

Numerical Shape from Shading and Occluding Boundaries

Katsushi Ikeuchi

*Electrotechnical Laboratory, Information Science Division,
1-1-4 Umezono, Sakura-mura, Niihari-gun, Ibaraki-ken 305,
Japan*

Berthold K.P. Horn

*Massachusetts Institute of Technology, Artificial Intelligence
Laboratory, 545 Technology Square, Cambridge, MA 02139,
U.S.A.*

ABSTRACT

An iterative method for computing shape from shading using occluding boundary information is proposed. Some applications of this method are shown.

We employ the stereographic plane to express the orientations of surface patches, rather than the more commonly used gradient space. Use of the stereographic plane makes it possible to incorporate occluding boundary information, but forces us to employ a smoothness constraint different from the one previously proposed. The new constraint follows directly from a particular definition of surface smoothness.

We solve the set of equations arising from the smoothness constraints and the image-irradiance equation iteratively, using occluding boundary information to supply boundary conditions. Good initial values are found at certain points to help reduce the number of iterations required to reach a reasonable solution. Numerical experiments show that the method is effective and robust. Finally, we analyze scanning electron microscope (SEM) pictures using this method. Other applications are also proposed.

1. Introduction

This paper explores the relationship between image brightness and object shape. Much of the work in machine vision does not explicitly exploit the information contained in the brightness values recorded in the image, using

these values only to segment the image, based on the difference of average brightnesses in adjacent regions. A great deal of information is contained in the image brightness values, however, since it has been shown that image brightness is related to surface orientation [Horn, 1975].

The problem of computing the shape of an object from the shading in an image can be thought of as the problem of reconstructing one surface (height of the object above some reference plane) from another surface (brightness in the image plane). Information at each point of the image is used to compute the orientation of the corresponding point on the object, using the assumption that the surface is smooth. This is in contrast with other machine-vision methods which analyze surface depth based on discontinuities in surface orientation and boundary information alone.

1.1. Historical background

The photometric approach to determining surface orientation from image brightness was first formulated in the form of a non-linear first-order partial differential equation in two unknowns [Horn, 1975]. This equation can be solved using a modified characteristic strip-expansion method. This method assumes that the surface is smooth, as does the method described in this paper. We will see, however, that slightly different interpretations of the term 'smooth' are employed.

The reflectance map, introduced later [Horn, 1977], represents the relationship between surface orientation and surface brightness. The map is defined in gradient space, which appeared in the work on scene analysis of line drawings [Huffman, 1971; Mackworth, 1973; Draper, 1980]. We assume orthographic image projection and take the viewing direction as parallel to the z -axis. The shape of the object can be described by its height, z , above the xy -plane. It is convenient to use the short-hand notation p and q for the first partial derivatives of z with respect to x and y :

$$p = \partial z / \partial x \quad \text{and} \quad q = \partial z / \partial y.$$

The pq -plane is referred to as *gradient space*, since every point in it corresponds to a particular surface gradient. Distance from the origin of gradient space equals the slope of the surface, while the direction is the direction of steepest ascent.

Fixed scene illumination, surface-reflectance properties, and imaging geometry can be incorporated into an explicit model that allows image brightness to be related directly to surface orientation. Thus we can associate with each point in gradient space the brightness of a surface patch with the specified orientation. The result, usually depicted by means of iso-brightness contours, is called the *reflectance map*, and denoted $R(p, q)$. The reflectance map can be obtained experimentally using a test object or a sample mounted on a

goniometer stage. It can also be determined theoretically if the surface-reflectance is known as a function of the incident, emittance, and phase angles. The reflectance map can be computed, for example, if the so-called Bi-directional Reflectance Distribution Function (BRDF) [Nicodemus et al., 1977] and the distribution of light sources is known [Horn and Sjoberg, 1979].

The reflectance map is a convenient tool, since it provides a simple representation of the constraint inherent in one image-brightness measurement. Once the brightness, $E(x, y)$, is known at a point, one can ask what the surface orientation might be there. A measurement of image brightness restricts the possible surface orientations at the corresponding point on the surface of the object. This constraint is expressed by the *image-irradiance equation* [Horn, 1977]

$$R(p, q) = E(x, y),$$

where the gradient (p, q) denotes possible orientations and $E(x, y)$ is the brightness measured at the point (x, y) . In general, one measurement cannot give us both p and q . It can only fix a relationship between the two variables. Additional constraint is required for a unique solution. This usually takes the form of some assumption about the class of surfaces that may be allowed.

One effect of the introduction of the reflectance map was that it motivated research on other ways of solving the shape-from-shading problem. It also led to the development of iterative methods, similar to those used to solve second-order partial differential equations. Such methods, now called pseudo-local, relaxation, or cooperative computation methods, were suggested some time ago [Horn, 1970, p. 192], but could not be developed for this application without these new tools.

1.2. Characteristic strip expansion

The old method [Horn, 1975] can be interpreted in terms of the reflectance map, too. In the method of characteristic strip-expansion [Garabedien, 1964; Moon and Spencer, 1969; Carrier and Pearson, 1976; Courant and Hilbert, 1962; John, 1978] the partial differential equation given above is replaced by a set of five ordinary differential equations, one for each of x , y , z , p , and q :

$$dx/ds = R_p \quad \text{and} \quad dy/ds = R_q,$$

$$dz/ds = pR_p + qR_q,$$

$$dp/ds = E_x \quad \text{and} \quad dq/ds = E_y,$$

where R_p and R_q are the partial derivatives of R with respect to p and q , while E_x and E_y are the partial derivatives of E with respect to x and u [Horn, 1977; Bruss, 1979].

The parameter s varies monotonically along a particular characteristic strip.

When the characteristic strip is extended, one moves to a new point (x, y) in the image and a new point (p, q) in the reflectance map [Horn, 1977; Woodham, 1978]. The orientation of the surface is known at the end of the strip. One has to move in a particular direction, (dx, dy) , in the image plane in order to be able to compute the orientation at the new point (see solid arrow in Fig. 1a). The equations above tell us that this direction is the direction of steepest ascent, (R_p, R_q) , at the corresponding point in the reflectance map (see dotted arrow in Fig. 1b). At the new image point so determined, a different value of image brightness is found, corresponding to a new point in the reflectance map. This point in the reflectance map also lies in a well defined direction, (dp, dq) , from the previous point in the reflectance map (see solid arrow in Fig. 1b). From the equations above we know that this direction is the direction of steepest ascent, (E_x, E_y) , at the corresponding point in the image (see dotted arrow in Fig. 1a).

One important observation is that these directions will be computed incorrectly if the brightness measurements are corrupted by noise. The result is that the characteristic strips deviate more and more from their ideal paths as the computation progresses. Thus, while characteristic strip-expansion works in the absence of noise, it suffers from error accumulation in practice. One way to

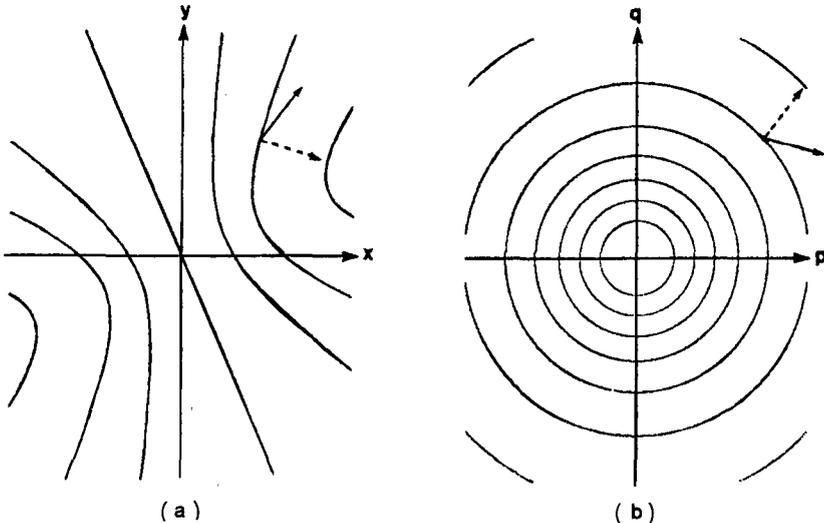


FIG. 1. Geometric interpretation of the equations arising from the application of the characteristic strip-expansion method to the image-irradiance equation. The image is shown in (a) as a set of iso-brightness contours and the reflectance map appears similarly in (b). It is known that the gradient is (p, q) on the surface of the object at the point corresponding to the image point (x, y) . In order to continue the solution from this point, one must take a small step on the image in a direction determined by the gradient of the reflectance map, while the direction of the corresponding step on the reflectance map is similarly determined by the image brightness gradient.

greatly reduce this effect is to expand neighboring strips simultaneously and to adjust the solution using the assumption that the surface is smooth between strips [Horn, 1970, p. 144]. The characteristic strip-expansion method modified in this fashion can also interpolate new strips when the existing ones separate too far and delete old ones when they approach too closely.

Strips may be started near a so-called singular point (see later) and grow outwards from there. Because of the directionality of the progress of the solution, information from the ends of the characteristic strips cannot be exploited by this method. This is an important shortcoming since crucial information is obtained from occluding boundaries (see later). Normally, these would be reached at the end of the computation, with no means available to influence the (completed) solution.

There have been two new approaches based on two different ways of introducing additional constraints. In the one case, additional images are obtained from the same position with changed lighting. This will be discussed in Section 1.4. The other approach stays in the single image domain, exploiting instead suitably formulated assumptions about the nature of the surface. The work in this area employs relaxation or cooperative computation methods, which depend on the propagation of local constraints to determine global solutions [Woodham, 1977; Strat, 1979; Brooks, 1979]. These methods can often be viewed essentially as iterative algorithms for solving large sets of simultaneous equations arising from a least squares minimization. Information flow is not just along the characteristic directions, taking into account the fact that individual measurements are noisy. Instead of singular points, boundary conditions on a closed curve are used to select one of an infinite number of possible solutions.

1.3. Previous numerical methods for shape from shading

We look at these algorithms in more detail now. Woodham's [Woodham, 1977] method requires two rules which reduce the number of possible surface orientations at a particular point. Suppose that two closely spaced image points P_1 and P_2 at (x_1, y_1) and (x_2, y_2) correspond to object points on the same section of a smooth surface. Further assume that the view angle, between surface normal and the line of sight, increases and that the direction of steepest ascent, polar angle in gradient space, decreases in going from P_1 and P_2 . Let C_1 and C_2 be the contours in gradient space corresponding to

$$R(p, q) = E(x_1, y_1) \quad \text{and} \quad R(p, q) = E(x_2, y_2),$$

where $R(p, q)$ is the reflectance map and $E(x_1, y_1)$ and $E(x_2, y_2)$ are the observed brightness values at the points P_1 and P_2 (see Fig. 2). Now the contour of permissible values for (p, q) at point P_1 can be restricted to those points on C_1 lying on or within the circle of maximum view-angle interpretation of P_2 .

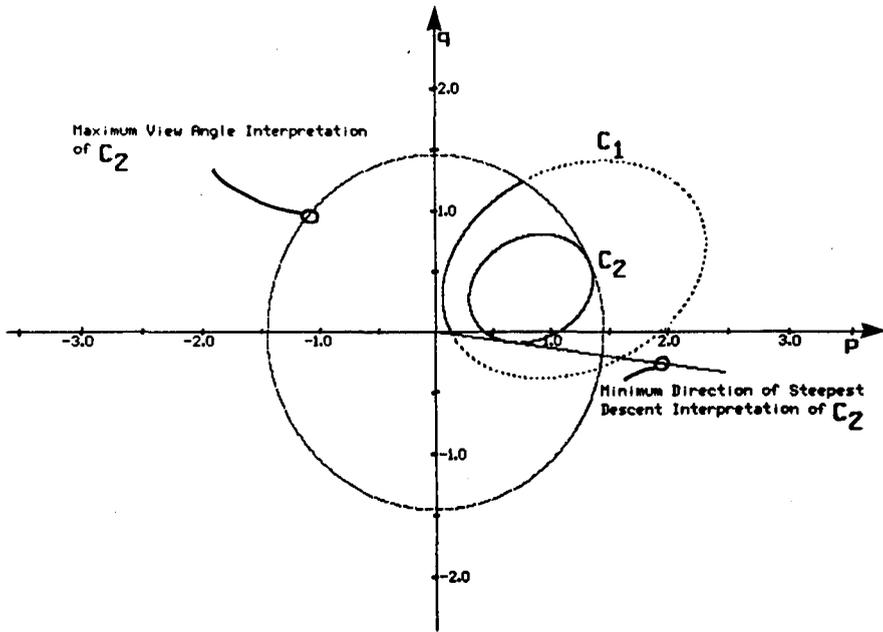


FIG. 2. The possible values of the gradients at the points P_1 and P_2 are restricted considerably if we assume that the view angle increases and the direction of steepest ascent decreases in going from P_1 to P_2 .

Further, the contour of permissible values for (p, q) at point P_1 can be restricted to those points on C_1 on or above the line of the minimum direction of steepest ascent interpretation of P_2 .

If a particular interpretation were applicable to the data, such an interpretation would provide a framework for ordering selected image points with respect to changes in both the view angle and the direction of steepest ascent. For example, if one can assume that the surface is elliptical or hyperbolic at a point (that is, the sign of the Gaussian curvature [Cohn-Vossen, 1952; Pogorelov, 1956; Moon and Spencer, 1969; do Carmo, 1976] is known) one can guarantee the sign of the change to the view angle or the sign of the change to the direction of steepest ascent by choosing the direction of the small step (dx, dy) from P_1 to P_2 appropriately [Woodham, 1977; Woodham, 1979]. Thus, local assumptions about surface shape provide monotonicity relations between selected image points. Thus we can start a shape-from-shading algorithm using these rules on the selected image points.

A somewhat different approach makes use of a particular way of formulating a smoothness constraint. To derive the constraint, we will assume for now that the second partial derivatives of the surface height z exist and are continuous. Then the partial derivative of z with respect to x and y is independent of the

order in which the differentiations are performed:

$$\partial p / \partial y = \partial^2 z / \partial y \partial x = \partial^2 z / \partial x \partial y = \partial q / \partial x.$$

Evidently in any simply-connected region, B say,

$$\iint \{ \partial p / \partial y - \partial q / \partial x \} = 0,$$

since the integrand is zero at every point. Thus, by Green's theorem [Hildebrand, 1965], the integral of (p, q) along a closed curve (the boundary, ∂B , of the region) also equals zero:

$$\oint \{ p \, dx + q \, dy \} = 0.$$

This is eminently sensible, since the line integral gives the difference between the height at the end of the line and its beginning, and we assumed that we remain on the same section of the surface (see Fig. 3).

If we use a numerical method based on imperfect data to compute the orientation of the surface at every point, we can expect that this integral will in fact not be exactly equal to zero. One way of imposing a smoothness constraint then is to find a solution which minimizes the errors in these loop integrals while also satisfying the image-irradiance equation as closely as possible.

This idea was pursued by Strat [Strat, 1979] as well as Brooks [Brooks, 1979]. Brooks showed that he could restrict attention to integrals around small loops in the image plane. He listed the possible orientations (quantized in a suitable way) at each point, then iteratively eliminated those conflicting with the loop integral criterion. This was accomplished using a relaxation method not unlike that developed by Waltz [Waltz, 1975] for labelling line drawings.

Strat minimized the weighted sum of the errors in integrals around small loops and the errors in the image-irradiance equations. He developed a system for iteratively solving the large, sparse set of equations arising from this minimization formulation. His method is in many ways similar to the one presented here except insofar as he used gradient space and so could not deal

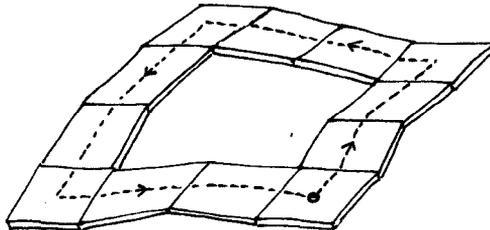


FIG. 3. Discrete version of the loop integral of (p, q) . The total change in elevation as one goes around the loop should be zero if the patches belong to a differentiable, single-valued surface.

with occluding boundaries. As a result, his algorithm requires that surface orientation be given on a closed curve. This limits its applicability.

1.4. Photometric stereo methods

The other new approach uses further images to provide additional constraint. These images are taken from the same position, but with changed lighting conditions. This so-called *photometric stereo* approach [Woodham, 1977; Horn et al. 1978; Woodham, 1980] allows one to determine surface orientation locally without smoothness assumptions.

Since the images are obtained from the same position, a particular point on the object will appear at the same spot in each image. This means that one does not have the problem of identifying projections of a particular surface feature in multiple views, as happens in ordinary stereo. A different reflectance map applies to each image, however, since the lighting is different for each view. For a given point in the image we have one brightness value corresponding to each of these reflectance maps. Suppose, for example, that at a particular point, (x_0, y_0) , two measurements of image brightness are available. Then we have two (non-linear) equations for p and q ,

$$R_1(p, q) = E_1(x_0, y_0), \quad R_2(p, q) = E_2(x_0, y_0),$$

where R_1 and R_2 are the reflectance maps appropriate to the two lighting situations, while E_1 and E_2 are the observed image-brightness values. The intersection of the corresponding gradient-space contours provides one with the sought after surface orientation. This is the essential idea of photometric stereo. Naturally, the above pair of nonlinear equations in two unknowns may have more than one solution, in which case additional information (such as a third image) may be needed to find the unique answer (see Fig. 4).

The graphical construction shown here can be replaced with a simple lookup-table procedure [Silver, 1980; Ikeuchi, 1981]. Quantized brightness values are used as indices into the table; the entries in the table contain the corresponding surface orientations. Detection of errors is facilitated (if more than two images are used) by blank entries which represent incompatible combinations of brightness values. Values of p and q are found for every point in the image, yielding surface orientation for the corresponding surface patches. While a description of the shape of an object in this form may be suitable for recognition and to find the attitude of an object in space, it is at times helpful to provide height above some reference plane instead. While z can obviously be found by integration of p and q , the best way to do this when it is known that the data is corrupted by measurement noise has not yet been described.

In industrial applications one has to deal with metallic parts which often exhibit specular or mirror-like reflections. In this case a distributed light source

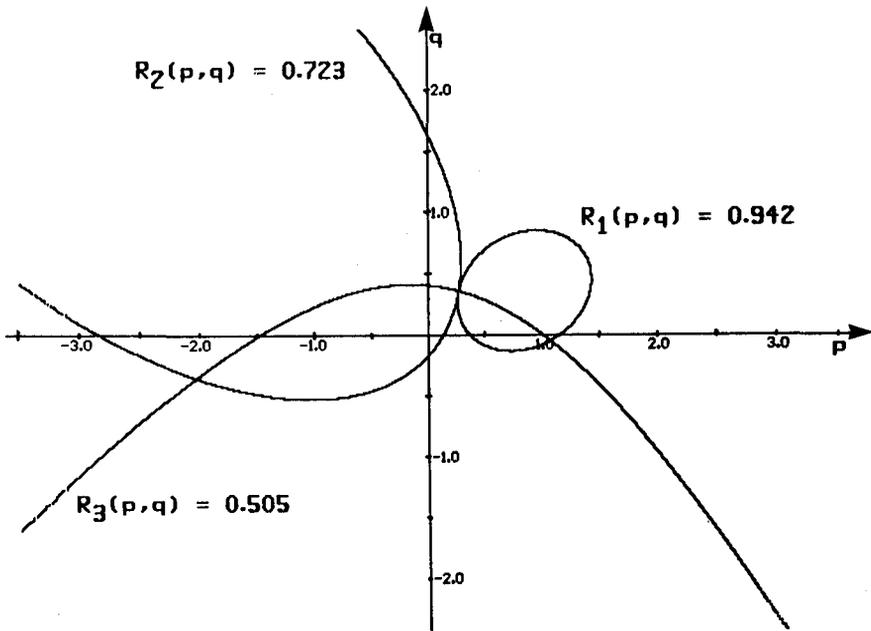


FIG. 4. Determining the surface orientation at a given image point using the photometric stereo method. Here, three measurements of image brightness are obtained using different illumination conditions. Three superimposed reflectance map contours, corresponding to the different measurements, intersect at the point sought.

with spatially varying brightness may be used [Ikeuchi, 1981]. For matte surfaces one can simply use a number of point sources [Silver, 1980]. So far experimentation has been confined to single objects, avoiding possible inaccuracies due to interreflection or mutual illumination [Spacek, 1979].

1.5. Motivation of this research

The outline of the projection of an object in the image plane is called its *silhouette*. Parts of this silhouette may correspond to sharp edges on the surface, others to places where the surface curves around smoothly. The smooth parts of the surface corresponding to parts of the silhouette are referred to as *occluding boundaries*. The locus of (barely) visible points on the surface where the tangent plane contains the viewer is called the *occluding boundary*. Occluding boundaries supply important information about the shape of an object and one can attempt to recover the shape of the object from this information alone using rather strong assumptions [Marr, 1977]. A simple interpolation algorithm works, particularly if the surface is (locally) spherical or cylindrical [Barrow and Tenenbaum, 1981]. This can also be approached from the statistical point of view of recovering the most likely surface [Witkin, 1980].

The true surface, however, can be found only if we also pay attention to the shading information. The occluding boundary can then be used as the boundary condition for the reconstruction. There is some question about whether the problem is well posed in this form, since the partial derivatives of height become infinite on the boundary. It should be clear, though, that if a numerical method is to succeed at all, then the representation of surface orientation must allow these boundary conditions to be expressed correctly. Little is known now about the uniqueness of solutions obtained in this way. It seems likely, however, that only one solution is possible under certain rather general assumptions about the nature of the reflectance map [Bruss, 1979; Bruss, 1980].

2. The Gaussian Sphere and Reflectance Functions

We can identify surface orientations with points on a unit sphere, called the *Gaussian sphere* [Hilbert and Cohn-Vossen, 1952]. Let the viewer be far above the north pole, along the z -axis, where the z -axis is taken as the extended line from the center of the sphere to its north pole. Assume that we take a patch of the surface of an object and place it in the center of the sphere, without changing its attitude in space. The patch will face some point of the sphere. That is, a unit-surface normal erected on the patch will touch the sphere at this point. The orientation of a horizontal portion of a surface, for example, is represented by a point at the north pole, since its surface normal points directly at the viewer. Portions of the surface seen at a glancing angle by the viewer correspond to points on the equator, while surface elements turned away from the viewer are associated with points in the southern hemisphere. Thus points in the northern hemisphere are of most interest to us here.

2.1. The Gaussian sphere and apparent brightness of a surface patch

If we assume orthographic projection of the three-dimensional world, then the angle between the line of sight and the surface normal is independent of the position of the patch on the object. As far as the geometry of light reflection is concerned we need to concern ourselves only with the orientation of a surface patch as represented by a point on the Gaussian sphere. Surface patches of an object which share the same orientation are associated with the same point on the sphere, independent of their position in space (see Fig. 5).

Surface brightness depends on the surface material and the geometry of light reflection: the angles between the line of sight, the surface normal, and the incident light rays. Since each point on the Gaussian sphere expresses one particular such geometric relationship, we can associate a unique value of surface brightness with each point on its surface. Say, for example, that a surface patch perpendicular to the viewer appears with unit brightness. In that situation we assign this value to the north pole. Using this method, we can assign brightness values to all points on the Gaussian sphere. Shown in Fig. 6 is

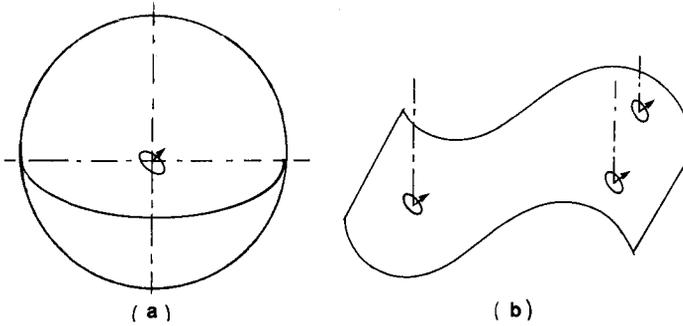


FIG. 5. Surface patches of an object (b), which share the same orientation are associated with the same point on the Gaussian sphere (a), independent of their position on the object.

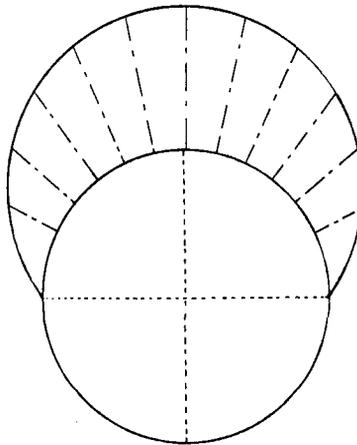


FIG. 6. Cross section through Gaussian sphere showing brightness values associated with points on the sphere. The radial distance of the curve from the surface of the sphere is proportional to the brightness. The particular reflectance function shown here happens to vary as $\sqrt{a^2 \cos^2 e + 1} + (a \cos e - 1)$, where e is the emittance angle, between the surface normal and the line of sight, and $a = 0.72$.

a cross section of the Gaussian sphere with brightness values represented by the height of a curve above the surface.

2.2. Projections of the Gaussian sphere

It may not be convenient to use this representation of the dependence of surface brightness on orientation, since the Gaussian sphere is three-dimensional. Graphic presentation of this information, for example, is difficult. It is therefore advantageous to project the Gaussian sphere onto a plane. (This is possible since we are interested only in the surface of the Gaussian sphere.)

There are, of course, many ways to perform this mapping operation. Even if we were to confine ourselves to conformal projections, we would have many to choose from, including the Mercator, Lambert, and stereographic projections used in geodesy and cartography [Thomas, 1952; Raisz, 1962]. We prefer continuous projections like these, since they map neighboring points into neighboring points. This is important because we plan to generate surface representations iteratively from boundary information. Roughly speaking, to be able to do this, we want the following to hold: a point C lying between two other points A and B is projected into a point $F(C)$ which lies approximately between $F(A)$ and $F(B)$.

It can easily be shown that,

$$\mathbf{n} = (-p, -q, 1)/\sqrt{1+p^2+q^2}$$

is a unit (outward) normal to the surface [Horn, 1977; Horn, 1981]. This gives us the coordinates of the point on the Gaussian sphere corresponding to a surface patch with gradient (p, q) , and immediately suggests a projection onto a plane with axes labelled p and q . Appropriately enough, the plane of projection is called gradient space in this case, as we have mentioned before [Huffman, 1971; Mackworth, 1973; Draper, 1980].

Geometrically we can think of this as the projection of the Gaussian sphere by rays from its center onto a plane tangent to the north pole (with the sense of the p and q axes reversed from that of the x and y axes). This projection, called the *gnomonic* projection, has the property that great circles are mapped into lines. (It is used for this reason in navigation to determine the shortest path between two points on the earth.) This also makes it a so-called *geodesic* projection. In our case here, the north pole is mapped into the origin, the top hemisphere is mapped into the whole pq -plane, and points on the equator end up at infinity. Points on the lower hemisphere are not projected onto the pq -plane.

Since each point in gradient space corresponds to a particular surface orientation, one can associate a unique brightness value with it. The result, as mentioned earlier, is the reflectance map [Horn, 1977]. The gnomonic projection is convenient for the following reasons: (1) the coordinates correspond to the first partial derivatives of surface height, z , (2) we can compute surface height by integrating p and q , and (3) a line integral of (p, q) along a closed loop is always zero. Unfortunately, gradient space also has a serious drawback. As we shall see, points on the equator of the Gaussian sphere correspond to surface patches on the occluding boundary. With the gradient-space projection, the equator maps to infinity. As a result, occluding boundary information cannot be expressed in gradient space.

2.3. Stereographic projection

One solution to this problem is the use of the stereographic projection [Sohon,

1941]. Hipparchus (*circa* 150 B.C.) is credited with the invention of this projection (which appears to have been employed in the astrolabe-planisphere solution of the astronomical triangle). We can think of this projection in geometric terms also as projection onto a plane tangent at the north pole. This time, however, the center of projection is the south pole, not the center of the sphere (see Fig. 7). We label the axes of the stereographic plane f and g to distinguish them from those of gradient space. It can be shown that these coordinates are related to the partial derivatives as follows:

$$f = 2p[\sqrt{1 + p^2 + q^2} - 1]/(p^2 + q^2),$$

$$g = 2q[\sqrt{1 + p^2 + q^2} - 1]/(p^2 + q^2).$$

This projection is conformal (preserves angles between lines and shapes of small figures) and maps circles on the Gaussian sphere into circles on the plane [Hilbert and Cohn-Vossen, 1952]. The whole sphere, not just the northern hemisphere, is mapped onto the plane this time. Only the south pole ends up at infinity, while the equator is mapped into a circle of radius two (see Fig. 8). We may also assign a brightness value to each point in the stereographic plane, just as we did with points in gradient space.

The two projections considered so far fall in the category of *azimuthal* projections, since latitude on the sphere simply becomes the azimuth in the plane. Another convenient projection in this class is the so-called *azimuthal equidistant* projection. Here the distance from the origin in the plane equals the distance along the surface of the sphere measured from the north pole (that is, the co-latitude). There is no obvious geometric interpretation in terms of rays emanating from a fixed center of projection. One can think of this mapping instead as one obtained by rolling the sphere on the plane, along meridians, always starting with the north pole aligned with the origin. If we call

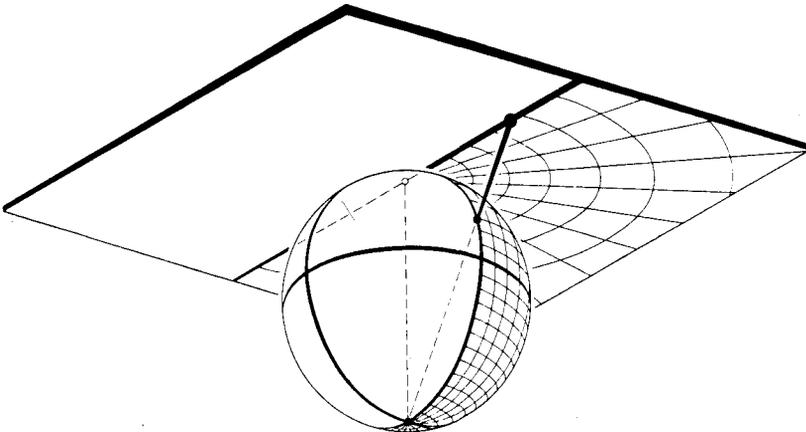


FIG. 7. The stereographic mapping projects each point on the surface of the sphere, along a ray from one pole, onto a plane tangent to the opposite pole.

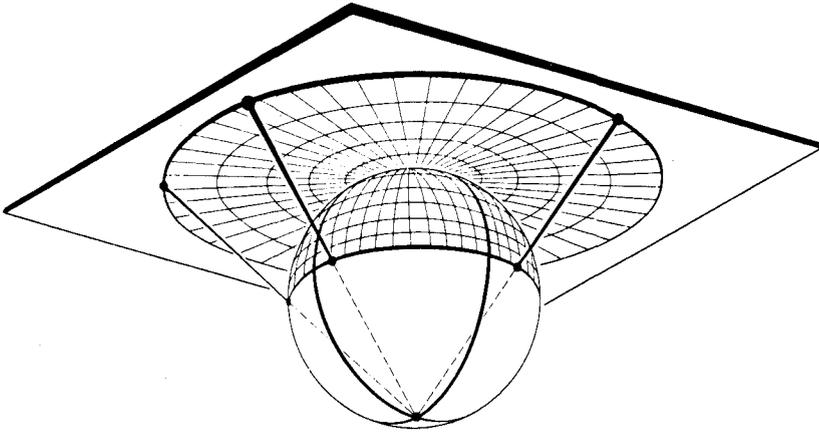


FIG. 8. The stereographic projection maps the equator onto a circle of radius two. The equator of the Gaussian sphere corresponds to points on the occluding boundary of the object.

the coordinates of the plane a and b in this case, we have

$$a = [p/\sqrt{p^2 + q^2}] \tan^{-1} \sqrt{p^2 + q^2},$$

$$b = [q/\sqrt{p^2 + q^2}] \tan^{-1} \sqrt{p^2 + q^2}.$$

As with the gnomonic projection, we project only the northern hemisphere. But, unlike the gnomonic projection, points on the equator do not end up at infinity; instead they are mapped onto a circle of radius $\pi/2$.

There are, of course, numerous other projections one might use. K. Tanaka, for example, used another azimuthal projection, the so-called *orthographic* projection in early work on hill-shading [Horn, 1981].

3. Smoothness Constraint and Boundary Conditions

In the sequel we use the stereographic projection. The form of the image-irradiance equation will be slightly different, since we use f and g instead of p and q . However, this makes no difference to the basic idea that a measurement of surface brightness restricts the possible surface orientations. We must develop a suitable smoothness constraint expressible in the coordinate system natural to the stereographic plane. We also have to determine what boundary conditions are available to constrain the solution of the image-irradiance equation. Finally, it will be helpful to find a way of determining good initial values of surface orientation for at least some of the points, in order to get the numerical solution off to a good start.

3.1. Formulation of surface smoothness constraints

If we employ the stereographic projection, we cannot use the closed loop constraint proposed by Strat [Strat, 1979] and Brooks [Brooks, 1979], because

the closed loop constraint depends upon a special characteristic of the gnomonic projection. Namely, the closed loop integral can be expressed directly in terms of p and q , the coordinates of gradient space. Unfortunately, this is not true in the case of either the stereographic projection or the azimuthal equidistant projection. If a constraint expresses the property of surface smoothness, it ought to be valid regardless of the projection we happen to use. It seems reasonable to express the smoothness criterion in terms of a relationship between neighboring points.

We now turn to the definition of surface smoothness. The standard definition of a smooth function is one which has continuous first partial derivatives. This agrees with our intuitive feeling of what constitutes a smooth surface. First of all, smoothness requires that there be no depth discontinuities. That is to say, height, as a function of the image coordinates, should be continuous (height is class C^0). But we also require that surface orientation be continuous, for if a surface has a sharp edge or wrinkle, the surface is not considered smooth. In other words, smoothness also requires that the first partial derivatives of height be continuous (height is class C^1).

Should one also require that the second partial derivatives of height be continuous, or equivalently, that the derivatives of orientation are continuous? The following illustration shows that this is not necessary. Imagine a planar surface attached to a portion of a cylindrical surface so that the two are tangent where they touch (see Fig. 9). Here surface orientation is constant on the planar part, but varies on the cylindrical part. Thus surface orientation is continuous, but its derivatives are not. Nevertheless, people regard this surface as smooth.

It is interesting to note at this point that the method of characteristic strip expansion can be shown to apply in general when the function sought is of class C^2 . In the case of a first-order partial differential equation in two independent variables, however, it is known that the solution is unique even when the function is only class C^1 [Courant and Hilbert, 1962, p. 145]. Similarly, the derivation of the closed loop constraint shown above requires that the function be class C^2 . It turns out to be sufficient, however, that the function be single-valued and (once) differentiable.

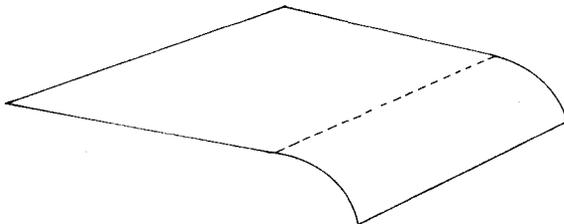


FIG. 9. A surface constructed by attaching a portion of a cylindrical surface to a plane, in such a way that they are tangent on the contact line, is considered smooth.

3.1.1. Continuity of surface orientation

What is the definition of continuous? A function is continuous if neighboring points map into neighboring points. If a function is vector valued, being continuous implies that each component of the function is continuous [do Carmo, 1976]. This allows us to consider each component of surface orientation separately. For example, in the stereographic plane, we can formulate two constraints which express continuity of f and g separately. We use the standard definition of continuity [do Carmo, 1976]:

A function F is continuous at (x_0, y_0) if, given any $\epsilon > 0$, there exists a δ such that when

$$(x - x_0)^2 + (y - y_0)^2 < \delta^2,$$

then,

$$|F(x, y) - F(x_0, y_0)| < \epsilon.$$

If, given a particular ϵ , we can find a single value of δ for all points in the region of interest, then the function is uniformly continuous.

In practice, we will be dealing with a discrete grid of points with values defined at the nodes of the grid. If we take the grid interval δ_0 smaller than δ , then we are guaranteed that:

$$|F(x_0 + \delta_0, y_0) - F(x_0, y_0)| < \epsilon \quad \text{and} \quad |F(x_0, y_0 + \delta_0) - F(x_0, y_0)| < \epsilon.$$

At this point we should remember that it is surface orientation which we are assuming varies continuously as a function the image coordinates. Thus equations like the one above apply to f and g .

3.1.2. Summation of error terms

One more check is necessary before we can use our smoothness constraint. To measure how far the computed surface departs from smoothness, we will be looking at error terms like

$$(f_{i+1,j} - f_{ij})^2 \quad \text{and} \quad (g_{i+1,j} - g_{ij})^2.$$

We form the sum of the squares of the differences over all adjacent pairs of nodes in the grid. When the grid interval becomes small, the number of nodes per unit area becomes large. For our purposes, the sum of all the differences squared should nevertheless remain small as we decrease the grid interval. For this to be true we actually require that the derivatives of orientation exist and are bounded. Suppose that the largest absolute value of the partial derivatives of f and g with respect to x and y is denoted by D_m . Then the square of an error term will be less than

$$[D_m \delta_0]^2.$$

Since the number of nodes per unit area equals $1/\delta_0^2$, we find that the sum of squares of the errors is less than D_m . Note that discontinuities of the derivatives of orientation, as occur on the dotted line in Fig. 9, do not constitute a problem.

3.2. Constraints from boundary information

In some areas of an image, we can determine surface orientations directly. This is true, for example, at so-called singular points and specular points. Useful initial values for the iterative solution process are found this way. More importantly, at an occluding boundary, we can determine surface orientation from the silhouette. This supplies us with the all-important boundary conditions.

3.2.1. Occluding boundary

At an occluding boundary we can determine the surface normal uniquely [Marr, 1977; Barrow and Tenenbaum, 1981]. The following two facts are the starting points of our discussion of the determination of the surface normals from the silhouette.

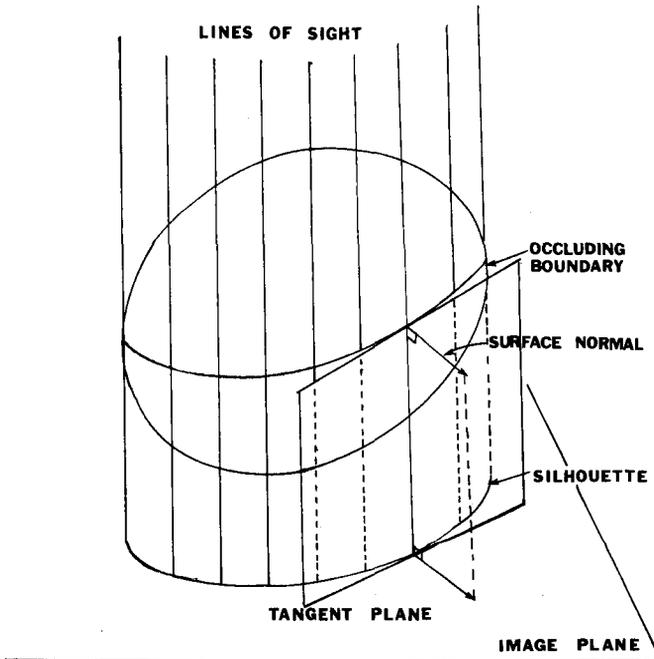


FIG. 10. The line of sight is tangent to the surface at the point where it touches the occluding boundary. The tangent plane at that point cuts the image plane in a line. This line is tangent to the silhouette in the image plane.

First: since we observe an occluding boundary there, the line connecting the surface to the viewer must be tangential to the surface patch at that point (see Fig. 10). Thus the line of sight lies in the plane tangent to the surface at the point where it grazes the surface. It is therefore perpendicular to the surface normal at that point.

Second: the line of sight is perpendicular to the image plane (Since we assumed orthographic image projection, all lines of sight are parallel to the z -axis and thus perpendicular to the image plane). It follows that the tangent plane is perpendicular to the image plane and therefore projected as a line in the image plane. This line is tangent to the silhouette in the image plane.

We see then that a normal to the silhouette in the image plane is parallel to the normal to the surface at the corresponding point on the occluding boundary. We can in this fashion obtain surface orientations for all points on the occluding boundary.

3.2.2. *Self-shadow boundary*

Consider the situation where there is a single point source. The locus of (barely) illuminated points on the surface where the tangent plane contains the light source is called the *self-shadow boundary*. On the self-shadow boundary, light rays play the role of the lines of sight in the previous discussion. Namely, the light grazes the surface there; the rays are perpendicular to the surface normal. The trouble is that now the tangent plane is not projected as a line in the image plane (see Fig. 11). Consequently we cannot determine surface orientations uniquely on a self-shadow boundary. Approximations to the correct surface orientations can be computed if we make some assumptions. This is helpful, because these can serve as initial values for the iterative algorithm. One possible assumption we might make is that the self-shadow boundary lies in a plane perpendicular to the light source. This occurs, for example, when the object is spherical, and is not a bad approximation when the object is ellipsoidal.

Let the light source lie in a direction given by the unit vector n_s . The projection of the self-shadow boundary in the image is called the *shadow edge*. Let a vector in the image plane perpendicular to the shadow edge be n_b (drawn from the lighted towards the shadowed side). The self-shadow boundary lies in a plane perpendicular to both n_b and n_s and must therefore be parallel to

$$n_s \times n_b.$$

The surface normal on the shadow boundary must be perpendicular to this line, as well as to the incident rays. It must therefore be parallel to

$$(n_s \times n_b) \times n_s = n_b - (n_b \cdot n_s)n_s.$$

The above vector vanishes when the incident rays are parallel to the image plane, indicating that in this case we cannot determine anything about the

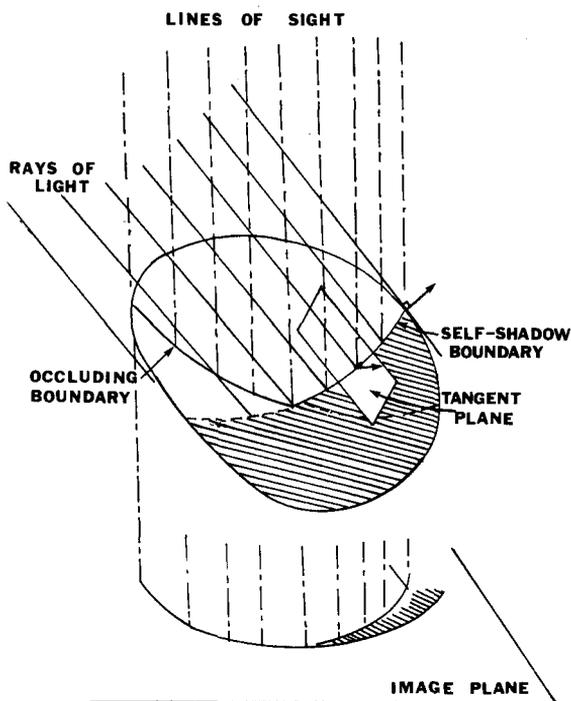


FIG. 11. The tangent planes on the self-shadow boundary do not project as lines into the image plane. The surface orientation of points on the self-shadow boundary cannot be determined uniquely.

surface normals on the shadow boundary, even with the assumption that this boundary lies in a plane perpendicular to the incident rays. Conversely, when the source is at the viewer, the surface normal is fully determined even without this assumption, since the shadow boundary then coincides with the occluding boundary.

Errors in the determination of the surface normal on the shadow boundary do not lead to errors in the final result, since they affect only the initial values. Presumably, better initial values do reduce the number of iterations required to obtain an accurate solution, however. Note that the condition that the surface normal is perpendicular to the incident rays on the self-shadow boundary is actually already implicit in the equation $R(p, q) = 0$.

3.2.3. Specular point and singular point

We can also determine surface orientation at so-called singular points and specular points. Assume for now that the illumination comes from a single point-source of light. Consider first a surface patch oriented so that its normal

vector points directly at this light source. (It will be maximally illuminated, but not necessarily maximally luminous.) Let us call the point on the Gaussian sphere corresponding to this orientation the *source spot*, since it essentially specifies the direction towards the light source. Note, by the way, that the source spot ends up in the southern hemisphere when the object is illuminated from behind. Such a source spot will be projected into a point in the plane by the stereographic projection. This is not the case if we use the gnomonic or azimuthal-equidistant projections, since these project only the northern hemisphere.

Next, assume that we are dealing with a surface material which exhibits mirror-like or specular reflection. A surface patch on such a surface may be oriented just right to reflect rays from the light source towards the viewer. This leads to a so-called *specular point* in the image. This happens only when the incident angle, i , equals the emittance angle, e , and when the incident ray, the surface normal, and the reflected ray lie in the same plane. This orientation corresponds to the point mid-way between the source spot on the Gaussian sphere and the north pole.

If we let (p_s, q_s) be the point in gradient space which corresponds to the source spot on the Gaussian sphere, then a surface patch oriented for specular reflection has the gradient (p_m, q_m) , given by:

$$p_m = p_s[\sqrt{1 + p_s^2 + q_s^2} - 1]/(p_s^2 + q_s^2),$$

$$q_m = q_s[\sqrt{1 + p_s^2 + q_s^2} - 1]/(p_s^2 + q_s^2).$$

This can be shown by noting that (p_m, q_m) must lie in the same direction in gradient space as (p_s, q_s) , and that the angle between $(0, 0, 1)$ and $(-p_m, -q_m, 1)$ must be equal to the angle between $(-p_m, -q_m, 1)$ and $(-p_s, -q_s, 1)$. Similar formulae apply in the stereographic plane:

$$f_m = f_s[\sqrt{1 + f_s^2 + g_s^2} - 1]/(f_s^2 + g_s^2),$$

$$g_m = g_s[\sqrt{1 + f_s^2 + g_s^2} - 1]/(f_s^2 + g_s^2).$$

Finally, for the azimuthal equidistant projection,

$$a_m = a_s/2 \quad \text{and} \quad b_m = b_s/2.$$

For an ideal point source, an ideal specular surface (and an ideal imaging system), the specular point would be infinitely bright and infinitesimal in extent. For slightly extended sources and surfaces that are not perfectly smooth, the specular reflection is spread out somewhat, with a large, but finite maximum brightness. Surfaces with slightly undulating surface microstructure give rise to *glossy high-lights* [Horn, 1981].

For surfaces which do not exhibit specular reflection there often still is a unique surface orientation which gives rise to maximum (or minimum) brightness. A Lambertian surface, for example, is brightest when the light rays strike

it perpendicularly. Suppose then that there is a particular value of brightness which corresponds uniquely to a particular surface orientation. A *singular point* is a point in the image where this value of brightness is observed. Note that specular points are special cases of singular points. Evidently, we can determine the surface orientation uniquely at a singular point.

Singular points, by the way, were used to provide starting values in an implementation of the characteristic strip-expansion method [Horn, 1975]. Note that information about the orientation of the surface at these special points is implicit in the image-irradiance equation. What we gain here are helpful initial values for the iterative method.

3.3. Minimization of errors

It helps to first consider the continuous case, ignoring for now the tessellation of the image plane into picture cells. We try to find functions $f(x, y)$ and $g(x, y)$ which make the errors in the image-irradiance equation small while also being as 'smooth' as possible. We can try to minimize

$$e = \iint \{(f_x^2 + f_y^2) + (g_x^2 + g_y^2) + \lambda [E(x, y) - R_s(f, g)]^2\} dx dy,$$

where $f_x, f_y, g_x,$ and g_y are the first partial derivative of f and g with respect to x and y . The errors in the image irradiance equation are weighted by the factor λ relative to the measure of departure from 'smoothness'. This factor can be made large when the reflectance map is known accurately and the brightness measurements are precise.

The minimization of an integral of the form

$$\iint F(f, g, f_x, f_y, g_x, g_y) dx dy$$

is a problem in the calculus of variation [Courant and Hilbert, 1953; Hildebrand, 1965; Carrier and Pearson, 1976]. When the integrand depends on two functions of two independent variables as well as their first partial derivatives, then the Euler equations are,

$$F_f - \partial/\partial x(F_{f_x}) - \partial/\partial y(F_{f_y}) = 0,$$

$$F_g - \partial/\partial x(F_{g_x}) - \partial/\partial y(F_{g_y}) = 0.$$

Applying these formulae we obtain:

$$\nabla^2 f = \lambda [E(x, y) - R_s(f, g)] \partial R_s / \partial f,$$

$$\nabla^2 g = \lambda [E(x, y) - R_s(f, g)] \partial R_s / \partial g,$$

where

$$\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$$

is the Laplacian operator. These equations will provide helpful guidelines when we tackle the discrete case in the next section. It can be shown, by looking at higher order variations, that if f and g obey the above partial differential equations, then the stationary value of the sum of errors is actually a minimum.

It may also be of interest that we would have ended up with the biharmonic operator here (instead of the Laplacian) if we had tried to minimize the sum of squares of the Laplacian of f and g (instead of the sum of squares of the first partial derivatives of f and g). We will not discuss why the two functions, f and g , should be 'consistent', that is, correspond to the surface orientations of a smooth surface.

As an extension of the above analysis we may consider the minimization of

$$e' = \int \int \{(f_x^2 + f_y^2) + (g_x^2 + g_y^2)\} dx dy,$$

subject to the constraint

$$E(x, y) = R_s(f, g).$$

Introduction of the Lagrangian multiplier λ leads to the same equations as the ones considered above. Elimination of λ gives us this set of equations:

$$(\partial R_s / \partial g) \nabla^2 f = (\partial R_s / \partial f) \nabla^2 g,$$

$$E(x, y) = R_s(f, g).$$

We will not use the results of this constrained minimization exercise since we do expect errors in the measurements of image brightness. Instead, we will set the value of the parameter λ in inverse proportion to the root-mean-square of the noise in the image brightness measurements.

4. Proposed Algorithm and Numerical Experiments

We will construct an iterative algorithm using two kinds of constraints: one from the image-irradiance equations and the other from the smoothness condition. In addition, we depend on the occluding boundary to provide boundary conditions and a few special points to supply helpful initial values.

We will have to check whether the algorithm converges. Since we do not have a theoretical analysis of the conditions required for convergence, we resort to numerical experimentation. With exact, complete information, we expect that the algorithm will converge, and converge to the correct solution. In practice there are errors, however, due both to noise in the measurements and to discrepancies between the mathematical model and the real situation. Further, due to obscuration, for example, the information may be incomplete. Under these circumstances one would hope that the algorithm would be robust enough to converge to some solution, preferably one close to the correct one.

Consequently we have carried out numerical experiments under two different conditions: in the first case, all information given to the algorithm is

exact: the reflectance map is correct, and complete boundary information is available. In the second, some of the information is erroneous: the reflectance map given may be incorrect, either because of inappropriate assumptions about the properties of the surface material or because of incorrect information about the position of the light source. We also may be lacking effective boundary conditions on part of the border, perhaps because of partial obscuration of the object by another. The algorithm would not be very useful if it did not converge, or converged to a solution very different from the correct one, when faced with these difficulties.

4.1. Proposed algorithm

We will use an iterative algorithm to determine surface orientations using the image-irradiance equations and the smoothness criterion as constraints. We can measure the departure from smoothness as follows:

$$s_{ij} = [(f_{i+1,j} - f_{ij})^2 + (f_{ij+1} - f_{ij})^2 + (g_{i+1,j} - g_{ij})^2 + (g_{ij+1} - g_{ij})^2]/4.$$

The error in the image-irradiance equation, on the other hand, can be stated this way:

$$r_{ij} = [E_{ij} - R_s(f_{ij}, g_{ij})]^2,$$

where E_{ij} is the observed image brightness at the node (i, j) , while R_s is the reflectance map with f , and g , the surface orientation components as arguments. We will seek a solution which minimizes the sum of the error terms over all nodes:

$$e = \sum_i \sum_j (s_{ij} + \lambda r_{ij}).$$

The factor λ weights the errors in the image-irradiance equation relative to the departures from surface smoothness.

We can, by the way, write a formula for the error, just like the one given above, using the components of the gradient, p and q , or the azimuthal equidistant coordinates, a and b , instead of the stereographic coordinates f and g . For that matter, we can do this directly with coordinates specified on the Gaussian sphere.

We are going to differentiate e with respect to f_{ij} and g_{ij} (note that each f_{ij} and g_{ij} occurs in four terms of the sum for e). We obtain,

$$\partial e / \partial f_{ij} = 2(f_{ij} - f_{ij}^*) - 2\lambda [E_{ij} - R_s(f_{ij}, g_{ij})] \partial R_s / \partial f,$$

$$\partial e / \partial g_{ij} = 2(g_{ij} - g_{ij}^*) - 2\lambda [E_{ij} - R_s(f_{ij}, g_{ij})] \partial R_s / \partial g,$$

where f^* and g^* are the local averages of f and g :

$$f_{ij}^* = [f_{i+1,j} + f_{ij+1} + f_{i-1,j} + f_{ij-1}]/4,$$

$$g_{ij}^* = [g_{i+1,j} + g_{ij+1} + g_{i-1,j} + g_{ij-1}]/4.$$

The partial derivatives of e with respect to $f_{i,j}$ and $g_{i,j}$ will all equal zero, if the set of values we have for $f_{i,j}$ and $g_{i,j}$ constitute a solution to the minimization problem. Assume that they are; then the task is to solve the resulting large, sparse set of equations [Hildebrand, 1965; Conte and de Boor, 1972; Hamming, 1972]. Iterative techniques like the Gauss-Jacobi method, the Gauss-Seidel method, or successive over-relaxation [Carrier and Pearson, 1976, p. 277; Ames, 1977, p. 103], can be used if we first rearrange the equations as follows:

$$f_{i,j} = f_{i,j}^* + \lambda [E_{i,j} - R_s(f_{i,j}, g_{i,j})] \partial R_s / \partial f,$$

$$g_{i,j} = g_{i,j}^* + \lambda [E_{i,j} - R_s(f_{i,j}, g_{i,j})] \partial R_s / \partial g.$$

We can think of these equations as suggesting an adjustment of f and g , in a direction given by the gradient of R_s , by an amount proportional to the residual error in the image-irradiance equation. (The magnitude of the adjustment is also controlled by the parameter λ .) Further, we can see that the value for the orientation at a point will be filled in from the surround when the gradient of R_s is zero, as happens at a singular point.

The basic idea now is that the difference between corresponding values of f and corresponding values of g on successive iterations will decrease as we proceed. Thus we can perhaps use the values from the n th iteration on the righthand side to compute the values for the $(n+1)$ st iteration on the lefthand side. Under certain conditions this method converges to the solution of the original set of equations. Thus we might use the rule:

$$f_{i,j}^{n+1} = f_{i,j}^{*n} + \lambda [E_{i,j} - R_s(f_{i,j}^n, g_{i,j}^n)] \partial R_s / \partial f,$$

$$g_{i,j}^{n+1} = g_{i,j}^{*n} + \lambda [E_{i,j} - R_s(f_{i,j}^n, g_{i,j}^n)] \partial R_s / \partial g,$$

where $f_{i,j}^n$ and $g_{i,j}^n$ are used in evaluating R_s (and the partial derivatives of R_s). To avoid a particular kind of numerical instability we resort to a slightly different set of formulae

$$f_{i,j}^{n+1} = f_{i,j}^{*n} + \lambda [E_{i,j} - R_s(f_{i,j}^{*n}, g_{i,j}^{*n})] \partial R_s / \partial f,$$

$$g_{i,j}^{n+1} = g_{i,j}^{*n} + \lambda [E_{i,j} - R_s(f_{i,j}^{*n}, g_{i,j}^{*n})] \partial R_s / \partial g,$$

where $f_{i,j}^{*n}$ and $g_{i,j}^{*n}$ are used in evaluating R_s (and the partial derivatives of R_s). The flow of information when one uses this method can be illustrated schematically as shown in Fig. 12. Note that the computations for a given iteration are local. Global consistency is achieved by the propagation of constraints over many iterations.

The numerical instability alluded to above can best be thought of in terms of a checkerboard. On each iteration the new values for the white squares are computed using the old values on the black squares and *vice versa*. Small noise components can be amplified into a checkerboard pattern in this fashion as the

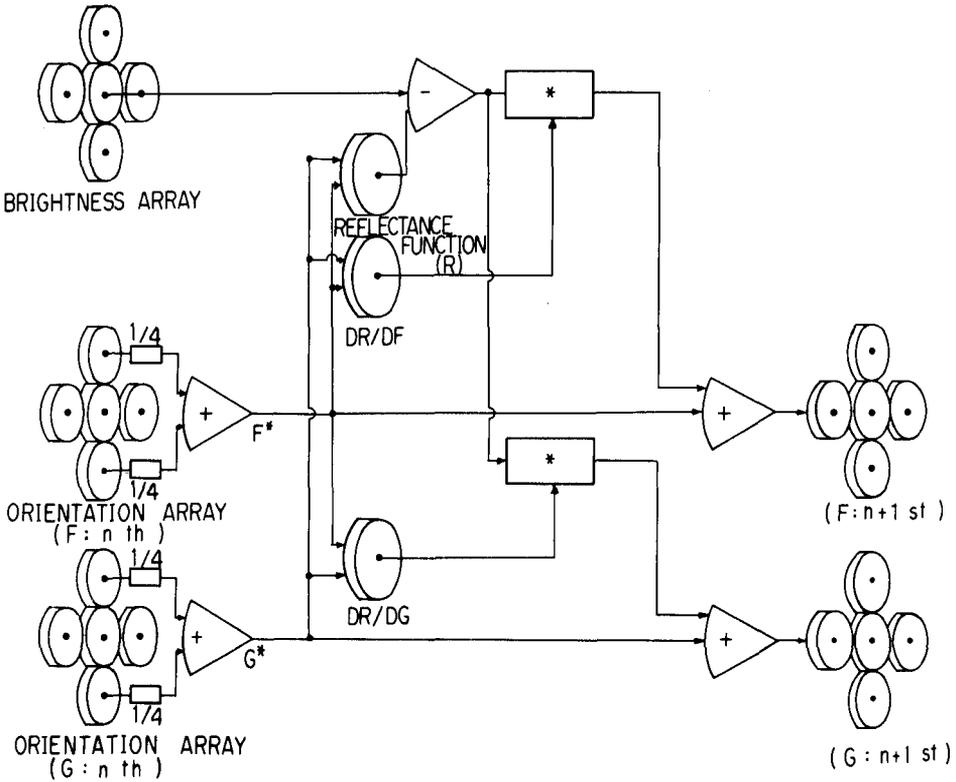


FIG. 12. This schematic diagram shows the flow of information in the iterative computation of surface orientation. It also suggests possible hardware implementations of the algorithm.

iterations progress. Say, for example, that the orientations (f, g) and (f^*, g^*) lie on opposite sides of the correct solution. Then the iterative equations above will produce an adjustment relative to (f^*, g^*) based on the error in the image-irradiance equation at (f, g) , which goes in the wrong direction. (Fig. 13 illustrates how this can occur.) This undesirable process can be damped out by using the local average, rather than the value at the cell itself in evaluating R .

Using a better stencil for computing the local average improves stability too. A *stencil* is a pattern of weights by which neighboring values are multiplied. One may, for example, add $(4/5)$ of the average of the four neighbors that share an edge with a particular point to $(1/5)$ of the average of the neighbors that touch on a corner. This suggestion is based on a better discrete approximation of the Laplacian than the one we have been implicitly using here [Richtmyer and Morton, 1967; Milne 1970; Carrier and Pearson, 1976; Ames, 1977].

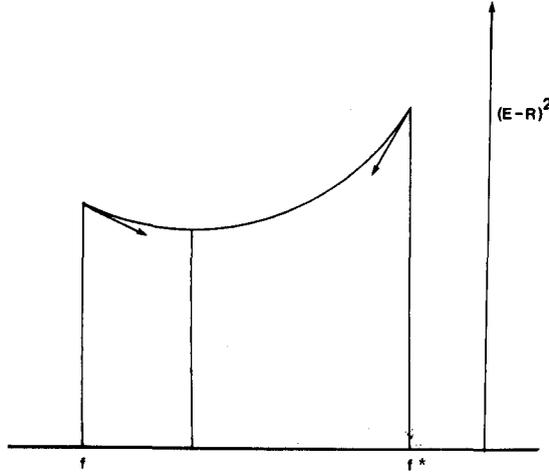


FIG. 13. This illustration shows why the simple iterative formulae may be subject to a certain kind of numerical instability. The correction to f^* (on the right) is computed from the gradient of the total error using the orientation f (on the left). The correction will have the wrong direction if the minimum lies between the two.

4.2. Numerical experiments

To start off with, we will use synthetic images based on models of objects, so that we can compare the results produced by our algorithm with the known shape. To check that the algorithm works correctly, we will give it the exact brightness distribution, complete boundary information, and the correct reflectance map. Then we will see what happens if we omit some of the boundary information, or if we estimate the light-source direction incorrectly, or assume the wrong reflectance properties. Finally, the method is applied to real images.

The shape information computed will be presented in two forms: as needle diagrams and oblique views of the surface. In a *needle diagram* we present an orthographic projection of the normals to the surface as they would be seen by the viewer [Horn, 1982]. This may be the most direct way of portraying graphically the information computed by the algorithm. The surface height above some reference plane can be estimated by integrating this surface orientation information. We can then show an oblique view of a network of lines drawn over the resulting surface by intersecting it with vertical planes parallel to the two coordinate axes. This mode of presentation is referred to as a *block-diagram* in cartography [Horn, 1981].

For several of the examples we will assume that the surface is made of a material which acts as a Lambertian reflector. A Lambertian surface is a diffuse reflector with the property that a particular surface patch looks equally bright

from all viewing directions and that its brightness is proportional to the illumination falling on it. Due to foreshortening of the surface as seen from the light source, the illumination (flux per unit area) varies as the cosine of the incident angle, i (between the incident rays and the surface normal). For a Lambertian surface, then, the brightness, too, will vary as the cosine of the incident angle. For surface patches turned away from the light source, the incident angle becomes larger than 90° , and so the cosine becomes negative. Since brightness cannot be negative, we use the formula

$$\max[0, \cos i]$$

for computing brightness.

The first task is to determine the reflectance map. If we let \mathbf{n} be a unit-surface normal and \mathbf{n}_s a unit vector pointing in the direction of the light source, then the cosine of the incident angle is just the dot-product of these two vectors,

$$\cos i = \mathbf{n} \cdot \mathbf{n}_s.$$

The expression for the unit-surface normal in terms of p and q , shown earlier, allows one to express the cosine of the incident angle in terms of the first partial derivatives of the surface orientation [Horn, 1977; Horn, 1981]:

$$\cos i = \frac{(1 + p_s p + q_s q)}{[\sqrt{1 + p^2 + q^2} \sqrt{1 + p_s^2 + q_s^2}]}.$$

Using the coordinates of the stereographic plane instead:

$$\cos i = \frac{[16(f_s f + g_s g) + (4 - f^2 - g^2)(4 - f_s^2 - g_s^2)]}{[(4 + f^2 + g^2)(4 + f_s^2 + g_s^2)]}.$$

If we use the azimuthal equidistant projection we have for $\cos i$:

$$\begin{aligned} & \cos \sqrt{a^2 + b^2} \cos \sqrt{a_s^2 + b_s^2} + \sin \sqrt{a^2 + b^2} \sin \sqrt{a_s^2 + b_s^2} \\ & \times \frac{(a_s a + b_s b)}{[\sqrt{a^2 + b^2} \sqrt{a_s^2 + b_s^2}]} \end{aligned}$$

In several of the examples we use a sphere as a test object, or an ellipsoid obtained by anisotropically scaling a sphere. To make a synthetic image we need to know the surface orientation at each point in the image. The implicit equation for a sphere with its center at the origin can be written

$$x^2 + y^2 + z^2 - R^2 = 0.$$

Computing the gradient of the lefthand side of this equation tells us that the surface normal at the point (x, y, z) is parallel to the vector (x, y, z) and that

$$\partial z / \partial x = -x/z \quad \text{and} \quad \partial z / \partial y = -y/z.$$

From these values of p and q , we can calculate f and g , or a and b . The brightness at every point in the image can then be found from

$$E(x, y) = \max[0, \cos i]$$

and the equations for $\cos i$ given above.

4.2.1. Example 1. Lambertian sphere with source near the viewer

Consider a Lambertian sphere illuminated by a single, distant light source, located near the viewer. The occluding boundary is also the self-shadow boundary in this case. The whole occluding boundary is visible to the viewer, while none of the shadowed areas is. The image-brightness data is derived analytically from a model of the surface and the known reflectance properties of a Lambertian surface. The algorithm is then applied to the synthesized image. (Fig. 14 shows a pseudo gray-level image of the object.)

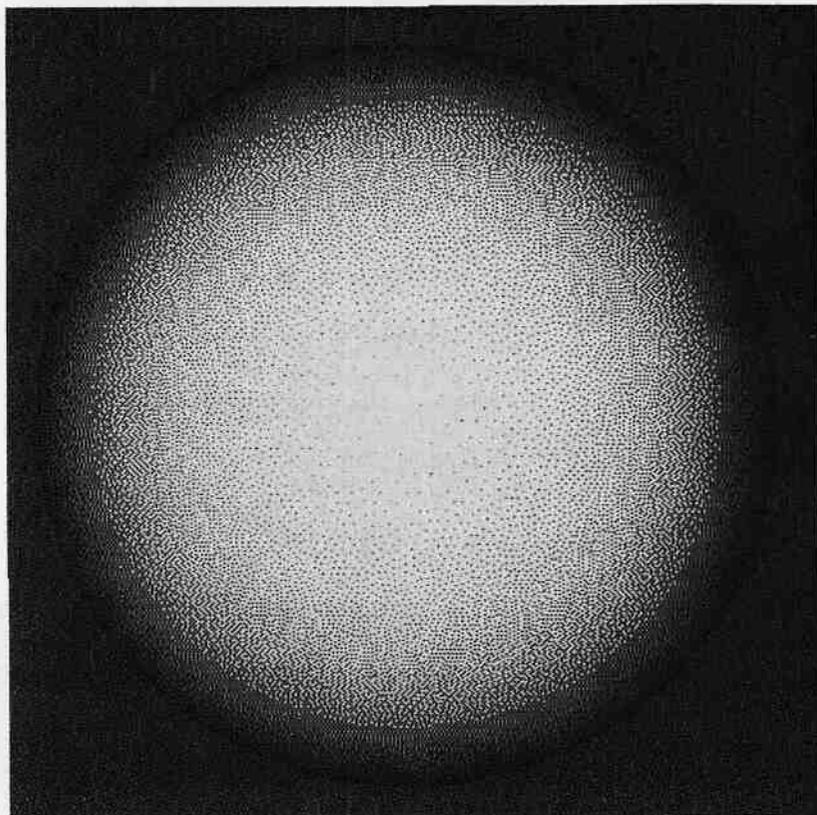


FIG. 14. Pseudo gray-level image of a Lambertian sphere illuminated by a light source near the viewer. This is the synthetic image used as input to the algorithm.

The discussion earlier showed that at an occluding boundary, the surface normal is parallel to the normal to the silhouette at the corresponding point in the image. We can therefore determine the surface orientations at the occluding boundary. This boundary information is shown in Fig. 15a in the form of a needle diagram.

The algorithm requires at least as many iterations as there are nodes across the image. This can be explained as follows: At a point near one side of a

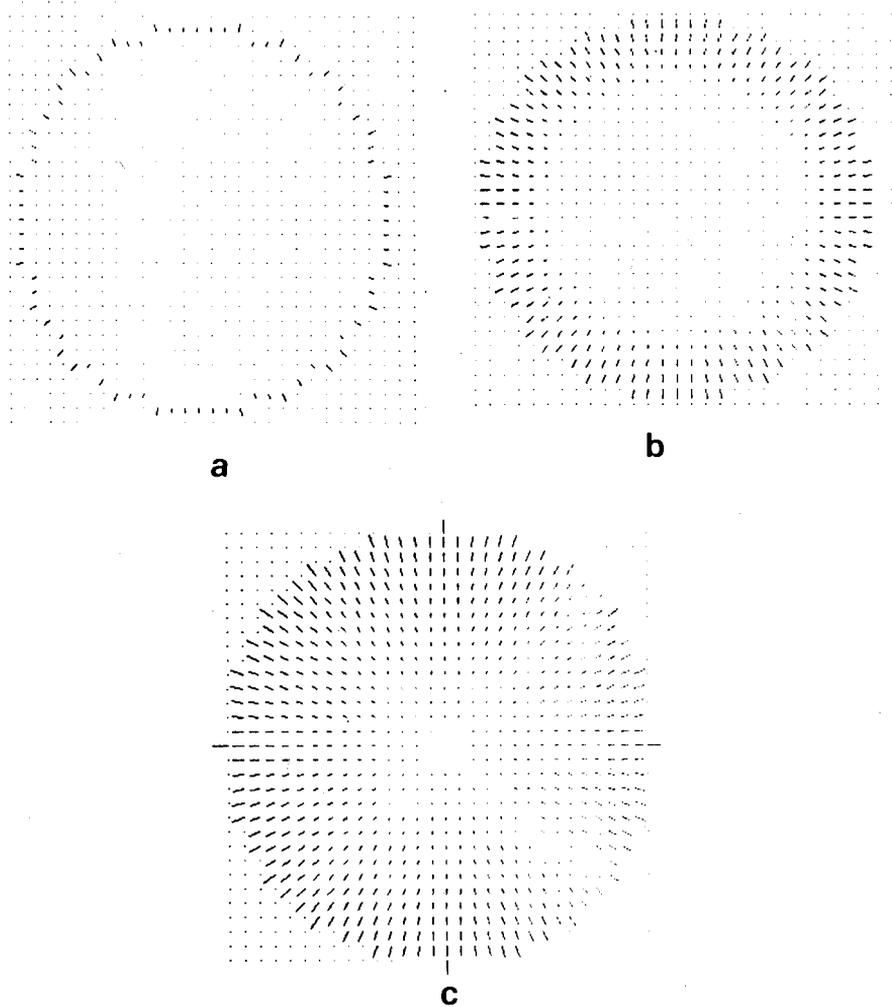


FIG. 15. Needle diagrams showing various stages of the iterative solution: (a) This is the initial orientation array, obtained from boundary information. (b) Result after 5 iterations. (c) Result after 30 iterations.

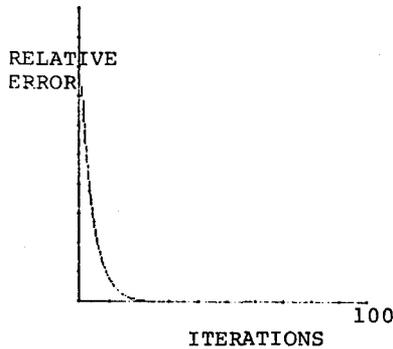


FIG. 16. This shows how the root-mean-square of error in orientation decreases with the number of iterations.

region surrounded by an occluding boundary, one needs information from the opposite side to obtain the exact orientation. At each iteration, information available at one point propagates only to its immediate neighbors. This means that it takes as many iterations as there are nodes across the object for a point near the boundary to begin to be influenced by the opposite boundary. In this case, the number of nodes across is thirty and this implies that we may need more than thirty iterations. Fig. 15b shows the results after 5 iterations, while Fig. 15c is the result after 30 iterations. Fig. 16 is the root-mean-square error (the difference between the true values of orientation and those computed by the algorithm) plotted as a function of the number of iterations.

The error gradually decreases, and after thirty iterations, the relative error is less than 0.01%. Fig. 17 is a view of the surface generated from the surface orientation data that was used to make the needle diagram.

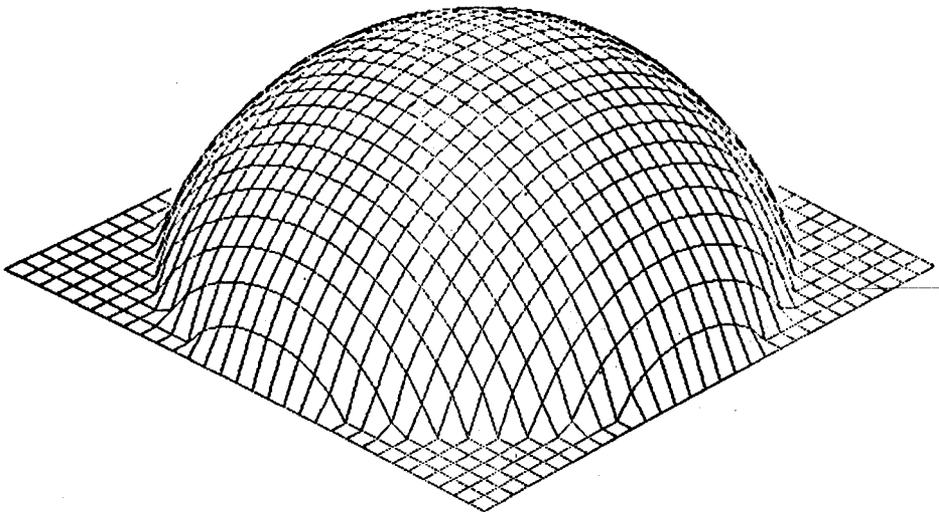


FIG. 17. The surface generated by the algorithm from the synthetic image of the sphere.

4.2.2. Example 2. Effect of initial values

We proposed the use of orientations at singular points as initial values. It may be desirable that until the boundary information propagates to a point, orientation values at that point not begin to change. At a singular point, the partial derivatives of R are always zero. This means that the constraints of the image-irradiance equations are ineffective there. The orientations at each point can be initialized to $(0.0, 0.0)$. The partial derivatives for that orientation typically are non-zero, however, with the result that the iterative equations indicate adjustments at each grid node before information from the boundary has had a chance to propagate to the node. It is not clear if the number of iterations required for a reasonable solution is affected adversely by this phenomenon.

This second example illustrates the effect of initial values. An egg-shaped object is illuminated from a direction near the extension of one of its short axes, and the viewer is looking in the direction of the long axis. We use an egg shape now to demonstrate that our algorithm works on objects other than spheres. The ratio of the length of the long axis to that of the short axis is 3 in this example. The precise position of the light source is given by $(f_s, g_s) = (0.5, 0.0)$ (the angle between the incident rays and the z -axis is 53.1°).

The surface orientations on the self-shadow boundary are given initially to simplify matters. Fig. 18 shows the resulting needle diagram and Fig. 19 the relative errors, illustrating the utility of initial values at singular points. When we use singular values as initial values, we get to a reasonable solution after fewer iterations than when all points are initialized to have the orientation $(0.0, 0.0)$. A view of the surface computed is shown in Fig. 20.

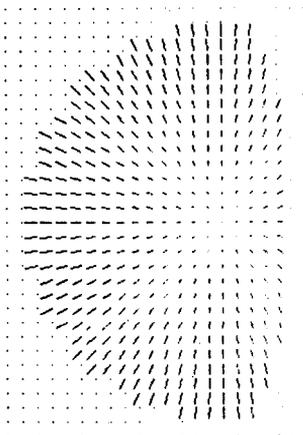


FIG. 18. The needle diagram generated for the egg-shaped object.

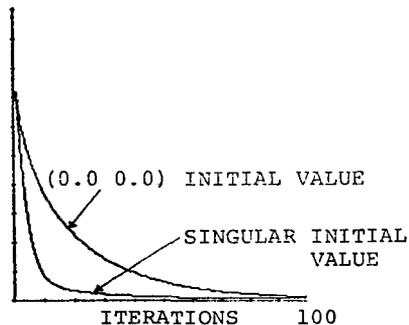


FIG. 19. The root-mean-square error in orientation for the egg-shaped object. The error is lower after a given number of iterations when the singular point is used to provide initial values.

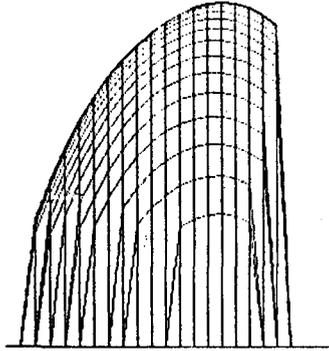


FIG. 20. The surface generated by the algorithm from the synthetic image of the egg-shaped object.

4.2.3. Example 3. *Negative Gaussian curvature*

The two previous examples involved surfaces whose Gaussian curvature is positive [Cohn-Vossen, 1952; Pogorelov, 1956; Moon and Spencer, 1969; do Carmo, 1976]. The third example involves a hyperbolic shape, one with negative Gaussian curvature. Unfortunately, there are no occluding boundaries on this surface. Instead, we provide the algorithm with orientations on a closed curve as boundary conditions. The light source is again near the viewer in this example. Fig. 21 is the needle diagram generated for this surface.

It takes about the same number of iterations to obtain a reasonable solution as in the first example. This result illustrates that the algorithm is not sensitive to whether a surface has negative or positive Gaussian curvature. In other

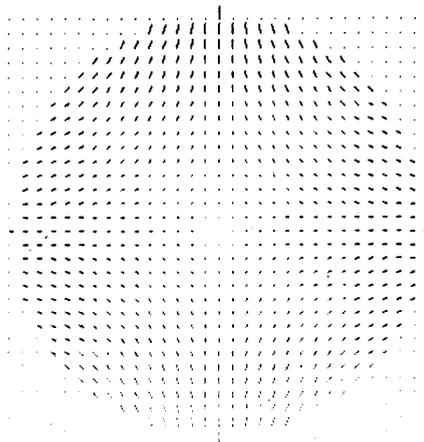


FIG. 21. The needle diagram of the solution obtained for the hyperbolic surface.

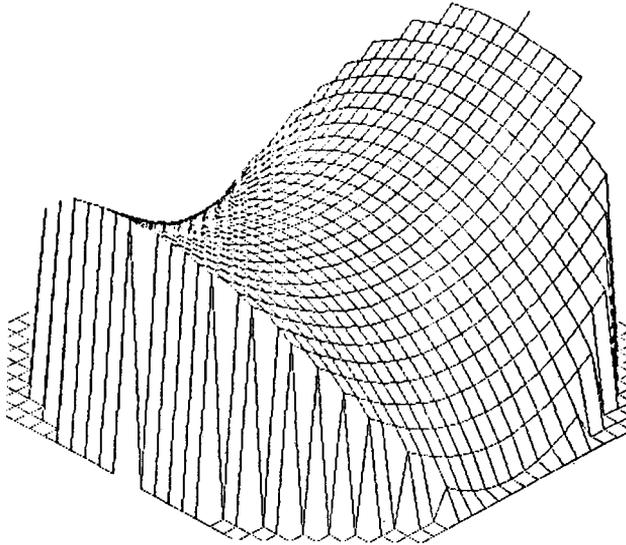


FIG. 22. The surface generated by the algorithm from the synthetic image of the hyperbolic surface.

words, we can use this algorithm without paying attention to the global characteristics of the surface, whether it be concave or convex, and whether its surface consists of hyperbolic or elliptical points [Woodham, 1977]. Fig. 22 is a view of the surface generated by the algorithm.

4.2.4. Example 4. *Effect of incomplete boundary information*

If no boundary information is available at all, then we can say very little about the shape of the surface. We would, however, like to get some estimate of the surface shape when only some of the boundary information is missing.

In the example shown here, we do not provide explicit boundary information on the self-shadow boundary. There are two motivations for considering this situation. One is that, as mentioned earlier, we cannot determine surface orientations uniquely there anyway. The other is that it sometimes happens that we cannot see all of the surface of an object, and we have to do without information on part of the boundary, leaving it unconstrained. The algorithm should try to obtain reasonable surface orientations even in situations like that (although one might expect that there may not be a unique solution to the shape-from-shading problem in this case).

We imagine that a Lambertian sphere is illuminated from the direction $(f_s, g_s) = (0.5, 0.0)$. No information is provided on the self-shadow boundary. Fig. 23 shows the relative errors in the solution obtained by the algorithm, plotted against the number of iterations.

The conclusion is that the information about orientations on the self-shadow boundary helps the algorithm arrive at a good solution after fewer steps, but is

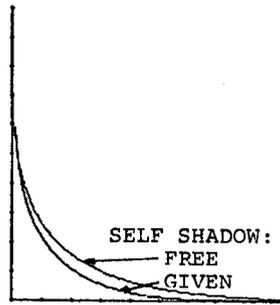


FIG. 23. The root-mean-square error in orientation when boundary information is incomplete decreases more slowly than when it is complete. A reasonable solution is obtained in both cases.

not necessary for convergence. A closed boundary does not seem to be a necessary condition for the algorithm to arrive at a solution. If an object is partially obscured by another, it may still be possible to obtain a 'reasonable' solution, one which does not contradict any of the information provided by the image.

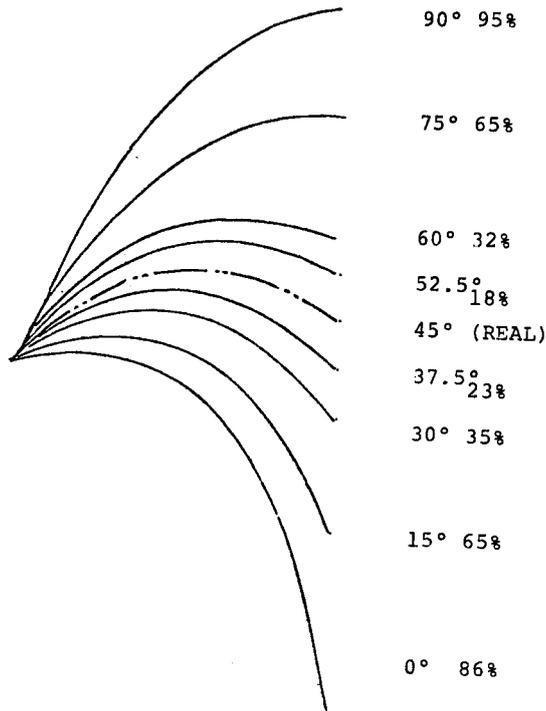


FIG. 24. The effect of inaccurate estimation of the position of the light source. The source is actually at 45°, while the program is misled into believing it is at a variety of angles, from 0° to 90°. The interrupted line shows a cross section through the correct solution.

4.2.5. Example 5. *Effect of error in estimating light-source position*

In practice, we may not be able to determine the light-source position precisely. In this example the light-source position is given incorrectly. It is reasonable to insist that the algorithm still work, although, of course, the solution cannot be expected to be accurate.

A synthesized image of a sphere is made with the illumination coming from a direction which makes an angle of 45° with the z -axis. Then the algorithm tries to determine surface orientations under the assumption that the light source is at some other position, ranging from 0° to 90° . As one can see from Fig. 24, an error of 7.5° in estimating the source direction causes no more than 20% error in surface height.

4.2.6. Example 6. *Effect of error in assumed reflectance properties*

In this experiment, the algorithm uses a reflectance map based on incorrect assumptions about the reflectance properties of the surface. Egg shapes, whose axis ratios are 5, 3, and 1 are illuminated from a light source placed near the viewer. The surface has brightness linearly dependent on the incident angle (rather than the cosine of the incident angle):

$$1 - 2i/\pi.$$

The algorithm, however, assumes that the surface is Lambertian. The output still resembles the correct result as shown by the cross sections in Fig. 25.

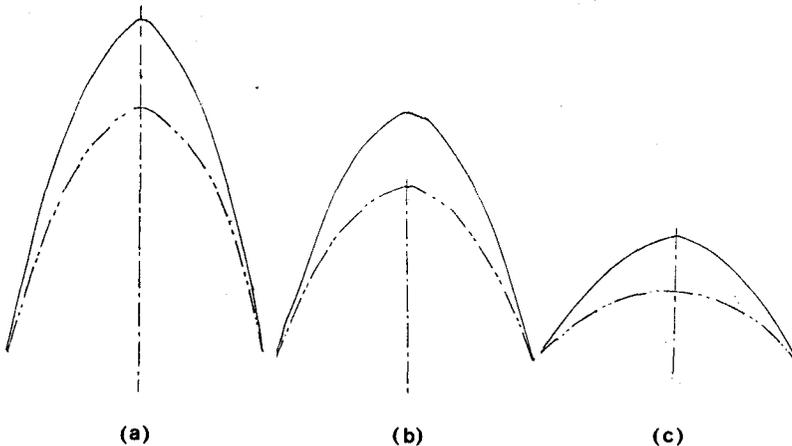


FIG. 25. Cross sections through shapes computed when the reflectance properties were different from those assumed by the program. (a) Egg shape with ratio of axes equal to 5. (b) Egg shape with ratio of axes equal to 3. (c) Egg shape with ratio of axes equal to 1. In each case the interrupted line shows a cross section through the correct solution.

4.2.7. Example 7. Analysis of a SEM picture

Here the algorithm is applied to a picture which was produced by a scanning electron microscope (SEM). Such devices are described in a book by Wells [Wells et al., 1974], and pictures made using them can be found in a number of popular books [Scharf, 1977; Grillone, 1978] and articles [Echlin, 1968]. SEM images are in many ways similar to images obtained in an optical system when the light source is symmetrically distributed about the viewer. The reflectance map is rotationally symmetric about the origin in both cases, and none of the visible areas are shadowed. One minor difference is that, in an SEM image, a surface patch oriented with its normal pointing at the viewer has minimum brightness, not maximum.

The object shown in the pseudo gray-level image that appears as Fig. 26 is a protuberance on a young leaf [Scharf, 1977, p. 96]. The reflectance map we used at first was based on the data of Laponsky [Laponsky and Whetten, 1960],

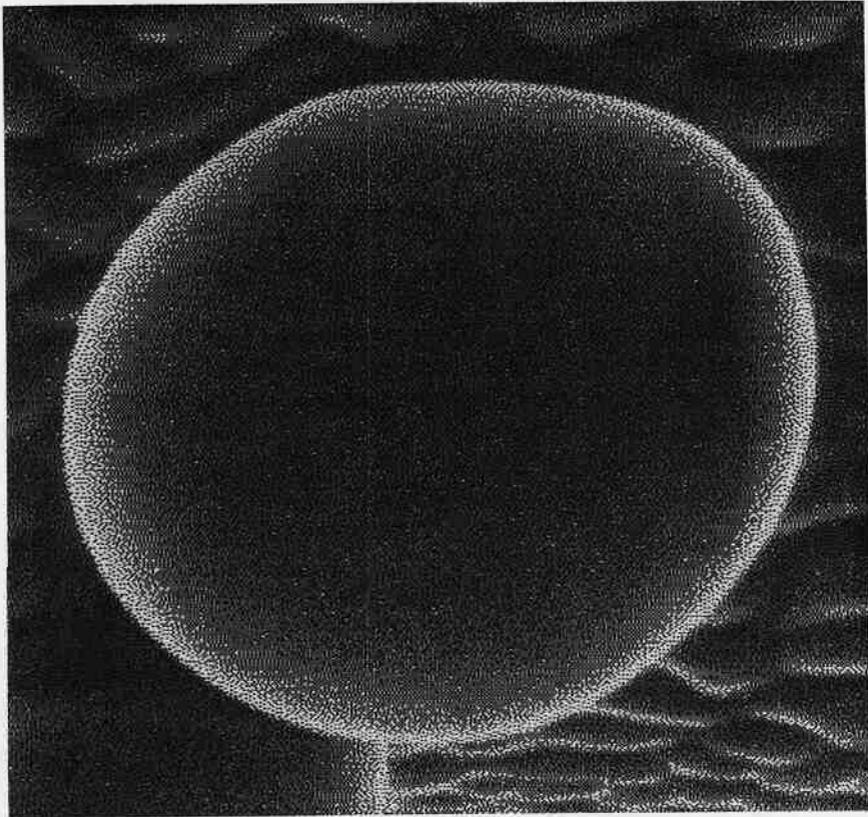


FIG. 26. A pseudo gray-level picture made from a scanning electron microscope image of a protuberance on a young leaf of *cannabis sativa indica*.

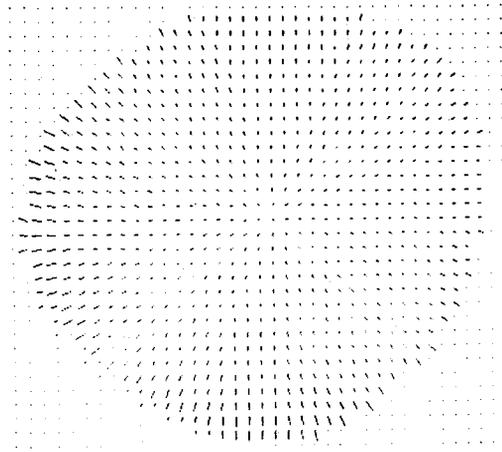


FIG. 27. The needle diagram computed for the object in the previous figure.

which suggests that secondary electron flux in a Scanning Electron Microscope varies approximately as the secant of the incident angle. Fig. 27 is the needle diagram which the algorithm computed, and views of the surface generated are shown in Fig. 28.

The computed surface appears to be a bit too flat, probably as a result of incorrect assumptions about the form of the reflectance map, as well as the fact that we did not take into account the changes in gray-level that are introduced

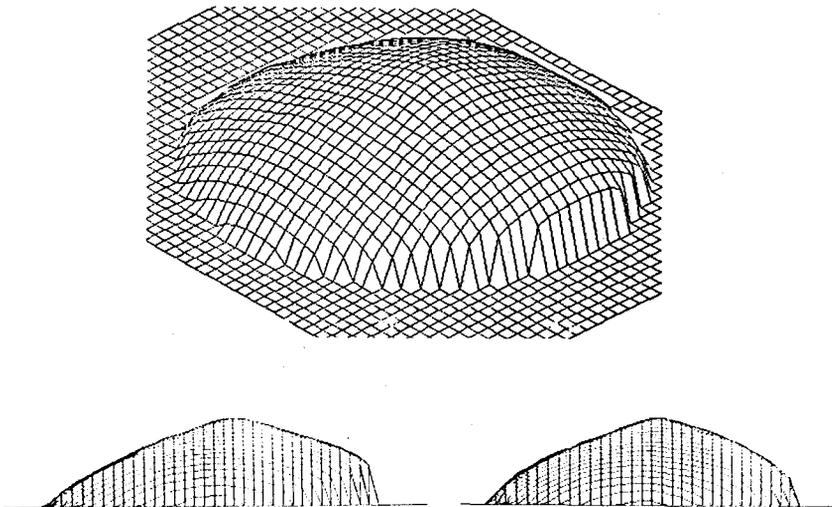


FIG. 28. Various views of the surface computed from the scanning electron microscope picture.

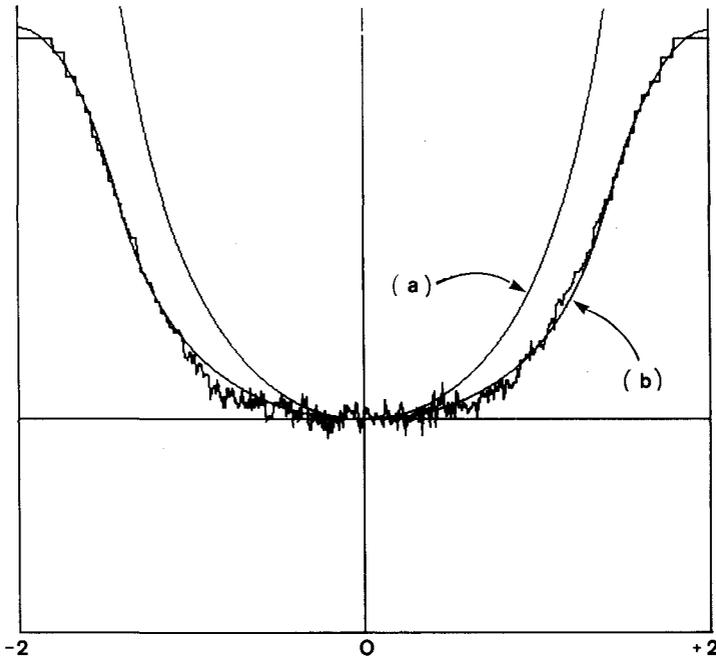


FIG. 29. Cross section through the reflectance map based on the assumption that the resin droplet is spherical. The horizontal coordinate equals $\sqrt{f^2 + g^2} = 2 \tan i/2$. Also shown for comparison are (a) the function $\sec i$ and (b) a function which equals $(1 + \sec i)/2$ for $i < 70^\circ$ and saturates for larger incident angles.

by the lithographic reproduction of the picture. The data of Müller [Müller, 1937] suggests that the secondary electron flux grows more slowly than the secant of the incident angle, at least for amorphous metals. Careful measurements on a more or less spherical resin droplet imaged in another picture of the same plant confirmed this for our example. Shown in Fig. 29 is a cross section through the reflectance map obtained from this 'calibration object'.

An approximation for the brightness, which remains finite for all incident angles,

$$e^{\alpha(1-\cos i)},$$

has been suggested by Bruining [Bruining, 1954]. The best fit of this function to our data occurred for $\alpha \approx 1$. A considerably better fit was provided by a function of the form

$$\sec ki$$

for $k \approx 0.8$. For incident angles less than 70° , however, the best fit was obtained from

$$(1 - s) + s \sec i$$

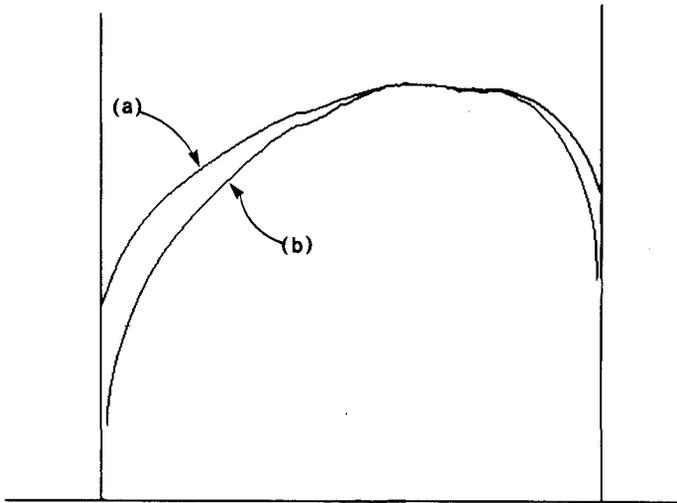


FIG. 30. Cross sections through the surfaces computed from the image of the protuberance using the two reflectance functions shown in Fig. 29.

for $s \approx \frac{1}{2}$. The only remaining discrepancy results from the fact that the observed brightness saturates at a finite value for incident angles close to 90° . An ad hoc correction for this was incorporated in our estimate, also shown in Fig. 29.

Shown in Fig. 30 are cross sections through two reconstructions. The upper one (a) corresponds to the result shown in Fig. 28, while the lower one (b) was obtained using the reflectance function $(1 + \sec i)/2$, modified to take into account the saturation of brightness for large incident angles. The shape computed using the latter reflectance function is in better accord with human judgment.

The lesson is that it is better to use a calibration object of known shape for the determination of the reflectance map than to depend on measurements obtained under somewhat different circumstances, or on a theoretically derived approximation.

5. Concluding Remarks

We proposed an algorithm for computing surface orientations from shading information using an iterative method. We then demonstrated that this algorithm does successfully compute surface orientations under a variety of conditions.

5.1. Relevance to applications in machine vision

One use of the algorithm is, of course, to compute shape from shading in a

single image. This algorithm is especially effective in the analysis of scanning electron microscope pictures of smooth parts of objects.

Another application is the improvement of the results produced by the photometric stereo method. We do not need any assumptions about surface smoothness to implement the photometric stereo method. However, when there is noise in the measured brightness values, the computed orientations will be in error. This error can be drastically reduced if we do introduce the constraint of surface smoothness. If, for example, there is a dirty spot on the object, the measured brightness values will be too low. The lookup table used in the photometric stereo method may then not have a surface orientation entry for these erroneous values. In this case, surface orientation can be filled in from the neighboring points by averaging. This is the simplest use of the method. Output from a photometric stereo system [Silver, 1980] has been smoothed successfully using this method.

An even better way to proceed is to minimize the weighted sum of the errors in the image-irradiance equations and the smoothness measure. The solution is then, in effect, pulled towards each of the constraint lines of the reflectance map and towards the local average of the surface orientations. It ends up in some compromise position which minimizes the overall 'strain'. The weights, by the way, can be set according to known values of noise in the measurements. This method has in fact been used in some recent work on photometric stereo. (Compare Fig. 8.2 with Fig. 8.4 in [Ikeuchi, 1981].) In the cases shown there, the algorithm not only smoothed the output, but also extended the area in which a solution could be found.

5.2. Relevance to basic research in machine vision

The algorithm still works reasonably when the reflectance map is only a crude approximation. As can be seen from Example 5, if one is prepared to tolerate errors of up to 20%, then one can be off by as much as 7.5° in estimating where the light source is. Similarly, Example 6 shows that if this kind of error is acceptable, it does not matter whether the surface is Lambertian or one with somewhat different reflectance properties. Thus, if one is willing to accept 20% estimation errors, only six or seven reflectance maps are needed for different light-source position, multiplied by two or three to allow for different surface materials. Note that we don't really need different reflectance maps for different light-source azimuths, since the reflectance map can be rotated to accommodate changes in azimuth of the light source. The nature of the reflectance map does, however, change when the zenith angle of the light source changes.

Propagation of constraints is a powerful method for attacking vision problems. Clarification of the constraints or relationships between neighboring nodes often gives us the key to the solution of a problem. The propagation of

constraints from neighboring nodes cuts down the space of possible solutions at a node [Zucker, 1976]. In vision, the number of nodes is large and the possible values lie in a continuous range. This means that the solution space is rather large. In this case, the propagation technique is more effective than the search technique. Even when the propagation technique cannot determine a unique solution, it can reduce the number of possible solutions so that subsequent search can be more effective [Waltz, 1975].

There are several examples in vision where this method has been employed effectively. An iterative method is used to invert the Laplace operator in the computation of lightness [Horn, 1974]. A *pseudo-local* operator is a global operator which can be computed by iteration of a local operator because it happens to be the inverse of a local operator [Horn, 1982]. In the computation alluded to here, the so-called *G* operator (pseudo-local) calculates a lightness value from those at neighboring points and the thresholded result of the so-called *L* operator (local) applied at the corresponding point in the image. The first stereo algorithm of Marr [Marr and Poggio, 1976] used this idea too. Here, the 'continuity rule' is an example of positive propagation, requiring that neighboring points have similar disparity values. The 'uniqueness rule' is an example of negative propagation, causing the value at a node to inhibit neighboring nodes from taking on the same value. Iterative computations similar to the ones presented in this paper also occur in the estimation of optical flow from image sequences [Horn and Schunck, 1981]. The shape from regular pattern algorithm [Ikeuchi, 1980] is another example which uses a very similar method. Needless to say, the relaxation technique [Zucker, 1976; Rosenfeld, 1978; Hummel and Zucker, 1980; Davis and Rosenfeld, 1981] is an implementation method for the propagation of constraints.

The propagation of constraint method is suggestive of parallel processing in a planar computational network. The relationship between nodes is homogeneous. Each node connects to its neighbors in the same fashion, independent of its position within the network. An essential operation in many of the algorithms mentioned here is the computation of a weighted average over the neighboring nodes. This is the kind of operation that even a simple array processor can perform easily.

ACKNOWLEDGMENT

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. 1 is from Horn's paper "Understanding Image Intensities". Figs. 2 and 4 are from R.J. Woodham's thesis, "A Machine Vision System for Understanding Surface Defects in Metal Castings". Fig. 3 is from Horn's paper "Sequins & Quills". Figs. 7 and 8 are out of D. Hilbert and S. Cohn-Vossen's book *Geometry and the Imagination* (Copyright 1952, Chelsea Publishing Company). Fig. 26 is based on a picture in David Scharf's book *Magnifications* (Copyright 1977, Schocken Books Incorporated). These figures are reproduced here by permission of the publishers. We also thank J. Jones for drawing some of the other figures.

The authors would like to extend their sincere appreciation to Professor Patrick H. Winston for encouragement. Discussions with K. Murota of the University of Tokyo, T. Kanade of Carnegie Mellon University, and A. Bruss, W. Silver, and R. Sjoberg of M.I.T. were helpful. Thanks go to M. Brady, K. Forbus, E. Grimson, E. Hildreth, K. Prendergast, C. Roberts and K. Stevens of M.I.T. for helping to proofread the manuscript. We are also grateful to the LISP-Machine Group of the Artificial Intelligence Laboratory for the construction of the LISP Machine. Without the LISP Machine, we could not have completed the experimental part of this research so rapidly.

REFERENCES

1. Ames, W.F. (1977), *Numerical Methods for Partial Differential Equations*, Academic Press, New York.
2. Barrow, H.G. and Tenenbaum, J.M. (1981), Interpreting line drawings as three-dimensional surfaces, *Artificial Intelligence* 17, 75–116 [this volume].
3. Brooks, M.J. (1979), Surface normals from closed paths, *Proceedings of the Sixth IJCAI*, Tokyo, Japan, 98–101.
4. Bruining, H. (1954), *Physics and Application of Secondary Electron Emission*, Pergamon, New York.
5. Bruss, A.R. (1979), Some properties of discontinuities in the image irradiance equation, AI Memo 517, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
6. Bruss, A.R. (1980), The image-irradiance equation, its solution and application, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
7. do Carmo, M.P. (1976), *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs.
8. Carrier, G.F. and Pearson, C.E. (1976), *Partial Differential Equations—Theory and Technique*, Academic Press, New York.
9. Conte, S.D. and de Boor, C. (1972), *Elementary Numerical Analysis*, McGraw-Hill, New York.
10. Courant, R. and Hilbert, D. (1953), *Methods of Mathematical Physics—Volume I*, Interscience, New York.
11. Courant, R. and Hilbert, D. (1962), *Methods of Mathematical Physics—Volume II*, Interscience, New York.
12. Davis, L.S. and Rosenfeld, A. (1981), Cooperative processes for low-level vision: a survey, *Artificial Intelligence*.
13. Draper, S.W. (1980), The use of gradient and dual space in vision, Internal Report, Department of Computer Science, Sussex University.
14. Echlin, P. (1968), Pollen, *Scientific American* 218 (4) 80–90.
15. Garabedian, P.R. (1964), *Partial Differential Equations*, Wiley, New York.
16. Grillone, L. and Gennaro, J. (1978), *Small Worlds Close Up*, Crown Publishers, New York.
17. Hamming, R.W. (1962), *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York.
18. Hilbert, D. and Cohn-Vossen, S. (1952), *Geometry and the Imagination*, Chelsea, New York.
19. Hildebrand, F.B. (1965), *Methods of Applied Mathematics*, Prentice-Hall, Englewood Cliffs.
20. Hildebrand, F.B. (1974), *Introduction to Numerical Analysis*, McGraw-Hill, New York.
21. Horn, B.K.P. (1970), Shape from shading: a method for obtaining the shape of a smooth opaque object from one view, Technical Report MAC-TR-79, Project MAC, Massachusetts Institute of Technology.
22. Horn, B.K.P. (1974), Determining lightness from an image, *Computer Graphics and Image Processing* 3(1) 111–299.
23. Horn, B.K.P. (1975), Obtaining shape from shading information, in: Winston, P.H. (Ed.), *Psychology of Computer Vision*, McGraw-Hill, New York, 115–155.

24. Horn, B.K.P. (1977), Understanding image intensities, *Artificial Intelligence* 8 (2) 201-231.
25. Horn, B.K.P. (1981), Hill-shading and the reflectance map, *Proceedings of the IEEE* 19 (1) 14-47.
26. Horn, B.K.P. (1981), Sequins & Quills—representations for surface topography, in: Bajcsy, R. (Ed.), *Representation of 3-Dimensional Objects*, Springer, Berlin.
27. Horn, B.K.P. and Schunck, B.G. (1981), Determining optical flow, *Artificial Intelligence* 17, 185-203 [this volume].
28. Horn, B.K.P. and Sjöberg, R.W. (1979), Calculating the reflectance map, *Applied Optics* 18 (11) 1770-1779.
29. Horn, B.K.P., Woodham, R.J. and Silver, W.M. (1978), Determining shape and reflectance using multiple images, AI Memo 490, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
30. Huffman, D.A. (1971), Impossible objects as nonsense sentences, in: Meltzer, R. and Michie, D. (Eds.), *Machine Intelligence* 6, Edinburgh University Press, Edinburgh, 295-323.
31. Hummel, R.A. and Zucker, S.W. (1980), On the foundations of relaxation labeling processes, Report No. 80-7, Computer Vision and Graphics Laboratory, Department of Electrical Engineering, McGill University, Montreal, Quebec.
32. Ikeuchi, K. (1980), Shape from regular patterns—an example of constraint propagation in vision, *Proceedings of the Fifth IJ CPR*, Miami Beach.
33. Ikeuchi, K. (1981), Determination of surface orientations of specular surfaces by using the photometric stereo method, accepted for publication by *IEEE-PAMI*.
34. John, F. (1978), *Partial Differential Equations*, Springer, New York.
35. Laponsky, A.B. and Whetten, N.R. (1960), Dependence of secondary emission from MgO single crystals on angle of incidence, *Physics Review* 120 (3) 801-806.
36. Mackworth, A.K. (1973), Interpreting pictures of polyhedral scenes, *Artificial Intelligence* 4 (2) 121-137.
37. Marr, D. (1977), Analysis of occluding contour, *Proc. Roy. Soc. London Ser. B* 197, 441-475.
38. Marr, D. and Poggio, T. (1976), Cooperative computation of stereo disparity, *Science* 194, 283-287.
39. Milne, W.E. (1970), *Numerical Solution of Differential Equations*, Dover, New York.
40. Moon, P. and Spencer, D.E. (1969), *Partial Differential Equations*, D.C. Heath, Lexington.
41. Müller, H.O. (1937), Die Abhängigkeit der Sekundärelektronenemission einiger Metalle vom Einfallswinkel des primären Kathodenstrahls, *Zeitschrift für Physik* 104, 475-486.
42. Nicodemus, F.E., Richmond, J.C., Hsia, J.J., Ginsberg, I.W. and Limperis, T. (1977), *Geometrical Considerations and Nomenclature for Reflectance*, NBS Monograph 160, U.S. Department of Commerce, National Bureau of Standards.
43. Pogorelov, A.V. (1956), *Differential Geometry*, Noordhoff, Groningen, The Netherlands.
44. Raisz, E. (1962), *Principles of Cartography*, McGraw-Hill, New York.
45. Richtmyer, R.D. and Morton, K.W. (1967), *Difference Methods for Initial-Value Problems*, Interscience, New York.
46. Rosenfeld, A. (1978), Iterative methods for image analysis, *Pattern Recognition* 10, 181-187.
47. Scharf, D. (1977), *Magnifications—Photography with the Scanning Electron Microscope*, Schocken Books, New York.
48. Silver, W. (1980), Determining shape and reflectance using multiple images, S.M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.
49. Sohon, F.W. (1941), *The Stereographic Projection*, Chelsea, New York.
50. Spacek, L.A. (1979), Shape from shading and more than one view, Internal Communication, Department of Computer Science, Essex University.
51. Strat, T.M. (1979), A numerical method for shape from shading from a single image, S.M. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

52. Thomas, P.D. (1952), *Conformal Projections in Geodesy and Cartography*, Special Publication No. 251, U.S. Department of Commerce, Coast and Geodetic Survey, Washington, D.C.
53. Waltz, D. (1975), Understanding line drawings of scenes with shadows, in: Winston, P.H. (Ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York, 19-91.
54. Wells, O.C., Boyde, A., Lifshin, E. and Rezanowich, A. (1974), *Scanning Electron Microscopy*, McGraw-Hill, New York.
55. Witkin, A.P. (1980), Shape from contour, Ph.D. Thesis, Department of Psychology, Massachusetts Institute of Technology.
56. Woodham, R.J. (1977), A cooperative algorithm for determining surface orientation from a single view, *Proceedings of the Fifth IJCAI*, Cambridge, MA, 635-641.
57. Woodham, R.J. (1978), Reflectance map techniques for analyzing surface defects in metal castings, Technical Report AI-TR-457, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
58. Woodham, R.J. (1979), Relating properties of surface curvature to image intensity, *Proceedings of the Sixth IJCAI*, Tokyo, Japan, 971-977.
59. Woodham, R.J. (1980), Photometric method for determining surface orientation from multiple images, *Optical Engineering* **19** (1) 139-144.
60. Zucker, S.W. (1976), Relaxation labelling and the reduction of local ambiguities, *Proceedings of the Third IJCP*, San Diego, CA.

Received March 1980; revised version received October 1980