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),

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

Now $\frac{\partial f}{\partial t}$, $\frac{\partial f}{\partial x}$, and $\frac{\partial f}{\partial y}$ are all measurable quantities, and $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are estimates of what we are looking for—the velocity in the x and y directions. Writing

$$\frac{dx}{dt} = u, \qquad \frac{dy}{dt} = v$$

gives

$$-\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}u + \frac{\partial f}{\partial y}v \tag{3.56}$$

or equivalently,

$$-\frac{\partial f}{\partial t} = \nabla f \cdot \mathbf{u} \tag{3.57}$$

where ∇f is the spatial gradient of the image and $\mathbf{u} = (u, v)$ the velocity.

The implications of (3.57) are interesting. Consider a fixed camera with a scene moving past it. The equations say that the *time* rate of change in intensity of a point in the image is (to first order) explained as the *spatial* rate of change in the intensity of the scene multiplied by the *velocity* that points of the scene move past the camera.

This equation also indicates that the velocity (u, v) must lie on a line perpendicular to the vector (f_x, f_y) where f_x and f_y are the partial derivatives with respect to x and y, respectively (Fig. 3.33). In fact, if the partial derivatives are very accurate the magnitude component of the velocity in the direction (f_x, f_y) is (from 3.57):

$$\frac{-f_t}{[(f_x^2 + f_y^2)]^{1/2}}$$

3.6.2 Calculating Optical Flow by Relaxation

Equation (3.57) constrains the velocity but does not determine it uniquely. The development of Section 3.5.4 motivates the search for a solution that satisfies Eq.

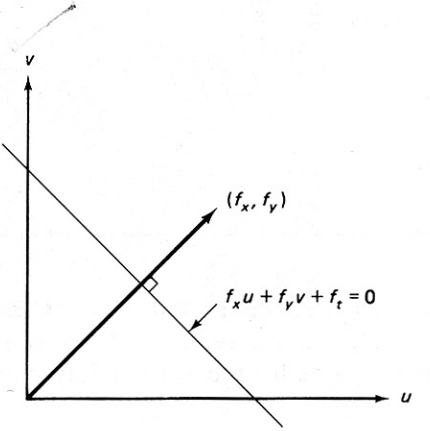

Fig. 3.33 Relation between (u, v) and (f_x, f_y).

(3.57) as closely as possible and also is locally smooth [Horn and Schunck 1980]. In this case as well, the Laplacians of the two velocity components, $\nabla^2 u$ and $\nabla^2 v$, can measure local smoothness.

Again using the method of Lagrange multipliers, minimize the flow error

$$E^2(x, y) = (f_x u + f_y v + f_t)^2 + \lambda^2[(\nabla^2 u)^2 + (\nabla^2 v)^2] \tag{3.58}$$

Differentiating this equation with respect to u and v provides equations for the change in error with respect to u and v, which must be zero for a minimum. Writing $\nabla^2 u$ as $u - u_{av}$ and $\nabla^2 v$ as $v - v_{av}$, these equations are

$$(\lambda^2 + f_x^2)u + f_x f_y v = \lambda^2 u_{av} - f_x f_t \tag{3.59}$$

$$f_x f_y u + (\lambda^2 + f_y^2)v = \lambda^2 v_{av} - f_y f_t \tag{3.60}$$

These equations may be solved for u and v, yielding

$$u = u_{av} - f_x \frac{P}{D} \tag{3.61}$$

$$v = v_{av} - f_y \frac{P}{D} \tag{3.62}$$

where

$$P = f_x u_{av} + f_y v_{av} + f_t$$

$$D = \lambda^2 + f_x^2 + f_y^2$$

To turn this into an iterative equation for solving $u(x, y)$ and $v(x, y)$, again use the Gauss-Seidel method.

Algorithm 3.4: Optical Flow [Horn and Schunck 1980].

$k = 0$.
Initialize all u^k and v^k to zero.
Until some error measure is satisfied, do

$$u^k = u_{av}^{k-1} - f_x \frac{P}{D}$$

$$v^k = v_{av}^{k-1} - f_y \frac{P}{D}$$

As Horn and Schunck demonstrate, this method derives the flow for two time frames, but it can be improved by using several time frames and using the final solution after one iteration at one time for the initial solution at the following time frame. That is:

Algorithm 3.5: Multiframe Optical Flow.

$t = 0$.
Initialize all $u(x, y, 0)$, $v(x, y, 0)$
for $t = 1$ *until* maxframes *do*

$$u(x, y, t) = u_{av}(x, y, t-1) - f_x \frac{P}{D}$$

$$v(x, y, t) = v_{av}(x, y, t-1) - f_y \frac{P}{D}$$

The results of using synthetic data from a rotating checkered sphere are shown in Fig. 3.34.

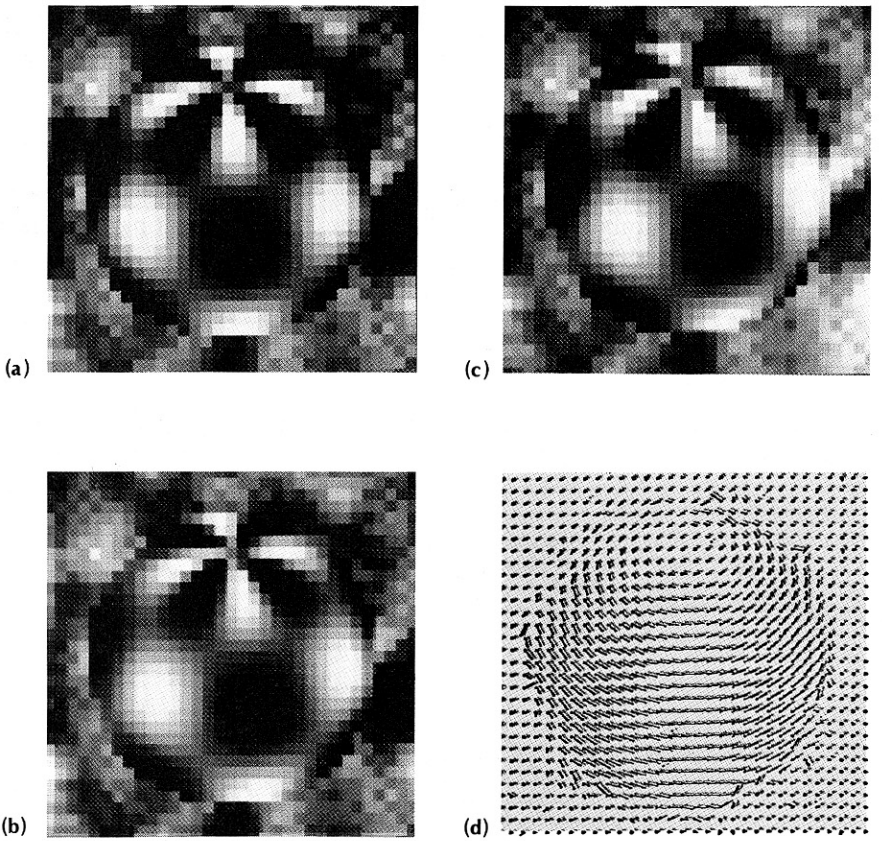

Fig. 3.34 Optical flow results. (a), (b) and (c) are three frames from the rotating sphere, (d) is the derived three-dimensional flow after 32 such time frames.

What is the best spatial resolution for an image? The sampling theorem states that the maximum spatial frequency in the image data must be less than half the sampling frequency in order that the sampled image represent the original unambiguously. However, the sampling theorem is not a good predictor of how easily objects can be recognized by computer programs. Often objects can be more easily recognized in images that have a very low sampling rate. There are two reasons for this. First, the computations are fewer because of the reduction in dimensionality. Second, confusing detail present in the high-resolution versions of the images may not appear at the reduced resolution. But even though some objects are more easily found at low resolutions, usually an object description needs detail only revealed at the higher resolutions. This leads naturally to the notion of a *pyramidal* image data structure in which the search for objects is begun at a low resolution, and refined at ever-increasing resolutions until one reaches the highest resolution of interest. Figure 3.35 shows the correspondence between pixels for the pyramidal structure.

In the next three sections, pyramids are applied to gray-level images and edge images. Pyramids, however, are a very general tool and can be used to represent any image at varying levels of detail.

3.7.1 Gray-level Consolidation

In some applications, redigitizing the image with a different sampling rate is a way to reduce the number of samples. However, most digitizer parameters are difficult to change, so that often computational means of reduction are needed. A straightforward method is to partition the digitized image into nonoverlapping

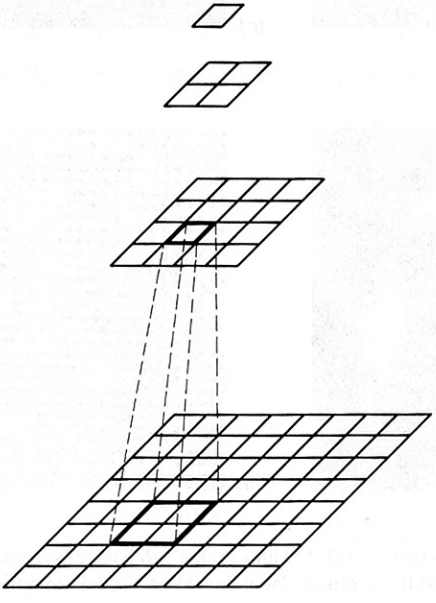

Fig. 3.35 Pyramidal image structure.

neighborhoods of equal size and shape and to replace each of those neighborhoods by the average pixel densities in that neighborhood. This operation is *consolidation.* For an $n \times n$ neighborhood, consolidation is equivalent to averaging the original image over the neighborhood followed by sampling at intervals n units apart.

Consolidation tends to offset the aliasing that would be introduced by sampling the sensed data at a reduced rate. This is due to the effects of the averaging step in the consolidation process. For the one-dimensional case where

$$f'(x) = \frac{1}{2}[f(x) + f(x + \Delta)] \tag{3.63}$$

the corresponding Fourier transform [Steiglitz 1974] is

$$H(u) = \frac{1}{2}\left[1 + e^{-j2\pi\Delta}\right]F(u) \tag{3.64}$$

which has magnitude $|H(u)| = \cos[\pi(u/u_0)]$ and phase $-\pi(u/u_0)$. The sampling frequency $u_0 = 1/\Delta$ where Δ is the spacing between samples. Thus the averaging step has the effect of attenuating the higher frequencies of $F(u)$ as shown in Fig. 3.36. Since the higher frequencies are involved in aliasing, attenuating these frequencies reduces the aliasing effects.

3.7.2 Pyramidal Structures in Correlation

With correlation matching, the use of multiple resolution techniques can sometimes provide significant functional and computational advantages [Moravec 1977]. Binary search correlation uses pyramids of the input image and reference

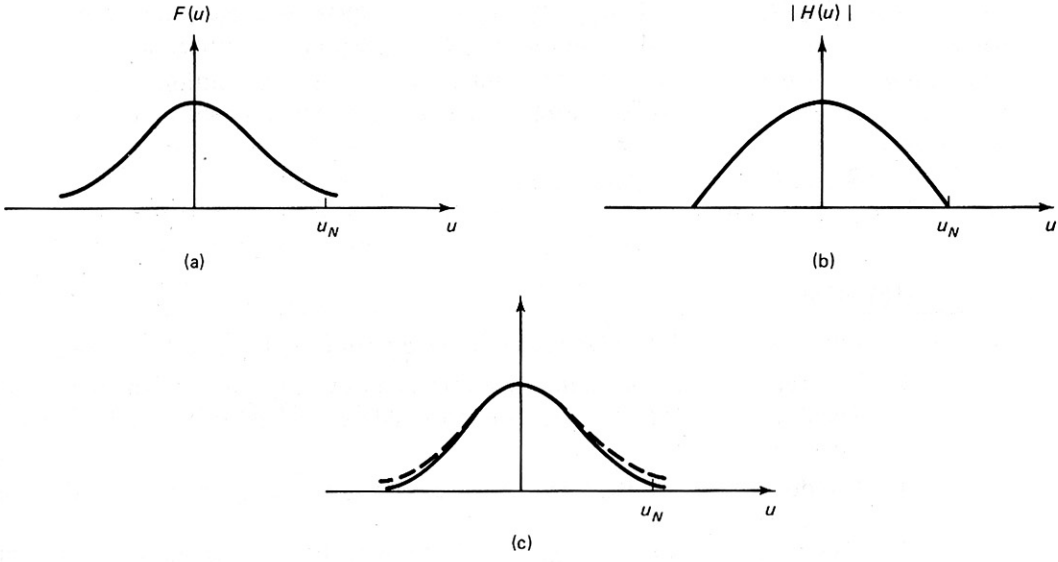

Fig. 3.36 Consolidation effects viewed in the spatial frequency domain. (a) Original transform. (b) Transform of averaging operator. (c) Transform of averaged image.

patterns. The algorithm partakes of the computational efficiency of binary (as opposed to linear) search [Knuth 1973]. Further, the low-resolution correlation operations at high levels in the pyramid ensure that the earlier correlations are on gross image features rather than details.

In binary search correlation a feature to be located is at some unknown location in the input image. The reference version of the feature originates in another image, the reference image. The feature in the reference image is contained in a window of $n \times n$ pixels. The task of the correlator is to find an $n \times n$ window in the input image that best matches the reference image window containing the feature. The details of the correlation processes are given in the following algorithm.

Algorithm 3.6: Binary Search Correlation Control Algorithm

Definitions

OrigReference: an $N \times N$ image containing a feature centered at (FeatureX, FeatureY).

OrigInput: an $M \times M$ array in which an instance of the Feature is to be located. For simplicity, assume that it is at the same resolution as OrigReference.

n: a window size; an $n \times n$ window in OrigReference is large enough to contain the Feature.

Window: an $n \times n$ array containing a varying-resolution subimage of OrigReference centered on the Feature.

Input: a $2n \times 2n$ array containing a varying-resolution subimage of OrigInput, centered on the best match for the Feature.

Reference: a temporary array.

Algorithm

1. Input := Consolidate OrigInput by a factor of $2n/M$ to size $2n \times 2n$.
2. Reference := Consolidate OrigReference by the same factor $2n/M$ to size $2nN/M \times 2nN/M$. This consolidation takes the Feature to a new (Feature$X$, Feature$Y$).
3. Window := $n \times n$ window from Reference centered on the new (Feature$X$, Feature$Y$).
4. Calculate the match metric of the window at the $(n + 1)^2$ locations in Input at which it is wholly contained. Say that the best match occurs at (BestMatch$X$, BestMatch$Y$) in Input.

5. Input := $n \times n$ window from Input centered at (BestMatch$X$, BestMatch$Y$), enlarged by a factor of 2.
6. Reference := Reference enlarged by a factor of 2. This takes Feature to a new (Feature$X$, Feature$Y$).
7. *Go to* 3.

Through time, the algorithm uses a reference image for matching that is always centered on the feature to be matched, but that homes in on the feature by being increased in resolution and thus reduced in linear image coverage by a factor of 2 each time. In the input image, a similar homing-in is going on, but the search area is usually twice the linear dimension of the reference window. Further, the center of the search area varies in the input image as the improved resolution refines the point of best match.

Binary search correlation is for matching features with context. The template at low resolution possibly corresponds to much of the area around the feature, while the feature may be so small in the initial consolidated images as to be invisible. The coarse-to-fine strategy is perfect for such conditions, since it allows gross features to be matched first and to guide the later high-resolution search for best match. Such matching with context is less useful for locating several instances of a shape dotted at random around an image.

3.7.3 Pyramidal Structures in Edge Detection

As an example of the use of pyramidal structures in processing, consider the use of such structures in edge detection. This application, after [Tanimoto and Pavlidis 1975], uses two pyramids, one to store the image and another to store the image edges. The idea of the algorithm is that a neighborhood in the low-resolution image where the gray-level values are the same is taken to imply that in fact there is no gray-level change (edge) in the neighborhood. Of course, the low-resolution levels in the pyramid tend to blur the image and thus attenuate the gray-level changes that denote edges. Thus the starting level in the pyramid must be picked judiciously to ensure that the important edges are detected.

Algorithm 3.7: Hierarchical Edge Detection

```
recursive procedure refine (k, x, y)
  begin
  if k < MaxLevel then
    for dx = 0 until 1 do
    for dy = 0 until 1 do
      if EdgeOp (k, x + dx, y + dy) > Threshold(x)
      then refine (k + 1, x + dx, y + dy)
  end;
```

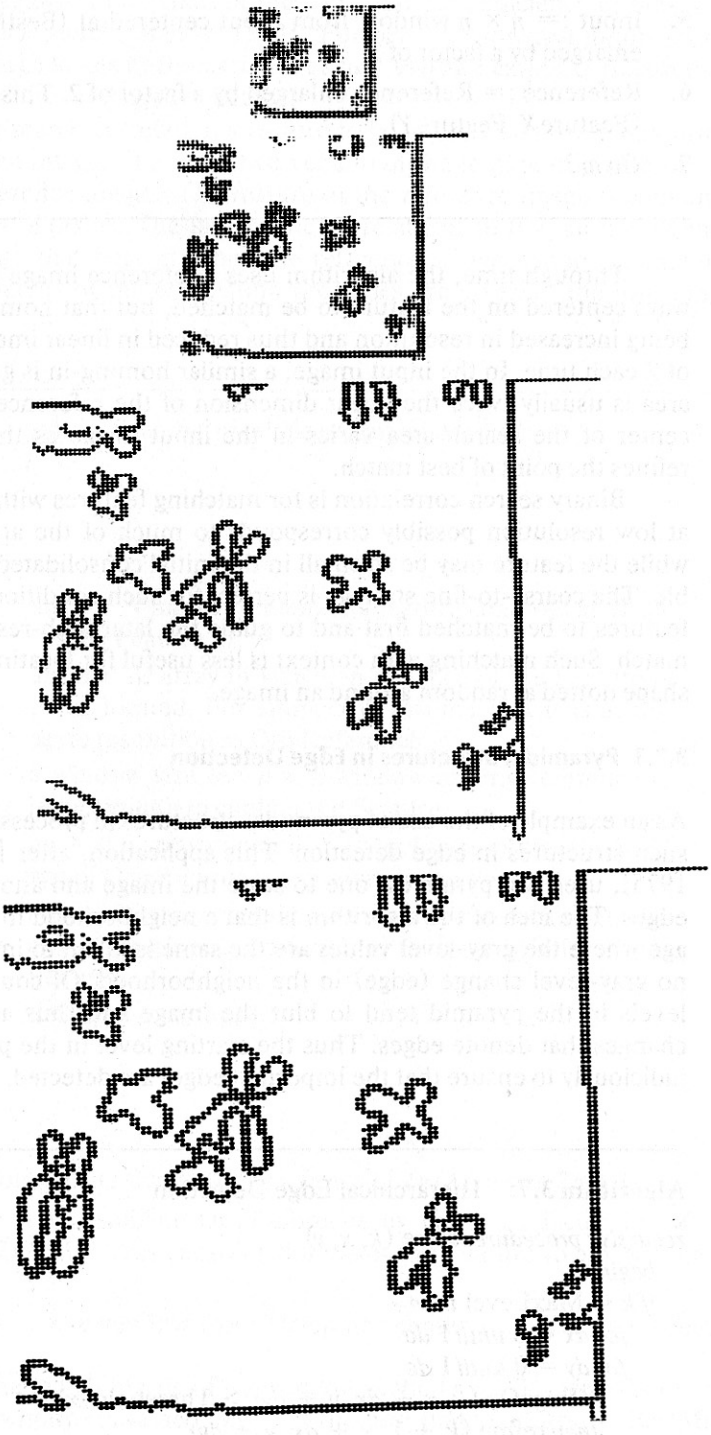

Fig. 3.37 Pyramidal edge detection.

```
procedure FindEdges:
  begin
  comment apply operator to every pixel in the
      starting level s, refining where necessary;
    for x := 0 until 2^S − 1 do
    for y := 0 until 2^S − 1 do
      if EdgeOp (s, x, y) > Threshold(s)
      then refine (s, x, y);
  end;
```

Figure 3.37 shows Tanimoto's results for a chromosome image. The table inset shows the computational advantage in terms of the calls to the edge operator as a function of the starting level s.

Similar kinds of edge detection strategies based on pyramids have been pursued by [Levine 1978; Hanson and Riseman 1978]. The latter effort is a little different in that processing within the pyramid is bidirectional; information from edges detected at a high-resolution level is projected to low-resolution levels of the pyramid.

EXERCISES

3.1 Derive an analytical expression for the response of the Sobel operator to a vertical step edge as a function of the distance of the edge to the center of the operator.

3.2 Use the formulas of Eqs. (3.31) to derive the digital template function for g_1 in a 5^3 pixel domain.

3.3 Specify a version of Algorithm 3.1 that uses the gradient edge operator instead of the "crack" edge operator.

3.4 In photometric stereo, three or more light source positions are used to determine a surface orientation. The dual of this problem uses surface orientations to determine light source position. What is the usefulness of the latter formulation? In particular, how does it relate to Algorithm 3.3?

3.5 Using any one of Algorithms 3.1 through 3.4 as an example, show how it could be modified to use pyramidal data structures.

3.6 Write a reflectance function to capture the "grazing incidence" phenomenon—surfaces become more mirror-like at small angles of incidence (and reflectance).

3.7 Equations 3.49 and 3.50 were derived by minimizing the local error. Show how these equations are modified when total error [i.e., $\sum_{x,y} E(x, y)$] is minimized.

REFERENCES

ABDOU, I. E. "Quantitative methods of edge detection." USCIPI Report 830, Image Processing Institute, Univ. Southern California, July 1978.

AKATSUKA, T., T. ISOBE, and O. TAKATANI. "Feature extraction of stomach radiograph." *Proc.*, 2nd IJCPR, August 1974, 324-328.

ANDREWS, H. C. and B. R. HUNT. *Digital Image Restoration.* Englewood Cliffs, NJ: Prentice-Hall, 1977.

ATTNEAVE, F. "Some informational aspects of visual perception." *Psychological Review 61*, 1954.

BARROW, H. G. and J. M. TENENBAUM. "Computational Vision." *Proc. IEEE 69*, 5, May 1981, 572-595

BARROW, H. G. and J. M. TENENBAUM. "Recovering intrinsic scene characteristics from images." Technical Note 157, AI Center, SRI International, April 1978.

BINFORD, T. O. "Visual perception by computer." *Proc.*, IEEE Conf. on Systems and Control, Miami, December 1971.

BLINN, J. E. "Computer display of curved surfaces." Ph.D. dissertation, Computer Science Dept., Univ. Utah, 1978.

FREI, W. and C. C. CHEN. "Fast boundary detection: a generalization and a new algorithm." *IEEE Trans. Computers 26*, 2, October 1977, 988-998.

GONZALEZ, R. C. and P. WINTZ. *Digital Image Processing.* Reading, MA: Addison-Wesley, 1977.

GRIFFITH, A. K. "Edge detection in simple scenes using a priori information." *IEEE Trans. Computers 22*, 4, April 1973.

HANSON, A. R. and E. M. RISEMAN (Eds.). *Computer Vision Systems (CVS).* New York: Academic Press, 1978.

HORN, B. K. P. "Determining lightness from an image." *CGIP 3*, 4, December 1974, 277-299.

HORN, B. K. P. "Shape from shading." In *PCV*, 1975.

HORN, B. K. P. and B. G. SCHUNCK. "Determining optical flow." AI Memo 572, AI Lab, MIT, April 1980.

HORN, B. K. P. and R. W. SJOBERG. "Calculating the reflectance map." *Proc.*, DARPA IU Workshop, November 1978, 115-126.

HUBEL, D. H. and T. N. WIESEL. "Brain mechanisms of vision." *Scientific American*, September 1979, 150-162.

HUECKEL, M. "An operator which locates edges in digitized pictures." *J. ACM 18*, 1, January 1971, 113-125.

HUECKEL, M. "A local visual operator which recognizes edges and lines." *J. ACM 20*, 4, October 1973, 634-647.

IKEUCHI, K. "Numerical shape from shading and occluding contours in a single view." AI Memo 566, AI Lab, MIT, revised February 1980.

KIRSCH, R. A. "Computer determination of the constituent structure of biological images." *Computers and Biomedical Research 4*, 3, June 1971, 315-328.

KNUTH, D. E. *The Art of Computer Programming.* Reading, MA: Addison-Wesley, 1973.

LEVINE, M. D. "A knowledge-based computer vision system." In *CVS*, 1978.

LIU, H. K. "Two- and three-dimensional boundary detection." *CGIP 6*, 2, 1977, 123-134.

MARR, D. and T. POGGIO. "Cooperative computation of stereo disparity." *Science 194*, 1976, 283-287.

MARR, D. and T. POGGIO. "A theory of human stereo vision." AI Memo 451, AI Lab, MIT, November 1977.

MERO, L. and Z. VASSY. "A simplified and fast version of the Hueckel operator for finding optimal edges in pictures." *Proc.*, 4th IJCAI, September 1975, 650-655.

MORAVEC, H. P. "Towards automatic visual obstacle avoidance." *Proc.*, 5th IJCAI, August 1977, 584.

NEVATIA, R. "Evaluation of a simplified Hueckel edge-line detector." Note, *CGIP 6*, 6, December 1977, 582-588.

PHONG, B-T. "Illumination for computer generated pictures." *Commun. ACM 18*, 6, June 1975, 311-317.

PINGLE, K. K. and J. M. TENENBAUM. "An accommodating edge follower." *Proc.*, 2nd IJCAI, September 1971, 1-7.

PRAGER, J. M. "Extracting and labeling boundary segments in natural scenes." *IEEE Trans. PAMI 2*, 1, January 1980, 16-27.

PRATT, W. K. *Digital Image Processing.* New York: Wiley-Interscience, 1978.

PREWITT, J. M. S. "Object enhancement and extraction." In *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld (Eds.). New York: Academic Press, 1970.

QUAM, L. and M. J. HANNAH. "Stanford automated photogrammetry research." AIM-254, Stanford AI Lab, November 1974.

ROBERTS, L. G. "Machine perception of three-dimensional solids." In *Optical and Electro-optical Information Processing*, J. P. Tippett et al. (Eds.). Cambridge, MA: MIT Press, 1965.

ROSENFELD, A. and A. C. KAK. *Digital Picture Processing.* New York: Academic Press, 1976.

ROSENFELD, A., R. A. HUMMEL, and S. W. ZUCKER. "Scene labelling by relaxation operations." *IEEE Trans. SMC 6*, 1976, 430.

RUSSELL, D. L. (Ed.). *Calculus of Variations and Control Theory.* New York: Academic Press, 1976.

SHAPIRA, R. "A technique for the reconstruction of a straight-edge, wire-frame object from two or more central projections." *CGIP 3*, 4, December 1974, 318-326.

SHIRAI, V. "Analyzing intensity arrays using knowledge about scenes." In *PCV*, 1975.

STEIGLITZ, K. *An Introduction to Discrete Systems.* New York: Wiley, 1974.

STOCKHAM, T. J., Jr. "Image processing in the context of a visual model." *Proc. IEEE 60*, 7, July 1972, 828-842.

TANIMOTO, S. and T. PAVLIDIS. "A hierarchical data structure for picture processing." *CGIP 4*, 2, June 1975, 104-119.

TRETIAK, O. J. "A parameteric model for edge detection." *Proc.*, 3rd COMPSAC, November 1979, 884-887.

TURNER, K. J. "Computer perception of curved objects using a television camera." Ph.D. dissertation, Univ. Edinburgh, 1974.

WECHSLER, H. and J. SKLANSKY. "Finding the rib cage in chest radiographs." *Pattern Recognition 9*, 1977, 21-30.

WHITTED, T. "An improved illumination model for shaded display." *Comm. ACM 23*, 6, June 1980, 343-349.

WOODHAM, R. J. "Photometric stereo: A reflectance map technique for determining surface orientation from image intensity." *Proc.*, 22nd International Symp., Society of Photo-optical Instrumentation Engineers, San Diego, CA, August 1978, 136-143.

ZUCKER, S. W. and R. A. HUMMEL. "An optimal three-dimensional edge operator." Report 79-10, McGill Univ., April 1979.

ZUCKER, S. W., R. A. HUMMEL, and A. ROSENFELD. "An application of relaxation labeling to line and curve enhancement." *IEEE Trans. Computers 26*, 1977. April No4 pp394