2).$$

In Azimuth Zenith plane,

$$(9)\quad r_s = r/2,$$
$$\phi_s = \phi.$$

A singular point means that at that point the reflectance map takes either a maximum value or a minimum value and that area is contained within a constant brightness contour on a reflectance map. For example, the origin of the reflectance map for a lambertian surface in the case where the source is near the viewer is such a point. It is obvious that at this point we can determine the surface orientation uniquely. Actually, Horn[1] used this singular point for the initial conditions of his method.

3.3 Image Intensity Equations

We used to use the image intensity equations as constraints[1]. Solutions of these equations consist of a set of possible surface orientations under a particular surface brightness at a point. More formally,

$$(10)\quad R(p,q) = I(x,y),$$

where (p,q) denotes the possible set of orientations and I(x,y) is the intensity value at point (x,y).

Since we are working on the stereographic plane rather than in gradient space, the equations are slightly different from Horn's equation. Namely, (p,q) are not partial derivatives of z with respect to x,y but rather (f,g) given in Eq. 2 in the stereographic plane and (r,ϕ) given in Eq. 3 in the azimuth-zenith plane. However, there is no difference in the basic idea that surface brightness restricts the possible surface orientations. Only the formalization is different.

4 PROPOSED ALGORITHM AND NUMERICAL EXPERIMENTS

We will construct an algorithm from the above mentioned constraints. Our iteration algorithm consists of two parts; one comes from the image equations and the other from the "average" constraints. We will use the constraints from the boundary as the initial conditions of the algorithm.

Then, we will check whether our algorithm converges or not. Theoretically, this constraints works. In practice, however, there is no guarantee of the convergence; it often happens that digitization errors cause the solution to oscillate, or the solution which we want may be a local minimum rather than a global minimum, and from the usual initial values we cannot reach the desired one. We have done some numerical experiments in order to see if our algorithm converges. The experiments were done under two different conditions; in the first case all information given to the algorithm is exact, in the second, some information is erroneous. In the former case, the reflectance map is correct. At all closed boundaries, surface orientations are given correctly. The algorithm knows the exact direction of the light source. This case checks to see whether the algorithm can obtain correct solutions.

Since the erroneous situation frequently occurs in practical applications, it is also worth while to check whether the algorithm works under that unfavorable situation. For the latter case, at some boundaries there is no effective information about surface orientations, a reflectance map is incorrect and the direction of the light source is different from the real situation. In practical applications, this situation can be interpreted as follows. Sometimes a part of an object may be overlapped and cannot be seen by a viewer. It may also occur if we use a reflectance map derived from some kind of mathematical model slightly different from the actual one. If this caused divergence of the algorithm, we could not use the algorithm in practical applications. Although the solution may be different from the real one, the important point is that the algorithm does not diverge. Obtaining a solution is more important than not being able to get a solution. We will also examine how much error is caused by the wrong information.

4.1 Proposed Algorithm

We will use a cooperative algorithm to determine surface orientations using image equations and "average constraints". It is reasonable to define the following error function $E_{i,j}$ at each mesh point, and to seek a solution which minimizes a summation of $E_{i,j}$ over all the mesh points in an object. Namely,

$$(11)\quad E_{i,j}=(I_{i,j}-R(F_{i,j},G_{i,j}))^2+\lambda\{(F_{i+1,j}+F_{i-1,j}+F_{i,j+1}+F_{i,j-1}-4F_{i,j})^2+(G_{i+1,j}+G_{i-1,j}+G_{i,j+1}+G_{i,j-1}-4G_{i,j})^2\},$$

where i,j is a mesh point, I is a brightness value, F,G are surface orientations,and λ is a scale factor to bring the arbitrary units of image intensity equations and "average constraints" in line. The first factor comes from the image intensity equations, and the second and third factors are derived from the surface smoothness constraints. If you use stereographic

projection, F=f and G=g. In the Azimuth-Zenith plane, F= r and G= ϕ. Needless to say, it also works in gradient space; in that situation, F=p and G=q. We will minimize

(12) $E = \Sigma_i\Sigma_j E_{i,j}$

We will take the partial derivative of E with respect to $F_{i,j}$ and $G_{i,j}$ at each mesh point, and set them to zero, because each $F_{i,j}$ and $G_{i,j}$ are independent variables and the partial derivatives are zero at a minimum value of E. Namely,

(13) $$\partial E/\partial F_{i,j}=2(I_{i,j}-R(F_{i,j},G_{i,j}))(-\partial R/\partial F_{i,j})+\lambda(4F^*_{i,j}-4F_{i,j})(-8F_{i,j})=0,$$
$$\partial E/\partial G_{i,j}=2(I_{i,j}-R(F_{i,j},G_{i,j}))(-\partial R/\partial G_{i,j})+\lambda(4G^*_{i,j}-4G_{i,j})(-8G_{i,j})=0,$$

where

(14) $$F^*_{i,j} = (F_{i+1,j} + F_{i-1,j} + F_{i,j+1} + F_{i,j-1})/4$$
$$G^*_{i,j} = (G_{i+1,j} + G_{i-1,j} + G_{i,j+1} + G_{i,j-1})/4.$$

If $(F_{i,j},G_{i,j})$ is a solution, the partial derivatives are zero. Our job is to solve equations Eq. 13.

We use the Gauss-Seidel Method in order to solve Eq. 13. This method is called "an optimistic method". According to that method, we rearrange Eq. 13 in the following form;

(15) $$F_{i,j} = F^*_{i,j} + (\lambda/16)(I_{i,j}-R(F_{i,j},G_{i,j}))\,\partial R/\partial F_{i,j}$$
$$G_{i,j} = G^*_{i,j} + (\lambda/16)(I_{i,j}-R(F_{i,j},G_{i,j}))\,\partial R/\partial G_{i,j}$$

Then, optimistically, the Gauss-Seidel Method assumes the following; At the limit there is no difference between solutions at the n+1 st iteration and solutions at n th iteration. Therefore, we calculate the right hand side of Eq. 15, by using solutions at the n th iteration and can regard the n+1 st solution as the left hand side. We will finally approach real solutions iteratively! Thus,

(16) $$F_{i,j}^{n+1} = F^*_{i,j}^{n} + (\lambda/16)(I_{i,j}-R(F_{i,j}^{n},G_{i,j}^{n}))\,\partial R/\partial F_{i,j}^{n}$$
$$G_{i,j}^{n+1} = G^*_{i,j}^{n} + (\lambda/16)(I_{i,j}-R(F_{i,j}^{n},G_{i,j}^{n}))\,\partial R/\partial G_{i,j}^{n}$$

From some initial values, we will calculate the n+1 st solutions from the n th solutions, using Eq. 16 .

Strictly speaking, we use a slightly different formula than Eq. 16 in order to prevent the "checker effect". We used $F^*_{i,j}$ and $G^*_{i,j}$ instead of $F_{i,j}^{n}$ and $G_{i,j}^{n}$, in calculating the second terms of Eq. 16. Namely,

(17) $$F_{i,j}^{n+1} = F^*_{i,j}^{n} + (\lambda/16)(I_{i,j}-R(F^*_{i,j}^{n},G^*_{i,j}^{n}))\,\partial R/\partial F_{i,j}^{n}|F^*_{i,j}^{n},G^*_{i,j}^{n}$$
$$G_{i,j}^{n+1} = G^*_{i,j}^{n} + (\lambda/16)(I_{i,j}-R(F^*_{i,j}^{n},G^*_{i,j}^{n}))\,\partial R/\partial G_{i,j}^{n}|F^*_{i,j}^{n},G^*_{i,j}^{n}$$

Eq 17 are our formula, and can be diagrammed as in Fig. 10.

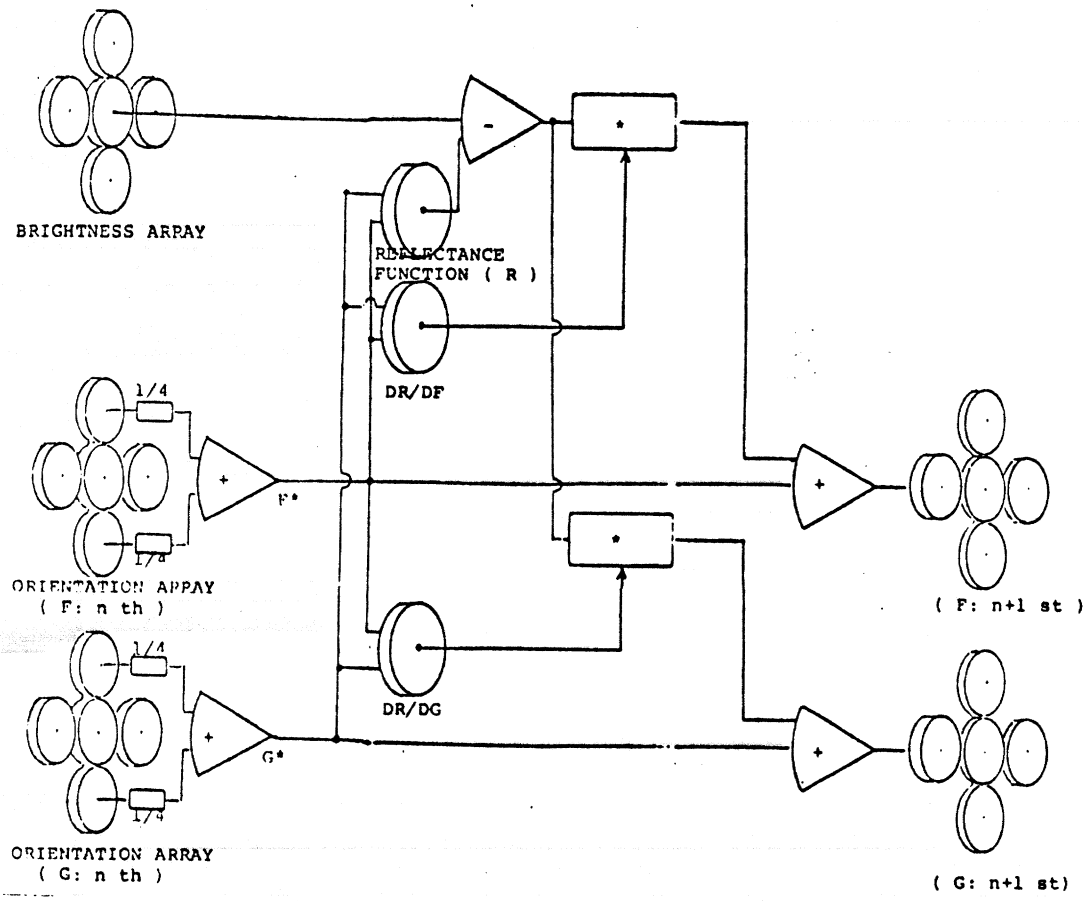
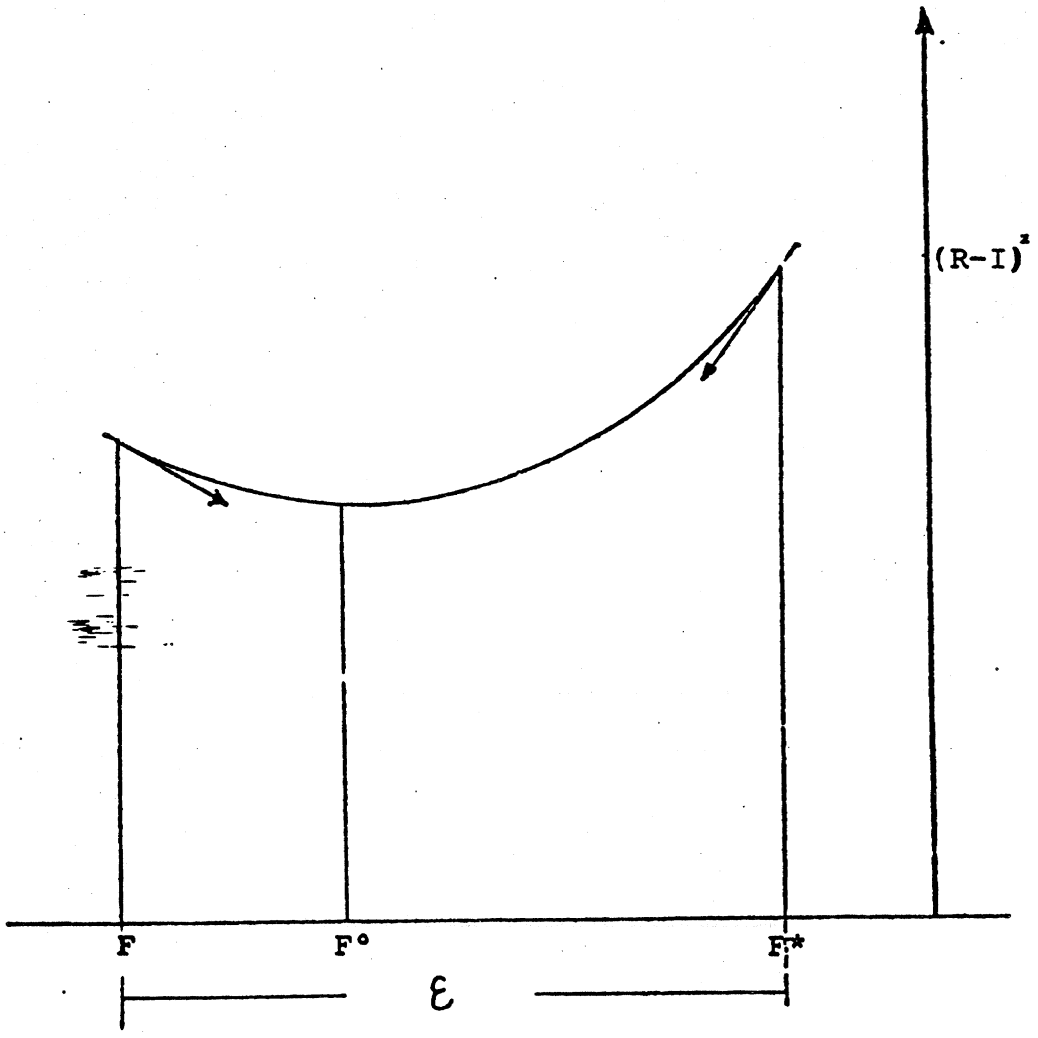


Fig. 10 Diagram of our algorithm.

The reason changing Eq. 16 to Eq. 17 is that Eq. 16 may cause the checker effect. This can be explained as follows. Theoretically, there is no difference between F and F' at the limit. F' can become F at the limit on a continuous plane. We, however, digitized the plane, and the discussion in Sec 3.1.2 tells us that there is a small difference ε according to mesh size δ (see Fig. 11). If there exists an optimal point between F and F', the old formula tells us to go to the right hand side from F' because the old formula calculates the direction using F rather than F'. If this situation occurs in all mesh point, the entire points result in oscillation. Fortunately, the new formula calculates the direction by using F' and moves from F'. So there is no possibility that the checker effect occurs in Eq. 17. At initial stages, F and F' are far from an optimal point, and there is no difference in using F or f'. This explanation is also verified by the fact that a checker effect always occurs at the nearly final stage of iteration rather than at the initial stage.



4.2 Numerical Experiments

At first, we will give the algorithm the exact brightness distribution and exact reflectance map to check whether our algorithm converges or not. Then we will see what happens if boundary information is not given, if the light source, if direction is estimated incorrectly, or if the reflectance characterisity of an object is estimated incorrectly.

4.2.1 Exact Information

(Experiment 1; To check whether the algorithm converges on the digitized image.)

At first let us consider the simplest example-a lambertian sphere is illuminated by a single distant light source with a viewer located near the light source so that only the occluding boundary and the illuminated area are observed. The brightness array (a synthesized image) will be derived analytically from the reflectance characteristic of a lambertian surface and the surface orientations. The algorithm will be applied to the synthesized image.

The first task is to determine the reflectance map. There are three candidate planes for the reflectance map; one is the stereographic plane, the second is the azimuth-zenith plane, and the third one is the gaussian sphere itself. Since a middle point ($(x_1+x_2)/2$, $(y_1+y_2)/2$, $(z_1+z_2)/2$) between (x_1,y_1,z_1) and (x_2,y_2,z_2) does not exist on the surface of the gaussian sphere, the average operation requires a special treatment. Without this point, however, there is no difference among the planes and the surface. Actually, convergence speed is the same in them. In the stereographic plane (f,g),

(18) $$R_{stereo}(f,g) = \{(1-f_s^2-g_s^2)(1-f^2-g^2)\}/\{(1+f_s^2+g_s^2)(1+f^2+g^2)\} + 4(ff_s+gg_s)/\{(1+f^2+g^2)(1+f_s^2+g_s^2)\}.$$

If we use the azimuth-zenith plane (r,ϕ),

(19) $$R_{azimuth-zenith}(r,\phi) = \cos\pi r\cos\pi r_s+\sin\pi r\sin\pi r_s(\cos\phi\cos\phi_s+\sin\phi\sin\phi_s).$$

On the gaussian sphere directly,

(20) $$R_{gauss}(x,y,z) = (xx_s+yy_s+zz_s),$$

provided that $x^2+y^2+z^2=1$, and $x_s^2+y_s^2+x_s^2=1$.

Next, we determine the surface orientation at each image point. Let the radius of the sphere be 1 without loss of generality. Then the equation of the sphere is

(21) $$x^2+y^2+z^2=1.$$

From $(\partial z/\partial x)=-(x/z)$ and $(\partial z/\partial y)=-(y/z)$, and using Eq. 19 or Eq. 20 , we can obtain (f,g) or (r,ϕ) at each point (x,y). The image brightness I is

(22) $$I(x,y) = \max\{ 0, R_{stereographic}(f(x,y),g(x,y)) \}$$

or

$$\max\{ 0,R_{azimuth-zenith}(r(x,y),\phi(x,y))\}$$

or

$$\max\{ 0,R_{gauss}(x,y,z(x,y)) \}.$$

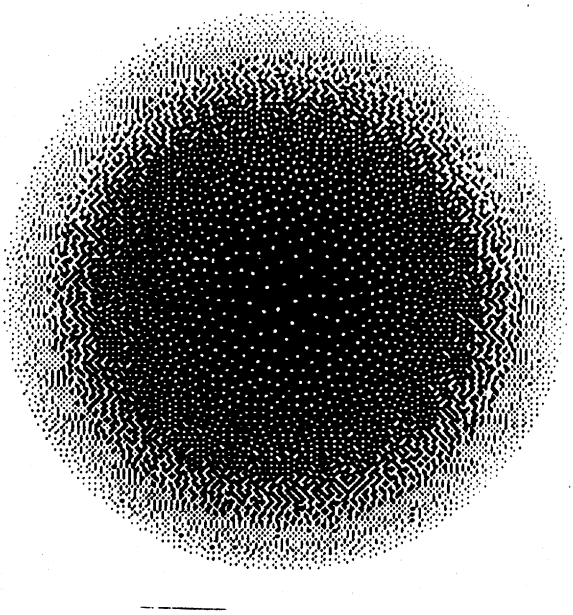


Fig. 12 A synthesized image of a sphere.

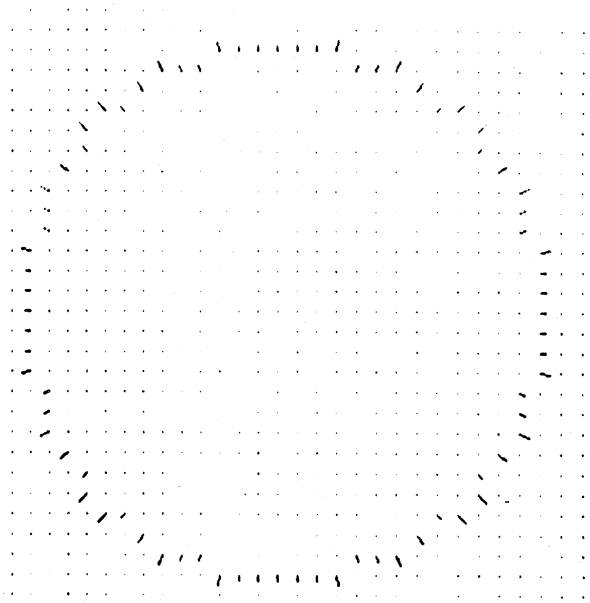


Fig. 13 Initial stage.

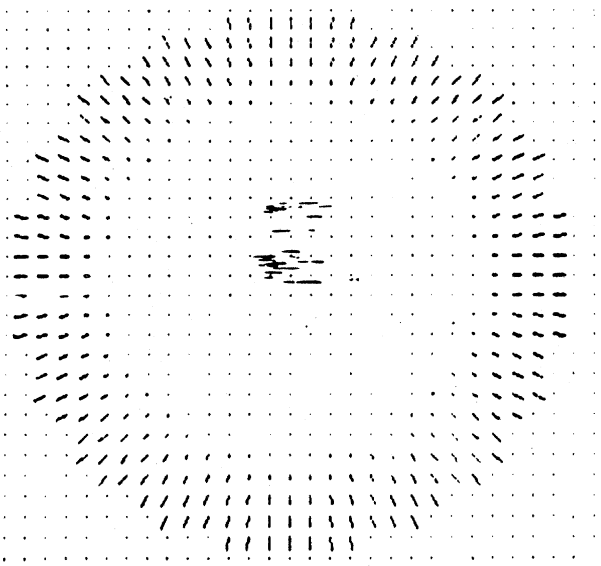


Fig. 14 Five times iteration.

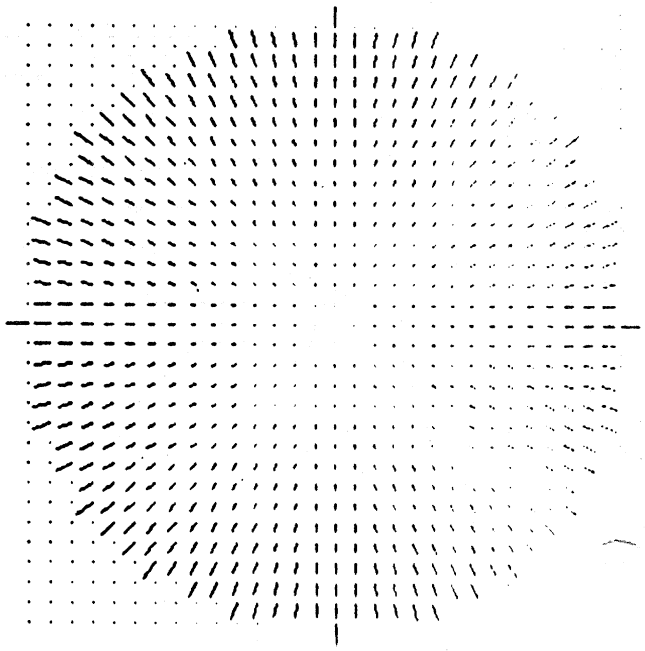


Fig. 15 A result after 30 times iterations

Fig 13 is the initial state of the algorithm. The discussion in Sec. 3.2.1 tells us that at an occluding boundary, surface normals are perpendicular to the contour line and lie on the image plane. We can determine surface orientations at the occluding boundary.

The algorithm requires at least as many iterations as the mesh number. This can be explained as follows. At a point near an occluding boundary, one needs information from the other occluding boundary to converge to the exact orientation. At each iteration, a particular piece of information propagates only to its neighboring points. This means that it takes as many iterations as mesh numbers for one boundary value to reach another boundary. In our case, the mesh size is thirty and this implies that we have to do more than thirty iterations. Fig 14 shows the results at 5 iterations. Fig 15 is the result after 30 iteration. Fig 16 indicates the difference between the sample value and the output of the algorithm at each iteration; it gradually decreases, and after twenty times, there are no errors (at thirty iterations, less than 0.01%). Fig. 17 is the generated surface. Namely, fig 12 is the input to the algorithm and fig. 17 is the output from the algorithm.

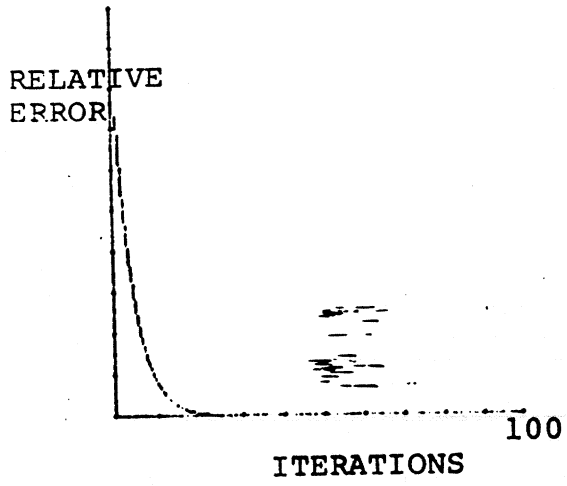


Fig. 16 Relative errors.

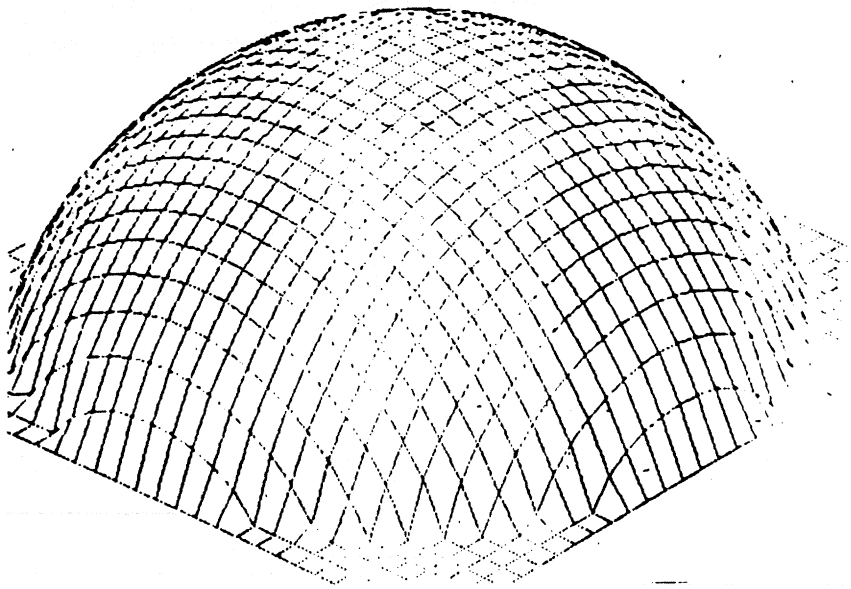


Fig. 17 A Generated surface.

(Experiment 2; Effect of initial values)

We propose to use orientations of singular values as the initial value. It is desirable that until the boundary information propagates to a point, orientation values at that point do not begin to change. At the singular point, the partial derivatives are always zero. This means that there is no effect from the constraints of the image intensity equations in Eq. 13. Though it is simple to set all points to (0.0,0.0) initially, there usually exists the partial derivatives at that point and each mesh begins to move without considering the boundary conditions at the first iteration. (At the first iterations, the boundary information propagates only to mesh points inside of the boundary. Deep inside, there is no information yet from the boundary.) Sometimes this situation requires more iterations to converge.

This second example illustrates the above mentioned initial value effect. An egg is illuminated in the direction near the extension of the short axis, and the viewer is looking in the direction of the long axis. The reason we use an egg shape is to demonstrate that our algorithm works for some object besides spheres. The ratio of the long axis to short axis is 3. The precise position of the light source is $f_s = 0.5$ and $g_s = 0.0$ or $r = 0.59$ (53.1 degree) and $\phi = 0.0$. In order to make the situation simple the surface orientations are given at the self shadow boundary initially. Fig. 19 shows the resulting needle diagram and fig. 20 show the relative errors, comparing the effect of initial values. When we set singular values as initial conditions, convergence speed is more rapid than (0.0,0.0) case.

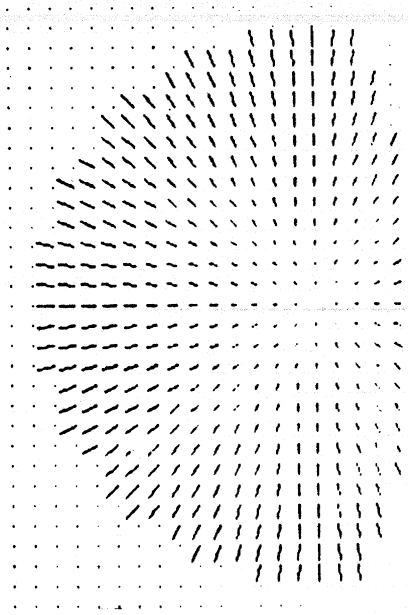


Fig. 18 Needle diagram

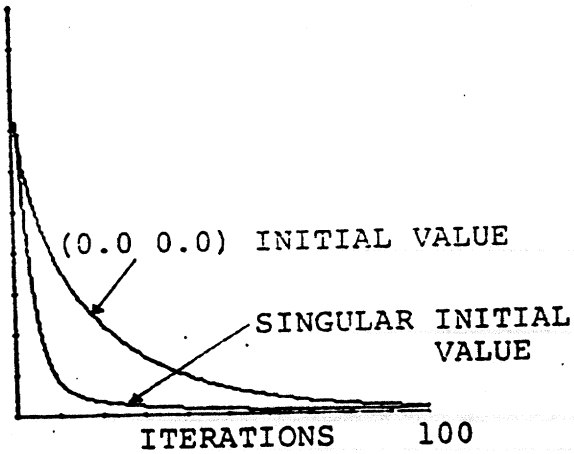


Fig. 19 Relative error.

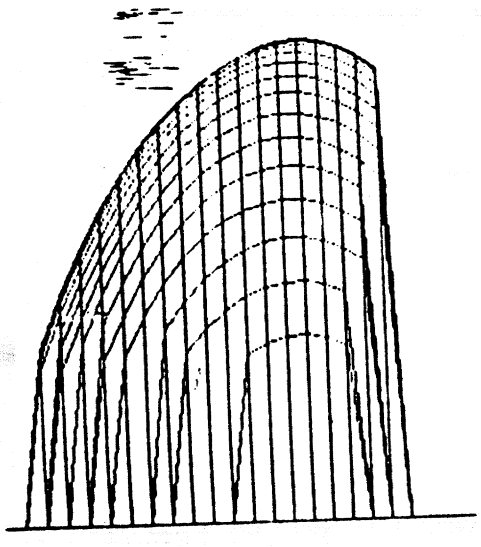


Fig. 20 A generated surface.

(Experiment 3; Negative Gaussian curvature)

The third example is a hyperbolic shape. The previous two examples treat surfaces whose Gaussian curvature is positive. On the other hand, a hyperbolic surface has negative Gaussian curvature making it worthwhile to check this situation. Unfortunately we cannot find occluding boundaries in this surface. We instead give orientations to the algorithm on a closed loop. The viewer is near the light source. Fig. 22 is the generated surface from the output of the algorithm. It takes about the same number of iterations as the first example. This result illustrates that the algorithm works whether a surface has negative or positive gaussian curvature.

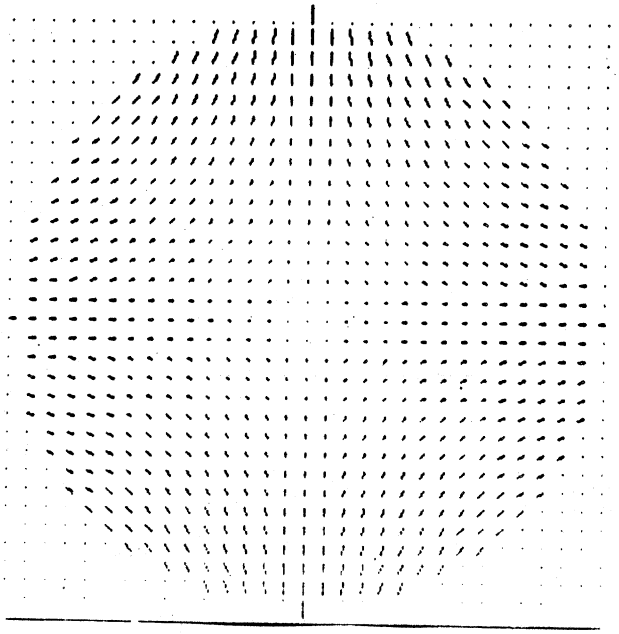


Fig. 21 Obtained needle diagram.

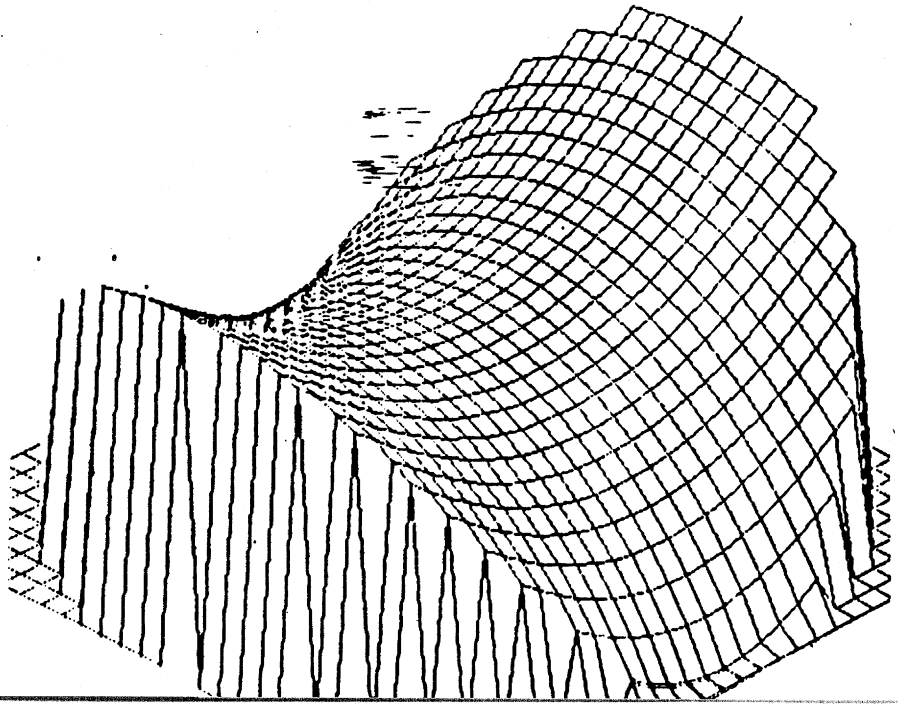


Fig. 22 A generated surface of hyperbolic.

4.2.2 Effect of Erroneous Information

All practical programs should be able to handle erroneous information. In practical applications, all information contains some noise. Some noise may come from devices; other errors may come from incorrect modeling of situations. If an algorithm is sensitive to noisy situations and cannot converge, the algorithm is of no use, even though the algorithm is beautiful and obtains good results in mathematical models. A practical program should be stable globally and manage to converge to a solution even if the solution is slightly different from the real one.

We have performed three kinds of experiments. The first one concerns open boundaries. We can only determine surface orientations at a self shadow boundary analytically at a few points. Often we cannot see the entire surface of an object. Thus, the algorithm must treat erroneous information which comes from an open boundary. In the second experiment we give the algorithm a different light source from the real one. This case is important because the light source direction is often incorrectly estimated. The third case is to use reflectance characteristics that are different in calculating synthesized image and determining surface orientations. For example, the algorithm tries to find solutions using reflectance map based on a Lambertian reflector, though the synthesized image is not calculated based on it. This test is important because there are no perfect Lambertian reflectors in the real world.

(Experiment 4; effect of open boundary)

If we have no boundary information, we cannot know anything about the surface. However, we would like a shape from shading algorithm to converge even when missing information. The answer it gives with inadequate boundaries may not be concert

We set a boundary line free; Needless to say, if we set all boundary lines free, it is impossible for the algorithm to converge. we set self shadow line free. There are two motivations for this situation. One is ,as mentioned above, that we cannot determine surface orientations analytically except at a few points. This example shows an extreme case. So if we can find a heuristic method, the algorithm will do better than this. This example is an lower limit. The other is that it sometimes happens that we cannot see all of the surface of an object. In that situation, we have to do without any information at that boundary and set that boundary free. This case also illustrates that unfortunate situation. Even in these situations, the algorithm should not give up trying to obtain surface orientation.

We imagine that an sphere is illuminated from the direction $(p_s,q_s) = (0.5,0.0)$. The surface is assumed to have Lambertian characteristics. At the shadow line, no information is given to the algorithm except that it is a free line. There are no constraints at the line. Fig. 23 shows how the algorithm obtained a real solution. Namely, if we can have surface orientations at the self shadow line, the information helps the algorithm to converge to a real solution more rapidly. A given closed boundary, however, is not a

necessary condition for the algorithm to converge. A real solution may be obtained under the situation that at some boundary, surface orientations are not clear. Even if an object is overlapped under another object, it is still possible that we can get a real solution.

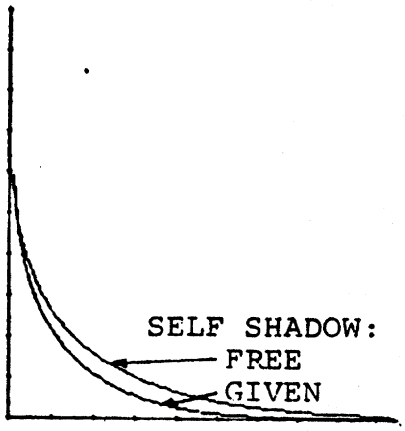


Fig. 23 Effect of an open boundary

(Experiment 5: effect of incorrect estimation of light source direction)

This example treats a situation in which we think the light source direction is different from the correct direction. In practical application we cannot determine light source direction precisely. It is reasonable for us to expect the algorithm to work with such rough information.

A sphere is illuminated from 45 degrees of the zenith angle, when we make a synthesized image, we use 45 degree as the light source direction. Then the algorithm tries to determine surface orientations under the assumption that the direction of the light source is the incorrect direction; from 0 degree to 90 degree of the zenith angle.

Fig. 24 shows the result. The broken line expresses the real solution. About 15 % estimation error of the source direction causes about 20 % error of surface orientations.

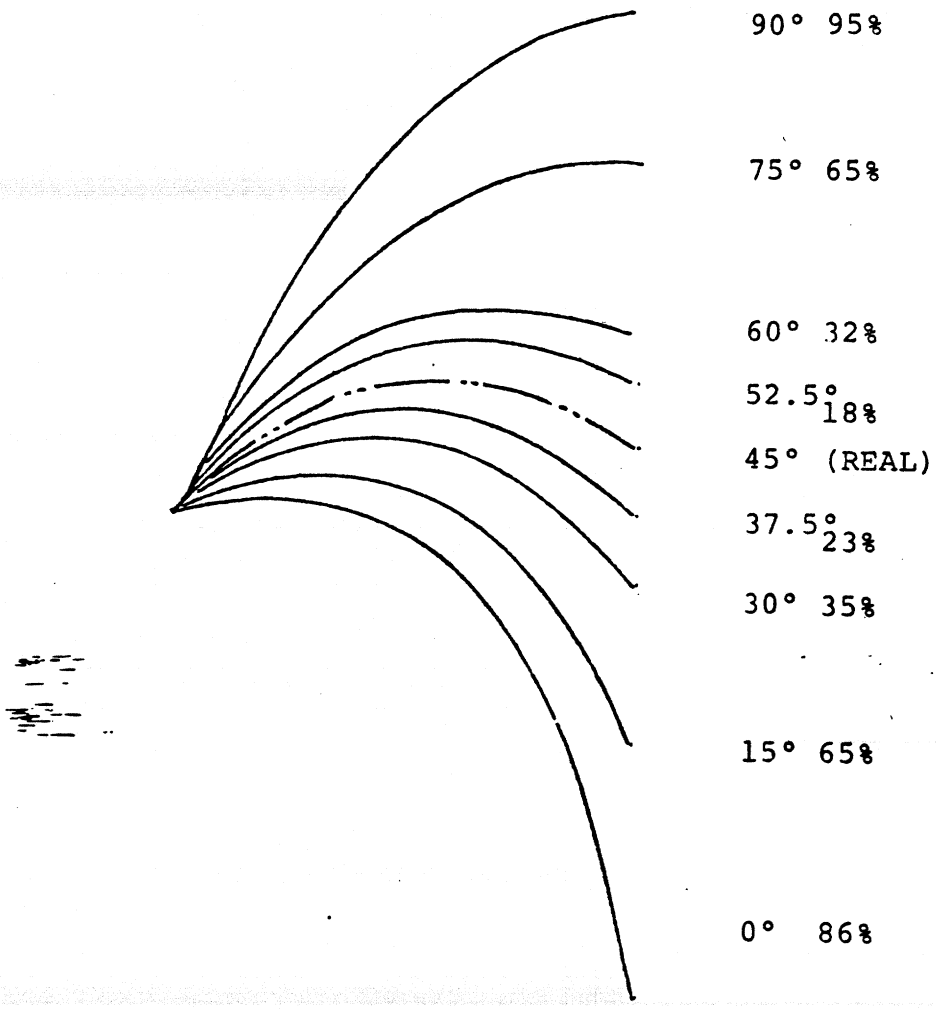
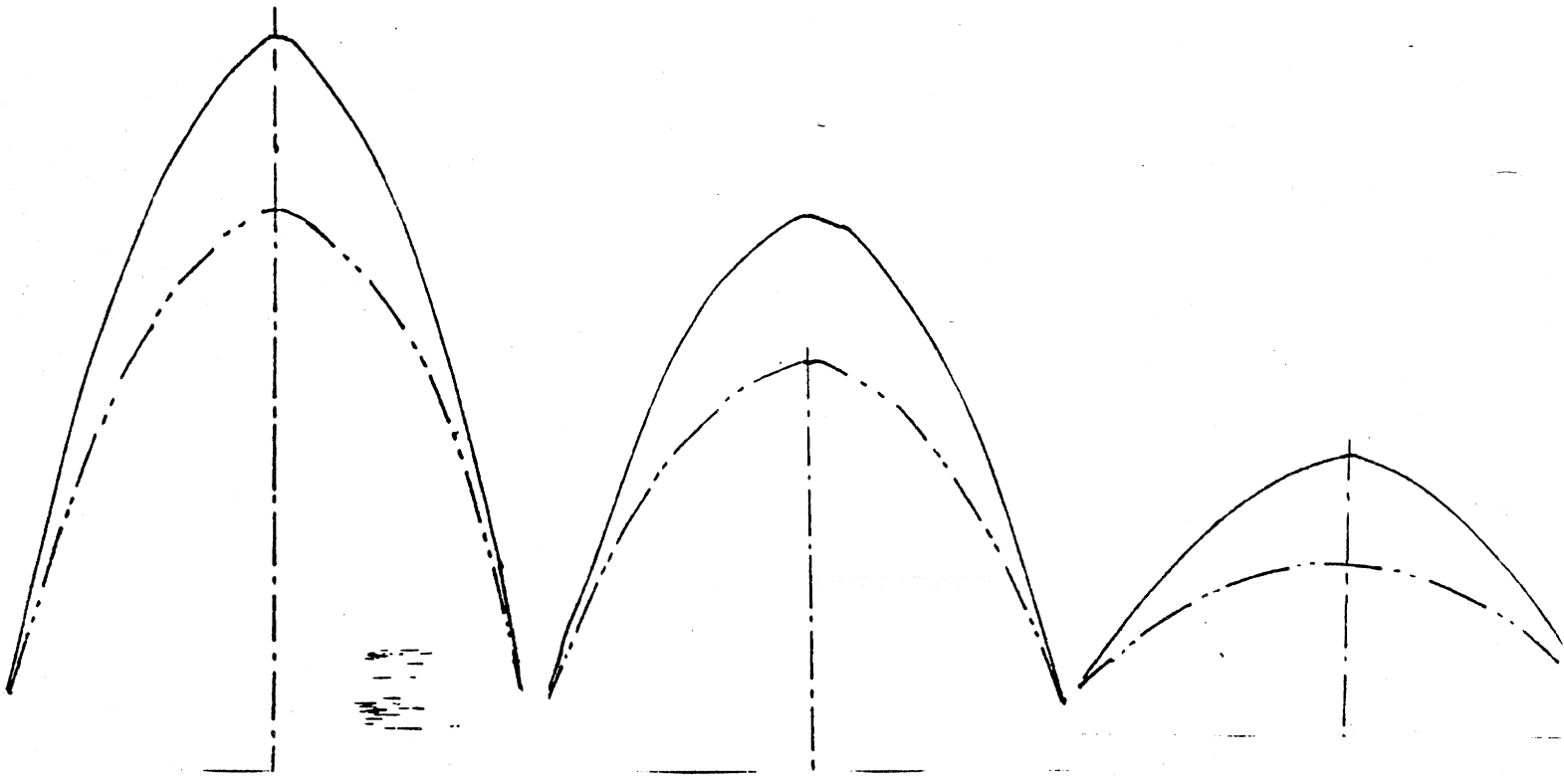


Fig. 24 Effect of incorrect estimation of light source direction

(Experiment 6; effect of incorrect estimation of reflectance characteristic)

We gave the algorithm a reflectance map based on different reflectance characteristics from the real one. Egg shapes (axis ratio is 5, 3 , 1) are illuminated from (0.0, 0.0). The surface has linear characteristics; the reflectance ratio is proportional to the zenith angle. However, we told the algorithm that the surface has a Lambertian characteristic. The output still resembles the original one. See Fig. 25.



An egg-shape of which axis rate is 5. An egg-shape of which axis rate is 3. An egg-shape of which axis rate is

Fig. 25 Effect of incorrect estimation of reflectance characteristic

(Experiment 7; analysis of a SEM picture)

The algorithm analyzes a picture which comes from the output of a SEM (Scanning Electron Microscope) picture. The object is something which exists in a young leaf. The reflectance map is based on a experiment of Laponsky [14]. Fig. 26 shows a binary image of the object. Fig. 27 is a obtained needle diagram. A generated surface is shown in Fig. 28.

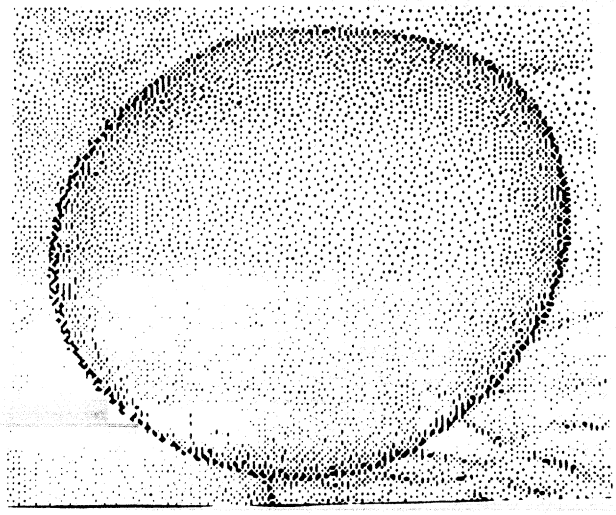


Fig. 26 A SEM picture of a bud on a young leaf.

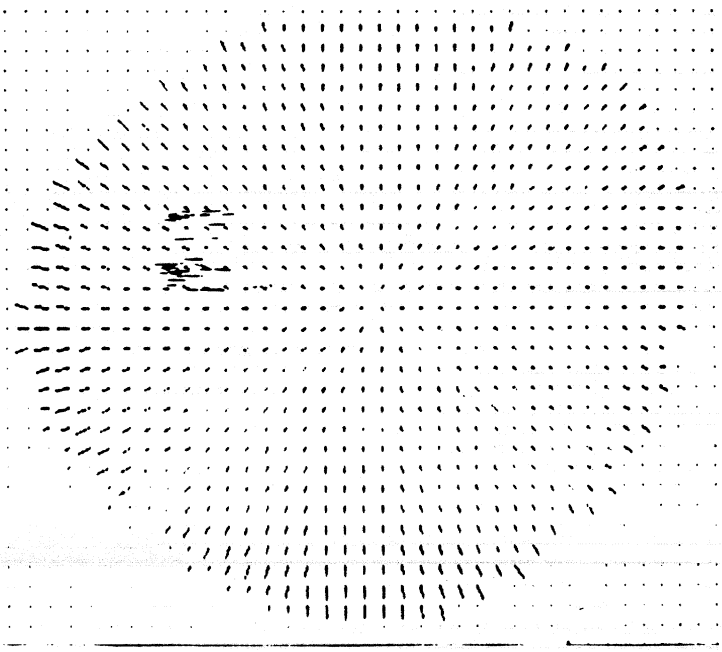


Fig. 27 Obtained needle diagram.

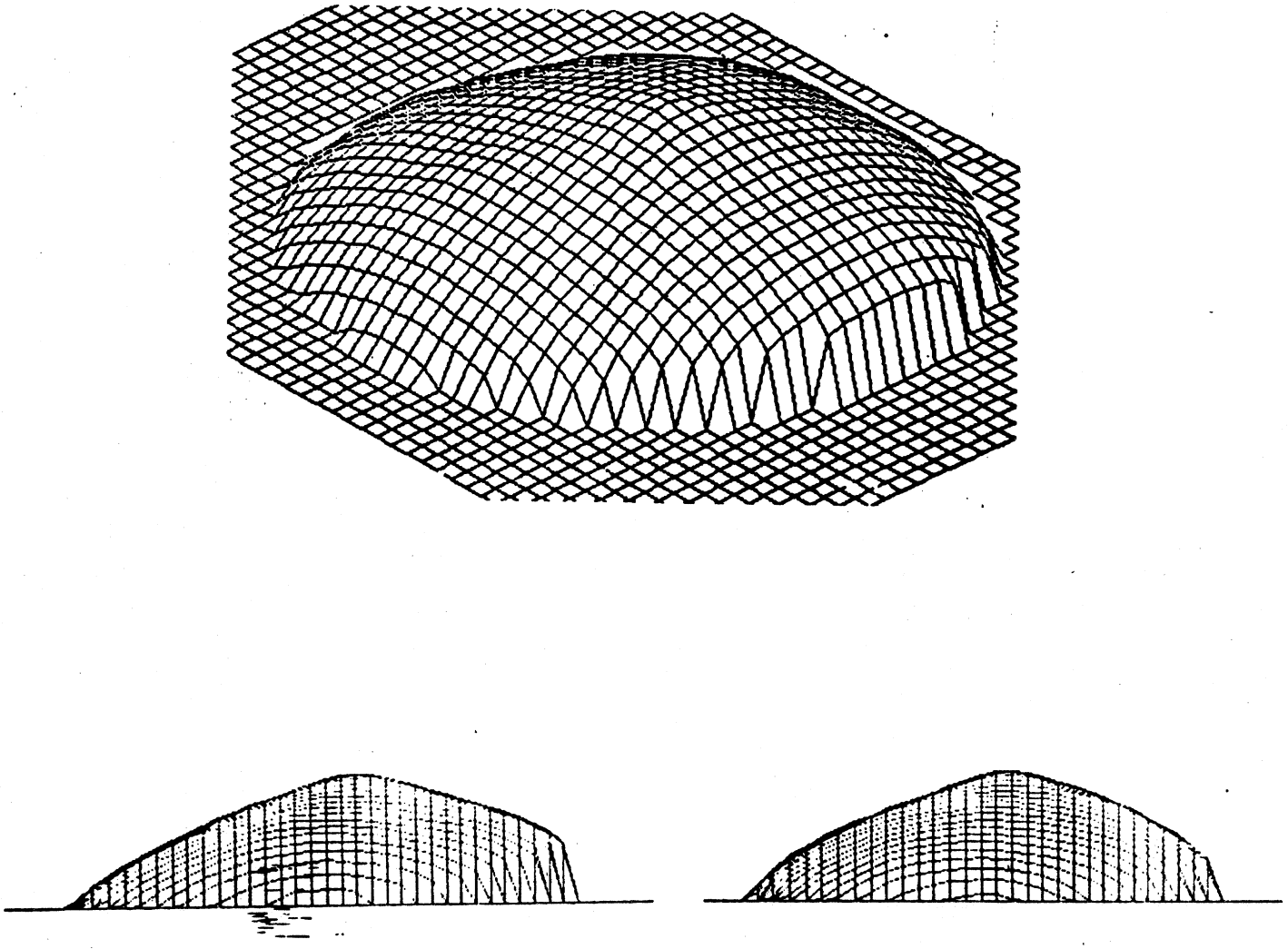


Fig. 28 A generated surface

5 DISCUSSION

5.1 Relevance to basic areas

One use of the algorithm is as a shape finder from shading information. This algorithm is effective especially in the analysis of SEM (Scanning Electron Microscope) pictures as shown in Experiment 7. We can treat SEM situations as that in which a viewer is near the camera. Namely, there is no self shadow line; we observe only an occluding boundary.

The other possible application is as a relaxation tool. Sometimes we prefer to smooth an output by using photometric properties and an assumption of surface smoothness. This algorithm tells us how much effect neighboring points have on a particular point, if we consider that the surface is smooth. For example, if there is a noise spot, the amount of brightness on this point becomes slightly low. Usually table lookup cannot help these situations. The simplest method is to take an average of orientations over the surrounding points. This program, however does more. It adjusts the final result considering both the brightness values and the average orientations.

Output from a photometric stereo system has been smoothed successfully using this method. The photometric stereo system can determine surface orientations based on a triple of brightness values under different light source conditions. If there exists a stain on a surface, the three brightness values become low. The triple of the brightness values does not exist in a lookup table, which converts a brightness triple into surface orientation. This results in the photometric stereo system not determining surface orientation at that point. This algorithm, however, considers neighborhood of the point. And if in this area the photometric stereo system has already found surface orientations, and the brightness value is not too low, the smoother determines surface orientation by considering "average constraints" and the three constraints which come from the intensity equations. Three terms from the intensity equations can be interpreted as being in competition to attract the point to each, and to bring the point to a reasonable place. If at the surrounding point the photometric stereo system cannot determine surface orientation, the smoother also does not determine surface orientations at that point. Namely, we cannot make average values and cannot obtain output from the smoother. Actually, by using this method; an output from the photometric stereo system is smoothed (Compare fig. 10.1.2 with fig. 10.1.4 in [11]. Fig. 10.1.2 is output directly from the photo metric stereo system. Fig. 10.1.4 is output from this smoother.) In that case this algorithm not only did smoothing but also extended a possible area where solutions were found.

5.2 Relevance to Basic Areas

A fairly rough reflectance map still works. As you can see in Experiment 5, if you can tolerate estimation errors around 50%, the reflectance map may be allowed up to 15 degrees estimation error in the direction of a light source. Also experiment 6 tells us that in that error range it does not matter whether a surface has a Lambertian characteristic or linear characteristics. Thus, if you can allow 20% estimation errors, you need only six or seven reflectance maps with respect to a light source direction, and only two or three reflectance maps with respect to reflectance characteristics. Contours on the reflectance map only depend on the change of zenith angle of the light source. The change of azimuth angle causes only a rotation of the map. You can rotate the gaussian sphere in order to make consistent the light source direction with the direction of the light source on the gaussian sphere. It is possible because a change of azimuth angle can be treated by rotating the sphere with respect to the viewer line. So there is no need for you to make reflectance maps which are different in the azimuth angle of the light source. Since only the zenith angle of the light source affects the distribution of brightness value on the gaussian sphere, we have prepare reflectance different in the zenith angle of the light source. We do not mind 20% errors. It allows us to divide 90 degrees by each 15 degrees. Finally, we have to prepare six reflectance maps. With respect to a reflectance characteristics; one is a matte surface and the other is a metal surface. These two is enough. Because all matte surface can be treated as a Lambertian surface. This causes only estimation errors around 20 %. We have not done a experiment about a specular surface. We can predict same level of errors based on the result from the characteristic of our algorithm.

The following four questions arise from the fact that a fairly rough reflectance map still works.

(1) how precisely can a human being determine a light source direction ?

(2) how precisely can he determine surface characteristics ?

(3) how precisely can he determine surface orientations ?

(4) how well can a child do ?

The relationship between the first point and the third point between the second point and the third point may reveal whether or not a human being uses a reflectance map technique, and, if so, how rough a reflectance map he may use. It would be interesting to compare the ability of detecting axes ratios of an egg shape between a human being and a program under incorrect estimation of the directing of a light source just as in Experiment 5.

Does all the visual ability come by nature? It seems to me that some of them are acquired by learning, especially, abilities like "shape from shading" and "shape from texture". On the other hand, "color perception" obviously come by nature. The fourth question may reveal whether or not the reflectance map is acquired by learning or by nature. In particular, if it is acquired by learning a child under some age may not be able to detect

surface orientations from shading information. The next question would be, if it is acquired by learning, at what period does a child obtain it. Possibly it is acquired at the third period [12] (the period of the sensory-motor schematism). At the third period, a child often wants to touch everything that he "sees". Also, according to Piaget, at that period, a child engages in "experiments for observations". That is, without any object, he tries to do a same task with changing initial conditions and observes the result many times. It might be that he is comparing the output of his vision system with real shape or real distance. He is trying to adjust his hand eye system and to establish it. Another interesting question is; how well the other animals do? It is an interesting to explore how many kinds of visual ability the other animal has.

6 CONCLUSION

We proposed an algorithm to compute surface orientations from shading information iteratively. We also showed that the algorithm computes surface orientations successfully.

ACKNOWLEDGMENT

The author would like to extend sincere appreciation to Prof. B.K.P. Horn for good advice and suggestions and Prof. P.H. Winston for encouragement. Discussions with K. Murota of the University of Tokyo, R. Sjoberg, and W. Silver of MIT were helpful. Thanks go to E. Hildreth, E. Grimson and K. Forbus of MIT for helping to proofread the manuscript. We are also grateful for the construction of the Lisp Machine by the Lisp Machine group at the AI Lab., MIT. Without the Lisp Machine, we could not have completed this research so rapidly.

REFERENCES

[1] Horn, B.K.P., "Obtaining shape from shading information," in *The Psychology of Computer Vision*, P.H. Winston(Ed.), McGraw-Hill, New York, 1975.

[2] Horn, B.K.P., "Understanding image intensities," *Artificial Intelligence*, 8, No. 2, 1977, 201-231.

[3] Horn, B.K.P. and Sjoberg, R.W., "Calculating the Reflectance Map," *Applied Optics*, Vol. 18,No. 11, 1979.

[4] Woodham, R.J., " A cooperative algorithm for determining surface orientation from a single view," *Proc. of IJCAI 1977*, 635-641.

[5] Strat, T.M., " A numerical method for shape-from-shading from a single image," M.S. thesis, EECS department, MIT,1979.

[6] Brooks, M.J., " Surface normals from closed path," *Proc. of IJCAI 1979*,

[7] Barrow, H.G. and Tenenbaum, J.M., " Recovering intrinsic scene characteristics from images," in *Computer Vision Systems*, Hanson, A. and Riseman, E. (Ed.), Academic Press, New York, 1979.

[8] Huffman, D.A., " Impossible objects as nonsense sentences," in *Machine Intelligence 6*, Meltzer, R. and Michie, D. (Ed.), Edinburgh University Press, 1971.

[9] Carmo, M.P.D., *Differential Geometry of Curves and Surface*, Prentice-Hall, New Jersey.

[10] Marr, D, " Analysis of occluding contour", *Proc. R. soc. London*, Vol. B.197, 1976, 441-475.

[11] Ikeuchi, K and Horn, B, " An application of the photometric stereo method," *Al-memo 539*, MIT-AI Lab.,1979.

[12] Piaget, J *Six Psychological Studies*, Random house, New York, 1967.

[13] Horn, B.K.P, " SEQUINS and QUILLS -- Representation for Surface Tomography," *Al-memo 536*, MIT-AI Lab.,1979.

[14] Laponsky, A.B. and Whetten, N.R, " Dependence of Secondary Emission from MgO Single Crystals on Angle of Incident, *Phys. Rev. Vol 120, No. 3*, 1960.