where

$$V_t = \{0, 1\}$$

$$V_n = \{a, b\}$$

$$R: \begin{matrix} a & b & b & b \\ b & b & b & b \\ b & b & b & b \end{matrix} \rightarrow \begin{matrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{matrix}$$

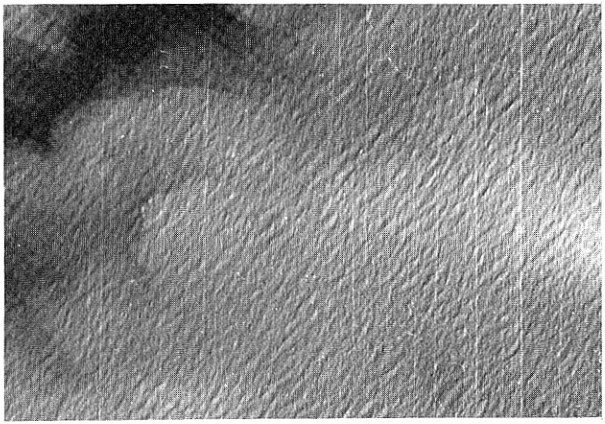
6.4 TEXTURE AS A PATTERN RECOGNITION PROBLEM

Many textures do not have the nice geometrical regularity of "reptile" or "wire braid"; instead, they exhibit variations that are not satisfactorily described by shapes, but are best described by statistical models. *Statistical pattern recognition* is a paradigm that can classify statistical variations in patterns. (There are other statistical methods of describing texture [Pratt et al. 1981], but we will focus on statistical pattern recognition since it is the most widely used for computer vision purposes.) There is a voluminous literature on pattern recognition, including several excellent texts (e.g., [Fu 1968; Tou and Gonzalez 1974; Fukunaga 1972], and the ideas have much wider application than their use here, but they seem particularly appropriate for low-resolution textures, such as those seen in aerial images [Weszka et al. 1976]. The pattern recognition approach to the problem is to classify instances of a texture in an image into a set of classes. For example, given the textures in Fig. 6.15, the choice might be between the classes "orchard," "field," "residential," "water."

The basic notion of pattern recognition is the *feature vector.* The feature vector $\mathbf{v}$ is a set of measurements $\{v_1 \cdots v_m\}$ which is supposed to condense the description of relevant properties of the textured image into a small, Euclidean *feature space* of m dimensions. Each point in feature space represents a value for the feature vector applied to a different image (or subimage) of texture. The measurement values for a feature should be correlated with its class membership. Figure 6.16 shows a two-dimensional space in which the features exhibit the desired correlation property. Feature vector values cluster according to the texture from which they were derived. Figure 6.16 shows a bad choice of features (measurements) which does not separate the different classes.

The pattern recognition paradigm divides the problem into two phases: training and test. Usually, during a training phase, feature vectors from known samples are used to partition feature space into regions representing the different classes. However, self teaching can be done; the classifier derives its own partitions. Feature selection can be based on parametric or nonparametric models of the distributions of points in feature space. In the former case, analytic solutions are sometimes available. In the latter, feature vectors are *clustered* into groups which are taken to indicate partitions. During a test phase the feature-space partitions are used to classify feature vectors from unknown samples. Figure 6.17 shows this process.

Given that the data are reasonably well behaved, there are many methods for clustering feature vectors [Fukunaga 1972; Tou and Gonzales 1974; Fu 1974].



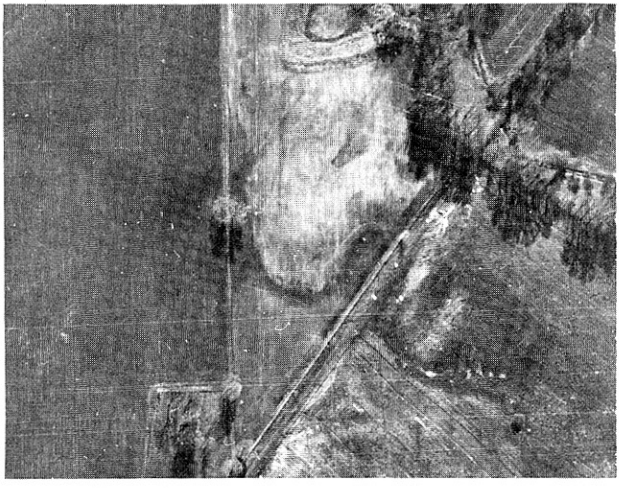


Fig. 6.15 Aerial image textures for discrimination.

Fig. 6.15 (cont.)

One popular way of doing this is to use prototype points for each class and a *nearest-neighbor* rule [Cover 1968]:

assign $\mathbf{v}$ to class w_i if i minimizes
$\min_i d(\mathbf{v}, \mathbf{v}_{w_i})$

where $\mathbf{v}_{w_i}$ is the prototype point for class w_i.

Parametric techniques assume information about the feature vector probability distributions to find rules that maximize the likelihood of correct classification:

assign $\mathbf{v}$ to class w_i if i maximizes
$\max_i p(w_i|\mathbf{v})$

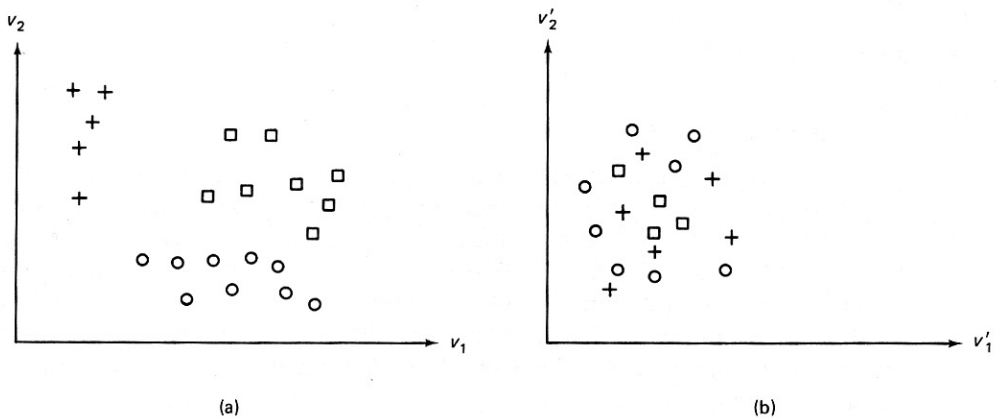


Fig. 6.16 Feature space for texture discrimination. (a) effective features (b) ineffective features.

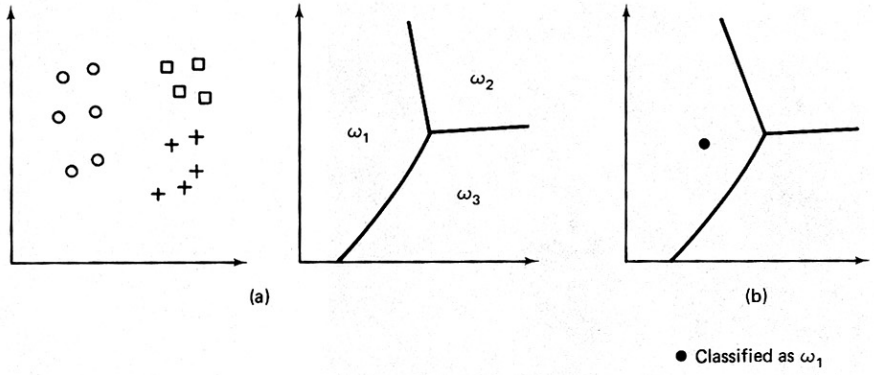


Fig. 6.17 Pattern recognition paradigm.

The distributions may also be used to formulate rules that minimize errors.

Picking good features is the essence of pattern recognition. No elaborate formalism will work well for bad features such as those of Fig. 6.15b. On the other hand, almost any method will work for very good features. For this reason, texture is a good domain for pattern recognition: it is fairly easy to define features that (1) cluster in feature space according to different classes, and (2) can separate texture classes.

The ensuing subsections describe features that have worked well. These subsections are in reverse order from those of Section 6.2 in that we begin with features defined on pixels—Fourier subspaces, gray-level dependencies—and conclude with features defined on higher-level texels such as regions. However, the lesson is the same as with the grammatical approach: hard work spent in obtaining high-level primitives can both improve and simplify the texture model. Space does not permit a discussion of many texture features; instead, we limit ourselves to a few representative samples. For further reading, see [Haralick 1978].

6.4.1 Texture Energy

Fourier Domain Basis

If a texture is at all spatially periodic or directional, its power spectrum will tend to have peaks for corresponding spatial frequencies. These peaks can form the basis of features of a pattern recognition discriminator. One way to define features is to search Fourier space directly [Bajcsy and Lieberman 1976]. Another is to partition Fourier space into bins. Two kinds of bins, radial and angular, are commonly used, as shown in Fig. 6.18. These bins, together with the Fourier power spectrum are used to define features. If F is the Fourier transform, the Fourier power spectrum is given by $|F|^2$.

Radial features are given by

$$v_{r_1 r_2} = \int\int |F(u, v)|^2 \, du \, dv \tag{6.5}$$

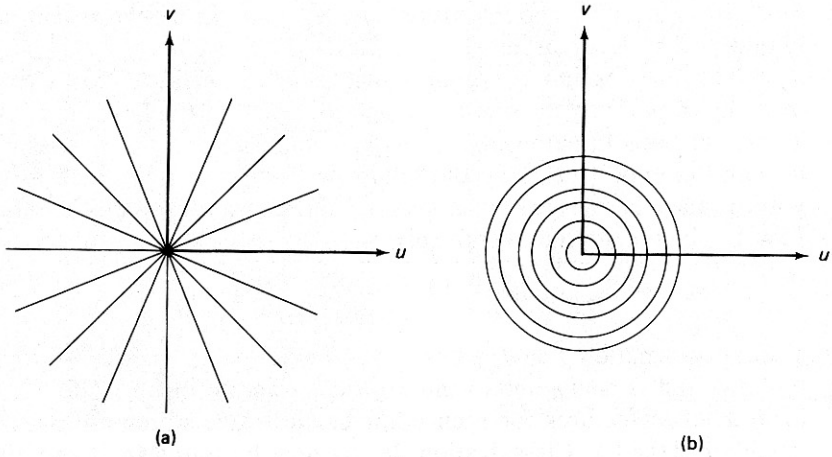


Fig. 6.18 Partitioning the Fourier domain into bins.

where the limits of integration are defined by

$$r_1^2 \leqslant u^2 + v^2 < r_2^2$$

$$0 \leqslant u, v < n-1$$

where $[r_1, r_2]$ is one of the radial bins and $\mathbf{v}$ is the vector (not related to v) defined by different values of r_1 and r_2. Radial features are correlated with texture coarseness. A smooth texture will have high values of $V_{r_1 r_2}$ for small radii, whereas a coarse, grainy texture will tend to have relatively higher values for larger radii.

Features that measure angular orientation are given by

$$\mathbf{v}_{\theta_1 \theta_2} = \int\int |F(u, v)|^2 \, du \, dv \tag{6.6}$$

where the limits of integration are defined by

$$\theta_1 \leqslant \tan^{-1}\left(\frac{v}{u}\right) < \theta_2$$

$$0 < u, v \leqslant n-1$$

where $[\theta_1, \theta_2)$ is one of the sectors and $\mathbf{v}$ is defined by different values of θ_1 and θ_2. These features exploit the sensitivity of the power spectrum to the directionality of the texture. If a texture has as many lines or edges in a given direction θ, $|F|^2$ will tend to have high values clustered around the direction in frequency space $\theta + \pi/2$.

Texture Energy in the Spatial Domain

From Section 2.2.4 we know that the Fourier approach could also be carried out in the image domain. This is the approach taken in [Laws 1980]. The advantage of this approach is that the basis is not the Fourier basis but a variant that is more

matched to intuition about texture features. Figure 6.19 shows the most important of Laws' 12 basis functions.

The image is first histogram-equalized (Section 3.2). Then 12 new images are made by convolving the original image with each of the basis functions (i.e., $f_k' = f * h_k$ for basis functions $h_1, \ldots, h_{12}$). Then each of these images is transformed into an "energy" image by the following transformation: Each pixel in the convolved image is replaced by an average of the absolute values in a local window of 15×15 pixels centered over the pixel:

$$f_k''(x, y) = \sum_{x', y' \text{ in window}} (|f_k'(x', y')|) \tag{6.7}$$

The transformation $f \rightarrow f_k''$, $k = 1, \ldots 12$ is termed a "texture energy transform" by Laws and is analogous to the Fourier power spectrum. The f_k'', $k = 1, \ldots 12$ form a set of features for each point in the image which are used in a nearest-neighbor classifier. Classification details may be found in [Laws 1980]. Our interest is in the particular choice of basis functions used.

Figure 6.20 shows a composite of natural textures [Brodatz 1966] used in Laws's experiments. Each texture is digitized into a 128×128 pixel subimage. The texture energy transforms were applied to this composite image and each pixel was classified into one of the eight categories. The average classification accuracy was about 87% for interior regions of the subimages. This is a very good result for textures that are similar.

6.4.2 Spatial Gray-Level Dependence

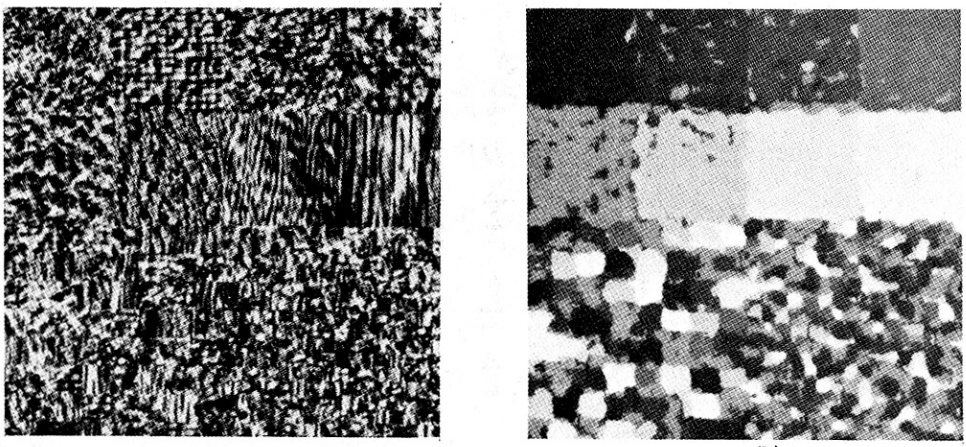
Spatial gray-level dependence (SGLD) matrices are one of the most popular sources of features [Kruger et al. 1974; Hall et al. 1971; Haralick et al. 1973]. The SGLD approach computes an intermediate matrix of measures from the digitized image data, and then defines features as functions on this intermediate matrix. Given an image f with a set of discrete gray levels I, we define for each of a set of discrete values of d and θ the intermediate matrix $S(d, \theta)$ as follows:

> $S(i, j|d, \theta)$, an entry in the matrix, is the number of times gray level i is oriented with respect to gray level j such that where
>
> $$f(\mathbf{x}) = i \quad \text{and} \quad f(\mathbf{y}) = j \quad \text{then}$$
>
> $$\mathbf{y} = \mathbf{x} + (d\cos\theta,\ d\sin\theta)$$

$$\begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & -4 & 0 & 2 \\ 1 & 0 & -2 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

Fig. 6.19 Laws' basis functions (these are the low-order four of twelve actually used).



(a) (b)

Fig. 6.20 (a) Texture composite. (b) Classification.

Note that we the gray-level values appear as indices of the matrix S, implying that they are taken from some well-ordered discrete set 0, ..., K. Since

$$S(d, \theta) = S(d, \theta + \pi).$$

common practice is to restrict θ to multiples of $\pi/4$. Furthermore, information is not usually retained at both θ and $\theta + \pi$. The reasoning for the latter step is that for most texture discrimination tasks, the information is redundant. Thus we define

$$S(d, \theta) = \tfrac{1}{2} [S(d, \theta) + S(d, \theta + \pi)]$$

The intermediate matrices S yield potential features. Commonly used features are:

1. *Energy*

$$E(d, \theta) = \sum_{i=0}^{K} \sum_{j=0}^{K} [S(i, j|d, \theta)]^2 \tag{6.8}$$

2. *Entropy*

$$H(d, \theta) = \sum_{i=0}^{K} \sum_{j=0}^{K} S(i, j|d, \theta) \log f(i, j|d, \theta) \tag{6.9}$$

3. *Correlation*

$$C(d, \theta) = \frac{\sum_{i=0}^{K} \sum_{j=0}^{K} (i - \mu_x)(j - \mu_y) S(i, j|d, \theta)}{\sigma_x \sigma_y} \tag{6.10}$$

4. *Inertia*

$$I(d, \theta) = \sum_{i=0}^{K} \sum_{j=0}^{K} (i - j)^2 S(i, j|d, \theta) \tag{6.11}$$

5. *Local Homogeneity*

$$L(d, \theta) = \sum_{i=0}^{K} \sum_{j=0}^{K} \frac{1}{1 + (i-j)^2} S(i, j|d, \theta) \tag{6.12}$$

where $S(i, j|d, \theta)$ is the (i, j) th element of (d, θ), and

$$\mu_x = \sum_{i=0}^{K} i \sum_{j=0}^{K} S(i, j|d, \theta) \tag{6.13a}$$

$$\mu_y = \sum_{i=0}^{K} j \sum_{j=0}^{K} S(i, j|d, \theta) \tag{6.13b}$$

$$\sigma_x^2 = \sum_{i=0}^{K} (i-\mu_x)^2 \sum_{j=0}^{K} f(i, j|d, \theta) \tag{6.13c}$$

and

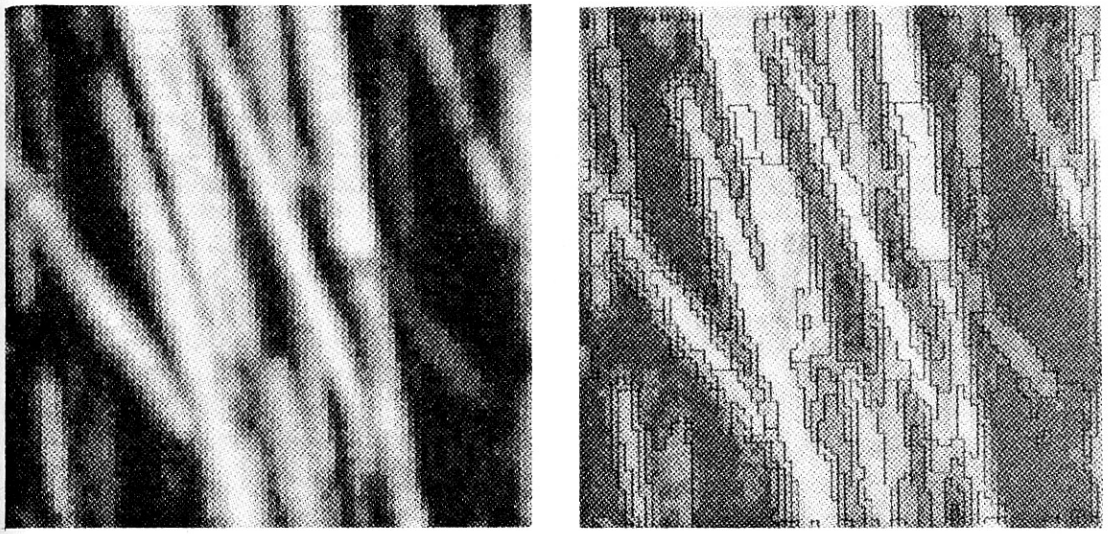
$$\sigma_y^2 = \sum_{i=0}^{K} (j-\mu_y)^2 \sum_{i=0}^{K} f(i, j|d, \theta) \tag{6.13d}$$

One important aspect of this approach is that the features chosen do not have psychological correlates [Tamura et al. 1978]. For example, none of the measures described would take on specific values corresponding to our notions of "rough" or "smooth." Also, the texture gradient is difficult to define in terms of SGLD feature values [Bajcsy and Lieberman 1976].

6.4.3 Region Texels

Region texels are an image-based way of defining primitives above the level of pixels. Rather than defining features directly as functions of pixels, a region segmentation of the image is created first. Features can then be defined in terms of the shape of the resultant regions, which are often more intuitive than the pixel-related features. Naturally, the approach of using edge elements is also possible. We shall discuss this in the context of texture gradients.

The idea of using regions as texture primitives was pursued in [Maleson et al. 1977]. In that implementation, all regions are ultimately modeled as ellipses and a corresponding five-parameter shape description is computed for each region. These parameters only define gross region shape, but the five-parameter primitives seem to work well for many domains. The texture image is segmented into regions in two steps. Initially, the modified version of Algorithm 5.1 that works for gray-level images is used. Figure 6.21 shows this example of the segmentation applied to a sample of "straw" texture. Next, parameters of the region grower are controlled so as to encourage convex regions which are fit with ellipses. Figure 6.22 shows the resultant ellipses for the "straw" texture. One set of ellipse parameters is $\mathbf{x}_0, a, b, \theta$ where $\mathbf{x}_0$ is the origin, a and b are the major and minor axis lengths and θ is the orientation of the major axis (Appendix 1). Besides these shape parameters, elliptical texels are also described by their average gray level. Figure 6.23 gives a qualitative indication of how ranges on feature values reflect different texels.



(a) Image (b) With Region Boundaries

Fig. 6.21 Region segmentation for straw texture.

6.5 THE TEXTURE GRADIENT

The importance of texture in determining surface orientation was described by Gibson [Gibson 1950]. There are three ways in which this can be done. These methods are depicted in Fig. 6.24. All these methods assume that the texture is embedded on a planar surface.

First, if the texture image has been segmented into primitives, the maximum rate of change of the projected size of these primitives constrains the orientation of

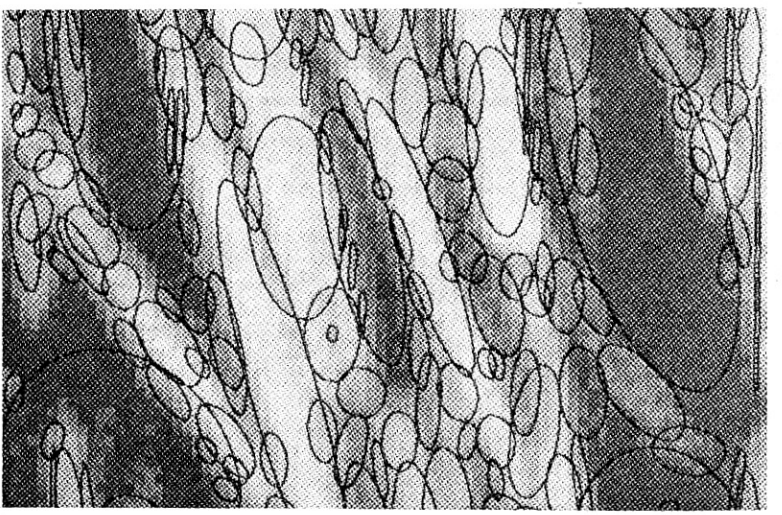


Fig. 6.22 Ellipses for straw texture.

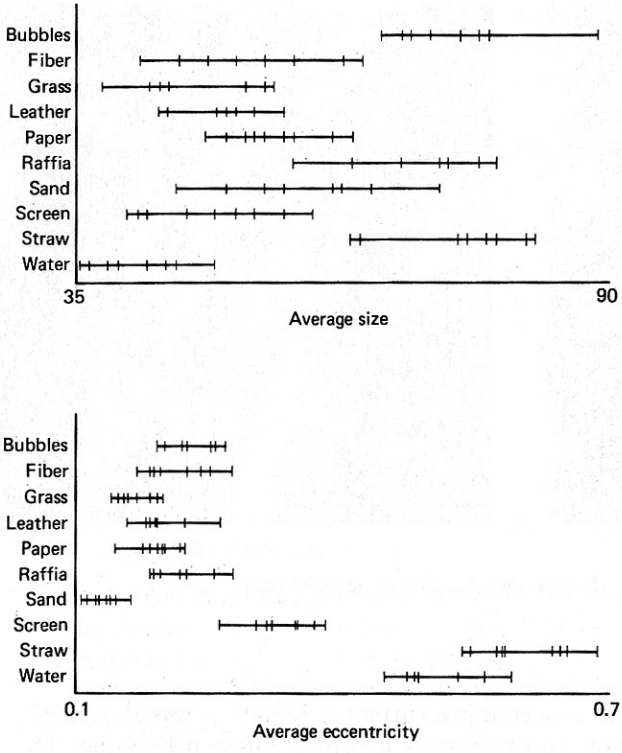


Fig. 6.23 Features defined on ellipses.

the plane in the following manner. The direction of maximum rate of change of projected primitive size is the direction of the *texture gradient.* The orientation of this direction with respect to the image coordinate frame determines how much the plane is rotated about the camera line of sight. The magnitude of the gradient can help determine how much the plane is tilted with respect to the camera, but knowledge about the camera geometry is also required. We have seen these ideas before in the form of gradient space; the rotation and tilt characterization is a polar coordinate representation of gradients.

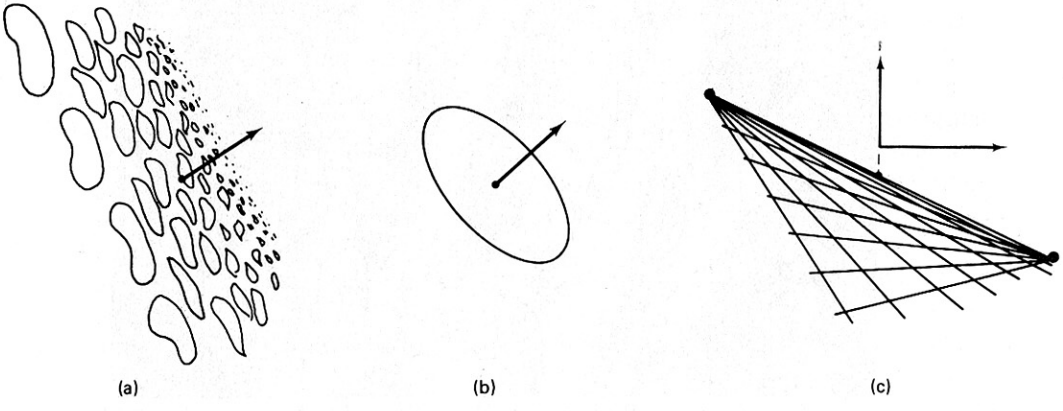


Fig. 6.24 Methods for calculating surface orientation from texture.

The second way to measure surface orientation is by knowing the shape of the texel itself. For example, a texture composed of circles appears as ellipses on the tilted surface. The orientation of the principal axes defines rotation with respect to the camera, and the ratio of minor to major axes defines tilt [Stevens 1979].

Finally, if the texture is composed of a regular grid of texels, we can compute vanishing points. For a perspective image, vanishing points on a plane P are the projection onto the image plane of the points at infinity in a given direction. In the examples here, the texels themselves are (conveniently) small line segments on a plane that are oriented in two orthogonal directions in the physical world. The general method applies whenever the placement tesselation defines lines of texels. Two vanishing points that arise from texels on the same surface can be used to determine orientation as follows. The line joining the vanishing points provides the orientation of the surface and the vertical position of the plane with respect to the z axis (i.e., the intersection of the line joining the vanishing points with $x = 0$) determines the tilt of the plane.

Line segment textures indicate vanishing points [Kender 1978]. As shown in Fig. 6.25, these segments could arise quite naturally from an urban image of the windows of a building which has been processed with an edge operator.

As discussed in Chapter 4, lines in images can be detected by detecting their parameters with a Hough algorithm. For example, by using the line parameterization

$$x \cos\theta + y \sin\theta = r$$

and by knowing the orientation of the line in terms of its gradient $\mathbf{g} = (\Delta x, \Delta y)$, a line segment $(x, y, \Delta x, \Delta y)$ can be mapped into r, θ space by using the relations

$$r = \frac{\Delta x x + \Delta y y}{\sqrt{\Delta x^2 + \Delta y^2}} \tag{6.14}$$

$$\theta = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) \tag{6.15}$$

These relationships can be derived by using Fig. 6.26 and some geometry. The Cartesian coordinates of the $r-\theta$ space vector are given by

$$\mathbf{a} = \left(\frac{\mathbf{g}\cdot\mathbf{x}}{\|\mathbf{g}\|^2}\right)\mathbf{g} \tag{6.16}$$

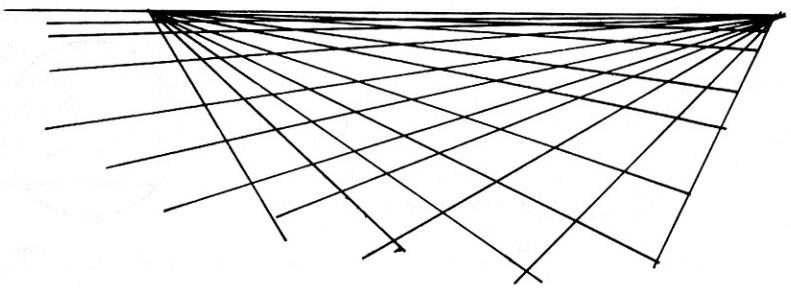


Fig. 6.25 Orthogonal line segments comprising a texture.

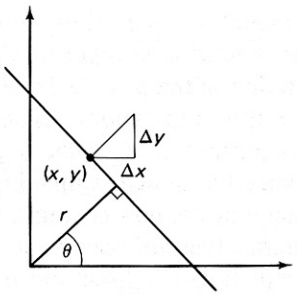


Fig. 6.26 r-θ transform.

Using this transformation, the set of line segments L_1 shown in Fig. 6.27 are all mapped into a single point in $r-\theta$ space. Furthermore, the set of lines L_2 which have the same vanishing point (x_v, y_v) project onto a circle in $r-\theta$ space with the line segment $((0, 0), (x_v, y_v))$ as a diameter. This scheme has two drawbacks: (1) vanishing points at infinity are projected into infinity, and (2) circles require some effort to detect. Hence we are motivated to use the transform $(x, y, \Delta x, \Delta y) \rightarrow \left(\frac{k}{r}, \theta\right)$ for some constant k. Now vanishing points at infinity are projected into the origin and the locus of the set of points L_2 is now a line. This line is perpendicular to the vector $\boldsymbol{x}_v$ and $\frac{k}{\|\mathbf{x}_v\|}$ units from the origin, as shown in Fig. 6.28. It can be detected by a second stage of the Hough transform; each point **a** is mapped into an $r'-\theta'$ space. For every **a**, compute all the r', θ' such that

$$a \cos\theta' + b \sin\theta' = r' \tag{6.17}$$

and increment that location in the appropriate r', θ' accumulator array. In this second space a vanishing point is detected as

$$r' = \frac{k}{\|\mathbf{x}_v\|} \tag{6.18}$$

$$\theta' = \tan^{-1}\left(\frac{y_v}{x_v}\right) \tag{6.19}$$

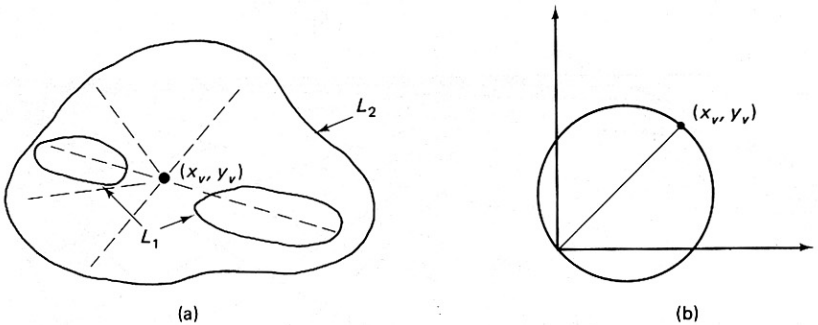


Fig. 6.27 Detecting the vanishing point with the Hough transform.

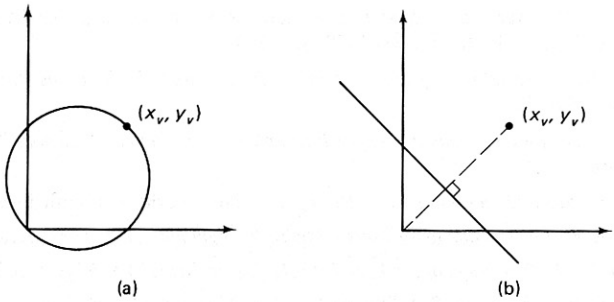


Fig. 6.28 Vanishing point loci.

In Kender's application the texels and their placement tesselation are similar in that the primitives are parallel to arcs in the placement tesselation graph. In a more general application the tesselation could be computed by connecting the centers of primitives.

EXERCISES

6.1 Devise a computer algorithm that, given a set of texels from each of a set of different "windows" of the textured image, checks to see of the resolution is appropriate. In other words, try to formalize the discussion of resolution in Section 6.2.

6.2 Are any of the grammars in Section 6.3 suitable for a parallel implementation (i.e., parallel application of rules)? Discuss, illustrating your arguments with examples or counterexamples from each of the three main grammatical types (shape, tree, and array grammars).

6.3 Are shape, array, and tree grammars context free or context-sensitive as defined? Can such grammars be translated into "traditional" (string) grammars? If not, how are they different; and if so, why are they useful?

6.4 Show how the generalized Hough transform (Section 4.3) could be applied to texel detection.

6.5 In an outdoors scene, there is the problem of different scales. For example, consider the grass. Grass that is close to an observer will appear "sharp" and composed of primitive elements, yet grass distant from an observer will be much more "fuzzy" and homogeneous. Describe how one might handle this problem.

6.6 The texture energy transform (Section 6.4.1) is equivalent to a set of Fourier-domain operations. How do the texture energy features compare with the ring and sector features?

6.7 The texture gradient is presumably a gradient in some aspect of texture. What aspect is it, and how might it be quantified so that texture descriptions can be made gradient independent?

6.8 Write a texture region grower and apply it to natural scenes.

REFERENCES

BAJCSY, R. and L. LIEBERMAN. "Texture gradient as a depth cue." *CGIP 5*, 1, March 1976, 52–67.

BRODATZ, P. *Textures: A Photographic Album for Artists and Designers.* Toronto: Dover Publishing Co., 1966.

CONNORS, R. "Towards a set of statistical features which measure visually perceivable qualities of textures." *Proc.*, PRIP, August 1979, 382–390.

COVER, T. M. "Estimation by the nearest neighbor rule." *IEEE Trans. Information Theory 14*, January 1968, 50–55.

FU, K. S. *Sequential Methods in Pattern Recognition and Machine Learning.* New York: Academic Press, 1968.

FU, K. S. *Syntactic Methods in Pattern Recognition.* New York: Academic Press, 1974.

FUKUNAGA, K. *Introduction to Statistical Pattern Recognition.* New York, Academic Press, 1972.

GIBSON, J. J. *The Perception of the Visual World.* Cambridge, MA: Riverside Press, 1950.

HALL, E. L, R. P. KRUGER, S. J. DWYER III, D. L. HALL, R. W. MCLAREN, and G. S. LODWICK. "A survey of preprocessing and feature extraction techniques for radiographic images." *IEEE Trans. Computers 20*, September 1971.

HARALICK, R. M. "Statistical and structural approaches to texture." *Proc.*, 4th IJCPR, November 1978, 45–60.

HARALICK, R. M., R. SHANMUGAM, and I. DINSTEIN. "Textural features for image classification." *IEEE Trans. SMC 3*, November 1973, 610–621.

HOPCROFT, J. E. and J. D. ULLMAN. *Introduction to Automata Theory, Languages and Computation.* Reading, MA: Addison-Wesley, 1979.

JAYARAMAMURTHY, S. N. "Multilevel array grammars for generating texture scenes." *Proc.*, PRIP, August 1979, 391–398.

JULESZ, B. "Textons, the elements of texture perception, and their interactions." *Nature 290*, March 1981, 91–97.

KENDER, J. R. "Shape from texture: a brief overview and a new aggregation transform." *Proc.*, DARPA IU Workshop, November 1978, 79–84.

KRUGER, R. P., W. B. THOMPSON, and A. F. TWINER. "Computer diagnosis of pneumoconiosis." *IEEE Trans. SMC 45*, 1974, 40–49.

LAWS, K. I. "Textured image segmentation." Ph.D. dissertation, Dept. of Engineering, Univ. Southern California, 1980.

LU, S. Y. and K. S. FU. "A syntactic approach to texture analysis." *CGIP 7*, 3, June 1978, 303–330.

MALESON, J. T., C. M. BROWN, and J. A. FELDMAN. "Understanding natural texture." *Proc.*, DARPA IU Workshop, October 1977, 19–27.

MILGRAM, D. L. and A. ROSENFELD. "Array automata and array grammars." *Proc.*, IFIP Congress 71, Booklet TA-2. Amsterdam: North-Holland, 1971, 166–173.

PRATT, W. K., O. D. FAUGERAS, and A. GAGALOWICZ. "Applications of Stochastic Texture Field Models to Image Processing." *Proc. of the IEEE.* Vol.69, No. 5, May 1981

ROSENFELD, A. "Isotonic grammars, parallel grammars and picture grammars." In *MI6*, 1971.

STEVENS, K.A. "Representing and analyzing surface orientation." In *Artificial Intelligence: An MIT Perspective,* Vol. 2, P. H. Winston and R. H. Brown (Eds.). Cambridge, MA: MIT Press, 1979.

STINY, G. and J. GIPS. *Algorithmic Aesthetics: Computer Models for Criticism and Design in the Arts.* Berkeley, CA: University of California Press, 1972.

TAMURA, H., S. MORI, and T. YAMAWAKI. "Textural features corresponding to visual perception." *IEEE Trans. SMC 8*, 1978, 460–473.

TOU, J. T. and R. C. GONZALEZ. *Pattern Recognition Principles.* Reading, MA: Addison-Wesley, 1974.

WESZKA, J. S., C. R. DYER, and A. ROSENFELD. "A comparative study of texture measures for terrain classification." *IEEE Trans. SMC 6*, 4, April 1976, 269–285.

ZUCKER, S. W. "Toward a model of texture." *CGIP 5*, 2, June 1976, 190–202.