the edges with normalized magnitudes greater than 0.25. Weak edges cause many gaps in the boundaries. The figures on the right side show the results of five iterations of edge relaxation. Here the confidence of the weak edges has been increased owing to the proximity of other edges, using the rules in Table 3.1.

Table 3.1

Decrement	*Increment*	*Leave as is*
0-0	*1-1*	*0-1*
0-2	*1-2*	*2-2*
0-3	*1-3*	*2-3*
		3-3

3.4 RANGE INFORMATION FROM GEOMETRY

Neither the perspective or orthogonal projection operations, which take the three-dimensional world to a two-dimensional image, is invertible in the usual sense. Since projection maps an infinite line onto a point in the image, information is lost. For a fixed viewpoint and direction, infinitely many continuous and discontinuous three-dimensional configurations of points could project on our retina in an image of, say, our grandmother. Simple cases are grandmothers of various sizes cleverly placed at varying distances so as to project onto the same area. An astronomer might imagine millions of points distributed perhaps through light-years of space which happen to line up into a "grandmother constellation." All that can be mathematically guaranteed by imaging geometry is that the image point corresponds to one of the infinite number of points on that three-dimensional line of sight. The "inverse perspective" transformation (Appendix 1) simply determines the equation of the infinite line of sight from the parameters of the imaging process modeled as a point projection.

However, a line and a plane not including it intersect in just one point. Lines of sight are easy to compute, and so it is possible to tell where any image point projects on to any known plane (the supporting ground or table plane is a favorite). Similarly, if two images from different viewpoints can be placed in correspondence, the intersection of the lines of sight from two matching image points determines a point in three-space. These simple observations are the basis of light-striping ranging (Section 2.3.3) and are important in stereo imaging.

3.4.1. Stereo Vision and Triangulation

One of the first ideas that occurs to one who wants to do three-dimensional sensing is the biologically motivated one of stereo vision. Two cameras, or one camera from two positions, can give relative depth or absolute three-dimensional location, depending on the elaboration of the processing and measurement. There has been

considerable effort in this direction [Moravec 1977; Quam and Hannah 1974; Binford 1971; Turner 1974; Shapira 1974]. The technique is conceptually simple:

1. Take two images separated by a baseline.
2. Identify points between the two images.
3. Use the inverse perspective transform (Appendix 1) or simple triangulation (Section 2.2.2) to derive the two lines on which the world point lies.
4. Intersect the lines.

The resulting point is in three-dimensional world coordinates.

The hardest part of this method is step 2, that of identifying corresponding points in the two images. One way of doing this is to use correlation, or template matching, as described in Section 3.2.1. The idea is to take a patch of one image and match it against the other image, finding the place of best match in the second image, and assigning a related "disparity" (the amount the patch has been displaced) to the patch.

Correlation is a relatively expensive operation, its naive implementation requiring $0(n^2m^2)$ multiplications and additions for an $m \times m$ patch and $n \times n$ image. This requirement can be drastically improved by capitalizing on the idea of variable resolution; the improved technique is described in Section 3.7.2.

Efficient correlation is of technological concern, but even if it were free and instantaneous, it would still be inadequate. The basic problems with correlation in stereo imaging have to do with the fact that things can look significantly different from different points of view. It is possible for the two stereo views to be sufficiently different that corresponding areas may not be matched correctly. Worse, in scenes with much obscuration, very important features of the scene may be present in only one view. This problem is alleviated by decreasing the baseline, but of course then the accuracy of depth determinations suffers; at a baseline length of zero there is no problem, but no stereo either. One solution is to identify world features, not image appearance, in the two views, and match those (the nose of a person, the corner of a cube). However, if three-dimensional information is sought as a help in perception, it is unreasonable to have to do perception first in order to do stereo.

3.4.2 A Relaxation Algorithm for Stereo

Human *stereopsis*, or fusing the inputs from the eyes into a stereo image, does not necessarily involve being aware of features to match in either view. Most human beings can fuse quite efficiently stereo pairs which individually consist of randomly placed dots, and thus can perceive three-dimensional shapes without recognizing monocular clues in either image. For example, consider the stereo pair of Fig. 3.23. In either frame by itself, nothing but a randomly speckled rectangle can be perceived. All the stereo information is present in the relative displacement of dots in the two rectangles. To make the right-hand member of the stereo pair, a patch of

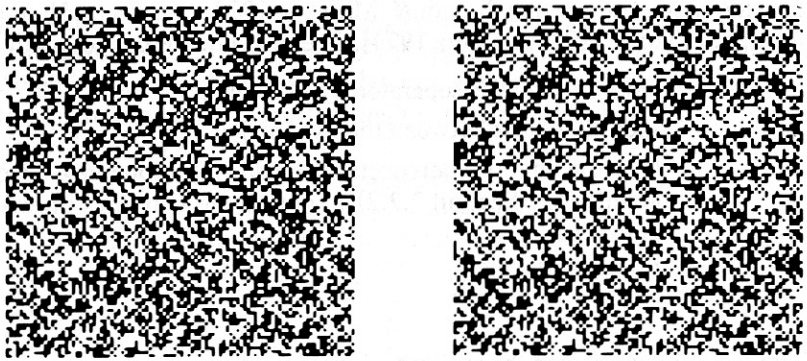

Fig. 3.23 A random-dot stereogram.

the randomly placed dots of the left-hand image is displaced sideways. The dots which are thus covered are lost, and the space left by displacing the patch is filled in with random dots.

Interestingly enough, a very simple algorithm [Marr and Poggio 1976] can be formulated that computes disparity from random dot stereograms. First consider the simpler problem of matching one-dimensional images of four points as depicted in Fig. 3.24. Although only one depth plane allows all four points to be placed in correspondence, lesser numbers of points can be matched in other planes.

The crux of the algorithm is the rules, which help determine, on a local basis, the appropriateness of a match. Two rules arise from the observation that most images are of opaque objects with smooth surfaces and depth discontinuities only at object boundaries:

1. Each point in an image may have only one depth value.
2. A point is almost sure to have a depth value near the values of its neighbors.

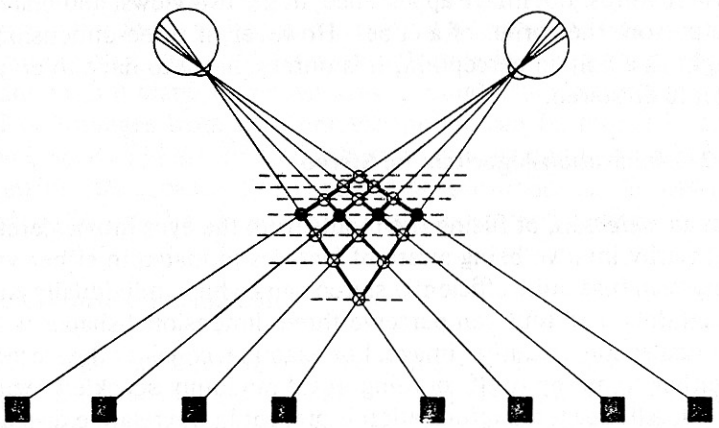

Fig. 3.24 The stereo matching problem.

Figure 3.24 can be viewed as a binary network where each possible match is represented by a binary state. Matches have value 1 and nonmatches value 0. Figure 3.25 shows an expanded version of Fig. 3.24. The connections of alternative matches for a point inhibit each other and connections between matches of equal depth reinforce each other. To extend this idea to two dimensions, use parallel arrays for different values of y where equal depth matches have reinforcing connections. Thus the extended array is modeled as the matrix $C(x, y, d)$ where the point x, y, d corresponds to a particular match between a point (x_1, y_1) in the right image and a point (x_2, y_2) in the left image. The stereopsis algorithm produces a series of matrices C_n which converges to the correct solution for most cases. The initial matrix $C_0(x, y, d)$ has values of one where x, y, d correspond to a match in the original data and has values of zero or otherwise.

Algorithm 3.2 [Marr and Poggio 1976]

Until C satisfies some convergence criterion, do

$$C_{n+1}(x, y, d) = \left\{ \sum_{x',y',d' \in S} C_n(x', y', d') - \sum_{x',y',d' \in \theta} C_n(x', y', d') + C_0(x, y, d) \right\} \quad (3.34)$$

where the term in braces is handled as follows:

$$\{t\} = \begin{cases} 1 & \text{if } t > T \\ 0 & \text{otherwise} \end{cases}$$

S = set of points x', y', d' such that $|\mathbf{x} - \mathbf{x}'| \leqslant 1$ and $d = d'$

θ = set of points x', y', d' such that $|\mathbf{x} - \mathbf{x}'| \leqslant 1$ and $|d - d'| = 1$

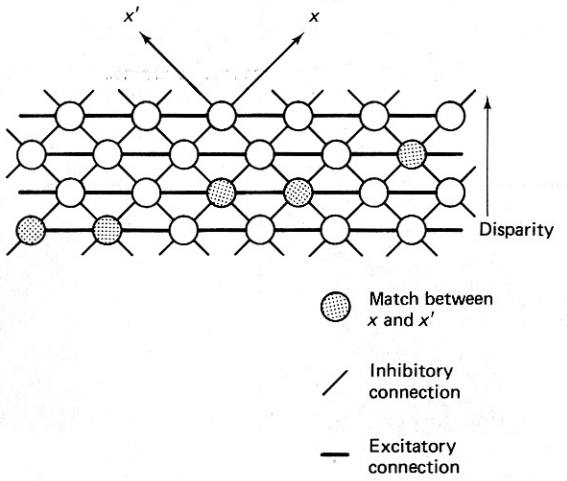

Fig. 3.25 Extension of stereo matching.

One convergence criterion is that the number of points modified on an iteration must be less than some threshold T. Fig. 3.26 shows the results of this computation; the disparity is encoded as a gray level and displayed as an image for different values of n.

A more general version of this algorithm matches image features such as edges rather than points (in the random-dot stereogram, the only features are

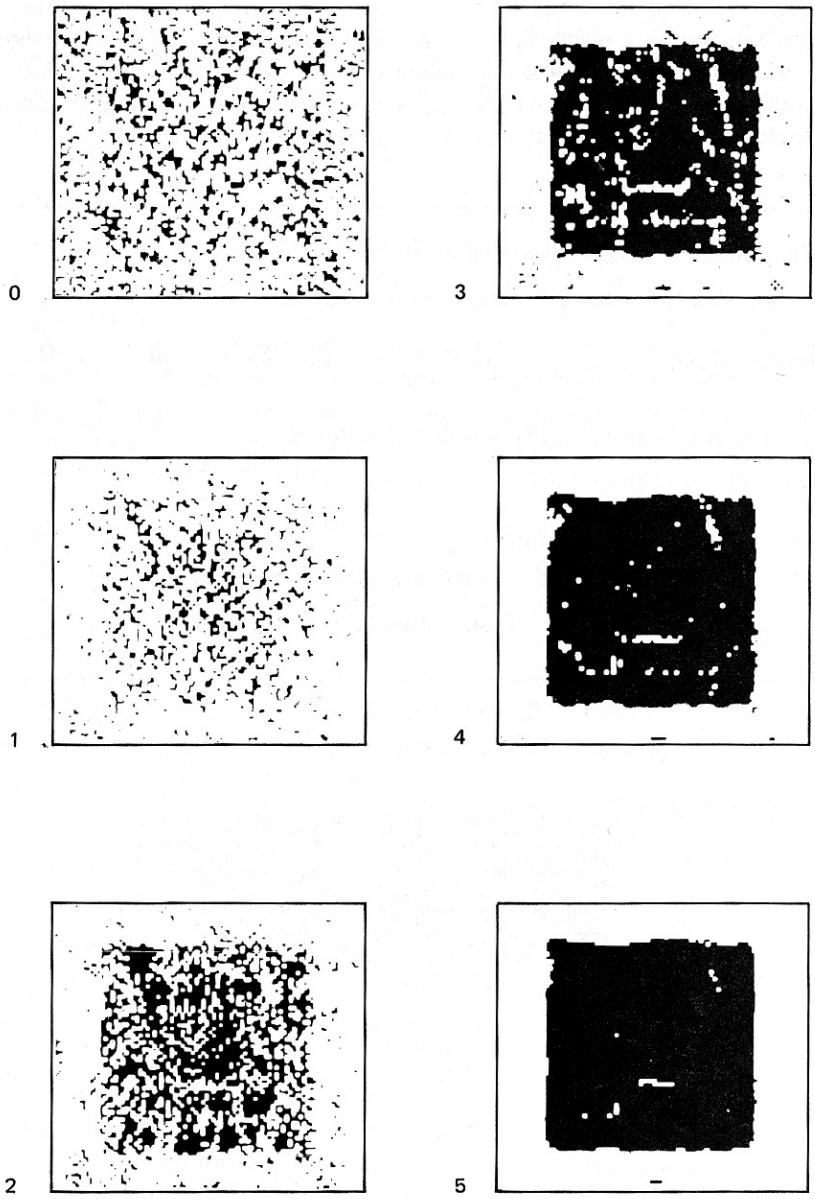

Fig. 3.26 The results of relaxation computations for stereo.

points), but the principles are the same. The extraction of features more complicated than edges or points is itself a thorny problem and the subject of Part II. It should be mentioned that Marr and Poggio have refined their stereopsis algorithm to agree better with psychological data [Marr and Poggio 1977].

3.5 SURFACE ORIENTATION FROM REFLECTANCE MODELS

The ordinary visual world is mostly composed of opaque three-dimensional objects. The intensity (gray level) of a pixel in a digital image is produced by the light reflected by a small area of surface near the corresponding point on the object.

It is easiest to get consistent shape (orientation) information from an image if the lighting and surface reflectance do not change from one scene location to another. Analytically, it is possible to treat such lighting as uniform illumination, a point source at infinity, or an infinite linear source. Practically, the human shape-from-shading transform is relatively robust. Of course, the perception of shape may be manipulated by changing the surface shading in calculated ways. In part, cosmetics work by changing the reflectivity properties of the skin and misdirecting our human shape-from-shading algorithms.

The recovery transformation to obtain information about surface orientation is possible if some information about the light source and the object's reflectivity is known. General algorithms to obtain and quantify this information are complicated but practical simplifications can be made [Horn 1975; Woodham 1978; Ikeuchi 1980]. The main complicating factor is that even with mathematically tractable object surface properties, a single image intensity does not uniquely define the surface orientation. We shall study two ways of overcoming this difficulty. The first algorithm uses intensity images as input and determines the surface orientation by using multiple light source positions to remove ambiguity in surface orientation. The second algorithm uses a single source but exploits constraints between neighboring surface elements. Such an algorithm assigns initial ranges of orientations to surface elements (actually to their corresponding image pixels) on the basis of intensity. The neighboring orientations are "relaxed" against each other until each converges to a unique orientation (Section 3.5.4).

3.5.1 Reflectivity Functions

For all these derivations, consider a distant point source of light impinging on a small patch of surface; several angles from this situation are important (Fig. 3.27).

A surface's reflectance is the fraction of a given incident energy flux (irradiance) it reflects in any given direction. Formally, the *reflectivity function* is defined as $r = \frac{dL}{dE}$, where L is exitant radiance and E is incident flux. In general, for anisotropic reflecting surfaces, the reflectivity function (hence L) is a function of all three angles i, e, and g. The quantity of interest to us is image irradiance, which is proportional to scene radiance, given by $L = \int r\, dE$. In general, the evaluation of this integral can be quite complicated, and the reader is referred to [Horn and

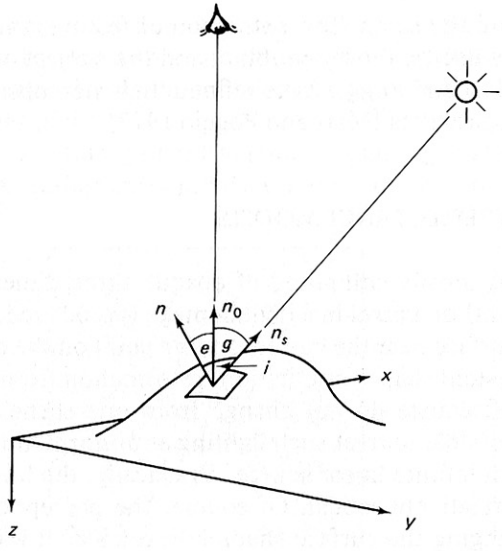

Fig. 3.27 Important reflectance angles: *i*, incidence; *e*, emittance; *g*, phase.

Sjoberg 1978] for a more detailed study. For our purposes we consider surfaces with simple reflectivity functions.

Lambertian surfaces, those with an ideal matte finish, have a very simple reflectivity function which is proportional only to the cosine of the incident angle. These surfaces have the property that under uniform or collimated illumination they look equally bright from any direction. This is because the amount of light reflected from a unit area goes down as the cosine of the viewing angle, but the amount of area seen in any solid angle goes up as the reciprocal of the cosine of the viewing angle. Thus the perceived intensity of a surface element is constant with respect to viewer position. Other surfaces with simple reflectivity functions are "dusty" and "specular" surfaces. An example of a dusty surface is the lunar surface, which reflects in all directions equally. Specular (purely mirror-like) surfaces such as polished metal reflect only at the angle of reflection = angle of incidence, and in a direction such that the incidence, normal, and emittance vectors are coplanar.

Most smooth things have a specular component to their reflection, but in general some light is reflected at all angles in decreasing amounts from the specular angle. One way to achieve this effect is to use the cosine of the angle between the predicted specular angle and the viewing angle, which is given by C where

$$C = 2\cos(i)\cos(e) - \cos(g)$$

This quantity is unity in the pure specular direction and falls off to zero at $\frac{\pi}{2}$ radians away from it. Convincing specular contributions of greater or less sharpness are produced by taking powers of C. A simple radiance formula that allows the simulation of both matte and specular effects is

$$L(i, e, g) = s(C)^n + (1 - s)\cos(i) \tag{3.35}$$

Here s varies between 0 and 1 and determines the fraction of specularly reflected light; n determines the sharpness of specularity peaks. As n increases, the specular peak gets sharper and sharper. Computer graphics research is constantly extending the frontiers of realistic and detailed reflectance, refractance, and illumination calculations [Blinn 1978; Phong 1975; Whitted 1980].

3.5.2 Surface Gradient

The reflectance functions described above are defined in terms of angles measured with respect to a local coordinate frame. For our development, it is more useful to relate the reflectivity function to surface gradients measured with respect to a viewer-oriented coordinate frame.

The concept of *gradient space*, which is defined in a viewer-oriented frame [Horn 1975], is extremely useful in understanding the recovery transformation algorithm for the surface normal. This gradient refers to the orientation of a physical surface, *not* to local intensities. It must not be confused with the *intensity* gradients discussed in Section 3.3 and elsewhere in this book.

Gradient space is a two-dimensional space of slants of scene surfaces. It measures a basic "intrinsic" (three-dimensional) property of surfaces. Consider the point-projection imaging geometry of Fig. 2.2, with the viewpoint at infinity (far from the scene relative to the scene dimensions). The image projection is then orthographic, nor perspective.

The surface gradient is defined for a surface expressed as $-z = f(x, y)$. The gradient is a vector (p, q), where

$$p = \frac{\partial(-z)}{\partial x} \qquad (3.36)$$

$$q = \frac{\partial(-z)}{\partial y}$$

Any plane in the image (such as the face plane of a polyhedral face) may be expressed in terms of its gradient. The general plane equation is

$$Ax + By + Cz + D = 0 \qquad (3.37)$$

Thus

$$-z = \frac{A}{C}x + \frac{B}{C}y + \frac{D}{C} \qquad (3.38)$$

and from (3.36) the gradient may be related to the plane equation:

$$-z = px + qy + K \qquad (3.39)$$

Gradient space is thus the two-dimensional space of (p, q) vectors. The p and q axes are often considered to be superimposed on the x and y image plane coordinate axes. Then the (p, q) vector is "in the direction" of the surface slant of imaged surfaces. Any plane perpendicular to the viewing direction has a (p, q) vector of (0,0). Vectors on the q (or y) axis correspond to planes tilted about the x axis in an "upward" or "downward" ("$y$ward") direction (like the tilt of a dressing table

mirror). The direction arctan (q/p) is the direction of fastest change of surface depth $(-z)$ as x and y change. $(p^2 + q^2)^{1/2}$ is the rate of this change. For instance, a vertical plane "edge on" to the viewer has a (p, q) of $(I\infty, 0)$.

The *reflectance map* $R(p, q)$ represents this variation of perceived brightness with surface orientation. $R(p, q)$ gives scene radiance (Section 2.2.3) as a function of surface gradient (in our usual viewer-centered coordinate system). (Figure 3.27 showed the situation and defined some important angles.) $R(p, q)$ is usually shown as contours of constant scene radiance (Fig. 3.28). The following are a few useful cases.

In the case of a Lambertian surface with the source in the direction of the viewer $(i = e)$, the gradient space image looks like Fig. 3.28. Remember that Lambertian surfaces have constant intensity for constant illumination angle; these constant angles occur on the concentric circles of Fig. 3.28, since the direction of tilt does not affect the magnitude of the angle. The brightest surfaces are those illuminated from a normal direction—they are facing the viewer and so their gradients are $(0, 0)$.

Working this out from first principles, the incident angle and emittance angle are the same in this case, since the light is near the viewer. Both are the angle between the surface normal and the view vector. Looking at the $x-y$ plane means a vector to the light source of $(0, 0, -1)$, and at a gradient point (p, q), the surface normal is $(p, q, -1)$. Also,

$$R = r_0 \cos i \tag{3.40}$$

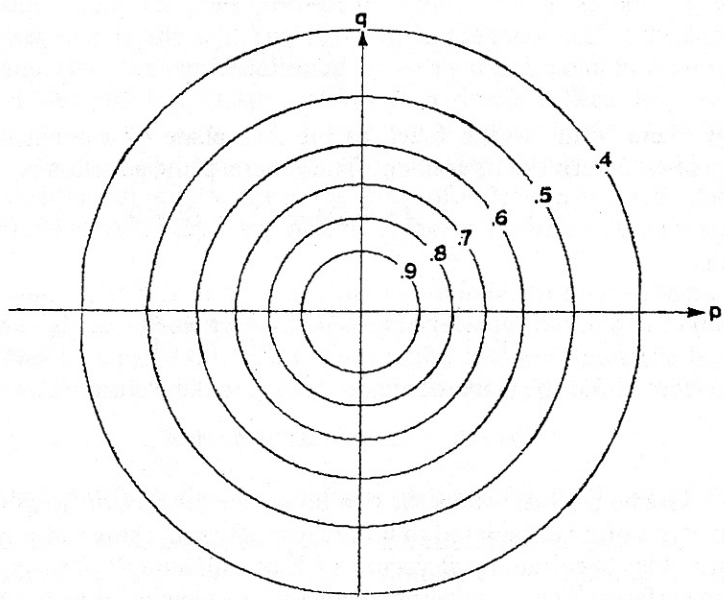

Fig. 3.28 Contours of constant radiance in gradient space for Lambertian surfaces; single light source near the viewpoint.

where r_o is a proportionality constant, and we conventionally use R to denote radiance in a viewer-centered frame. Let $\mathbf{n}_s$ and $\mathbf{n}$ be unit vectors in the source and surface normal directions. Since $\cos i = \mathbf{n}_s \cdot \mathbf{n}$

$$R = \frac{r_0}{(1 + p^2 + q^2)^{1/2}} \tag{3.41}$$

Thus $\cos(i)$ determines the image brightness, and so a plot of it is the gradient space image (Figs. 3.29 and 3.30).

For a more general light position, the mathematics is the same; if the light source is in the $(p_s, q_s, -1)$ direction, take the dot product of this direction and the surface normal.

$$R = r_o \mathbf{n} \cdot \mathbf{n}_s \tag{3.42}$$

Or, in other words,

$$R = \frac{r_o(p_s p + q_s q + 1)}{[(1 + p^2 + q^2)(1 + p_s^2 + q_s^2)]^{1/2}}$$

The phase angle g is constant throughout gradient space with orthographic projection (viewer distant from scene) and light source distant from scene.

Setting R constant to obtain contour lines gives a second-order equation, producing conic sections. In fact, the contours are produced by a set of cones of varying angles, whose axis is in the direction of the light source, intersecting a plane at unit distance from the origin. The resulting contours appear in Fig. 3.29. Here the dark line is the terminator, and represents all those planes that are edge-

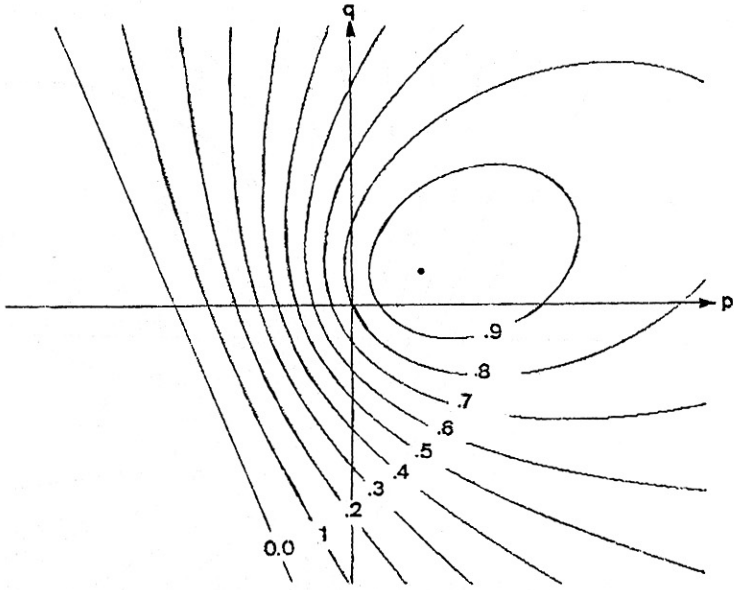

Fig. 3.29 Contours of constant radiance in gradient space. Lambertian surfaces; light not near viewpoint.

on to the light source; gradients on the back side of the terminator represent self-shadowed surfaces (facing away from the light). One intensity determines a contour and so gives a cone whose tangent planes all have that emittance. For a surface with specularity, contours of constant $I(i, e, g)$ could appear as in Fig. 3.30.

The point of specularity is between the matte component maximum brightness gradient and the origin. The brightest matte surface normal points at the light source and the origin points at the viewer. Pure specular reflection can occur if the vector tilts halfway toward the viewer maintaining the direction of tilt. Thus its gradient is on a line between the origin and the light-source direction gradient point.

3.5.3 Photometric Stereo

The reflectance equation (3.42) constrains the possible surface orientation to a locus on the reflectance map. Multiple light-source positions can determine the orientation uniquely [Woodham 1978]. Each separate light position gives a separate value for the intensity (proportional to radiance) at each point $f(\mathbf{x})$. If the surface reflectance r_o is unknown, three equations are needed to determine the reflectance together with the unit normal $\boldsymbol{n}$. If each source position vector is denoted by n_k, $k = 1, \ldots, 3$, the following equations result:

$$I_k(x, y) = r_o(\mathbf{n}_k \cdot \mathbf{n}), \qquad k = 1, \ldots, 3 \tag{3.43}$$

where I is normalized intensity. In matrix form

$$\mathbf{I} = r_o N \mathbf{n} \tag{3.44}$$

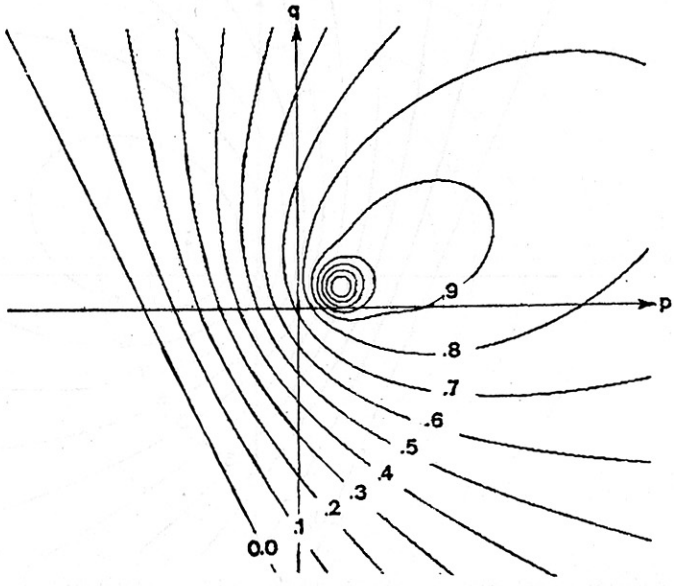

Fig. 3.30 Contours of constant radiance for a specular/matte surface.

where

$$\mathbf{I} = [I_1(x, y), I_2(x, y), I_3(x, y)]^T,$$

and

$$N = \begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{bmatrix} \tag{3.45}$$

and $I = fc$ where c is the appropriate normalization constant. If c is not known, it can be regarded as being part of r_o without affecting the normal direction calculation. As long as the three source positions $\mathbf{n}_1$, $\mathbf{n}_2$, $\mathbf{n}_3$ are not coplanar, the matrix N will have an inverse. Then solve for r_o and $\mathbf{n}$ by using (3.44), first using the fact that $\mathbf{n}$ is a unit vector to derive

$$r_0 = |N^{-1}\mathbf{I}| \tag{3.46}$$

and then solving for $\boldsymbol{n}$ to obtain

$$\mathbf{n} = \frac{1}{r_o} N^{-1}\mathbf{I} \tag{3.47}$$

Examples of a particular solution are shown in Fig. 3.31. Of course, a prerequisite for using this method is that the surface point not be in shadows for any of the sources.

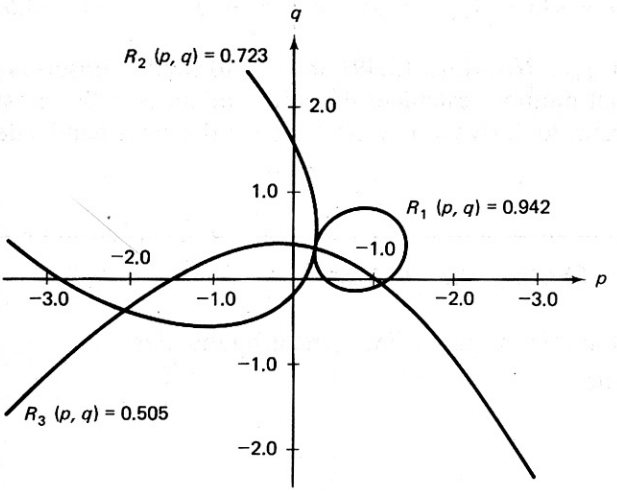

Fig. 3.31 A particular solution for photometric stereo.

3.5.4 Shape from Shading by Relaxation

Combining local information allows improved estimates for edges (Section 3.3.5) and for disparity (Section 3.4.2). In a similar manner local information can help in computing surface orientation [Ikeuchi 1980]. Basically, the reflectance equation

provides one constraint on the surface orientation and another is provided by the heuristic requirement that the surface be smooth.

Suppose there is an estimate of the surface normal at a point $(p(x, y), q(x, y))$. If the normal is not accurate, the reflectivity equation $I(x, y) = R(p, q)$ will not hold. Thus it seems reasonable to seek p and q that minimize $(I - R)^2$. The other requirement is that $p(x, y)$ and $q(x, y)$ be smooth, and this can be measured by their Laplacians $\nabla^2 p$ and $\nabla^2 q$. For a smooth curve both of these terms should be small. The goal is to minimize the error at a point,

$$E(x, y) = [I(x, y) - R(p, q)]^2 + \lambda[(\nabla^2 p)^2 + (\nabla^2 q)^2] \tag{3.48}$$

where the Lagrange multiplier λ [Russell 1976] incorporates the smoothness constraint. Differentiating $E(x, y)$ with respect to p and q and approximating derivatives numerically gives the following equations for $p(x, y)$ and $q(x, y)$:

$$p(x, y) = p_{av}(x, y) + T(x, y, p, q)\frac{\partial R}{\partial q} \tag{3.49}$$

$$q(x, y) = q_{av}(x, y) + T(x, y, p, q)\frac{\partial R}{\partial q} \tag{3.50}$$

where

$$T(x, y, p, q) = (1/\lambda)[I(x, y) - R(p, q)]$$

using

$$p_{av}(x, y) = \frac{1}{4}[p(x + 1, y) + p(x - 1, y) + p(x, y + 1) + p(x, y - 1)] \tag{3.51}$$

and a similar expression for q_{av}. Now Eqs. (3.49) and (3.50) lend themselves to solution by the Gauss-Seidel method: calculate the left-hand sides with an estimate for p and q and use them to derive a new estimate for the right-hand sides. More formally,

---

Algorithm 3.3: Shape from Shading [Ikeuchi 1980].

Step 0. k = 0. Pick an initial $p^0(x, y)$ and $q^0(x, y)$ near boundaries.

Step 1. k = k + 1; compute

$$p^k = p_{av}^{k-1} + T\frac{\partial R}{\partial p}$$

$$q^k = q_{av}^{k-1} + T\frac{\partial R}{\partial p}$$

Step 2. If the sum of all the E's is sufficiently small, stop. Else, go to step 1.

A loose end in this algorithm is that boundary conditions must be specified. These are values of p and q determined a priori that remain constant throughout each iteration. The simplest place to specify a surface gradient is at an occluding contour (see Fig. 3.32) where the gradient is nearly 90° to the line of sight. Unfortunately, p and q are infinite at these points. Ikeuchi's elegant solution to this is to use a different coordinate system for gradient space, that of a Gaussian sphere (Appendix 1). In this system, the surface normal is described relative to where it intersects the sphere if the tail of the normal is at the sphere's origin. This is the point at which a plane perpendicular to the normal would touch the sphere if translated toward it (Fig. 3.32b).

In this system the radiance may be described in terms of the spherical coordinates θ, ϕ. For a Lambertian surface

$$R(\theta, \phi) = \cos\theta \cos\theta_s + \sin\theta \sin\theta_s \cos(\phi - \phi_s) \tag{3.52}$$

At an occluding contour $\phi = \pi/2$ and θ is given by $\tan^{-1}(\partial y / \partial x)$, where the derivatives are calculated at the occluding contour (Fig. 3.32c).

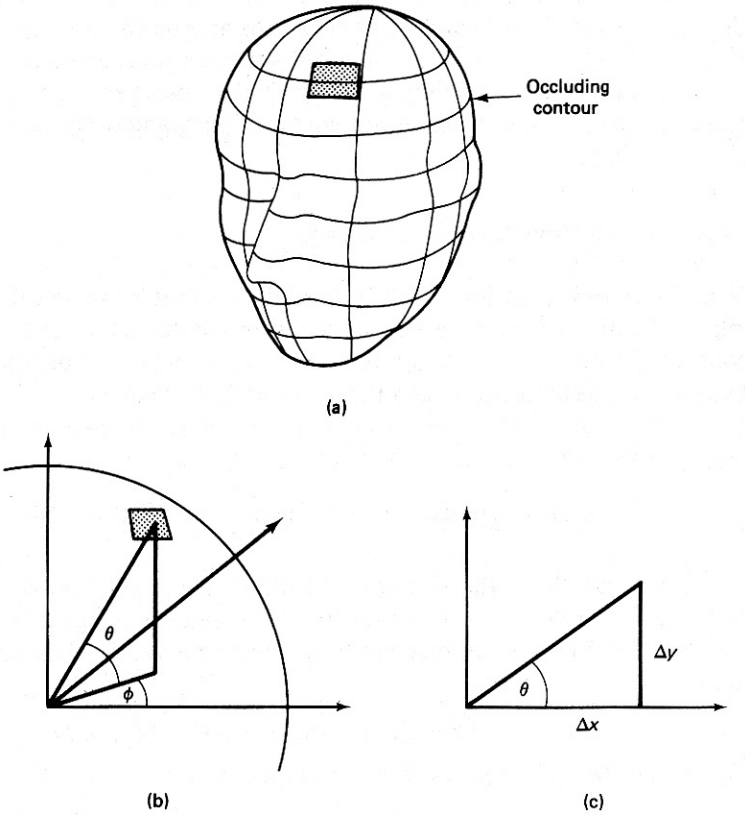

Fig. 3.32 (a) Occluding contour. (b) Gaussian sphere. (c) Calculating θ from occluding contour.

To use the (θ, ϕ) formulation instead of the (p, q) formulation is an easy matter. Simply substitute θ for p and ϕ for q in all instances of the formula in Algorithm 3.3.

3.6 OPTICAL FLOW

Much of the work on computer analysis of visual motion assumes a stationary observer and a stationary background. In contrast, biological systems typically move relatively continuously through the world, and the image projected on their retinas varies essentially continuously while they move. Human beings perceive smooth continuous motion as such.

Although biological visual systems are discrete, this quantization is so fine that it is capable of producing essentially continuous outputs. These outputs can mirror the continuous flow of the imaged world across the retina. Such continuous information is called *optical flow*. Postulating optical flow as an input to a perceptual system leads to interesting methods of motion perception.

The optical flow, or instantaneous velocity field, assigns to every point on the visual field a two-dimensional "retinal velocity" at which it is moving across the visual field. This section describes how approximations to instantaneous flow may be computed from the usual input situation in a sequence of discrete images. Methods of using optical flow to compute the observer's motion, a relative depth map, surface normals of his or her surroundings, and other useful information are given in Chapter 7.

3.6.1 The Fundamental Flow Constraint

One of the important features of optical flow is that it can be calculated simply, using local information. One way of doing this is to model the motion image by a continuous variation of image intensity as a function of position and time, then expand the intensity function $f(x, y, t)$ in a Taylor series.

$$f(x + dx, y + dy, t + dt) = \tag{3.53}$$
$$f(x, y, t) + \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial t} dt + \text{higher-order terms}$$

As usual, the higher-order terms are henceforth ignored. The crucial observation to be exploited is the following: If indeed the image at some time $t + dt$ is the result of the original image at time t being moved translationally by dx and dy, then in fact

$$f(x + dx, y + dy, t + dt) = f(x, y, t) \tag{3.54}$$

Consequently, from Eqs. (3.53) and (3.54),

$$-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} \tag{3.55}$$