CMU-CS-82-100

Using Shadows in Finding Surface Orientations

4 January 1982

Steven A. Shafer
Takeo Kanade

Computer Science Department
Carnegie-Mellon University

This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-81-K-1539.

Table of Contents

List of Figures

Abstract

Given a line drawing from an image with shadow regions identified, the shapes of the shadows can be used to generate constraints on the orientations of the surfaces involved. This paper describes the theory which governs those constraints under orthography.

A "Basic Shadow Problem" is first posed, in which there is a single light source, and a single surface casts a shadow on another (background) surface. There are six parameters to determine: the orientation (2 parameters) for each surface, and the direction of the vector (2 parameters) pointing at the light source. If some set of 3 of these are given in advance, the remaining 3 can then be determined geometrically. The solution method consists of identifying "illumination surfaces" consisting of illumination vectors, assigning Huffman-Clowes line labels to their edges, and applying the corresponding constraints in gradient space.

The analysis is extended to shadows cast by polyhedra and curved surfaces. In both cases, the constraints provided by shadows can be analyzed in a manner analogous to the Basic Shadow Problem. When the shadow falls upon a polyhedron or curved surface, similar techniques apply. The consequences of varying the position and number of light sources are also discussed. Finally, some methods are presented for combining shadow geometry with other gradient space techniques for 3D shape inference.

1. Introduction

1.1 The Shadow Geometry Problem

In many images, shadows are present (figure 1-1). When this is the case, the shadows provide some information which is useful for determining the 3D shapes and orientations of the objects in the scene.

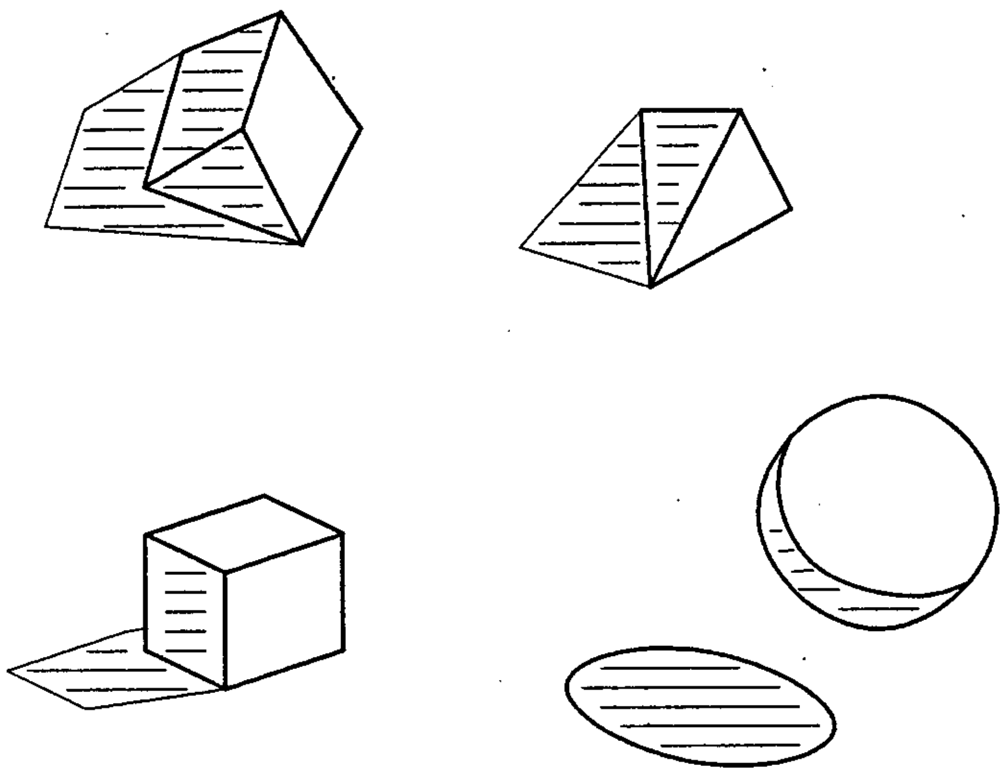


Figure 1-1: Shadows Provide Information for 3-D Shape Recovery

The interpretation of shadows in an image involves three distinct processes:

- Finding shadow regions in the image
- Solving the correspondence problem to determine which object has cast each shadow region
- Geometrically deducing information about the objects and surfaces involved on the basis of the identified object/shadow pairs

To date, most researchers have performed each step in sequence, although the latter steps certainly generate information which can be used to improve the former processes if they have been incomplete or noisy.

Techniques for the first step, *finding shadow regions*, have been proposed by many researchers, usually by looking for regions of low intensity with approximately the same hue as some neighboring region [12, 14]. A close examination of region colors will reveal that shadows due to the sun will have a slightly bluer hue than illuminated portions of the same surface. Lowe and Binford [9] proposed criteria which should be satisfied by edges of shadow regions; these can be used to suggest or try to confirm the hypothesis that a particular region is a shadow. Witkin [20] is also investigating shadow edges. Waltz [18] developed a method for labeling lines in line drawings as shadow edges, based on local geometric criteria at vertices.

The *correspondence problem* has been explored primarily by Lowe and Binford [9]. They describe several properties of this correspondence, and include descriptions of the special points of view from which degenerate cases arise. O'Gorman [13] proposed a heuristic method for finding correspondences in the blocks world under orthography.

Geometric interpretation of shadows is also performed by Lowe and Binford [9], who use shadows to determine height in overhead views of airplanes. They measure the distance in the image between the outline of an object and the outline of its shadow, and use similar triangles to conclude that this distance is proportional to the height of the object's edge above the ground. Quam [15] is also using shadows to determine depth information: These techniques have been employed in manual photo-interpretation of aerial photographs as well [17].

Waltz [18] used shadows to classify surfaces into several orientation categories depending upon the geometry of the shadows in a line drawing. His categories were qualitative, such as "front left" for an approximately vertical surface tipped to the left.

This paper presents a theory describing the constraints that shadows provide between surface orientations in line drawings, using shadow and surface outlines under orthographic projection. This can be thought of as a method for achieving the same kind of results as Waltz, but computing exact surface orientations rather than simply categorizing the surfaces into classes with similar orientations. The theory presented here subsumes the "shadow-plane" idea suggested by Mackworth [10] as a means for generating gradient-space constraints from shadows.

Shadows cast by and upon curved surfaces have been described by Witkin [19], who derived equations relating surface curvature to curvature of shadow edges in the image. The presentation in this paper is somewhat different, discussing surface *gradient* (local orientation) rather than *curvature* (rate of change of orientation).

1.2 This Presentation

This paper begins by defining the "Basic Shadow Problem", computing surface orientations from a line drawing depicting one surface casting a shadow on one other surface. The surfaces are assumed to be flat, the light source is assumed to be infinitely far away, and orthographic projection is used.

The consequences of varying the light source are then explored. These include changing the position to be in front of the camera instead of behind it, moving the light source to a point at a finite distance from the scene, and altering the number of light sources. The same Basic Shadow Problem occurs in all these cases, and the necessary modifications to the solution technique are presented.

The shaded surface -- the surface on which shadows appear -- is then generalized to be many planes (a polyhedron). The Basic Shadow Problem occurs within an algorithm to compute the orientations of each face intersecting the shadow edge.

The occluding surface -- the surface casting the shadow -- is generalized to be polyhedral. The Basic Shadow Problem is included in the shadow information available in this case, along with additional shadow-making edges.

The solution of shadow problems involving curved surfaces is then discussed. When curved surfaces are involved, additional information about the curvature is needed for an exact solution. The special case of a sphere is examined as an example in which knowledge about the surface curvature allows for the derivation of a unique solution with little *a priori* information required.

Some methods are presented for combining shadow geometry with other gradient space techniques, and with stereo image analysis.

Further plans include the elaboration of the above cases under perspective rather than orthography, and the construction of a program to perform the geometric reasoning outlined in this paper.

1.3 Introduction to Gradient Space and Line Labeling

This section presents an introduction to the gradient space and line labeling for readers who are not already familiar with these topics.

When constructing a 3D description of a scene from examination of an image, some coordinate system must be set up. The coordinate system used in this paper is illustrated in figure 1-2. Here, the x and y axes are aligned on the image plane in the horizontal and vertical directions, respectively, assigning the usual 2D Cartesian coordinate system to the image. The z axis points towards the viewer (or camera). This is the coordinate system used by Mackworth [10].

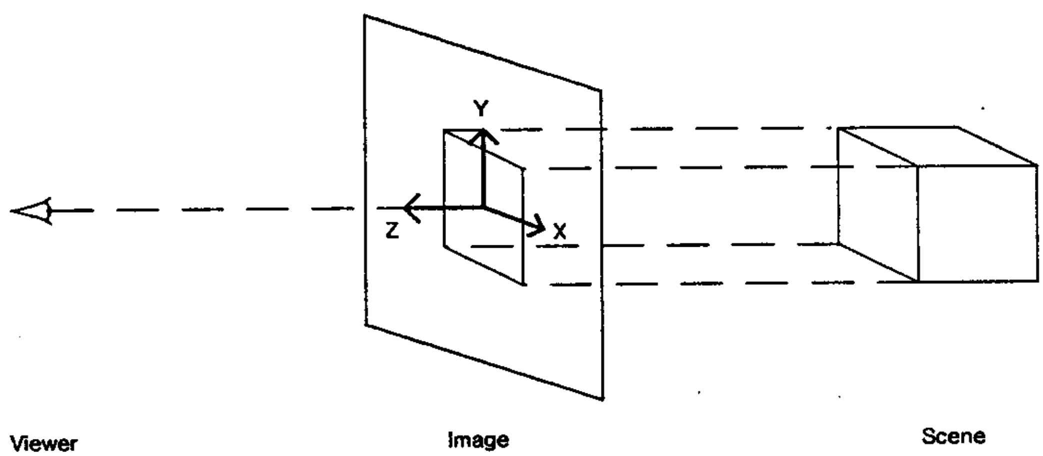


Figure 1-2: The X-Y-Z Coordinate System

In this paper, it will be presumed that the point (x,y,z) in the scene corresponds to the point (x,y) in the image. This is *orthography*. *Perspective* projection is not discussed in detail in this paper.

When describing the three-dimensional shape of an object, it is sufficient to specify the two-dimensional image together with the orientation (in three-space) of each surface in the scene. The problem of three-dimensional shape recovery is therefore equivalent to finding the orientation of each surface in the image. These orientations can be represented by points in a plane called the *gradient space* (figure 1-3) [5]. If a surface is represented by the equation

$$-z = f(x, y)$$

then its *gradient* is represented by the point:

$$(p, q) = (\partial f/\partial x, \partial f/\partial y)$$

This assigns a natural interpretation to points in gradient space: a surface which is "tipped" to the right is represented to a point on the right side of the origin; a surface tipped left has a gradient to the left of the origin. Similarly, a surface which is tipped up (or down) has its gradient above (or below) the origin. In figure 1-3, the gradients G_A (etc.) are shown for the surfaces S_A (etc.) in the line drawing at the right.

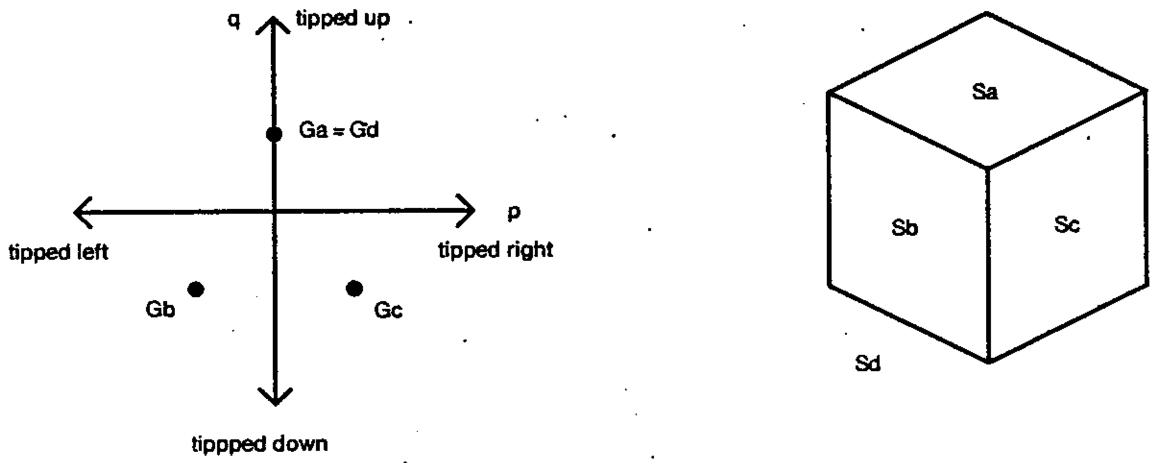


Figure 1-3: The Gradient Space

Before computing surface orientations, it is common to attempt to produce a *line drawing* from an image, in which all the surfaces are outlined. Huffman and Clowes [5, 2] showed that the edges (line segments) in a line drawing do not all represent the same three-dimensional surface configuration. The four types of edges they discovered are shown in figure 1-4, along with the half-planes containing the surfaces which meet at each type of edge. At a *convex* edge, the surfaces recede from the viewer as you travel farther from the edge. At a *concave* edge, the surfaces approach the viewer as you travel farther from the edge. At an occluding edge, only one of the two surfaces involved is directly visible in the image. Waltz [18] developed an algorithm for assigning these labels to the edges in a line drawing.

The *convex* and *concave* labels indicate relationships between the gradients of the surfaces which meet along an edge [10]. When two surfaces are joined along a convex edge, their gradients lie along a line in gradient space which is perpendicular to the edge in the image (figure 1-5). Furthermore, the relative positions of the surface gradients will be the same as the relative positions of the surfaces in the image. When two surfaces meet at a concave line, the gradients are still on a perpendicular line in gradient space, but the relative positions are reversed.

In general, if an edge $\boldsymbol{E} = (\Delta x, \Delta y)$ is contained on a surface with gradient $\mathbf{G} = (p, q)$, then the edge corresponds to the three-dimensional vector $(\Delta x, \Delta y, \Delta z)$ where

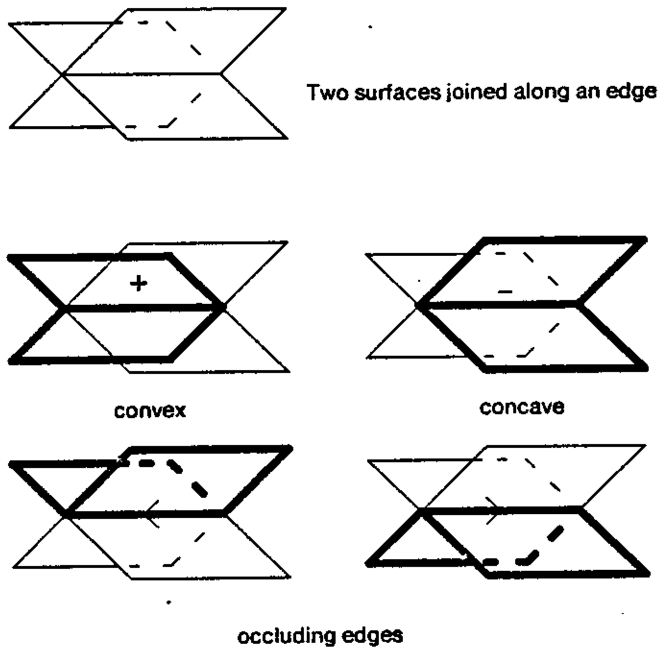


Figure 1-4: Line Labels and Surface Intersections

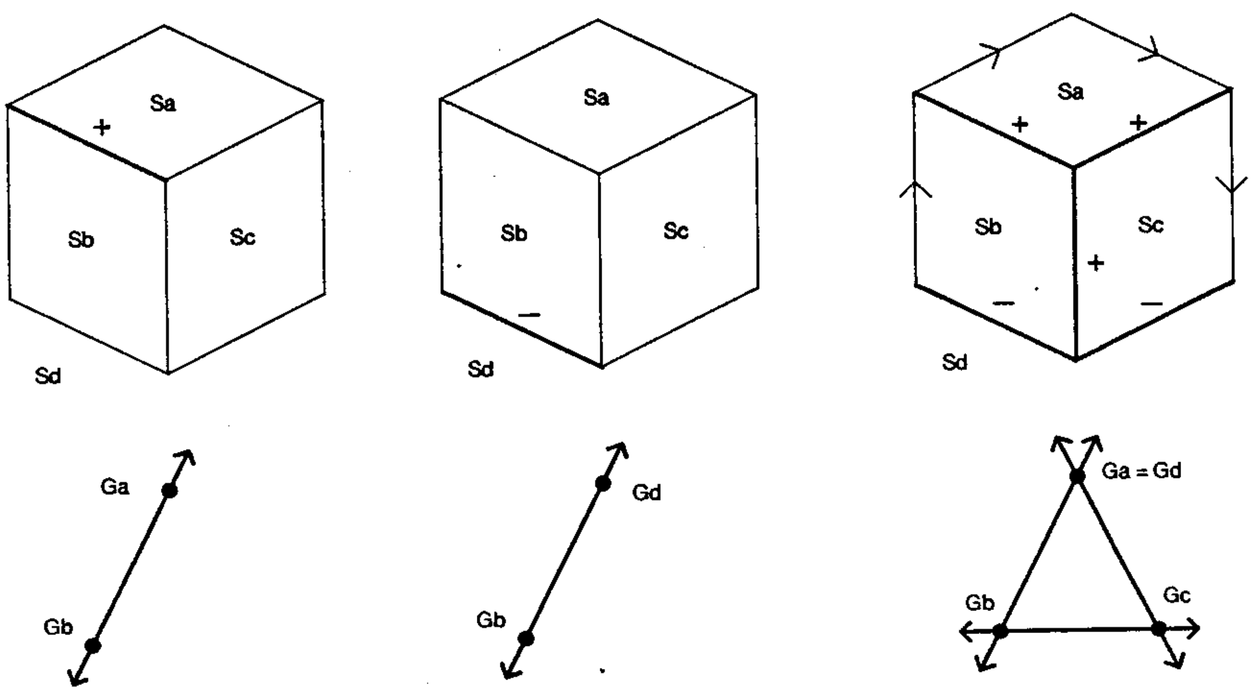


Figure 1-5: Line Labels and Gradient Space Relationships

$$-\Delta z = \mathbf{G} . \mathbf{E} \tag{1.1}$$

In this paper, a method is proposed for assigning Huffman-Clowes line labels to shadow-making edges and shadow edges in a line drawing, and for using the resulting gradient space relationships to determine surface orientations.

2. The Basic Shadow Problem

The *Basic Shadow Problem* is:

> Given a line drawing such as Figure 2-1, what constraints exist between the occluding surface S_O and the shaded surface S_S?

For simplicity, we will begin by assuming that the surfaces are both flat, and that orthographic projection is used. We will also, for the time being, presume that the light source is infinitely far away; this means that all *illumination vectors* (light rays emanating from the light source) are parallel.

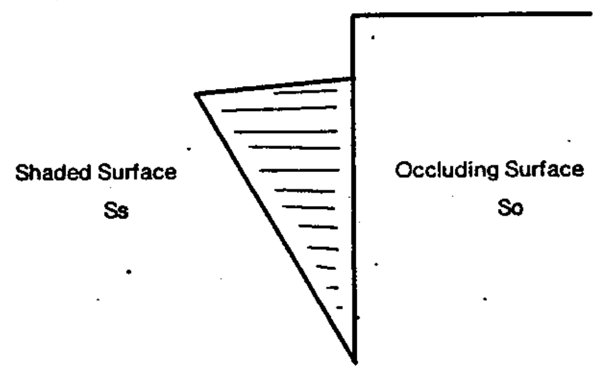


Figure 2-1: The Basic Shadow Problem

2.1 Solution of the Problem

To show the proper correspondences, the edges and vertices can be labeled as in figure 2-2, where edge E_{S1} is the shadow edge corresponding to E_{O1}, E_{S2} is the shadow of E_{O2}, and vertex V_{S12} is the shadow of V_{O12}.

Consider the physical interpretation of edge E_{S1}. Some light rays just graze past S_O at E_{O1}, and continue on to strike S_S along E_{S1}. This set of rays form a surface (a piece of a plane), in fact the plane containing E_{O1} and E_{S1}. This is a surface consisting of "illumination vectors"; call it surface S_{I1} (Figure 2-3).

Suppose we were to cut a piece of cardboard and fit it into the space occupied by S_{I1}. Then, this cardboard and S_O would be joined along E_{O1}, a *convex* edge. Using Huffman-Clowes line labeling [5], this edge can be given the label $+$. Similarly, E_{S1} joins S_S and S_{I1}, and is *concave*; it receives the label $-$.

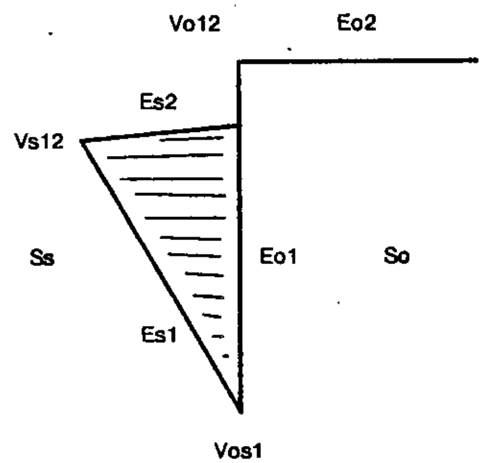


Figure 2-2: Basic Shadow Problem -- Correspondences Labeled

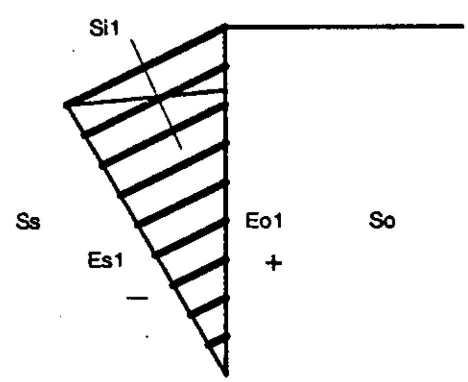


Figure 2-3: Basic Shadow Problem -- Illumination Surface 1

As Mackworth showed [10], these line labels can be mapped into constraints in the gradient space. The gradient of S_O (G_O) and the gradient of S_{I1} (G_{I1}) must be joined by a line perpendicular to E_{O1}; since the label of E_{O1} is $+$, G_O and G_{I1} have the same relative positions as S_O and S_{I1}. Similarly, G_{I1} and G_S are joined by a line perpendicular to E_{S1}, with relative positions reversed because of the $-$ label. These facts yield the relationship shown in figure 2-4 in the gradient space. However, we do not yet know the position of this figure in gradient space, nor the distances involved; only the angles are known.

S_{I1} is not the only illumination surface in the Basic Shadow Problem: the illumination surface S_{I2} joins edges E_{O2} and E_{S2} (Figure 2-5). Along E_{S2}, the $-$ label is assigned; along E_{O2}, the $-$ label refers to the junction of S_O and the upper half-plane of S_{I2}. The gradient space constraints are shown

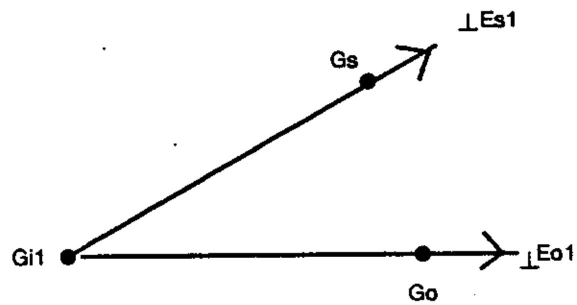


Figure 2-4: Gradient Space Constraints from Illumination Surface 1

in figure 2-6. Note that it is possible for E_{O2} and E_{S2} to be parallel, in which case the two rays shown in gradient space are coincident.

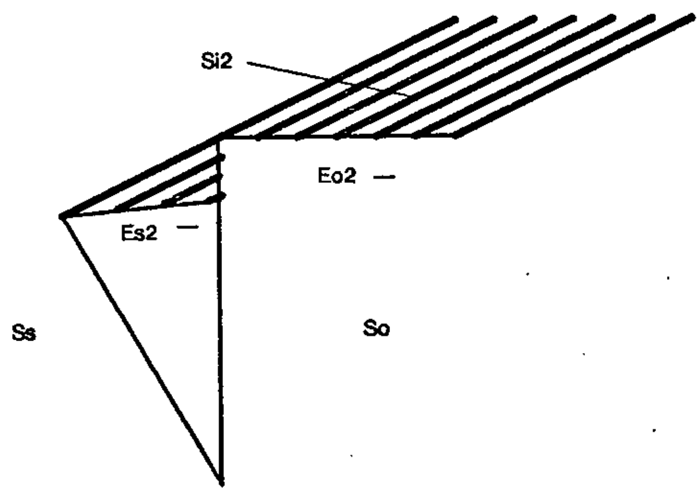


Figure 2-5: Basic Shadow Problem -- Illumination Surface 2

A third constraint in the gradient space arises from the fact that an edge E_{I1} can be drawn joining V_{O12} and V_{S12} (Figure 2-7). This edge lies in a line which passes through the light source, since V_{S12} is the shadow of V_{O12}. The vector *I* pointing at the light source can be represented in gradient space by a point G_I, which represents the intersection of a vector *I* from the origin with the plane $z = 1$. Since E_{I1} lies in the projection of this vector onto the image plane, the point G_I must lie along a line in gradient space, passing through the origin, and parallel to E_{I1} (Figure 2-8). It is not known, however, how far this point G_I is from the origin; suppose this is determined somehow (as described below), and call the distance k. It should be noted that k represents the relative change in z with a change in x or y along the illumination vector. It is defined by this equation:

Figure 2-6: Gradient Space Constraints From Illumination Surface 2

$$k = \text{sqrt}(\Delta x^2 + \Delta y^2) / \Delta z = \|E_{I1}\| / \Delta z \qquad (2.1)$$

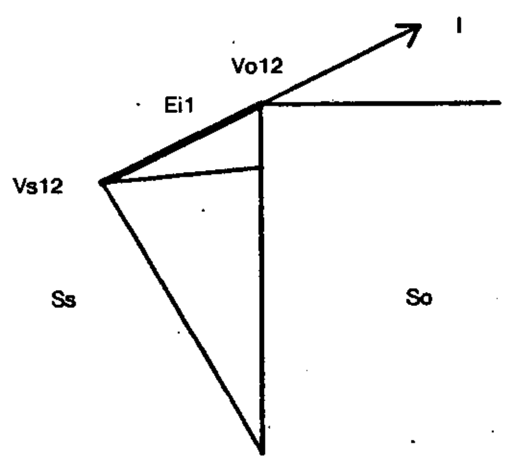


Figure 2-7: Basic Shadow Problem -- Illumination Vector

The line L_{illum} perpendicular to E_{I1}, and located at a distance $1/k$ from the origin, represents the locus of the gradients of all planes which contain the illumination vector I. This is the set of all illumination planes, and in particular contains both S_{I1} and S_{I2}; thus, G_{I1} and G_{I2} are points on the line L_{illum}. This property subsumes the property of G_{I1} and G_{I2} that they must be joined by a line perpendicular to E_{I1}, since E_{I1} can be given the label + or − (depending on which half-planes the line label refers to).

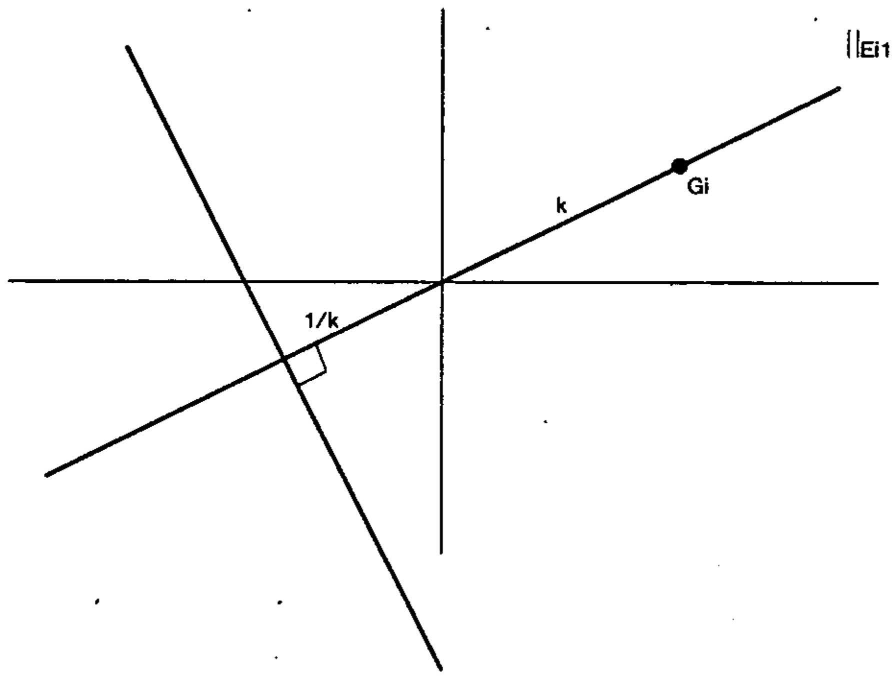


Figure 2-8: Gradient Space Constraints From Illumination Vector

The line L_{illum} is the same as the *terminator* described by Horn in [4]. It separates the gradient space into two half-planes; the half-plane containing G_I represents the gradients of all planes that will receive illumination, while the other half-plane contains the gradients of *self-shadowed* surfaces (facing away from the light source).

This is the extent of the information available from the line drawing in figure 2-1. Since each gradient is an ordered pair (p, q), the problem has six parameters to be computed:

- (2 parameters) G_O, the gradient of S_O
- (2 parameters) G_S, the gradient of S_S
- (2 parameters) G_I, the direction of the light source.

From the Basic Shadow Problem geometry, three constraints are provided:

- The angle G_O-G_{I1}-G_S, which comes from the angle E_{O1}-E_{S1}
- The angle G_O-G_{I2}-G_S, which comes from the angle between E_{O2} and E_{S2}

- The direction of the line L_{illum} (containing G_{I1} and G_{I2}), which comes from the direction of E_{I1}.

We would therefore expect that three parameters must be given in advance, and the other three can be computed from the geometry.

Let us suppose, for example, that the value k is given (the relative depth component of the direction of the light source), and that G_S is known (the relative orientation of the background with respect to the camera). The construction in the gradient space for computing G_O proceeds as follows (Figure 2-9):

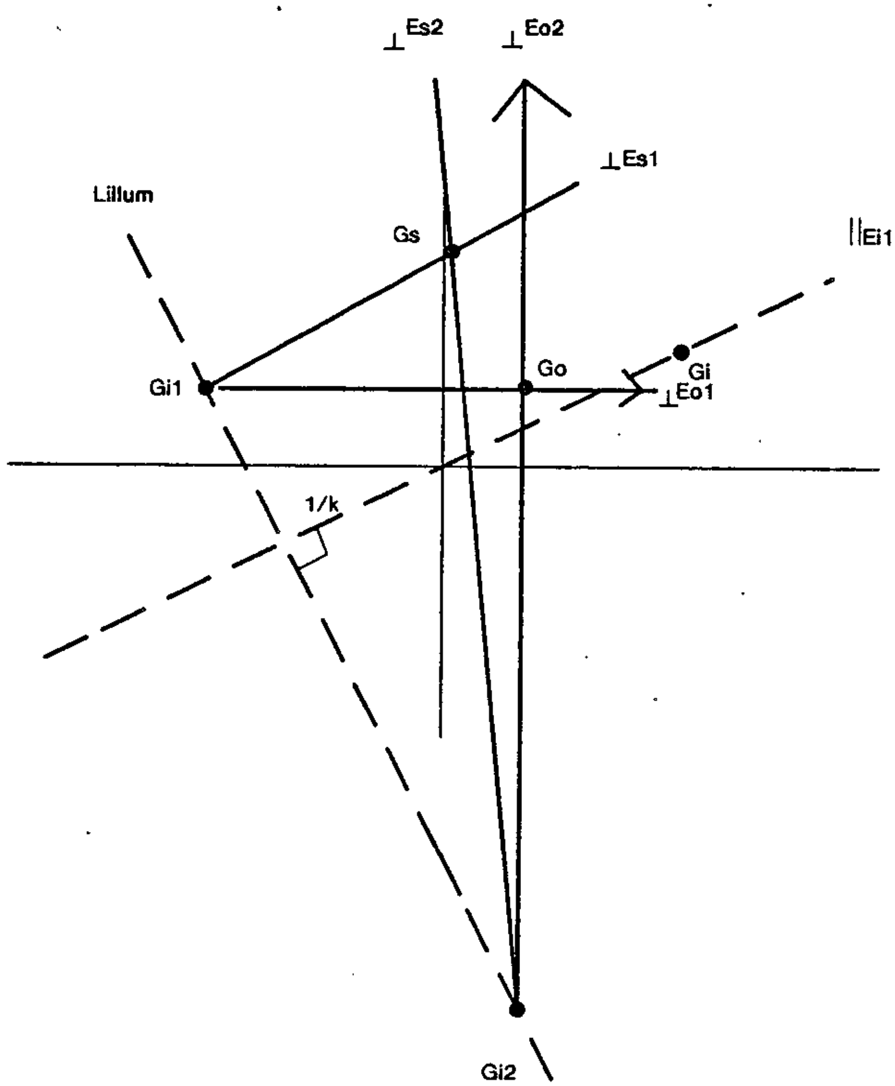


Figure 2-9: Solution to Basic Shadow Problem

1. Draw the line parallel to E_{I1} through the origin. Since k is known, G_i and L_{illum} can be found.

2. Plot G_S, which was given. Through this point, draw a line perpendicular to E_{S1}. Where it intersects L_{illum} must be G_{I1}. Through G_{I1}, draw a line perpendicular to E_{O1}. G_O must lie on this line.

3. From G_S, draw a line perpendicular to E_{S2}. Where it intersects L_{illum} will be G_{I2}. From there, draw a line perpendicular to E_{O2}. Since G_O must lie on this line, the intersection of this line with the final line from step (2) above must be G_O.

In Appendix I, the closed form solution for the Basic Shadow Problem is presented, using a vector formulation of the problem.

2.2 Relationships Among the Parameters Supplied in Advance

In the example above, G_S and k were needed before the construction could take place. In practice, a program for a specific application may not be able to compute these particular parameters.

It is possible to begin the construction with any three of the six pieces of information specified in advance, as long as none are redundant with each other, and none are redundant with the direction of E_{I1}.

It is possible, or perhaps likely, that a given line drawing will include the edge E_{OS} between S_O and S_S, as in figure 2-10. An interesting question arises as to whether this provides some additional constraint, which might perhaps relax the requirement that three pieces of information be provided in advance.

The edge E_{OS} turns out to be redundant with E_{O2} and E_{S2}, in the sense that given the latter, the former can be constructed, and vice versa. Suppose we are given E_{O2} and E_{S2}. These represent the intersections (in the *scene*) of planes S_O and S_{I2}, and S_S and S_{I2}, respectively. Now, either these two lines intersect or they do not. Suppose they intersect in a point. Call it V_{OS2}, since it is contained in surfaces S_O, S_S, and S_{I2}. This point is contained in both S_O and S_S, as is point V_{OS1} which is given in the line drawing. Therefore, the line E_{OS} must pass through these points. On the line drawing, find the intersection of E_{O2} and E_{S2}. Draw the line joining this point to V_{OS1}: this is E_{OS} (Figure 2-11).

Now, suppose that the two lines E_{O2} and E_{S2} do not intersect anywhere. Then there is no point V_{OS2} contained in all three surfaces S_O, S_S, and S_{I2}. So, E_{OS} cannot intersect either E_{O2} or E_{S2}. Since it is coplanar with these (on surfaces S_O and S_S, respectively), it must be parallel to both. Edge E_{OS} can therefore be drawn through V_{OS1}, parallel to E_{O2} (and E_{S2}).

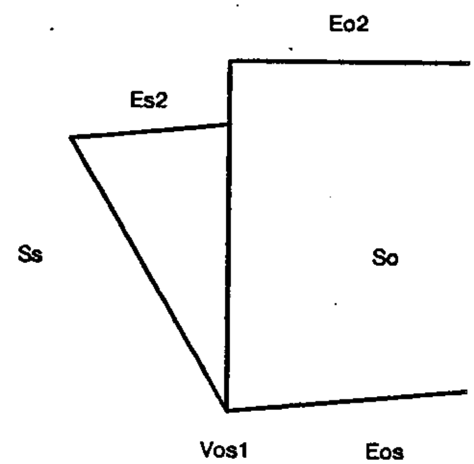


Figure 2-10: Basic Shadow Problem -- Edge E_{OS} Provided

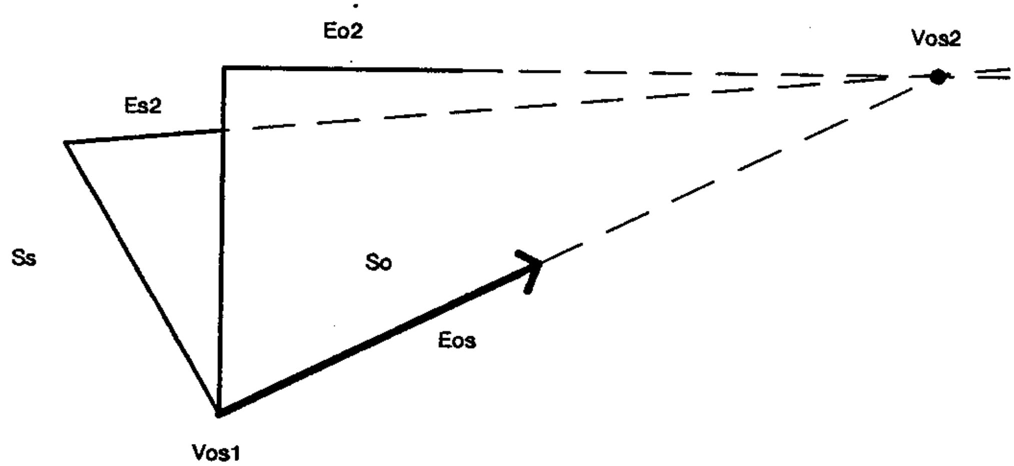


Figure 2-11: Redundancy of E_{OS} With E_{O2} and E_{S2}

By this reasoning, E_{OS} can be constructed from E_{O2} and E_{S2}. Similarly, if E_{OS} is given, either of E_{O2} and E_{S2} can be calculated from the other, to provide the geometric constraint described above for the solution of the Basic Shadow Problem. Of course, the solution can also proceed directly using the label – on E_{OS}, with identical results.

The solution of this problem should be compared with the solution to the problem if there are no shadows -- if just S_O is given, joined to S_S along edge E_{OS}. Here, there are four parameters (G_O and G_S) to compute, and one constraint from the image (E_{OS}), so three pieces of information are still needed in advance. With shadows, the same number of *a priori* parameters are needed, but one of them can be a description of the light source position instead of a description of a surface orientation. The significance of shadows is that they allow information about the light source to be used to solve the problem as a substitute for information about the surface orientations themselves.

2.3 Occurrence of the Basic Shadow Problem

It has not been assumed in this discussion that surfaces S_O and S_S must touch. In practice, the Basic Shadow Problem arises any time there are two surfaces which provide two shadow edge pairs and an enclosed illumination vector. Any additional shadow edge pairs on these two surfaces will be redundant, as will any visible edges along which these two surfaces intersect directly.

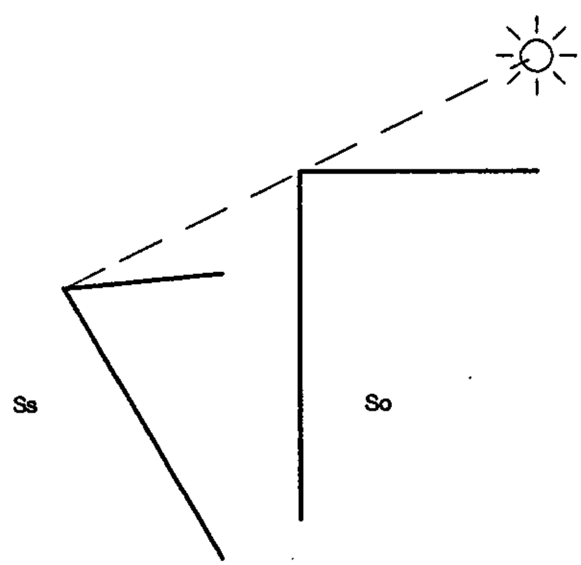


Figure 2-12: Occurence of the Basic Shadow Problem

3. Variations in Lighting

When the light source is not infinitely far away and behind and above the camera, the shadow geometry is slightly different.

3.1 Light Source In Front of the Camera

When the light source is in front of the camera (i.e. in the scene, where it might even appear in the image) and infinitely far away, the Basic Shadow Problem takes the form shown in figure 3-1. In this case, the first illumination surface S_{I1} joins edges E_{O1} and E_{S1}, giving both of these edges − labels. Illumination surface S_{I2} joins E_{O2} and E_{S2}. At E_{S2}, the label is clearly −. To label E_{O2}, it is necessary to extend S_{I2} above this edge, and apply the label to S_O and the upper half-plane of S_{I2}. The label will then be +.

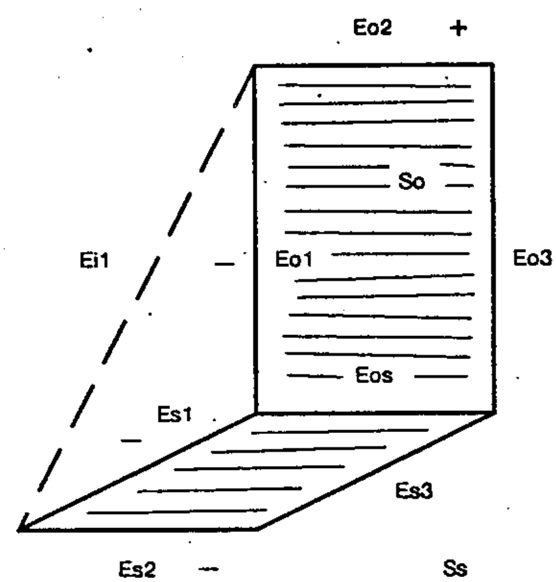


Figure 3-1: Geometry With Light Source In Front of Camera, Infinitely Far Away

The vector pointing toward the light source does not intersect the plane $z = 1$, but the vector pointing *away* from the light source (toward the camera) does. This has the effect of placing the point G_I in the gradient space on a line parallel to edge E_{I1} passing through the origin as before, but on the half-line towards surface S_S instead of towards surface S_O. This is related to the fact that the gradient space can only represent half of all possible surface orientations. The Gaussian Sphere [8] might be used to overcome this problem, although it is difficult to represent in a computer.

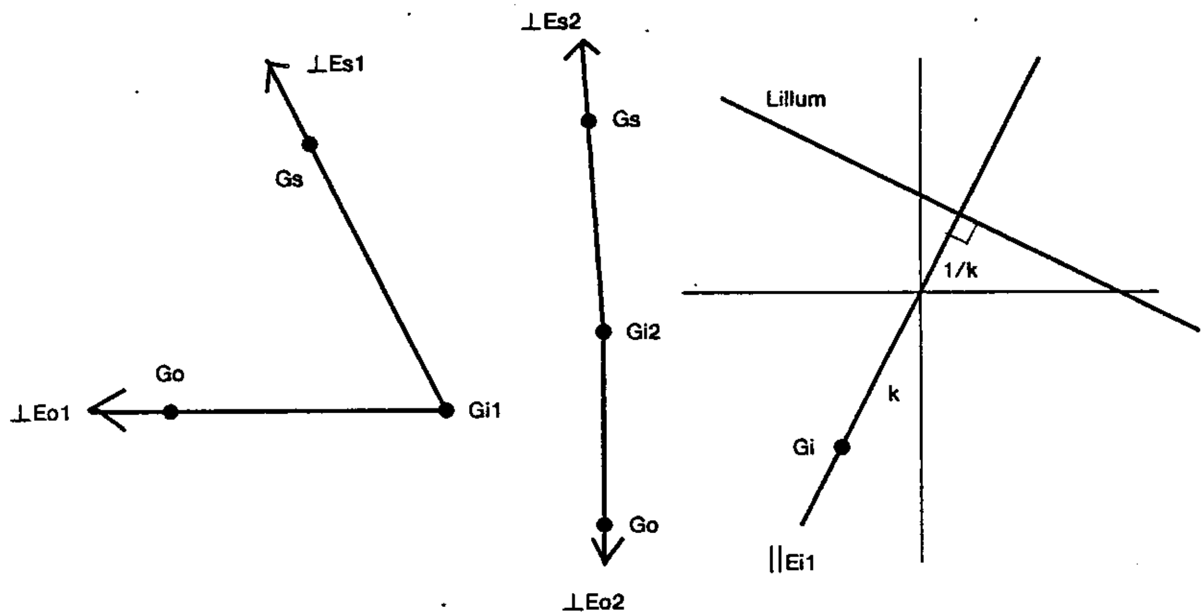


Figure 3-2: Gradient Space Constraints With Light Source In Front of Camera

All of the above gradient space constraints are shown in figure 3-2. The solution technique and parameterization are exactly the same as previously presented for the Basic Shadow Problem. The closed-form solution is that of Appendix I, with the convention that $k<0$ since S_O is self-shadowed (as explained in the Appendix).

The redundancy of edge E_{OS} is also the same: if E_{O2} and E_{S2} are parallel, then E_{OS} is parallel to them; if they intersect at a point, then E_{OS} intersects them both at that point. In this situation, if edges E_{O3} and E_{S3} are present, they are redundant with edges E_{O2} and E_{S2}. This can be easily seen, since edge E_{OS} can be calculated from the intersection of E_{O1} and E_{S1} and the intersection of E_{O3} and E_{S3}; since edge E_{OS} is known to be redundant with E_{O2} and E_{S2}, so must be E_{O3} and E_{S3}.

3.2 Light Source Behind and Below Camera

If the light source is behind the camera but below it, and infinitely far away, then the geometry is as shown in figure 3-3. In this case, the only difference from the Basic Shadow Problem is that edge E_{O2} receives the label $+$ instead of $-$; the labels of edges E_{O1}, E_{S1}, E_{S2}, and E_{OS} (if present) will be the same as previously described.

While the solution technique is the same as before, it should be noted that the point G_I, pointing

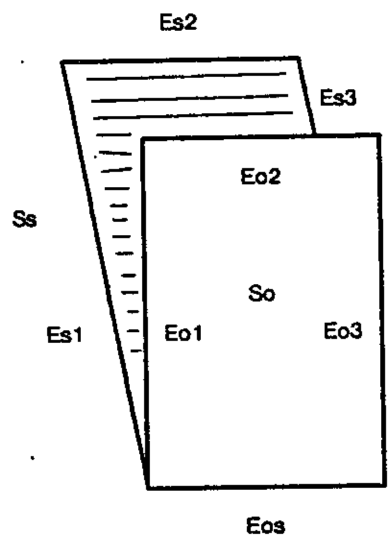


Figure 3-3: Light Source Behind and Below Camera, Infinitely Far Away.

towards the light source, will be in the lower half-plane of the gradient space instead of the upper half-plane.

In this situation, edge E_{OS} is still redundant with the pair of edges E_{O2} and E_{S2}; the pair of edges E_{O3} and E_{S3} is interchangeable with either of these.

3.3 Light Source Not Infinitely Distant

If the light source is a point not infinitely far away, then all illumination vectors will converge at the light source instead of being parallel (Figure 3-4).

Only two of the preceding arguments need to be changed in this case. The first difference is that the value k is dependent upon the particular illumination vector used, and each illumination vector will have its own value of k and its own line of illumination surface gradients L_{illum}.

The second change is that edges E_{O3} and E_{S3} are no longer interchangeable with E_{OS} or with E_{O2} and E_{S2}. The new information is actually provided not by the angle between the edges E_{O3} and E_{S3}, but by the new illumination vector E_{I2} seen between vertices V_{O23} and V_{S23}. This is shown in figure 3-4 for one case (light source below and behind camera); similar line labels and reasoning hold for the other cases presented previously.

In this arrangement, the exact position of the light source can be calculated. The lines E_{I1} and E_{I2}

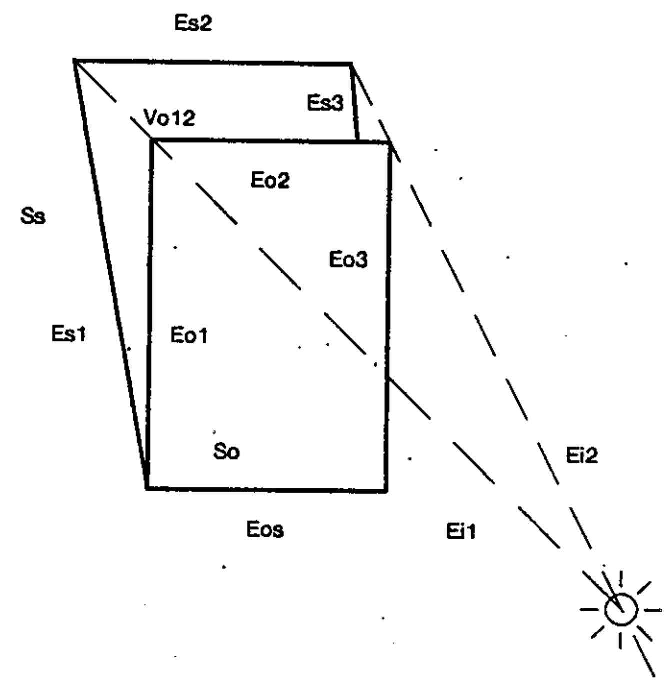


Figure 3-4: Point Light Source at Finite Distance

must intersect (in the scene); the light source is located at the point of intersection. Under orthography, as we are assuming here, the *x* and *y* coordinates of the light source will be the same as the *x-y* coordinates of the intersection of the lines in the image. So, these coordinates can easily be found. The relative *z* coordinate is then found using the *k* value for either of these vectors (E_{I1} or E_{I2}), using the definition of *k* presented above in equation (2.1): if (Δx, Δy, Δz) is an illumination vector from an object vertex to the light source (such as E_{I1} or E_{I2}), then Δx and Δy can be measured in the image, and

$$\Delta z = \mathrm{sqrt}(\Delta x^2 + \Delta y^2) / k$$

This shadow problem has seven parameters:

- (2 parameters each) Gradients G_O and G_S of surfaces S_O and S_S
- (3 parameters) Coordinates of light source position

Six of these (all except the relative *z* coordinate of the light source) can be calculated by exactly the same method used in the Basic Shadow Problem. To calculate the *z* coordinate of the light source, one additional piece of information must be utilized from the line drawing: the line E_{I2}. Since the number of *a priori* pieces of data needed does not change when the light source is at a finite distance, the remainder of this paper will omit further discussion of the extra parameter needed in this case. It will be indicated when the extra image constraint is available.

It can be determined from the line drawing whether the light source is in fact infinitely far away: if two illumination vectors (such as E_{I1} and E_{I2}) intersect, then the light source is at a finite distance, and all illumination vectors in the image must intersect at the same point. If any two illumination vectors are parallel, then all illumination vectors are parallel and the light source is infinitely far away. These observations can be used to arrive at constraints between various simple shadow problems that arise in different parts of the same image, involving different objects and surfaces.

3.4 Line Labels and Light Source Position

We are now in a position to describe how to compute the line labels to be assigned to the various edges of S_O and S_S, relating object surfaces to illumination surfaces. Each edge of S_O corresponds to a shadow edge on S_S. The line labels depend upon the relative position of the edge of S_O and the light source, and on whether S_O is illuminated (facing towards the light source) or self-shadowed (facing away from the light source).

In the discussion of cases below, note that each edge defines a line which cuts the image plane into two half-planes. Only one of these half-planes is occupied by the surface containing the edge. Similarly, only one half-plane is occupied by the light source; if the light source is infinitely far away, it can be classified as being in whichever half-plane the illumination vector I is pointing towards (as in Figure 2-7).

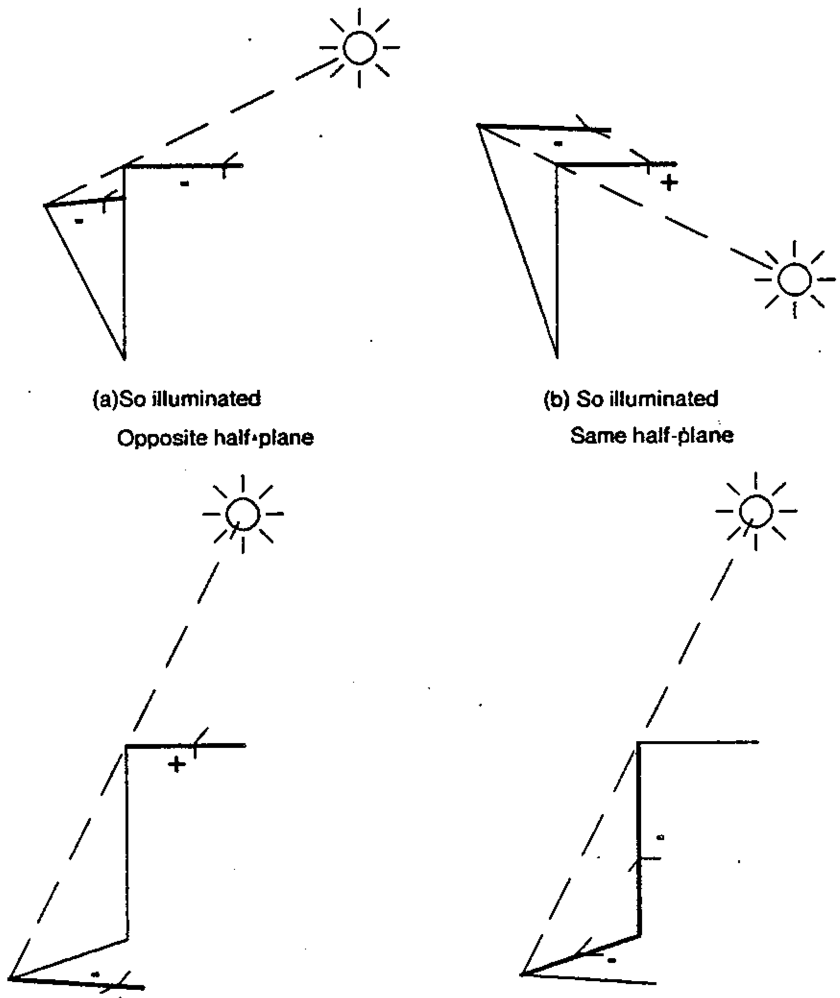
- *Case I: S_O illuminated; surface and light source in opposite half-planes.* In this case, the occluding edge and shadow edge both receive the label – (Figure 3-5(a)). This case corresponds to edges E_{O2} and E_{S2} in the Basic Shadow Problem (Figure 2-5).

- *Case II: S_O illuminated; same half-plane.* The occluding edge receives the label +; the shadow edge is labeled – (Figure 3-5(b)). This corresponds to edges E_{O1} and E_{S1} in the Basic Shadow Problem (Figure 2-3).

- *Case III: S_O self-shadowed; opposite half-planes.* The occluding edge is labeled +, referring to the *upper* half-plane of the illumination; the shadow edge is labeled –, referring to the *lower* half-plane of surface S_S. The reference marks in figure 3-5(c) indicate the half-planes involved.

- *Case IV: S_O self-shadowed; same half-plane.* Both edges receive the label – as shown in figure 3-5(d).

It is important to keep in mind that classical line-labeling methods such as that of Waltz [18] apply labels that refer to the *real* (object) surfaces which are bounded by a given edge. The line labels derived in this section apply to the relationships between one real surface and one *hypothesized* (illumination) surface along an edge.



(a)So illuminated
Opposite half-plane

(b) So illuminated
Same half-plane

(c) So self-shadowed
Opposite half-plane

(d) So self-shadowed
Same half-plane

Figure 3-5: Line Labels and Light Source Positions

3.5 Changing the Number of Light Sources

It is possible that several light sources will be present, as in figure 3-6. In this case, each light source produces two parameters in the problem (the direction of illumination), and adds two image constraints (an illumination vector and one non-redundant shadow edge pair). The number of *a priori* parameters needed will be the same, regardless of how many light sources are present.

However, for each light source, one of the *a priori* parameters may be the value k for that light source, based on knowledge of the three-dimensional direction of illumination. In general, if n light

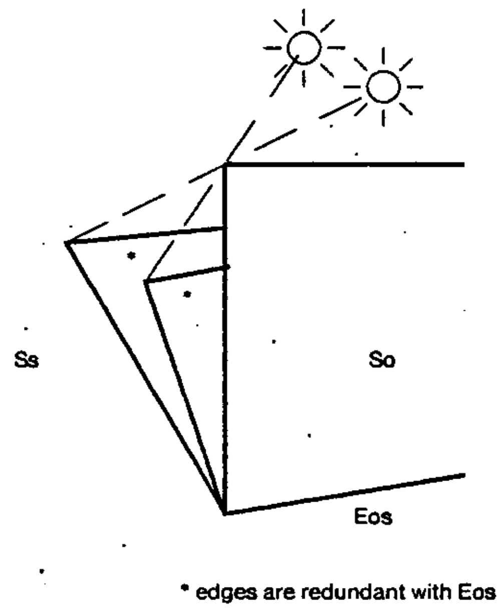


Figure 3-6: Basic Shadow Problem With Multiple Light Sources

sources are present and the value of k is known for each, the problem has $2n+4$ parameters, the image provides $3n+1$ constraints, and $3-n$ parameters are needed in advance. Thus, shadows allow you to use *a priori* knowledge about light source positions instead of *a priori* knowledge about surface orientations when computing the gradients of the visible surfaces.

In figure 3-7, there are no light sources or shadows. There are 4 parameters to compute (the gradients of the two surfaces). An image constraint will be provided in this case only if the two surfaces S_O and S_S touch along edge E_{OS}; if they do not, then an extra *a priori* parameter will be needed (i.e. 4 instead of 3).

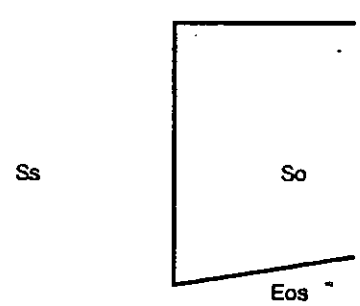


Figure 3-7: Two Surfaces With No Light Source

4. Shadows Falling On Polyhedra

When the shadow of S_O falls on several planes, it is possible to determine the surface orientations of all of them.

4.1 Shadows Falling On Polyhedra With No Shadow Edge Discontinuities

The shadow of S_O may fall on two surfaces, S_S and S_T (Figure 4-1). In this case, the first illumination surface S_{I1} contains edges E_{O1}, E_{S1}, and E_{T1}. Illumination surface S_{I2} contains edges E_{O2} and E_{S2}. Edge E_{I1} is an illumination vector, joining vertices V_{O12} and V_{S12}.

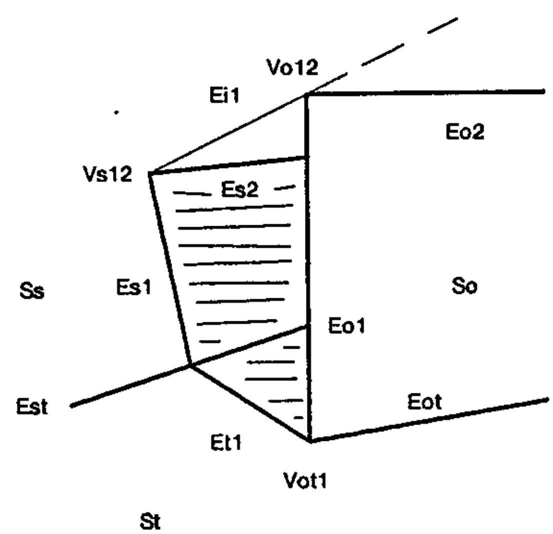


Figure 4-1: Shadow Falling On Two Surfaces

In this figure, a Basic Shadow Problem can be solved using surfaces S_O and S_S. The gradient space constraints are shown in figure 4-2. Parts (a) and (b) of this figure show constraints that are identical to those of the Basic Shadow Problem. In part (c), edge E_{T1} has provided a constraint on G_T in addition to the relation between G_O, G_S, and G_{I1}. Part (d) shows another constraint between G_S and G_T, from edge E_{ST}. Taken together, these two new constraints ((c) and (d)) can be used to compute G_T after the Basic Shadow Problem has been solved involving G_O and G_S. All of these constraints are derived from the line labels assigned to the figure as previously described in section 3.4. The edge E_{ST} is labeled $-$ if the shadow edge E_{S1} bends toward E_{O1} from E_{t1}, and $+$ if it bends away from E_{O1}.

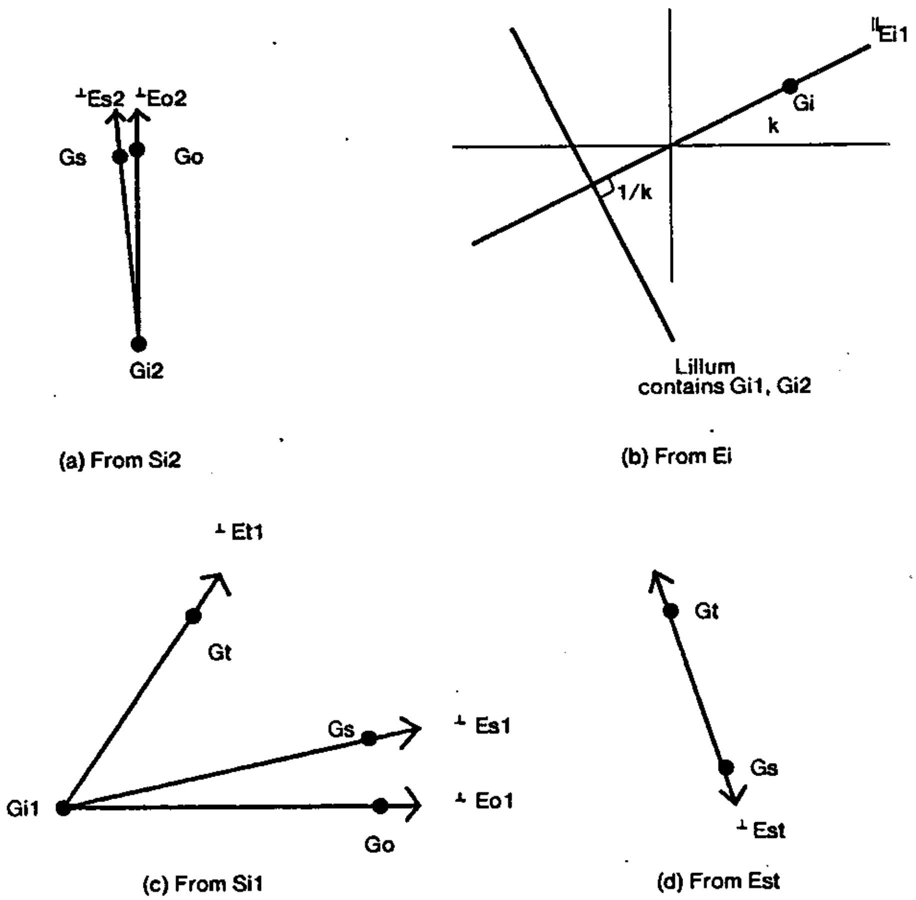


Figure 4-2: Gradient Space Constraints From Two Shaded Surfaces

In this problem, there are two additional parameters to compute (G_T) compared with the Basic Shadow Problem; there are also two additional pieces of information from the image (edges E_{ST} and E_{T1}). The relationships are those of Appendix I, with these additional relations involving G_T:

$$E_{ST} \in S_T, S_S \qquad -\Delta z_{ST} = G_T \cdot E_{ST} = G_S \cdot E_{ST}$$
$$E_{T1} \in S_T, S_{I1} \qquad -\Delta z_{T1} = G_T \cdot E_{T1} = G_{I1} \cdot E_{T1}$$

$$\begin{bmatrix} -\Delta z_{ST} \\ -\Delta z_{T1} \end{bmatrix} = \begin{bmatrix} E_{ST}^{\ T} \\ E_{T1}^{\ T} \end{bmatrix} G_T$$

$$G_T = \begin{bmatrix} E_{ST}^{\ T} \\ E_{T1}^{\ T} \end{bmatrix}^{-1} \begin{bmatrix} G_S \cdot E_{ST} \\ G_{I1} \cdot E_{T1} \end{bmatrix}$$

This problem, like the basic problem, requires that three pieces of information be supplied in advance.

This solution technique can be generalized to cases such as figure 4-3, in which there are several

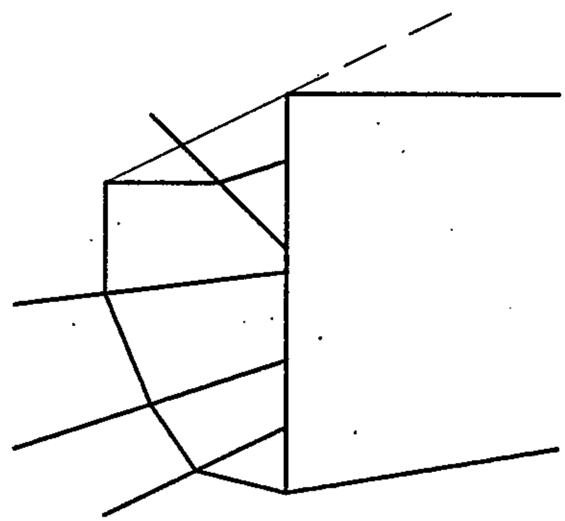


Figure 4-3: Shadow Falling On Many Surfaces

shaded surfaces. If there are n shaded surfaces which intersect the shadow edge with no discontinuities in the shadow edge, the problem will have a total of $2n+4$ parameters: $2n$ for the gradients of the shaded surfaces, 2 for G_O, and 2 for G_I. The image will supply $2n+1$ constraints; three parameters must be given in advance.

4.2 Breaks In the Shadow Edge

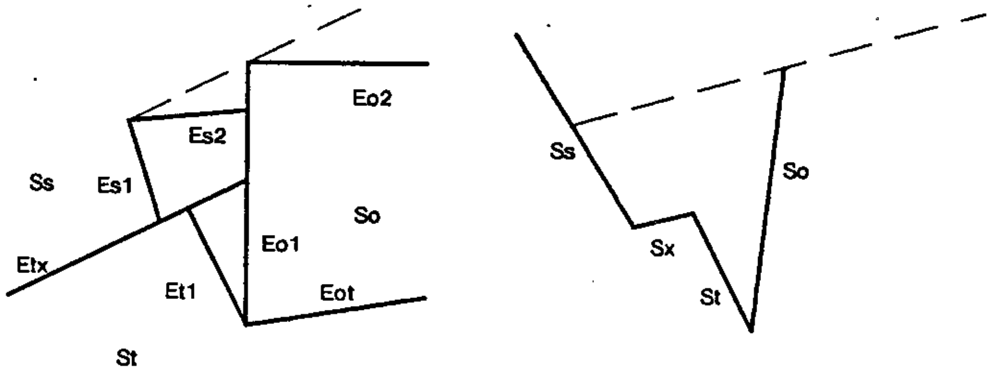
It is possible for the shadow edge to exhibit discontinuities when the shadow edge falls across occluding edges, as in figure 4-4.

The solution method is exactly as before, but this time there will be no constraint between surfaces S_S and S_T, since edge E_{ST} has been replaced by edge E_{TX} which provides no constraint between S_S and S_T. Therefore, the image provides one less constraint, and one additional non-redundant parameter must be supplied in advance in order to compute all the surface orientations. Of course, the gradient of surface S_X cannot be computed, since S_X is not visible in this image.

4.3 Constraints in the General Case

Suppose a shadow is cast by a single surface S_O, onto n shaded surfaces, and exhibiting d discontinuities.

- The problem has a total of $2n+4$ parameters to be computed:



(a) Image

(b) Side View

Figure 4-4: Shadow Edge With Discontinuites

- $2n$ for the gradients of the n shaded surfaces
- 2 for the gradient of the occluding surface S_O
- 2 for the direction G_I of the illumination

- The image provides $2n + 1 - d$ constraints:
  - $n + 1$ shadow edge segments E_{S1}, E_{T1}, E_{S2}, etc.
  - 1 from the illumination vector E_{I1}
  - n-d-1 from the intersections of the n shaded surfaces (E_{ST}, etc.)

- It is therefore required to supply $3 + d$ non-redundant parameters in advance:
  - 3 for the solution of the Basic Shadow Problem at the vertex
  - d to compensate for the d discontinuities

It is also the case that the edge E_{OT} (between the occluding surface and one shaded surface) is *non-redundant* if there are any discontinuities along the shadow edge caused by illumination surface S_{I1} (as in figure 4-4). Therefore, if this edge is present, the image provides an additional constraint, and only $2 + d$ parameters are needed in advance.

If the same figure is drawn without shadows (and including edge E_{OT}), then:

- The problem has $2n + 2$ parameters
- The image supplies n-d constraints
- $n + d + 2$ parameters must be supplied in advance

The improvement when shadows are present is that only $d + 2$ parameters are needed in advance, a difference of n parameters. This can be a very significant improvement when n is large, as when a shadow is cast on a convex polyhedron.

5. Shadows Cast By Polyhedra

When a shadow is cast by a polyhedron onto a single surface, three parameters must always be given in advance.

5.1 Polyhedra With Two Visible Surfaces

When a shadow is cast by a polyhedron as in figure 5-1, each shadow-making edge (E_{PX}, E_{OP}) must be the intersection of an illuminated surface and a self-shadowed surface of the polyhedron. In the figure, S_O is illuminated and S_P is self-shadowed. The edge E_{OP} between them is a shadow-making edge, and corresponds to shadow edge E_{S1}. Illumination surface S_{I1} contains these two edges. Similarly, it can be concluded that edge E_{PX} is a shadow-making edge, and must correspond to shadow edge E_{S2} (via illumination surface S_{I2}).

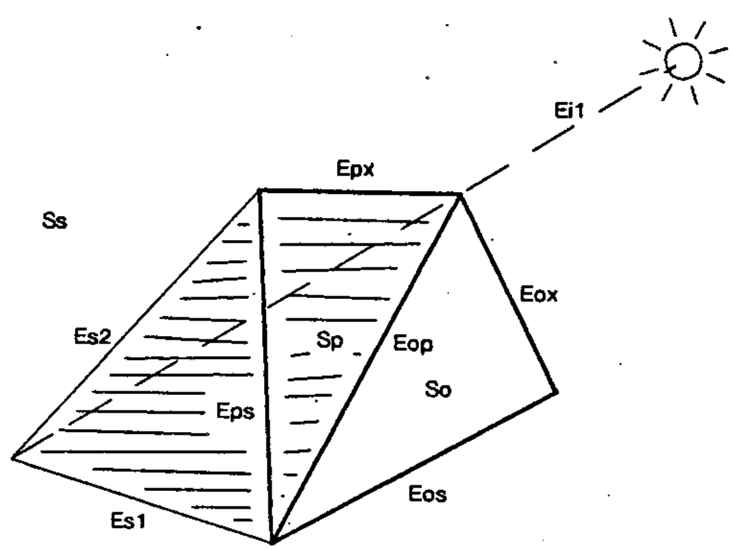


Figure 5-1: Shadow Cast By Simple Polyhedron

It can be deduced from the above observations that whatever surface intersects S_P along edge E_{PX} must be illuminated. It cannot, however, be concluded that the surface containing edge E_{PX} also contains edge E_{OX}. For this reason, no strong statements can be made about the surfaces that are not visible in the image.

In the figure, a Basic Shadow Problem exists involving surfaces S_P and S_S. The edge E_{Eps} is therefore redundant with the two shadow edge pairs (E_{OP} and E_{S1}, E_{PX} and E_{S2}). This is important, since it is typically difficult to resolve details such as edge E_{PS} within shaded portions of the image [9].

When the basic problem has been solved, the gradients of surfaces S_P and S_S will be known. The gradient of S_O can then be calculated by using the constraints provided by edges E_{OP} (with surface S_P) and E_{OS} (with surface S_S).

Little useful information is provided by edge E_{OX}, since it borders on only one visible or constructible surface (S_O). Edge E_{PX}, on the other hand, is very important, since it borders on two surfaces (visible surface S_P and the illumination surface S_{I2}).

In this problem, there are eight parameters to be computed (the gradients of surfaces S_O, S_P, and S_S, and the direction of the light source G_I). The image provides five constraints (two from the shadow edge pairs E_{OP}-E_{S1} and E_{PX}-E_{S2}, one from the illumination edge E_{I1}, and two from the edges E_{OP} and E_{OS}). Therefore, three parameters must be provided in advance in order to perform the computation.

If the figure were drawn with no shadows, there would be six parameters altogether (the gradients of the three surfaces), and three constraints in the image (from edges E_{OP}, E_{OS}, and E_{PS}). Three parameters would be required in this case, also. As in the Basic Shadow Problem itself, the shadow of a polyhedron does not provide additional constraints; it merely allows you to substitute information about the light source for *a priori* information about the surface orientations themselves, and allows you to utilize easy-to-find shadow edges instead of hard-to-find details within shaded areas of the image.

The above method of solution also applies when the light source is in a different position as in figure 5-2, which illustrates two illuminated surfaces of a polyhedron.

5.2 Adding a Self-Shadowed Surface

Suppose we add an additional self-shadowed surface to figure 5-1, as in figure 5-3. In this figure, both S_A and S_P are self-shadowed. We will suppose that the new surface S_A adjoins a shadow-making edge E_{AO}. (If the new surface S_A does not adjoin a shadow-making edge, it will be buried in the middle of the shaded area and will have no effect on the shape of the shadow.)

Two new parameters are present in the system: the gradient G_A of the new surface S_A. The image provides two new constraints that can be used to solve for these two parameters: the shadow edge pair E_{AX}-E_{S3}, and the edge E_{AO} between surfaces S_A and S_O. So, three parameters are still required in advance to solve the system completely.

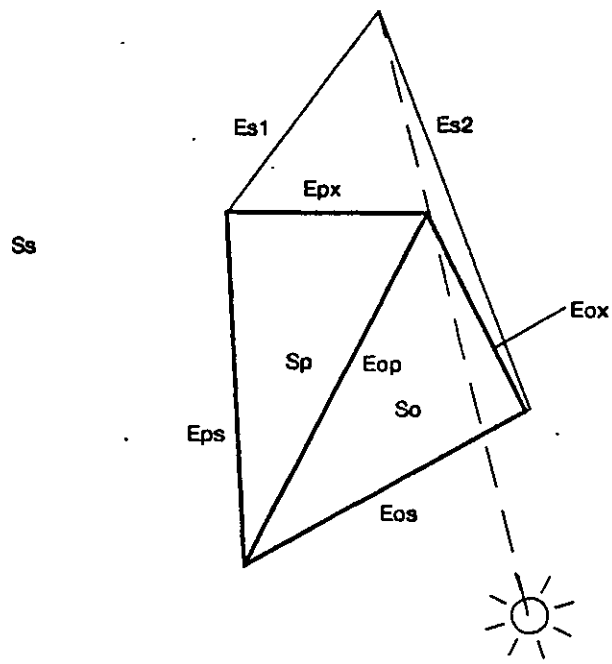


Figure 5-2: Light Source In a Different Position

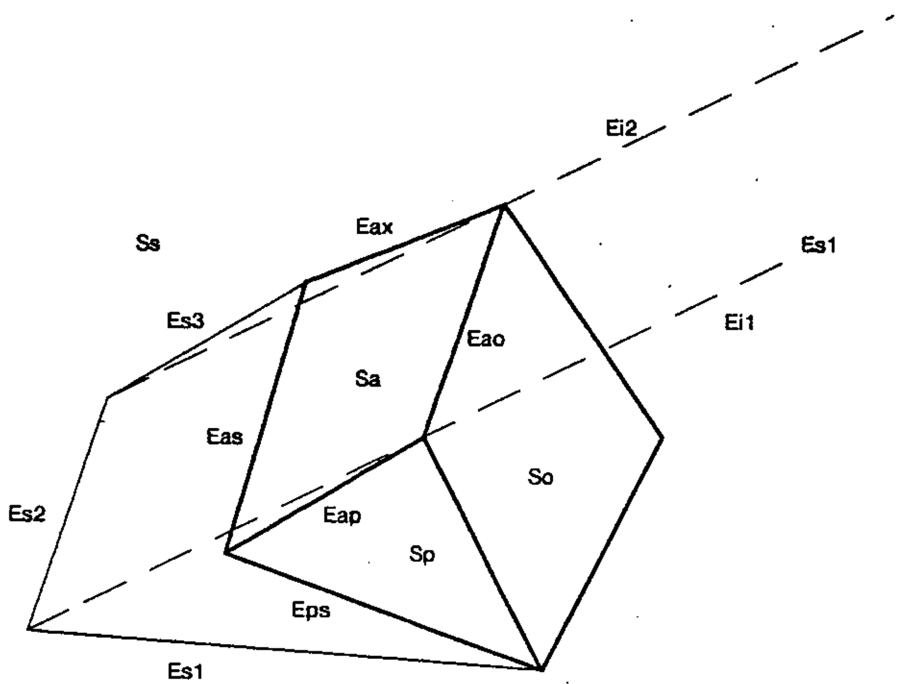


Figure 5-3: Polyhedron With Two Self-Shadowed Surfaces

If the figure is drawn without shadows, the same two parameters are required (G_A), and the two new constraints come from edges E_{AO} and E_{AS}.

The edge E_{AS} is redundant with the shadow edge pair E_{AO}-E_{S2} when shadows are present. One of the two edges E_{AP} and E_{PS} is needed, along with E_{OP}, to determine the gradient of surface S_P. Thus, two of the edges E_{SP}, E_{AS}, and E_{PS} are redundant, and only one is needed. Since these edges all lie in the shadowed area of the image, they will be difficult to extract reliably [13]. Shadows reduce the need to find edges within shadowed areas of the image.

It should also be noted that in this figure, the additional illumination edge E_{I2} can be used with E_{I1} to determine the exact location of the light source. This was not possible in the simple case (figure 5-1), because only one illumination edge was present.

5.3 Adding an Illuminated Surface

When the basic figure (Figure 5-1) is modified by adding an illuminated surface instead of a self-shadowed surface, a line drawing such as figure 5-4 is the result. In this figure, surfaces S_A and S_O are illuminated, while S_P is self-shadowed. (Again, if the surface does not adjoin a shadow-making edge, there will be no effect on the shape of the shadow and the consequent inferences to be made from shadow geometry. Therefore, we will assume that the new surface S_A does adjoin a shadow-making edge E_{AP}.)

The reasoning here is analogous to the case of an additional self-shadowed surface: two new parameters are needed (G_A), and there are two new constraints with shadows (the pair E_{PX}-E_{S3} and the edge E_{AO}), and two new constraints with no shadows (edges E_{AO} and E_{AP}). In any case, three parameters will be required in advance. As in the previous case, the new illumination vector E_{I2} can be used with E_{I1} to determine the exact location of the light source.

The shadow edge pair E_{AO}-E_{S2} from figure 5-3 has been replaced by the pair E_{AP}-E_{S2} in figure 5-4.

It is possible that additional *a priori* parameters will be needed in pathological cases. Figure 5-5 depicts an object with a surface adjoining the shadow-making edge which is not visible in the image (at E_{PX}). Here, an additional *a priori* parameter will be needed to determine the gradient of surface S_R. The additional parameter is needed because edge E_{QX} provides no constraint between surfaces S_Q and S_R. This situation is analogous to the discontinuities in the shadow edge discussed previously.

Another circumstance requiring additional *a priori* parameters is shown in figure 5-6. Here, vertex V_{OPQR} is not *trihedral* -- there are four surfaces meeting at that point (S_O, S_P, S_Q, and S_R). This adds

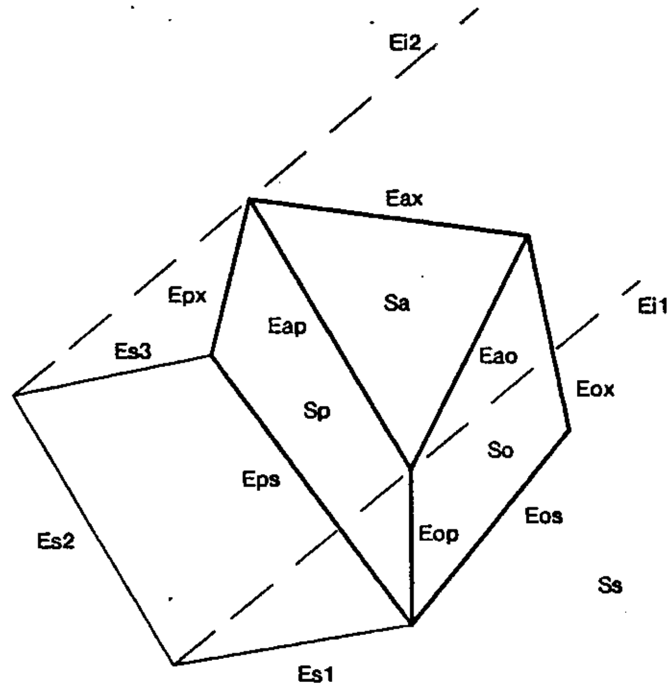


Figure 5-4: Polyhedron With Two Illuminated Surfaces

one degree of uncertainty involving the gradients of surfaces S_Q and S_R: one additional *a priori* parameter is needed to solve this problem.

5.4 The General Solution For Polyhedral Shadow Geometry

The results of the two previous extensions can be directly combined. In these arguments, it has never been assumed that the shadow edge E_{S3} and the corresponding shadow-making edge (E_{AX} or E_{PX}) meet at a vertex. Therefore, the results apply without change to line drawings with additional hidden surfaces, such as figure 5-7. In this figure, there is no strong information to be obtained from shadow edge E_{S4}.

In the combined case, a line drawing may depict i illuminated and s self-shadowed surfaces adjoining shadow-making edges, casting a shadow on one surface, with h hidden shadow-making surfaces and t non-trihedral vertices. The problem contains $2i+2s+4$ parameters (gradients of the shadow-making surfaces, G_S, and G_I). The image supplies $2i+2s\text{-}h\text{-}t+1$ parameters ($i+s$ shadow-making edges, $i+s\text{-}h\text{-}t$ nonredundant edges between two visible surfaces, and 1 illumination edge). For solution, $3+h+t$ additional parameters are therefore needed.

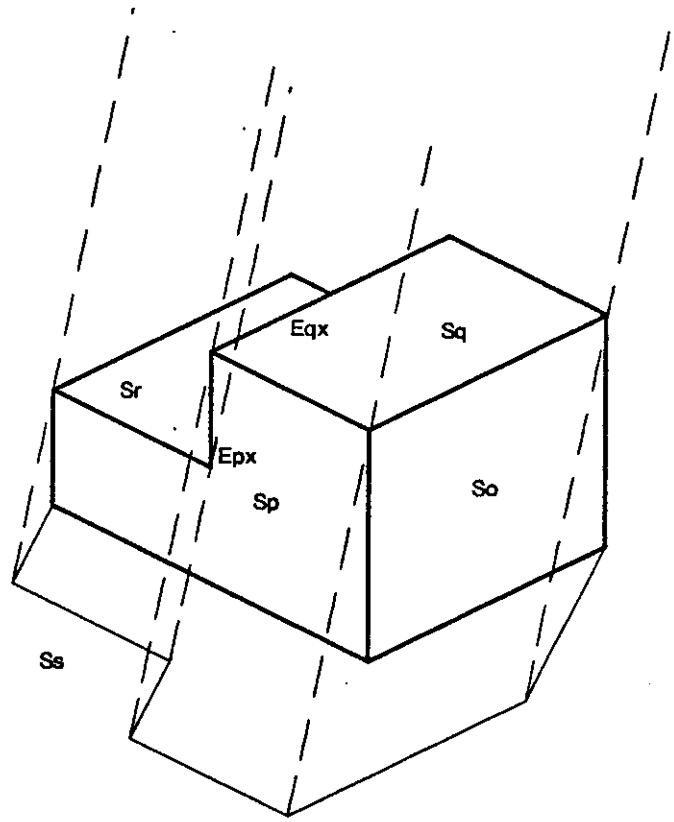


Figure 5-5: Additional Parameter Needed for Hidden Shadow-Making Surface

If no shadows are present, $3 + h + t$ parameters are still needed.

If $i>1$ or $s>1$, an additional illumination edge can be used to determine the exact position of the light source.

The above results can be composed with the results from the previous chapter for shadows cast upon polyhedra.

Suppose the image depicts i illuminated surfaces and s self-shadowed surfaces along the shadow-making edges of a polyhedron, casting a shadow whose corresponding edge intersects n surfaces of another polyhedron exhibiting d discontinuities, with h hidden shadow-making surfaces and t non-trihedral vertices.

- The problem has $2i + 2s + 2n + 2$ parameters:
    - $2i$ for the gradients of the i illuminated surfaces

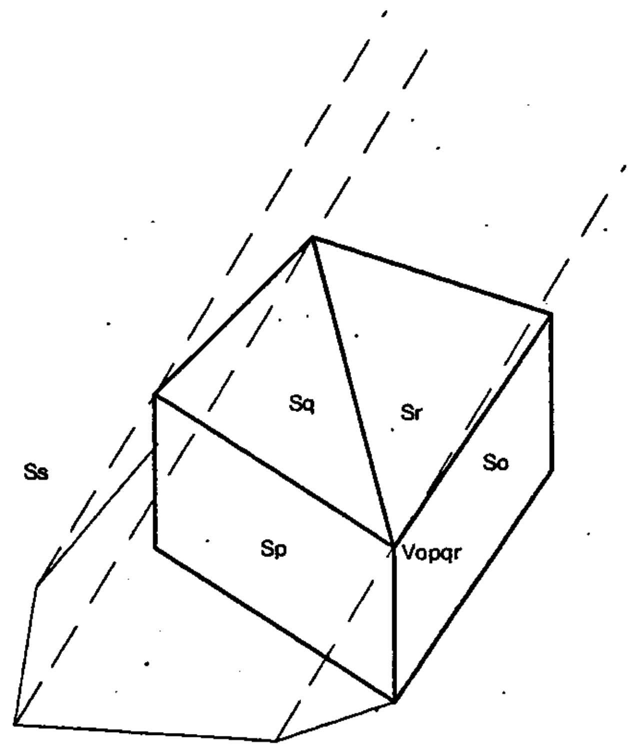


Figure 5-6: Additional Parameter Needed for Non-Trihedral Vertex

- o $2s$ for the gradients of the s self-shadowed surfaces
- o $2n$ for the gradients of the n background surfaces
- o 2 for the direction of illumination, G_I

- The image provides $2i+2s+2n\text{-}d\text{-}h\text{-}t\text{-}1$ constraints:

  - o 1 from the illumination vector
  - o 2 shadow-making/shadow edge pairs used to solve the Basic Shadow Problem at one vertex
  - o $i+s\text{-}2$ additional shadow-making edges
  - o $n\text{-}1$ additional shadow edges
  - o $i+s\text{-}h\text{-}t\text{-}1$ non-redundant edges between visible surfaces of the polyhedron casting the shadow
  - o 1 non-redundant edge between the shadow-making polyhedron and the shaded polyhedron
  - o $n\text{-}d\text{-}1$ edges at intersections of visible shaded surfaces

- Therefore, $3+d+h+t$ parameters must be provided *a priori*:

  - o 3 for the solution of the Basic Shadow Problem
  - o d to compensate for the d discontinuities in the shadow edge due to invisible shaded surfaces
  - o h to compensate for the h hidden shadow-making surfaces

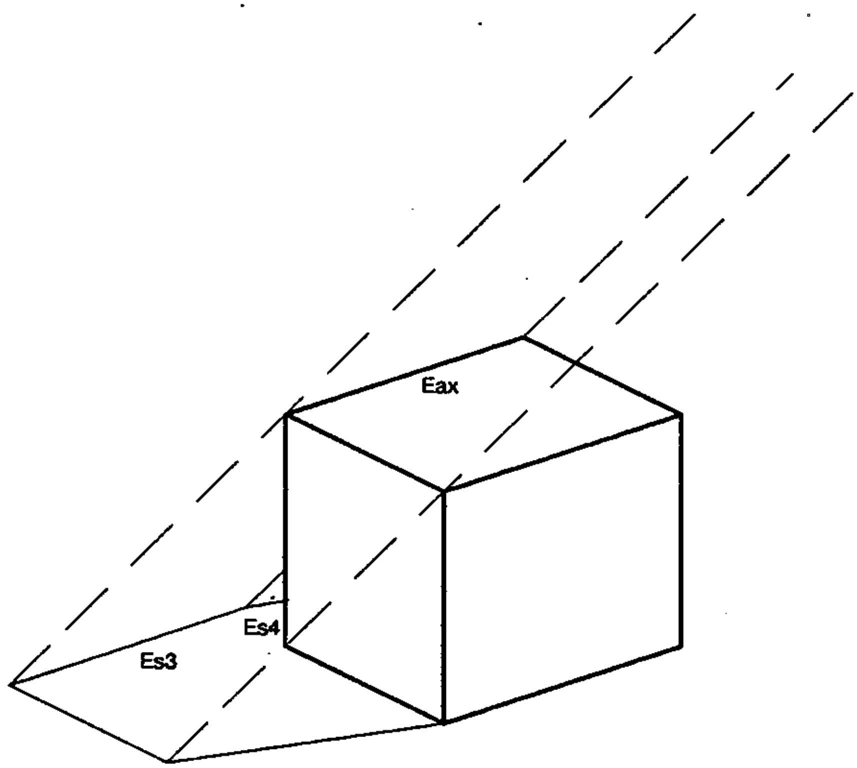


Figure 5-7: Polyhedron With Additional Invisible Surfaces

- t to compensate for the t non-trihedral vertices

Without shadows, the problem contains $2i+2s+2n$ parameters, the image supplies $2i+2s+n$-d-h-t-2 parameters, and $n+d+h+t+2$ parameters must be supplied before the computation.

If i>1 or s>1, an additional illumination vector can be used to determine the exact position of the light source.

The contribution of shadows for computing surface orientations from line drawings is thus made clear:

- Shadows provide an increasing amount of information when the shadow edge intersects many visible, differently oriented surfaces of the background.
- Shadows allow you to substitute one parameter describing the direction of illumination to replace one parameter describing a surface orientation before performing the required calculations.
- Shadows allow you to substitute (usually) highly visible shadow edges and shadow-

making edges for many of the unreliable edges within shaded portions of the image, while providing the same amount of information.

In addition, when several shadow problems appear in different portions of the same image, they share some constraints. For example, suppose several polyhedral blocks are scattered over a single surface. If the gradient of the surface and the direction of illumination are known, then three constraints are provided for *each* of the shadow problems. This will allow the exact solutions to be found for all the problems, if no shadow edge discontinuities or non-trihedral vertices are present.

6. Shadows Involving Curved Surfaces

In this chapter, the involvement of curved surfaces in shadow geometry will be explored. Whether the curvature lies in the occluding surface (object) or the shaded surface, additional information is required to determine the exact surface orientation along the shadow-making arc or the shadow edge arc.

Witkin [19] has also used shadows to determine curved surface orientation. He developed a relation between the curvature of a shadow edge in the scene and the curvature of the shadow edge in the image, then derived surface orientations, using surface texture gradients to provide the additional constraint necessary. The discussion below differs from Witkin's in that surface *orientation* rather than *curvature* (rate of change of orientation) is the basis of the theory.

For discussing curved surfaces, it is necessary to generalize the relation between line labels and surface gradients. Suppose two (possibly curved) surfaces S_A and S_B intersect along arc E_{AB} (Figure 6-1).

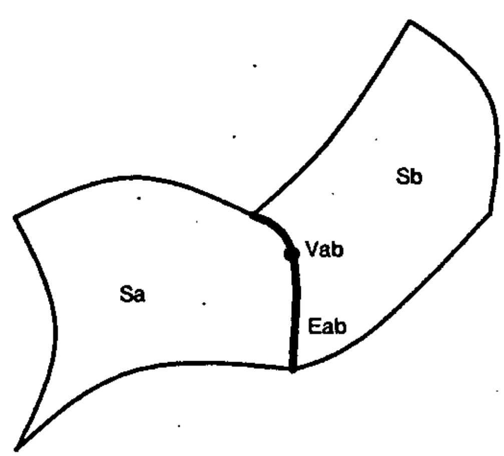


Figure 6-1: Curved Surfaces Intersecting Along an Arc

The surfaces are defined by

$$S_A:\ -z = f_A(x, y) \qquad S_B:\ -z = f_B(x, y)$$

At a point V_{AB} on E_{AB},

$$-z = f_A(x, y) = f_B(x, y)$$

Differentiating by x using the rule

$$\frac{df(x,y)}{dx} = \frac{\partial f}{\partial x} + \frac{dy}{dx}\frac{\partial f}{\partial y}$$

we have

$$-\frac{dz}{dx} = \frac{\partial f_A}{\partial x} + \frac{dy}{dx}\frac{\partial f_A}{\partial y} = \frac{\partial f_B}{\partial x} + \frac{dy}{dx}\frac{\partial f_B}{\partial y}$$

If G_A is the gradient of S_A at V_{AB}, and G_B is the gradient of S_B at V_{AB}, then

$$G_A = (p_A, q_A) = (\frac{\partial f_A}{\partial x}, \frac{\partial f_A}{\partial y}) \text{ and } G_B = (p_B, q_B) = (\frac{\partial f_B}{\partial x}, \frac{\partial f_B}{\partial y})$$

Substituting,

$$-\frac{dz}{dx} = p_A + \frac{dy}{dx}q_A = p_B + \frac{dy}{dx}q_B = G_A \cdot (1, \frac{dy}{dx}) = G_B \cdot (1, \frac{dy}{dx})$$

If $E = (\Delta x, \Delta y)$ is a vector tangent to E_{AB} at V_{AB} in the image, corresponding to the three-dimensional vector $(\Delta x, \Delta y, \Delta z)$ in the scene, then the above equation can be multiplied by Δx:

$$-\Delta x \frac{dz}{dx} = \Delta x\, G_A \cdot (1, \frac{dy}{dx}) = \Delta x\, G_B \cdot (1, \frac{dy}{dx})$$

Since

$$\Delta z = \Delta x \frac{dz}{dx} \text{ and } \Delta y = \Delta x \frac{dy}{dx},$$

we have

$$-\Delta z = G_A \cdot (\Delta x, \Delta y) = G_B \cdot (\Delta x, \Delta y) = G_A \cdot E = G_B \cdot E$$

This is the curved-surface analogue of the relation $-\Delta z = \mathbf{G} \cdot \mathbf{E}$ described earlier for planar surfaces: the planar-surface edge $\mathbf{E}$ is replaced by the tangent vector E to the arc of intersection of two curved surfaces. As a consequence, G_A and G_B lie along a line in gradient space perpendicular to the tangent to the arc of intersection in the image.

6.1 Curvature in the Shaded Surface

Suppose a flat surface is casting a shadow on a curved surface, as in figure 6-2. Here, vertex V_{S12} is the shadow of vertex V_{O12}. Surface S_{I1}, the first illumination surface, casts the shadow of edge E_{O1} on arc E_{S1} of the curved surface S_S. Surface S_{I2} similarly casts the shadow of edge E_{O2} on arc E_{S2}.

Suppose V_{SX} is an arbitrary point on the arc E_{S1}. Can we determine the gradient G_X of S_S at this point?

Arc E_{S1} is the arc of intersection between the curved surface S_S and the illumination surface S_{I1} (defined by edge E_{O1} of surface S_O). Therefore, as previously explained, gradients G_X (of S_S at V_{SX})

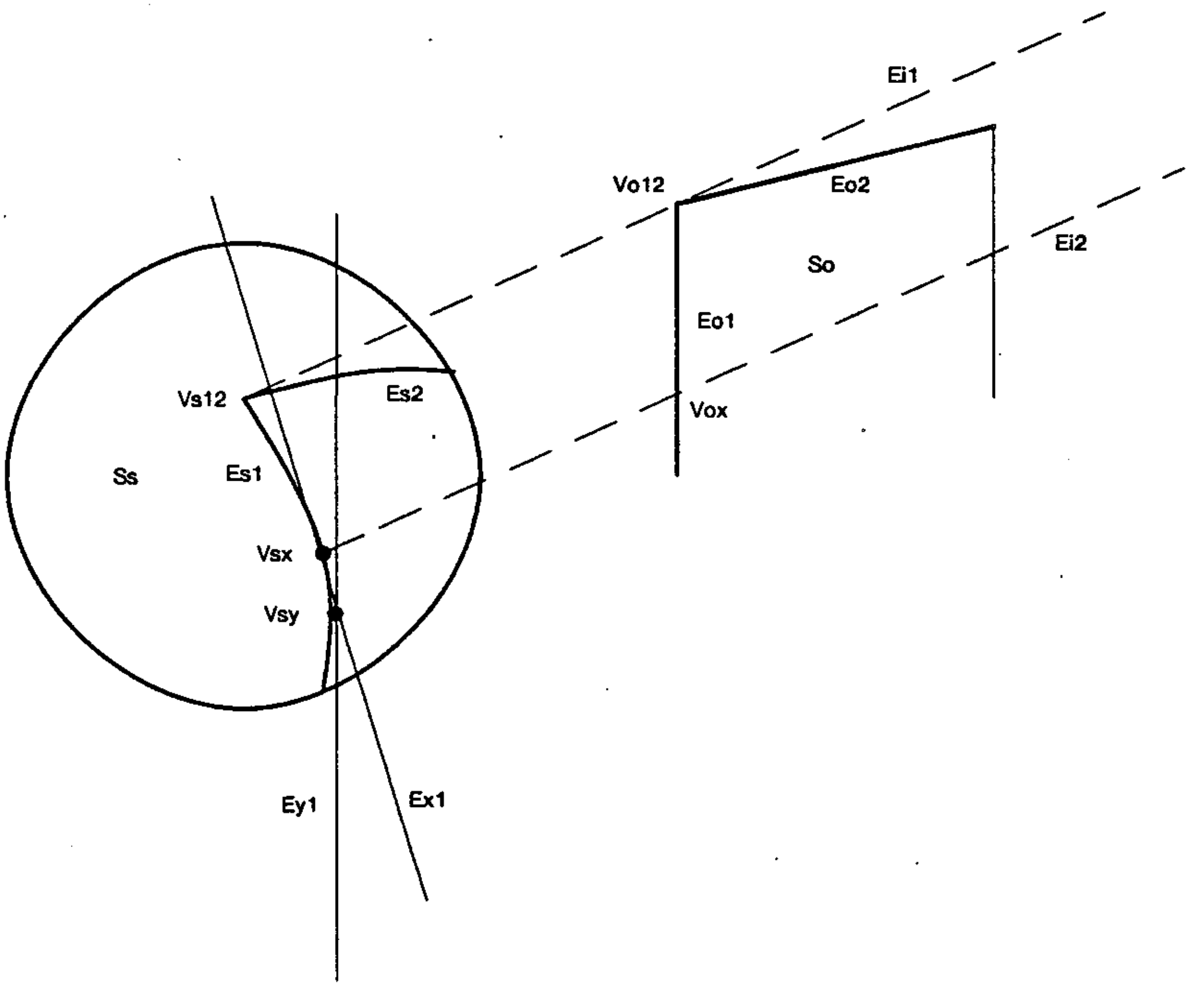


Figure 6-2: Shadow Cast On a Curved Surface

and G_{I1} (of S_{I1}) must lie along a line in gradient space perpendicular to the tangent line E_{X1} to E_{S1} at V_{SX}. This constraint is illustrated in figure 6-3.

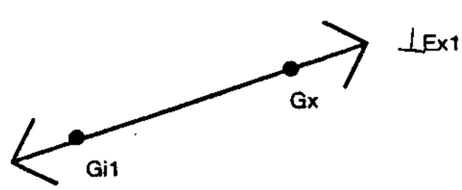


Figure 6-3: Gradient Space Constraint Between G_X and G_{I1}

This reasoning can be used to find the two tangent lines at vertex V_{S12}, and use them in a Basic

Shadow Problem with edges E_{O1} and E_{O2} of the occluding surface S_O. If S_V is the plane tangent to S_S at V_{S12}, the Basic Shadow Problem actually involves surfaces S_V and S_O. For this computation, three *a priori* parameters will be required, and the gradients G_O, G_V, G_{I1}, G_{I2}, and G_I will be computed.

It is not possible to compute the gradients G_X (and G_Y, etc.) without additional information. However, it is possible to establish a one-dimensional constraint on each such gradient. Since the gradient G_{I1} of illumination surface S_{I1} was computed as part of the Basic Shadow Problem at vertex V_{S12}, the constraints provided by the tangent lines E_{X1} and E_{Y1} cause gradient space constraints as shown in figure 6-4. Similar reasoning allows constraints on the gradients at points along arc E_{S2} to be computed, using the gradient G_{I2} of illumination surface S_{I2}.

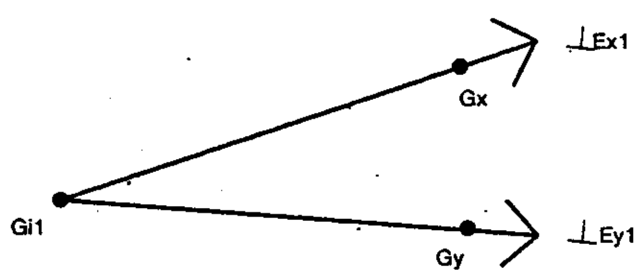


Figure 6-4: Gradient Space Constraints On Tangent Planes To S_S

For an investment of three parameters given in advance, then, the gradients of S_O and S_V can be computed, as well as a one-dimensional constraint on the gradient for each point along arcs E_{S1} and E_{S2}. Additional constraint for the gradients along these arcs might come from another source such as Horn's "shape from shading" technique [4] or *a priori* knowledge of the shape of the object bounded by surface S_S.

In this shadow problem, if another illumination vector is available (possibly from the shadow of another vertex of S_O), the exact position of the light source can then be determined.

The information available from using shadows in this problem is not redundant with information available from the same line drawing without shadows.

6.2 Shadows Cast By Curved Surfaces

When a curved object casts a shadow on a flat surface as in figure 6-5, the shadow edge E_{IS} corresponds to the shadow of the "arc of extinction" E_{IO} which divides surface S_O into an illuminated part and a self-shadowed part. There exists a curved illumination surface S_I, composed of illumination vectors, tangent to S_O along E_{IO} and intersecting the shaded surface S_S along E_{IS}. S_I is a cylinder, whose axis is parallel to the direction of illumination.

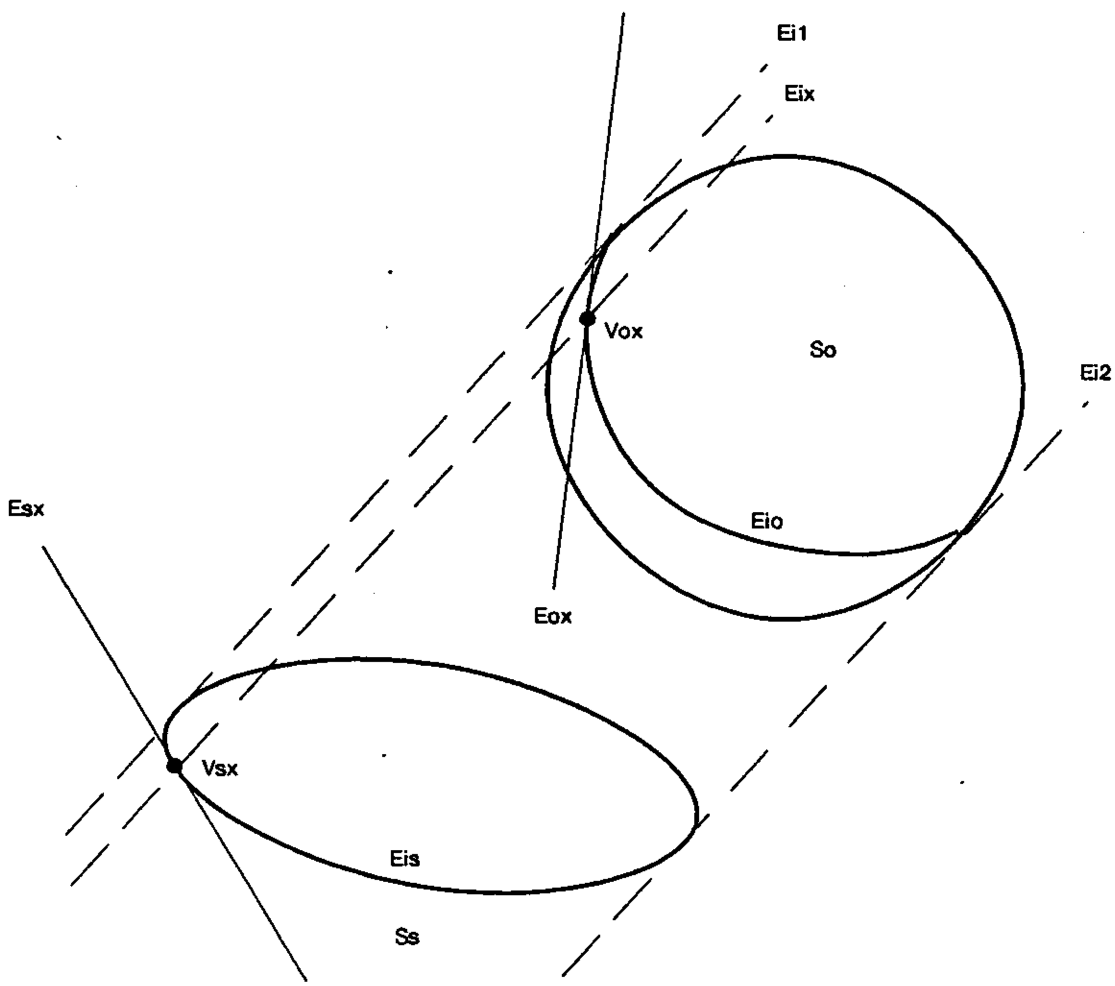


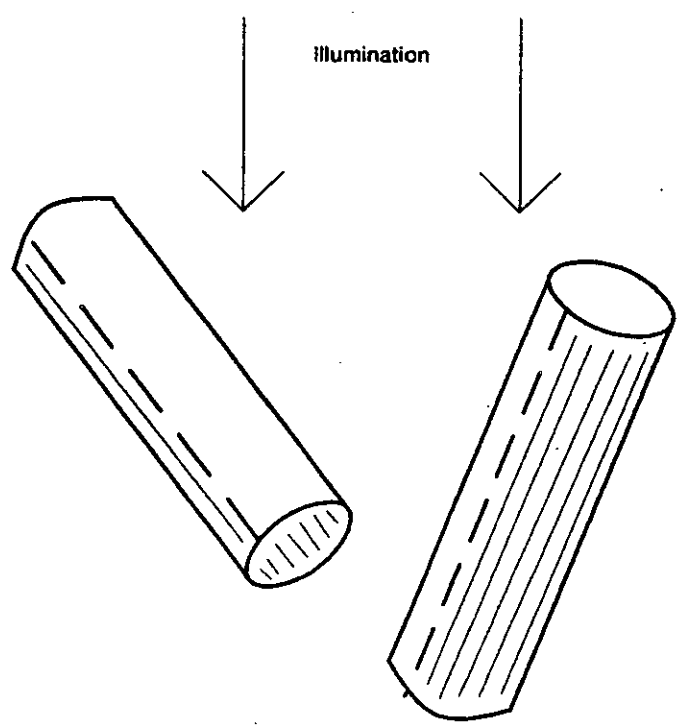
Figure 6-5: Shadow Cast By a Curved Surface

There is a special significance to the line in the image tangent to both E_{IS} and the outline of S_O: it is an illumination vector, such as E_{I1} in figure 6-5. If two such tangent lines are visible (as with E_{I1} and E_{I2} in figure 6-5) or some other feature is visible in both E_{IO} and E_{IS}, then a second illumination vector can be found. From two illumination vectors, the exact position of the light source can be computed and the shadow point V_{SX} can be determined for each point V_{OX} on arc E_{IO}.

The surface S_I is composed entirely of illumination vectors; its gradient at each point must therefore lie along the line L_{illum} in gradient space. To determine this line, the value k for the light source position must be given.

If the light source is not infinitely far away, each illumination vector such as E_{IX}, has a different value of k and determines a different line L_{illum} in gradient space. However, all the values of k can be computed from the position of the light source, given a single value of k such as that for E_{I1}. We will therefore assume, for simplicity, that the light source is infinitely far away, and that a single line L_{illum} exists.

Unfortunately, no stronger statements can be made about the gradient of S_O from examination of the arc E_{IO}. In particular, the direction of the tangent line E_{OX} bears no relationship to the gradient of S_O. This is illustrated in figure 6-6, which depicts two cylinders tangent to the same illumination plane. The arcs of extinction (dotted lines) have completely unrelated directions in the image.



Cylinders in the same plane

Figure 6-6: Arcs of Extinction are Unrelated To Surface Orientation

However, it is possible to use the shadow E_{IS} to compute the gradients of the tangent surfaces along E_{IO}. The gradient G_X of S_O at V_{OX} is the same as the gradient of S_I at V_{OX}, since S_I is tangent to S_O at that point. We have two constraints on G_X from properties of S_I:

1. S_I is an illumination surface, so G_X lies on L_{illum}.

2. The gradient (G_X) of S_I at V_{OX} is the same as the gradient of S_I at V_{SX} (the shadow of V_{OX}), since S_I is a cylinder. As previously shown, G_X and G_S (the gradient of the shaded surface S_S) must lie along a line in the gradient space which is perpendicular to E_{SX}, the line tangent to E_{IS} at V_{SX}.

The constraints on G_X are illustrated in figure 6-7.

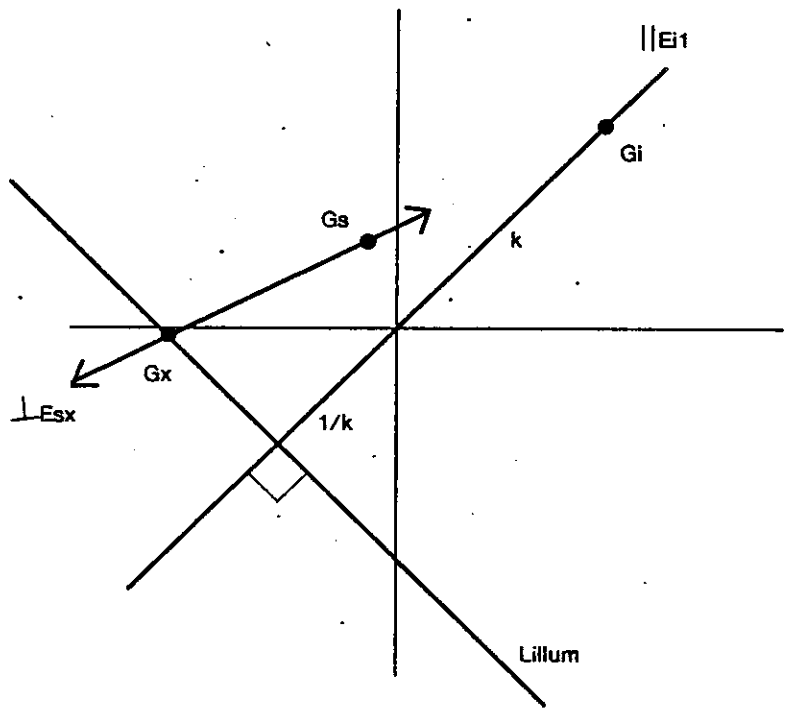


Figure 6-7: Gradient Space Constraints on G_x

So, suppose we are given three parameters -- k and the gradient G_S of surface S_S. From these, it is possible to compute the gradient G_X of the tangent plane to S_O for each point V_{OX} along the arc of extinction E_{IO}:

1. In gradient space, draw L_{illum} from E_{I1} and k.

2. In the image, find the shadow V_{SX} of V_{OX} by following an illumination vector from V_{OX} to its intersection with the shadow arc E_{IS}. Draw the tangent line E_{SX} through V_{SX}.

3. In gradient space, draw the line perpendicular to E_{SX} through G_S. This line intersects L_{illum} at G_X, as illustrated in figure 6-7.

This can also be presented in a closed form solution for G_X. Using the definition of k,

$$\Delta z_{IX} = \|E_{IX}\| / k$$

Since E_{IX} is contained in S_I,

$$-\Delta z_{IX} = G_X \cdot E_{IX}$$

Also, if E_{SX} is a vector tangent to E_{IS} at V_{SX},

$$-\Delta z_{SX} = G_X \cdot E_{SX} = G_S \cdot E_{SX}$$

Combining these,

$$\begin{bmatrix} -\Delta z_{IX} \\ -\Delta z_{SX} \end{bmatrix} = \begin{bmatrix} E_{IX}^T \\ E_{SX}^T \end{bmatrix} G_X$$

$$G_X = \begin{bmatrix} E_{IX}^T \\ E_{SX}^T \end{bmatrix}^{-1} \begin{bmatrix} -\Delta z_{IX} \\ -\Delta z_{SX} \end{bmatrix} = \begin{bmatrix} E_{IX}^T \\ E_{SX}^T \end{bmatrix}^{-1} \begin{bmatrix} -\|E_{IX}\|/k \\ G_S \cdot E_{SX} \end{bmatrix}$$

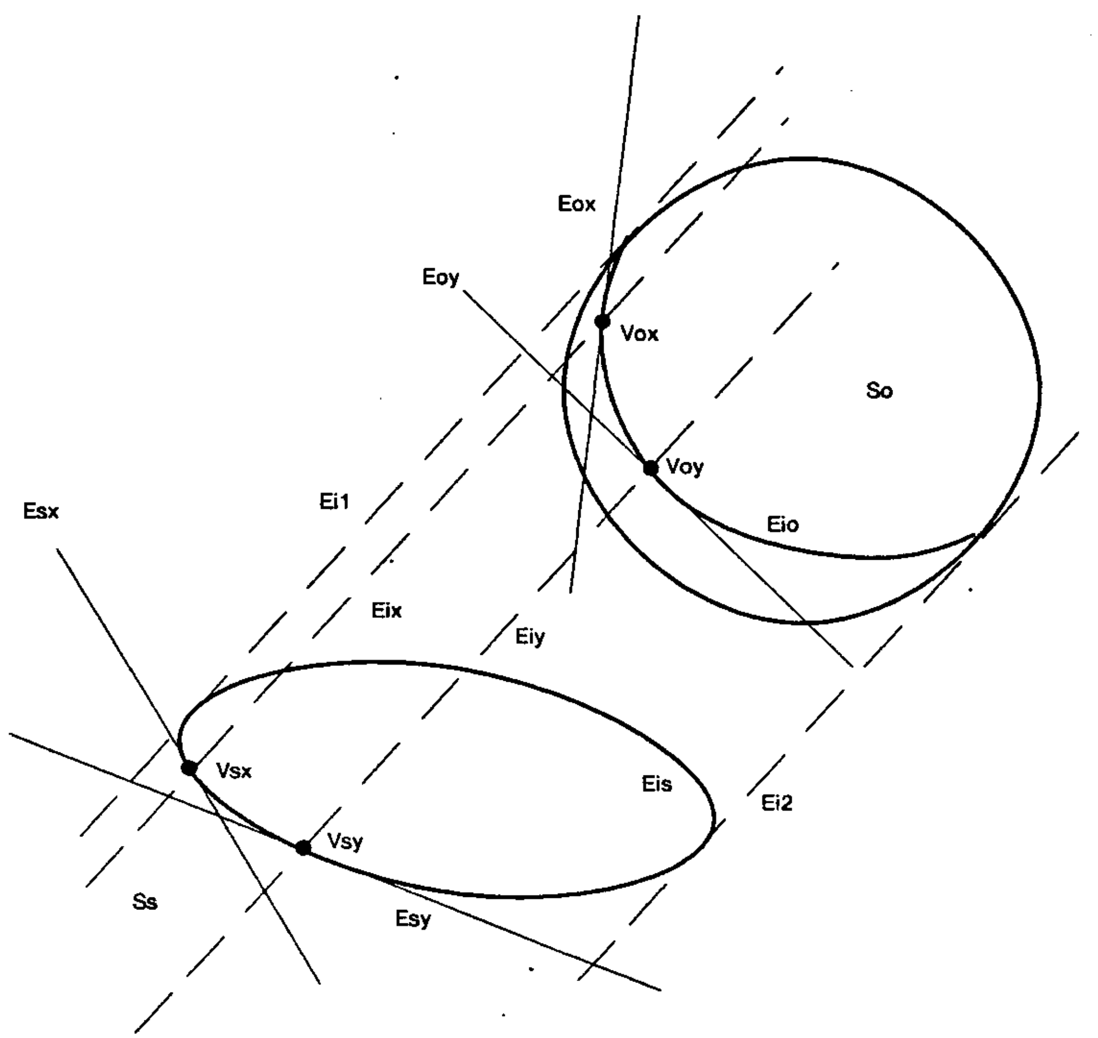


Figure 6-8: Using E_{IO} to Calculate the Gradient of S_S

It is also possible to use knowledge about the shape of the curved object S_O when G_S is not known in advance. Suppose that two vectors E_{OX} and E_{OY} tangent to the arc of extinction E_{IO} at points V_{OX} and V_{OY} are known. Let points V_{SX} and V_{SY} be the shadows of V_{OX} and V_{OY}, let E_{IX} and E_{IY} be the illumination vectors joining V_{OX} to V_{SX} and V_{OY} to V_{SY}, and let E_{SX} and E_{SY} be vectors tangent to the shadow edge E_{IS} at V_{SX} and V_{SY} (Figure 6-8).

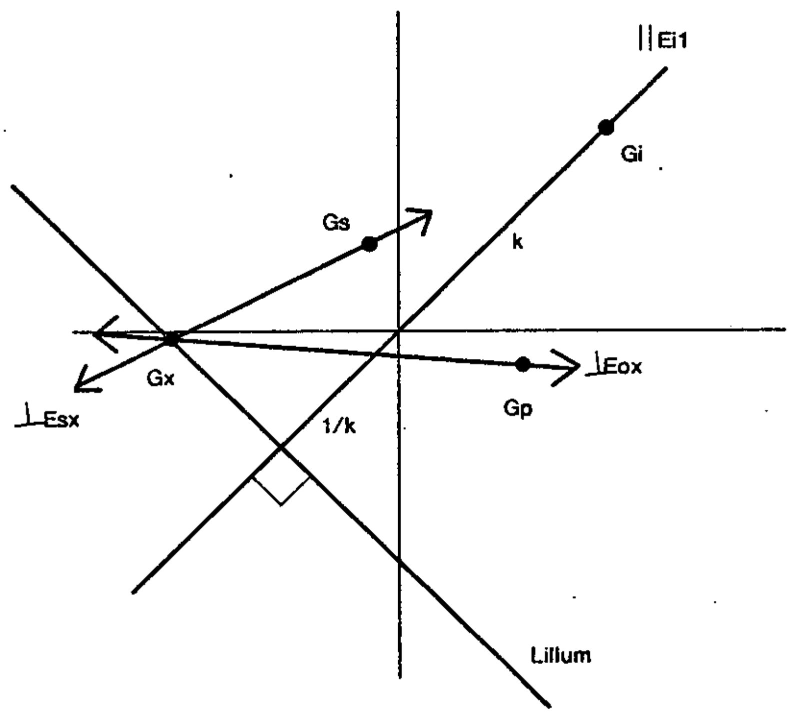


Figure 6-9: Gradient Space Constraints From V_{OX} For Computing G_S

If $(\Delta x_{OX}, \Delta y_{OX}, \Delta z_{OX})$ is the three-dimensional vector corresponding to E_{OX}, with similar definitions for the other vectors, then Δz_{OX} and Δz_{OY} are known in advance. As previously shown, if G_X is the gradient of S_I (and S_O) at V_{OX}, then

$$-\Delta z_{OX} = G_X \cdot E_{OX}$$

Since E_{IX} is an illumination vector,

$$\Delta z_{IX} = \|E_{IX}\| \,/\, k$$

and, since E_{IX} is contained in S_I at V_{OX},

$$-\Delta z_{IX} = G_X \cdot E_{IX}$$

Combining,

$$\begin{bmatrix} -\Delta z_{OX} \\ -\Delta z_{IX} \end{bmatrix} = \begin{bmatrix} E_{OX}^T \\ E_{IX}^T \end{bmatrix} G_X$$

$$G_X = \begin{bmatrix} E_{OX}^T \\ E_{IX}^T \end{bmatrix}^{-1} \begin{bmatrix} -\Delta z_{OX} \\ -\|E_{IX}\|/k \end{bmatrix}$$

So, G_X (and, similarly, G_Y, the gradient of S_I at V_{OY}), can be determined exactly.

Now, since S_I and S_S intersect at V_{SX} along E_{SX},

$$-\Delta z_{SX} = G_X \cdot E_{SX} = G_S \cdot E_{SX}$$

Similarly,

$$-\Delta z_{SY} = G_Y \cdot E_{SY} = G_S \cdot E_{SY}$$

So,

$$\begin{bmatrix} -\Delta z_{SX} \\ -\Delta z_{SY} \end{bmatrix} = \begin{bmatrix} E_{SX}^T \\ E_{SY}^T \end{bmatrix} G_S$$

$$G_S = \begin{bmatrix} E_{SX}^T \\ E_{SY}^T \end{bmatrix}^{-1} \begin{bmatrix} -\Delta z_{SX} \\ -\Delta z_{SY} \end{bmatrix} = \begin{bmatrix} E_{SX}^T \\ E_{SY}^T \end{bmatrix}^{-1} \begin{bmatrix} G_X \cdot E_{SX} \\ G_Y \cdot E_{SY} \end{bmatrix}$$

and therefore, G_S can be determined exactly. Now, G_S can be used as previously shown to determine the gradient of S_O at each point on the arc of extinction E_{IO}. Here, knowledge of k and the direction tangent to E_{IO} at two points has sufficed to determine the gradient of S_S and the gradient of S_O at all points along E_{IO}.

In the special case that S_O is spherical, for example, the entire arc of extinction E_{IO} lies in a plane S_P whose surface normal is an illumination vector. Therefore, the gradient $G_P = G_I$. In this case, the entire problem can be solved with only one parameter (k) given in advance, since Δz_{OX} and Δz_{OY} can be calclated directly:

$$G_I = E_{I1} \frac{k}{\|E_{I1}\|}$$

$$-\Delta z_{OX} = E_{OX} \cdot G_I = \frac{k\,(E_{OX} \cdot E_{I1})}{\|E_{I1}\|}$$

and $$-\Delta z_{OY} = \frac{k\,(E_{OY} \cdot E_{I1})}{\|E_{I1}\|}$$

The shadow information just described is not redundant with information available in the same line drawings when no shadows are present.

7. Shadow Geometry and Other Shape Inference Techniques

Shadow geometry can be combined with other techniques for determining 3D interpretations from images.

7.1 Other Gradient Space Techniques

In Appendix I, the closed-form solution for the Basic Shadow Problem is presented. The solution is stated in the form:

$$G_O = f(G_S, k)$$

When k is given in advance, G_O is shown to be an affine transform (two-dimensional linear transform) of G_S.

Stated in this form, it is very convenient to use shadow geometry in conjunction with other techniques for determining surface gradients. For example, in figure 7-1, a line drawing is shown in which the intensities of the surfaces are known. If the surfaces are Lambertian or have known reflectance functions, Horn's "shape from shading" technique [4] can be used to determine a contour in gradient space along which G_S must lie, and a similar contour for G_O. Now, if the contour for G_S is transformed in its entirety by the function f provided by shadow geometry (as discussed above), a new contour for G_O is provided in gradient space (figure 7-2). Since G_O must lie along two contours, it must lie at one of the points of intersection of these contours. Now, for each such point, the corresponding point G_S can be determined using the inverse of transform f.

Shadow geometry can similarly be combined with Kanade and Kender's "skewed symmetry" [7], as in figure 7-3. Here, skewed symmetry provides a hyperbolic contour for each of the two surface gradients G_O and G_S; shadow geometry can be used to transform the contour for G_S into an additional contour for G_O. The points of intersection of the contours for G_O are then the possible values of G_O, and the corresponding values of G_S can be found as above.

7.2 Shape Recovery for Curved Surfaces

Some techniques have appeared in the literature for reconstructing the orientation of a curved surface at every point, using relaxation techniques [1, 6]. These techniques typically begin with the surface orientation at every point along the outline of the surface (S_O in figure 7-4). These values form a boundary condition which drives the relaxation process.

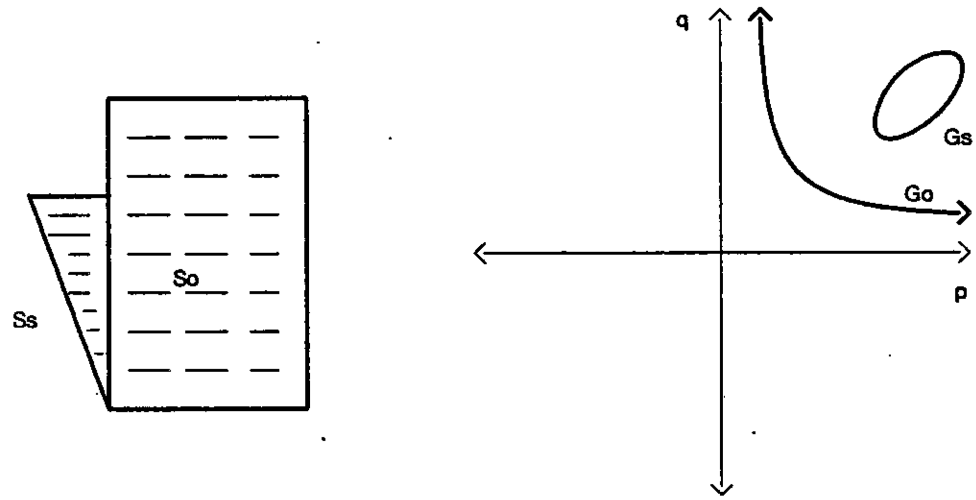


Figure 7-1: Shape From Shading

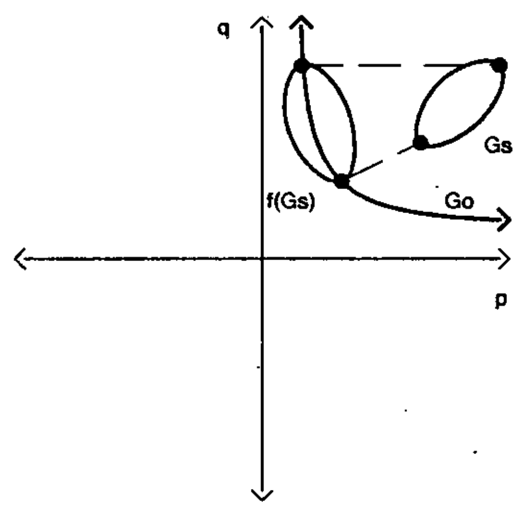


Figure 7-2: Shadow Geometry and Shape From Shading

In this paper, we have seen that it is possible to determine the surface orientation for the tangent planes at each point along the arc of extinction E_{IO}, using three *a priori* parameters (such as the *k* value for the light source and the orientation of the surface on which the shadow appears). These values can be used to provide stronger boundary conditions for relaxation techniques.

Surface orientations along the arc of extinction are valuable for another reason. Relaxation techniques must make some presumptions about the curvature of the surface (e.g. surface of

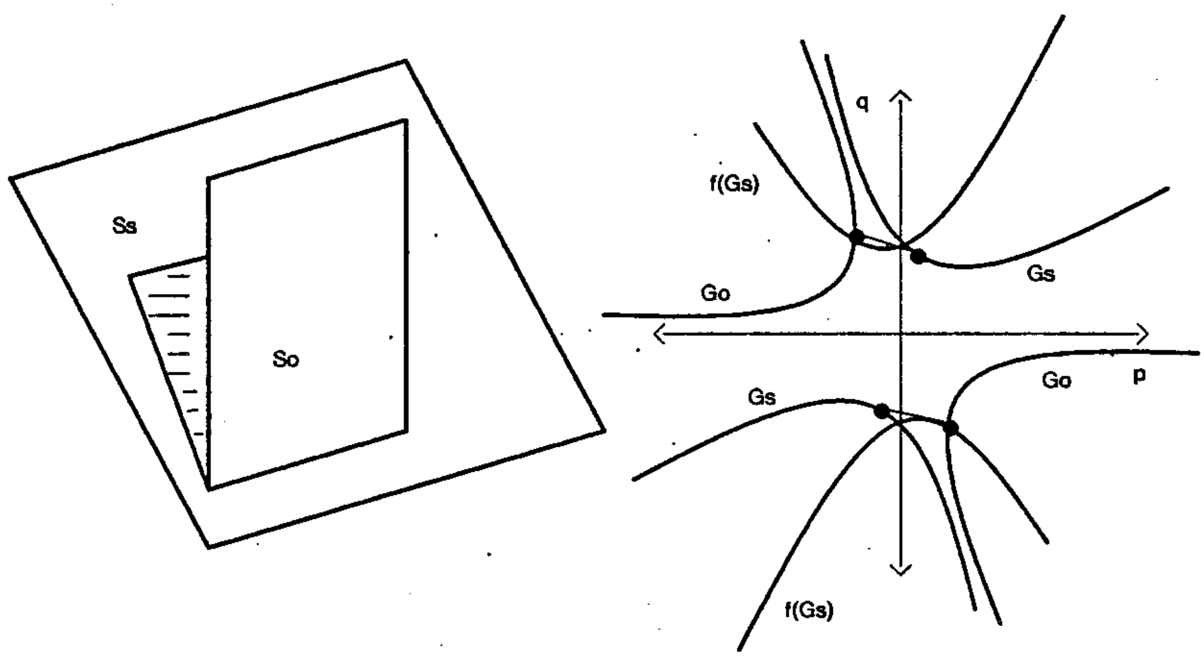


Figure 7-3: Shadow Geometry and Skewed Symmetry

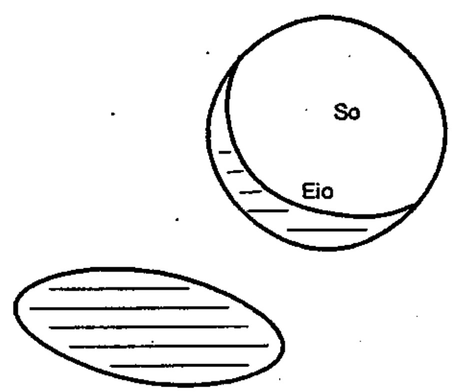


Figure 7-4: Shadow Geometry and Curved Surface Shape Recovery

minimum curvature, cubic or other surface of revolution). Since all of these models of curvature are consistent with the tangent gradients along the outline of So, it is not possible to decide which model is appropriate when the only boundary condition comes from the outline of S_O. However, when the arc of extinction is also used, it may be possible to select one from several possible models of surface curvature, or to measure systematic deviation from a particular model for a specific object.

7.3 Shadow Geometry and Stereo

It is known that, when a stereo pair of images is available but the camera positions are unknown, five pairs of corresponding points in the two images can be used to compute the three-dimensional description within a single unknown scaling factor [3].

With shadow geometry, it is possible to compute this scaling factor, thus solving the system uniquely, when the value of *k* for the light source is known. It is only necessary to find a single shadow-making point V_O, and its shadow point V_S, as in figure 7-5. In this figure, *d* is the distance (on the *x-y* image plane) between V_O and V_S.

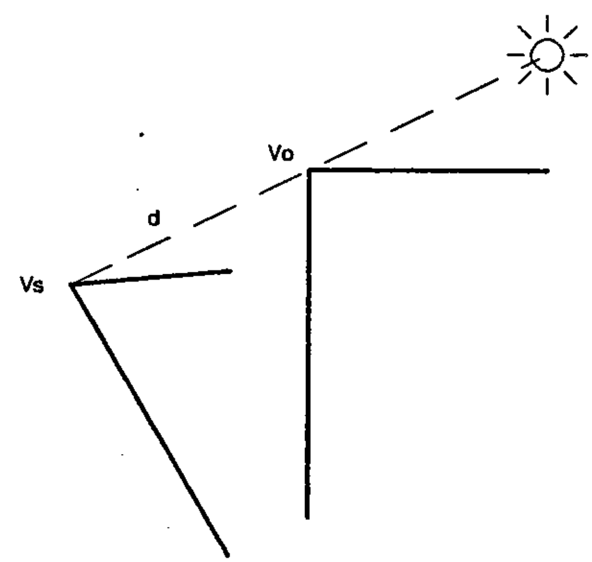


Figure 7-5: Shadow Geometry and Stereo

As we have seen in equation (2.1), *k* has the value:

$$k = \mathrm{sqrt}(\Delta x^2 + \Delta y^2) / \Delta z = d / \Delta z$$

So, Δz can be computed for the line segment joining V_O and V_S; from this, the value of the unknown scaling factor can be calculated. This is the same idea that underlies *light striping* [16].

In this technique, *a priori* knowledge about the light source position (in 3D) is used to provide the additional constraint needed for unique interpretation of images in a stereo pair. Shadow geometry has provided the means for converting knowledge about illumination direction into an absolute distance, as required for the solution of the stereo problem. In essence, the shadows provide the image of the occluding surface from an additional point of view (located at the light source) [9].

8. Conclusions

This paper has presented a theory describing relationships among surface orientations in line drawings with shadows. The relationships arise from hypothesizing the existence of "illumination surfaces" connecting shadow edge pairs, assigning appropriate line labels to shadow and shadow-making edges, and applying the resulting constraints in the gradient space.

This technique falls short of providing exact solutions to shadow geometry problems. The line drawing must be augmented with information such as the orientations or curvature of specific surfaces or the position of the light source if exact surface orientations are to be found.

It has been shown, however, that shadow geometry provides important benefits for image understanding:

- Shadows allow you to subsitute information about the light source position instead of *a priori* knowledge about surface orientations.
- Shadows allow you to determine geometric information from highly visible shadow edge pairs instead of using many of the unreliable edges within shaded portions of an image.
- An increasing amount of information is provided by the shadow edge when the shadow falls on many visible, differently oriented surfaces.
- Shadows provide some constraint when curved surfaces are involved.
- Shadows provide constraint between surfaces even when they do not touch in the scene (or image).
- Shadows allow the solution to one shadow problem to be used in the solution of other shadow problems, since typical shadow problems are mutually constrained (e.g. same light source, same background surface).

In addition, some observations have been made about the solution of the correspondence problem for shadows, which must be solved before surface orientations can be inferred.

8.1 Future Work

Work remains to be done on the following topics:

- Generalization of the entire method for images under perspective.
- Complex interactions between multiple light sources, polyhedra, and curved surfaces.

- Relationships between shadow geometry and other gradient space constraints, such as Horn's "shape from shading" [4], Kanade's "skewed symmetry" [7], and Kender's "shape from texture" paradigm [7].

- Finding shadow edges and solving the correspondence problem for shadow edge pairs.

A program should be written to employ the techniques presented here. The first application might be in conjunction with an interactive photo-interpretation program [11], in which the segmentation and correspondence problems will be solved by the human operator.

Additional questions have been raised in the course of this research concerning the more general problem of determining surface orientations from (or using) a line drawing. In an image such as 8-1, only three *a priori* pieces of information are needed besides the line drawing itself to determine the surface orientations uniquely. However, constraints can be found by all of these techniques:

- Shape from shading [4]

- Skewed symmetry and gravity [7]

- Shadow geometry

This problem is actually over-constrained. It should be possible to find a method for using the redundant information to improve erroneous segmentation and reduce the overall uncertainty in the shape recovery process;however, this topic has not yet been explored.

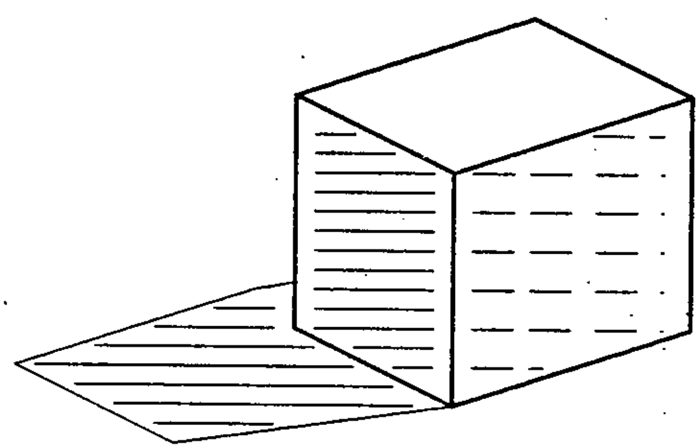


Figure 8-1: Image of Cube: Interpretation is Over-Constrained

The line drawing itself provides many constraints, but the following questions remain to be addressed:

- Given a line drawing with labeled edges, how many pieces of additional *a priori* information are required to determine all of the surface orientations?

- If additional information is needed, where can it be obtained? (Shape from Shading? Skewed Symmetry? Shadow Geometry? Other techniques?) Can this question be answered during the actual process of image analysis, or must the answer be known when the program is written?

- If the required amount of information is present, it may have been obtained from diverse sources. How can it all be combined algorithmically to determine the surface orientations?

- If too much information is present, the solution is over-constrained. In this case, the redundant information should be useful for correcting errorful segmentation or for improving the reliability of the solution. How can this be done?

8.2 Acknowledgements

John Kender, at Columbia University, provided suggestions which led to several of the generalizations in this paper. His observations regarding curved surfaces provided the starting point for the ideas described in this paper regarding shadows cast by curved surfaces.

In addition, this paper has benefited from review and suggestions by Marty Herman, Dave McKeown, Raj Reddy, and David Smith, all members of the Image Understanding project at CMU.

9. Bibliography

[1] Barrow, H. G. and Tenenbaum, J. M.
Reconstructing Smooth Surfaces From Partial, Noisy Information.
In Baumann, L. S. (editor), *ARPA IUS Workshop*, pages 76-86. November, 1979.

[2] Clowes, M. B.
On Seeing Things.
Artificial Intelligence 2:79-116, 1971.

[3] Gennery, D. B.
Stereo-Camera Calibration.
In Baumann, L. S. (editor), *ARPA IUS Workshop*, pages 101-107. November, 1979.

[4] Horn, B. K. P.
Understanding Image Intensities.
Artificial Intelligence 8:201-231, 1977.

[5] Huffman, D. A.
Impossible Objects as Nonsense Sentences.
In Meltzer, B. and Michie, D. (editor), *Machine Intelligence 6*, chapter 19pages 295-323. American Elsevier Pub. Co., New York, 1971.

[6] Ikeuchi, K.
Numerical Shape From Shading and Occluding Contours in a Single View.
AIM 566, MIT, November, 1979.

[7] Kanade, T. and Kender, J.
Mapping Image Properties into Shape Constraints: Skewed Symmetry, Affine-Transformable Patterns, and the Shape-from-Texture Paradigm.
Technical Report, Carnegie-Mellon University, Computer Science Department, July, 1980.

[8] Kender, J. R.
The Gaussian Sphere: A Unifying Representation of Surface Orientation.
In Baumann, L. S. (editor), *ARPA IUS Workshop*, pages 157-160. April, 1980.

[9] Lowe, D. G. and Binford, T. O.
The Interpretation of Geometric Structure from Image Boundaries.
In Baumann, L. S. (editor), *ARPA IUS Workshop*, pages 39-46. April, 1981.

[10] Mackworth, A. K.
Interpreting Pictures of Polyhedral Scenes.
Artificial Intelligence 4:121-137, 1973.

[11] McKeown, D. M. and Kanade, T.
Database Support for Automated Photo-Interpretation.
In Baumann, L. S. (editor), *ARPA IUS Workshop*, pages 7-13. April, 1981.

[12] Nagao, M., Matsuyama, T., and Ikeda, Y.
Region Extraction and Shape Analysis in Aerial Photographs.
Computer Graphics and Image Processing 10:195-223, 1979.

[13] O'Gorman, F.
Light Lines and Shadows.
Technical Report, School of Social Sciences, U. of Sussex, December, 1975.

[14] Ohlander, R. B.
Analysis of Natural Scenes.
PhD thesis, Carnegie-Mellon University, April, 1975.

[15] Quam, L.
Work in progress at SRI. Technical report expected.

[16] Shirai, Y. and Suwa, M.
Recognition of Polyhedrons With a Range Finder.
In *IJCAI-71*, pages 80-87. Imperial College, London, September, 1971.

[17] Smith, H. T. U.
Aerial Photographs and Their Applications.
D. Appleton-Century Co., New York, 1943.

[18] Waltz, D.
Understanding Line Drawings of Scenes with Shadows.
In Winston, P. H. (editor), *The Psychology of Computer Vision*, chapter 2pages 19-91.
McGraw-Hill, 1975.

[19] Witkin, A.
Shape From Contour.
AI-TR 589, MIT, November, 1980.

[20] Witkin, A.
Work in progress at SRI. Technical report expected.

I. Solution to the Basic Shadow Problem

The Basic Shadow Problem can be formulated in terms of vector relationships, for development of a closed-form analytic solution. Here, we will describe the gradient G_O of the occluding surface S_O in terms of the gradient G_S of the shaded surface S_S and the relative z-component of the direction of illumination, k.

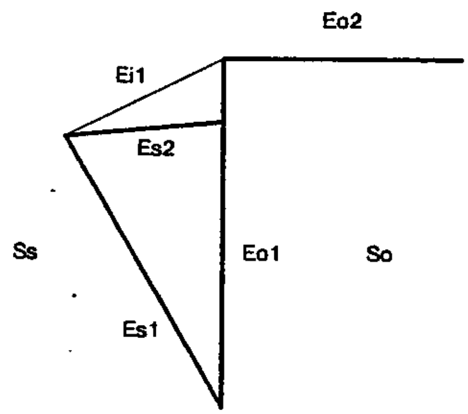


Figure 9-1: The Basic Shadow Problem

In figure 9-1, let edge $E_{I1} = (\Delta x_{I1}, \Delta y_{I1})$ correspond to the three-dimensional scene vector $(\Delta x_{I1}, \Delta y_{I1}, \Delta z_{I1})$. Three-dimensional counterparts can similarly be defined for the other vectors E_{O1}, E_{O2}, E_{S1}, and E_{S2}.

The following relations hold:

1. By the definition of k, $k = \|E_{I1}\| \,/\, \Delta z_{I1}$. It should be noted that k is positive (or zero) if S_O is illuminated, and negative if S_O is self-shadowed. With this definition, all the equations presented here hold for any direction of illumination.

2. $E_{I1} \in S_{I1}, S_{I2}$, so $-\Delta z_{I1} = G_{I1} \cdot E_{I1} = G_{I2} \cdot E_{I1}$

3. $E_{S1} \in S_{I1}, S_S$, so $-\Delta z_{S1} = G_{I1} \cdot E_{S1} = G_S \cdot E_{S1}$

4. $E_{S2} \in S_{I2}, S_S$, so $-\Delta z_{S2} = G_{I2} \cdot E_{S2} = G_S \cdot E_{S2}$

5. $E_{O1} \in S_{I1}, S_O$, so $-\Delta z_{O1} = G_{I1} \cdot E_{O1} = G_O \cdot E_{O1}$

6. $E_{O2} \in S_{I2}, S_O$, so $-\Delta z_{O2} = G_{I2} \cdot E_{O2} = G_O \cdot E_{O2}$

I.1 Gradient of the First Illumination Surface, G_{I1}

The constraints on G_{I1} are expressed by (1) and (2) above:

$$-\Delta z_{I1} = -\|E_{I1}\| / k = G_{I1} \cdot E_{I1}$$

and by (3):

$$-\Delta z_{S1} = G_{I1} \cdot E_{S1} = G_S \cdot E_{S1}$$

These can be combined into a single equation using matrices. The top row of the computation represents the first equation, and the bottom row represents the second equation:

$$\begin{bmatrix} -\Delta z_{I1} \\ -\Delta z_{S1} \end{bmatrix} = \begin{bmatrix} G_{I1} \cdot E_{I1} \\ G_{I1} \cdot E_{S1} \end{bmatrix} = \begin{bmatrix} E_{I1}^T \\ E_{S1}^T \end{bmatrix} G_{I1}$$

$$G_{I1} = \begin{bmatrix} E_{I1}^T \\ E_{S1}^T \end{bmatrix}^{-1} \begin{bmatrix} -\Delta z_{I1} \\ -\Delta z_{S1} \end{bmatrix} = \begin{bmatrix} E_{I1}^T \\ E_{S1}^T \end{bmatrix}^{-1} \begin{bmatrix} -\|E_{I1}\| / k \\ G_S \cdot E_{S1} \end{bmatrix}$$

This equation defines G_{I1} in terms of G_S, k, and several edges (measurable in the image).

It is possible to compute the coordinates of G_I in terms of the coordinates of the various vectors. To begin, we can use the fact that

$$\begin{bmatrix} a\,b \\ c\,d \end{bmatrix}^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \Big/ \begin{vmatrix} a\,b \\ c\,d \end{vmatrix}$$

to give the equation

$$G_{I1} = \begin{bmatrix} \Delta y_{S1}/D & -\Delta y_{I1}/D \\ -\Delta x_{S1}/D & \Delta x_{I1}/D \end{bmatrix} \begin{bmatrix} -\|E_{I1}\| / k \\ p_S \Delta x_{S1} + q_S \Delta y_{S1} \end{bmatrix}$$

where

$$D = \begin{vmatrix} E_{I1}^T \\ E_{S1}^T \end{vmatrix} = \Delta x_{I1} \Delta y_{S1} - \Delta y_{I1} \Delta x_{S1}$$

Then,

$$G_{I1} = \begin{bmatrix} -p_S \Delta x_{S1} \Delta y_{I1}/D - q_S \Delta y_{S1} \Delta y_{I1}/D - \|E_{I1}\| \Delta y_{S1}/kD \\ p_S \Delta x_{S1} \Delta x_{I1}/D + q_S \Delta y_{S1} \Delta x_{I1}/D + \|E_{I1}\| \Delta x_{S1}/kD \end{bmatrix}$$

$$= \begin{bmatrix} Ap_S + Bq_S + C/k \\ Ep_S + Fq_S + G/k \end{bmatrix} = \begin{bmatrix} A\,B\,C \\ E\,F\,G \end{bmatrix} \begin{bmatrix} p_S \\ q_S \\ 1/k \end{bmatrix}$$

where

$$A = -\Delta x_{S1} \Delta y_{I1} / D$$
$$B = -\Delta y_{S1} \Delta y_{I1} / D$$
$$C = -\Delta y_{S1}\, \mathrm{sqrt}(\Delta x_{I1}^2 + \Delta y_{I1}^2) / D$$
$$E = \Delta x_{S1} \Delta x_{I1} / D$$
$$F = \Delta y_{S1} \Delta x_{I1} / D$$
$$G = \Delta x_{S1}\, \mathrm{sqrt}(\Delta x_{I1}^2 + \Delta y_{I1}^2) / D$$

and

$$D = \Delta x_{I1}\Delta y_{S1} - \Delta y_{I1}\Delta x_{S1}$$

Here, the assumption has been made that $D \neq 0$, i.e. that $E_{I1} \nparallel E_{S1}$.

I.2 Gradient of the Second Illumination Surface, G_{I2}

G_{I2} is determined in a manner analogous to the determination of G_{I1} presented above:

$$G_{I2} = \begin{bmatrix} Hp_S + Iq_S + J/k \\ Lp_S + Mq_S + N/k \end{bmatrix} = \begin{bmatrix} H & I & J \\ L & M & N \end{bmatrix} \begin{bmatrix} p_S \\ q_S \\ 1/k \end{bmatrix}$$

where

$$H = -\Delta x_{S2}\Delta y_{I1} / P$$
$$I = -\Delta y_{S2}\Delta y_{I1} / P$$
$$J = -\Delta y_{S2}\, \mathrm{sqrt}(\Delta x_{I1}^2 + \Delta y_{I1}^2) / P$$
$$L = \Delta x_{S2}\Delta x_{I1} / P$$
$$M = \Delta y_{S2}\Delta x_{I1} / P$$
$$N = \Delta x_{S2}\, \mathrm{sqrt}(\Delta x_{I1}^2 + \Delta y_{I1}^2) / P$$

and

$$P = \Delta x_{I1}\Delta y_{S2} - \Delta y_{I1}\Delta x_{S2}$$

with the assumption that $P \neq 0$, i.e. $E_{I1} \nparallel E_{S2}$.

I.3 Gradient of the Occluding Surface, G_O

To determine G_O, we use relations (5) and (6) presented above:

$$-\Delta z_{O1} = G_{I1} \cdot E_{O1} = G_O \cdot E_{O1}$$
$$-\Delta z_{O2} = G_{I2} \cdot E_{O2} = G_O \cdot E_{O2}$$

In combined form,

$$\begin{bmatrix} -\Delta z_{O1} \\ -\Delta z_{O2} \end{bmatrix} = \begin{bmatrix} E_{O1}^T \\ E_{O2}^T \end{bmatrix} G_O$$

$$G_O = \begin{bmatrix} E_{O1}^T \\ E_{O2}^T \end{bmatrix}^{-1} \begin{bmatrix} -\Delta z_{O1} \\ -\Delta z_{O2} \end{bmatrix} = \begin{bmatrix} E_{O1}^T \\ E_{O2}^T \end{bmatrix}^{-1} \begin{bmatrix} G_{I1} \cdot E_{O1} \\ G_{I2} \cdot E_{O2} \end{bmatrix}$$

$$= \begin{bmatrix} \Delta y_{O2}/W & -\Delta y_{O1}/W \\ -\Delta x_{O2}/W & \Delta x_{O1}/W \end{bmatrix} \begin{bmatrix} p_{I1}\Delta x_{O1} + q_{I1}\Delta y_{O1} \\ p_{I2}\Delta x_{O2} + q_{I2}\Delta y_{O2} \end{bmatrix}$$

where

$$W = \begin{vmatrix} E_{O1}^T \\ E_{O2}^T \end{vmatrix} = \Delta x_{O1}\Delta y_{O2} - \Delta y_{O1}\Delta x_{O2}$$

The terms p_{I1}, q_{I1}, p_{I2}, and q_{I2} can be expanded in terms of p_S, q_S, and k, to yield:

$$G_O = \begin{bmatrix} Qp_S + Rq_S + S/k \\ Tp_S + Uq_S + V/k \end{bmatrix} = \begin{bmatrix} Q\ R\ S \\ T\ U\ V \end{bmatrix} \begin{bmatrix} p_S \\ q_S \\ 1/k \end{bmatrix}$$

where

$$\begin{bmatrix} Q\ R\ S \\ T\ U\ V \end{bmatrix} = \frac{1}{W} \begin{bmatrix} \Delta x_{O1}\Delta y_{O2} & \Delta y_{O1}\Delta y_{O2} & -\Delta y_{O1}\Delta y_{O2} & -\Delta y_{O1}\Delta y_{O2} \\ -\Delta x_{O1}\Delta x_{O2} & -\Delta y_{O1}\Delta x_{O2} & \Delta x_{O1}\Delta x_{O2} & \Delta x_{O1}\Delta y_{O2} \end{bmatrix} \begin{bmatrix} A\ B\ C \\ E\ F\ G \\ H\ I\ J \\ L\ M\ N \end{bmatrix}$$

and

A through *N* are defined as before

$W = \Delta x_{O1}\Delta y_{O2} - \Delta y_{O1}\Delta x_{O2}$

Expansion of the coefficients *A* through *N* does not yield additional simplification in the above equation.

Here, the assumption has been made that $W \neq 0$, i.e. $E_{O1} \nparallel E_{O2}$.