Lecture 10

Texture Analysis and Hough Transform

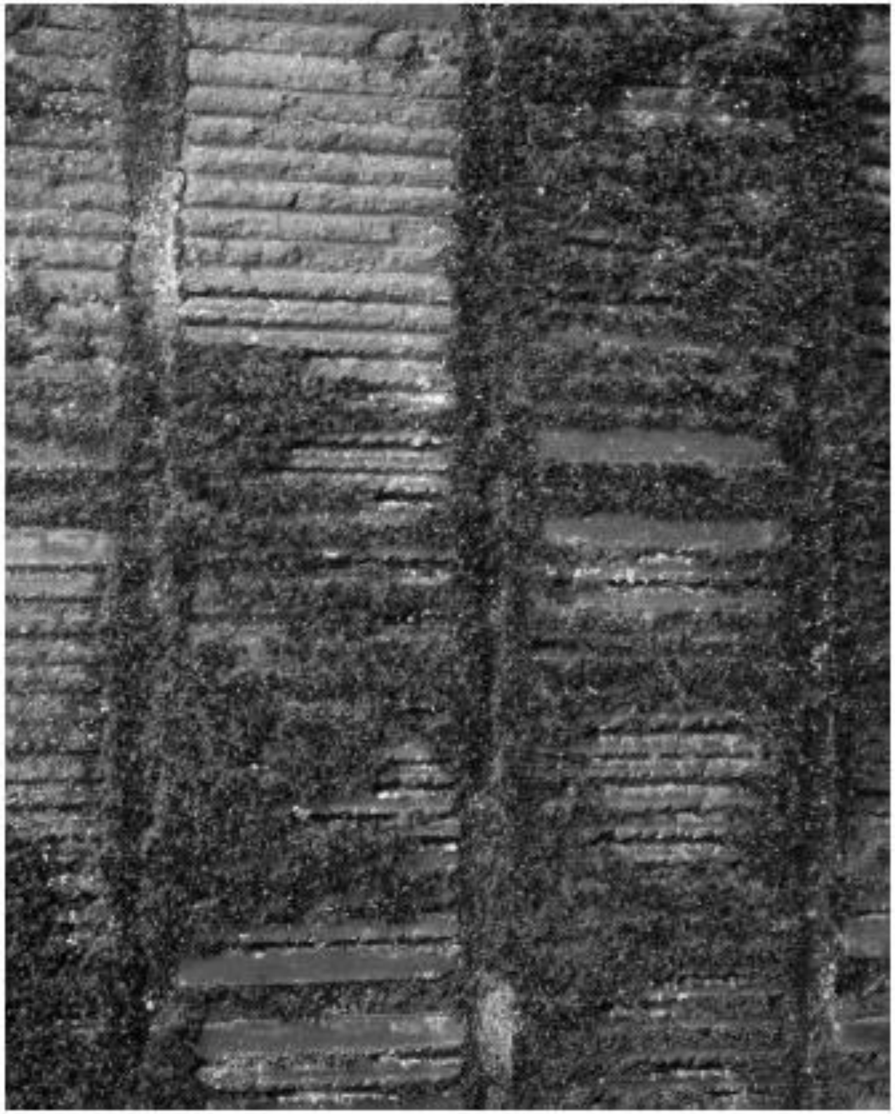
Texture

- Organized patterns of quite regular subelements called textons.
- Texture is a property of sufficiently large regions

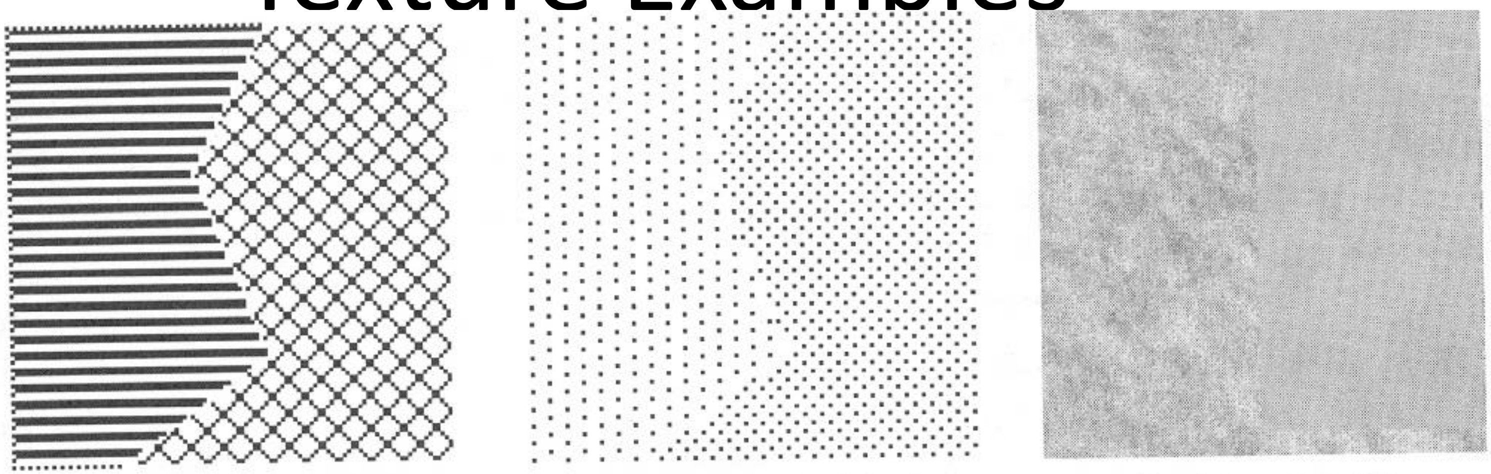
Applications:

- Texture based segmentation
- Texture synthesis
- Texture analysis and texture based matching
- Shape (surface orientation) from texture

Texture Examples



Texture Examples



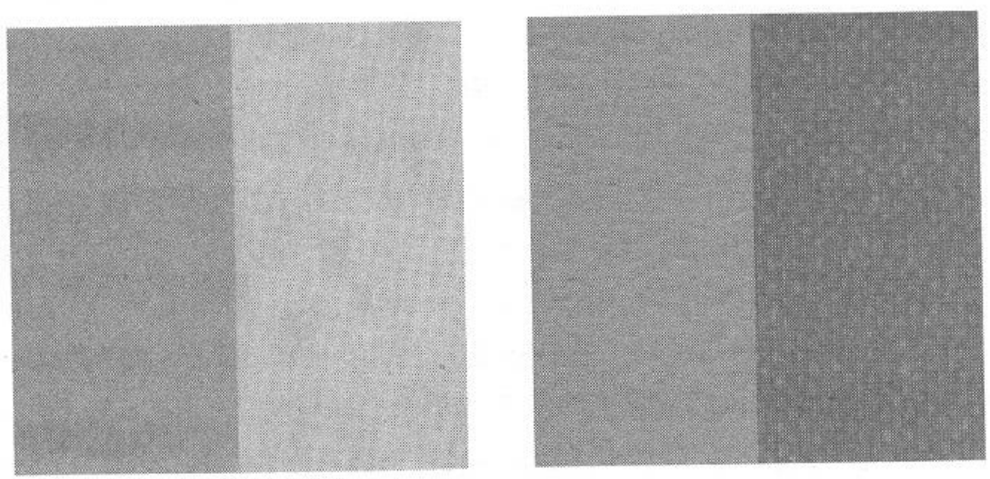
Test image T1
(a)

Test image T2
(b)

Test image T3
(c)

(a,b): Artificial textures

(c,d,e): Naturally occurring textures



Test image T4
(d)

Test image T5
(e)

Representing textures

Statistical

yields characterization of textures as smooth, coarse grainy, etc.

Spectral

are based on Fourier spectrum and are primarily used to detect the global periodicity in an image by identifying high energy narrow peaks in the spectrum.

Statistical approaches

- Based on the histogram measures of image
- Based on the Grey Level Co-occurrence Matrix (GLCM) and related measurement

Histogram based texture description

Using statistical moments of grey level histogram of the image or region

Let $p(z_i)$ is the histogram of the grey levels z_i of an image

The nth moment about the mean is given by:

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i)$$

Where mean is

$$m = \sum_{i=0}^{L-1} z_i p(z_i)$$

The 0^{th} moment = 1, and the 1^{st} moment = 0,

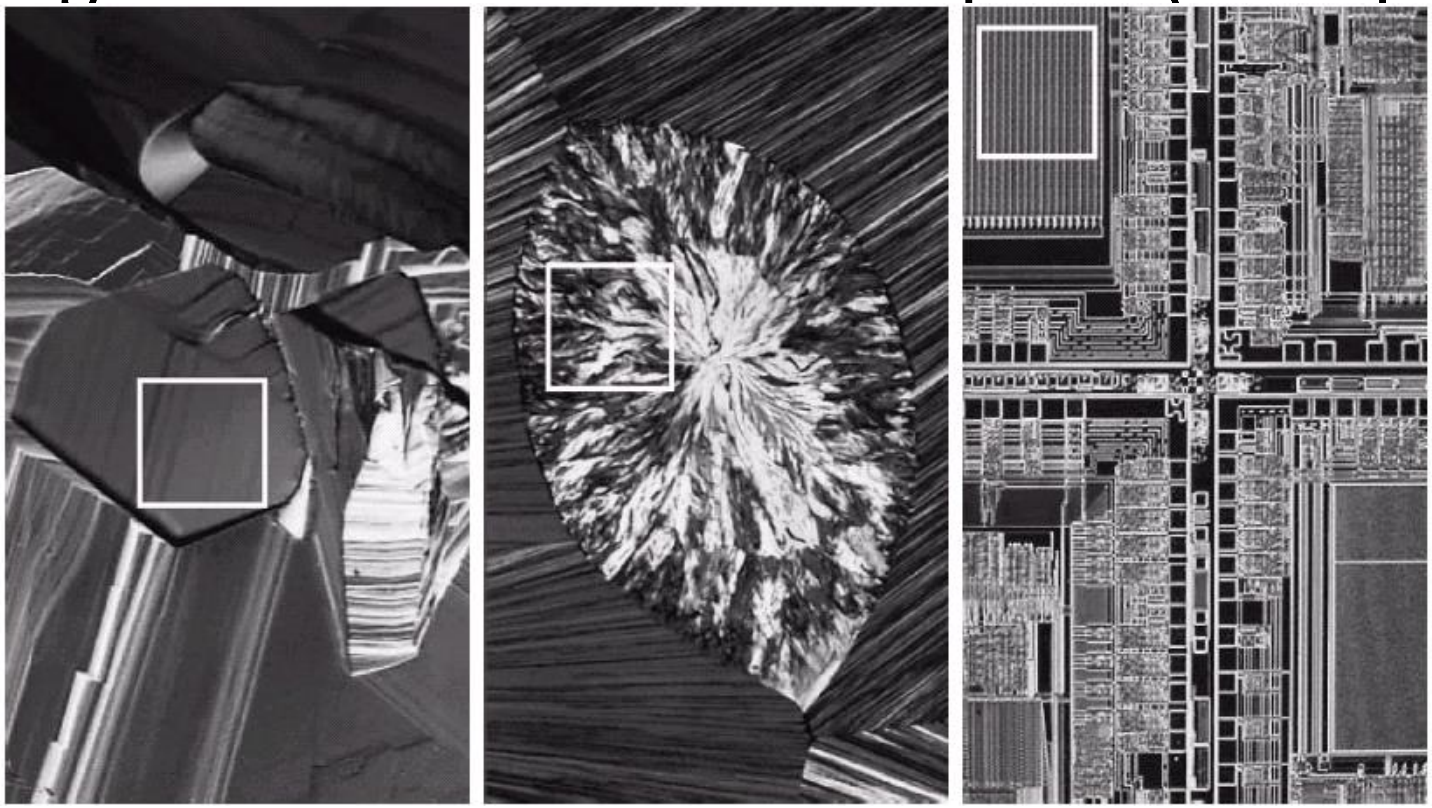
The variance is the second moment and is given by

$$\sigma^2(z) = \mu_2(z) = \sum_{i=0}^{L-1} (z_i - m)^2 p(z_i)$$

Histogram based texture description

- For texture description the following parameters are useful
  - Variance and related measures: descriptor of relative smoothness, use normalized variance $$R = 1 - \frac{1}{1+\sigma^2(z)}$$
  - Skewness of histogram $$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$$
  - Relative flatness of histogram $$\mu_4(z) = \sum_{i=0}^{L-1} (z_i - m)^4 p(z_i)$$
  - Uniformity $$U = \sum_{i=0}^{L-1} p^2(z_i)$$
  - Average Entropy $$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$$

Histogram based texture description (example)



Texture	Mean	Standard deviation	R (normalized)	Third moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

GLCMs

- For texture description the following parameters of GLCM are measured and analyzed

Maximum probability

$$\max_{i,j}(c_{ij})$$

Contrast

$$\sum_i \sum_j (i-j)^2 c_{ij}$$

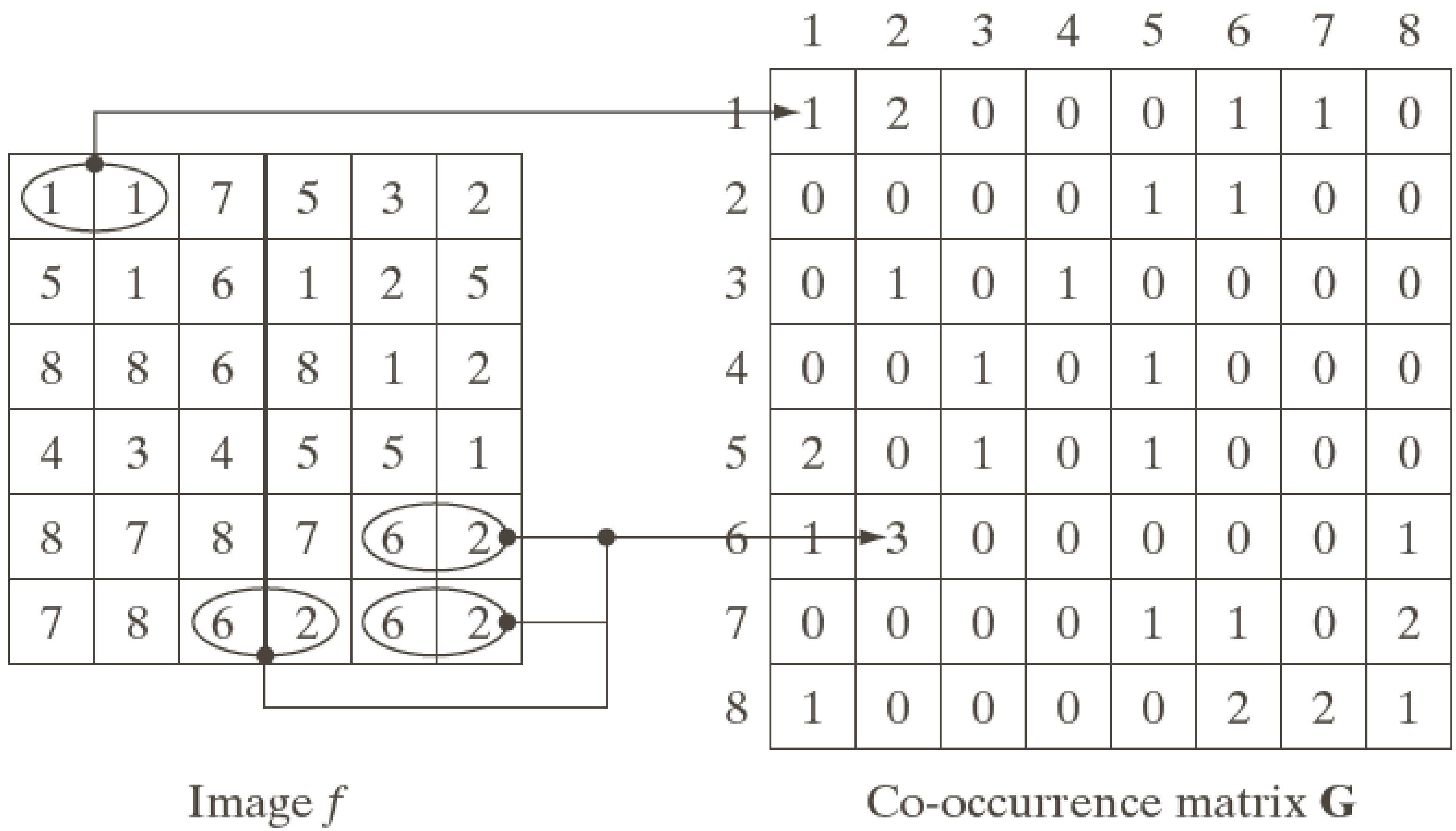
Uniformity

$$\sum_i \sum_j c_{ij}^2$$

Entropy

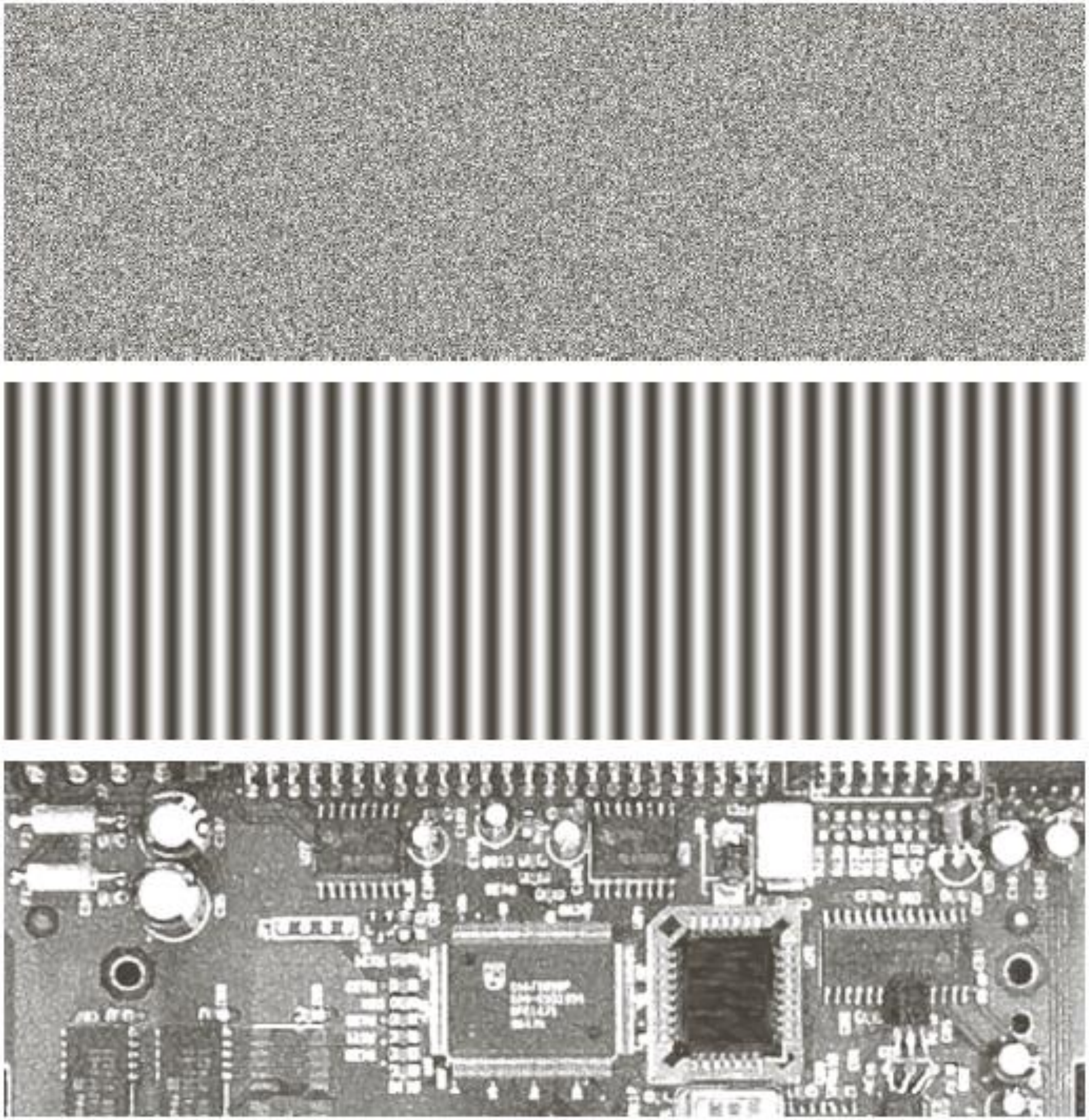
$$-\sum_i \sum_j c_{ij} \log_2 c_{ij}$$

FIGURE 11.29
How to generate a co-occurrence matrix.



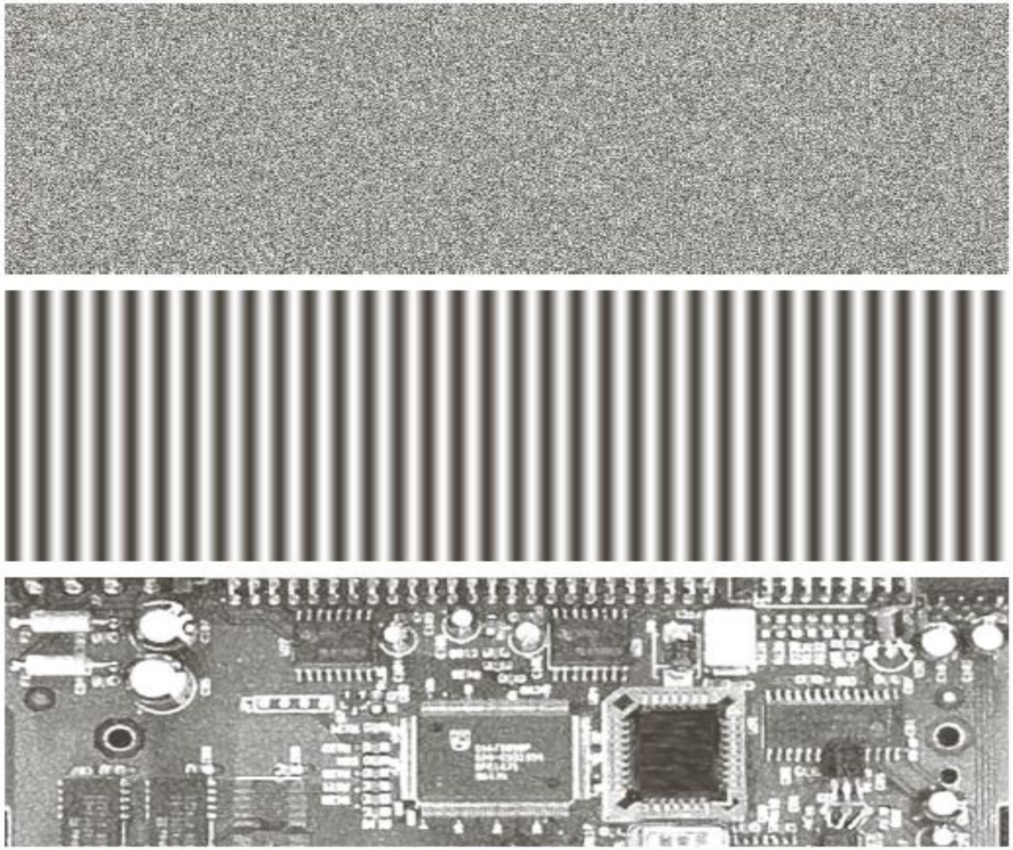
Descriptor	Explanation	Formula
Maximum probability	Measures the strongest response of **G**. The range of values is [0, 1].	$\max_{i,j}(p_{ij})$
Correlation	A measure of how correlated a pixel is to its neighbor over the entire image. Range of values is 1 to −1, corresponding to perfect positive and perfect negative correlations. This measure is not defined if either standard deviation is zero.	$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r\sigma_c}$ $\sigma_r \neq 0;\ \sigma_c \neq 0$
Contrast	A measure of intensity contrast between a pixel and its neighbor over the entire image. The range of values is 0 (when **G** is constant) to $(K-1)^2$.	$\sum_{i=1}^{K}\sum_{j=1}^{K}(i-j)^2 p_{ij}$
Uniformity (also called Energy)	A measure of uniformity in the range [0, 1]. Uniformity is 1 for a constant image.	$\sum_{i=1}^{K}\sum_{j=1}^{K}p_{ij}^2$
Homogeneity	Measures the spatial closeness of the distribution of elements in **G** to the diagonal. The range of values is [0, 1], with the maximum being achieved when **G** is a diagonal matrix.	$\sum_{i=1}^{K}\sum_{i=1}^{K}\frac{p_{ij}}{1+\lvert i-j\rvert}$
Entropy	Measures the randomness of the elements of **G**. The entropy is 0 when all p_{ij}'s are 0 and is maximum when all p_{ij}'s are equal. The maximum value is $2\log_2 K$. (See Eq. (11.3-9) regarding entropy).	$-\sum_{i=1}^{K}\sum_{i=1}^{K}p_{ij}\log_2 p_{ij}$

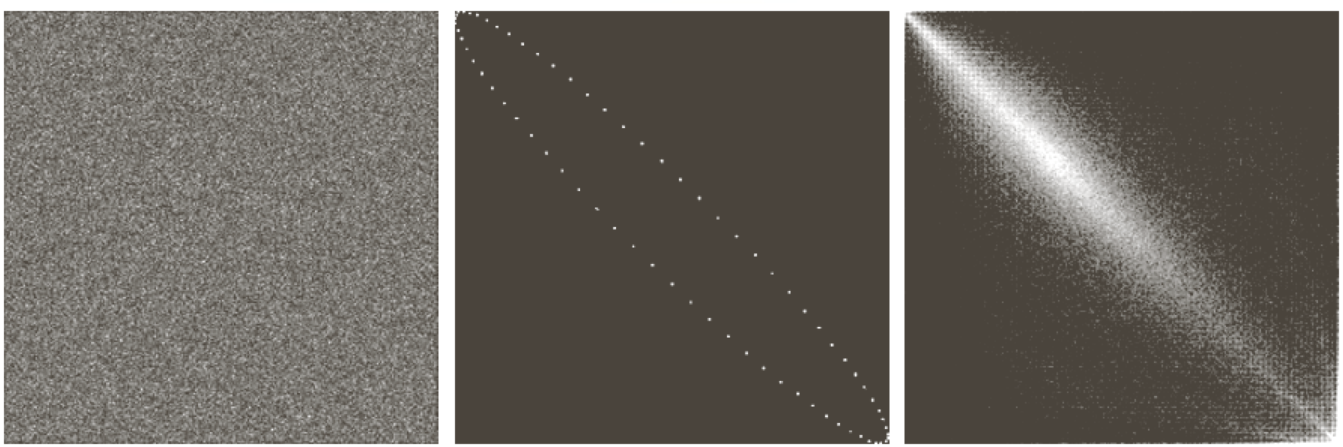
TABLE 11.3 Descriptors used for characterizing co-occurrence matrices of size $K \times K$. The term p_{ij} is the ijth term of **G** divided by the sum of the elements of **G**.



a
b
c

FIGURE 11.30 Images whose pixels have (a) random, (b) periodic, and (c) mixed texture patterns. Each image is of size 263×800 pixels.





a b c

FIGURE 11.31 256×256 co-occurrence matrices, $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, corresponding from left to right to the images in Fig. 11.30.

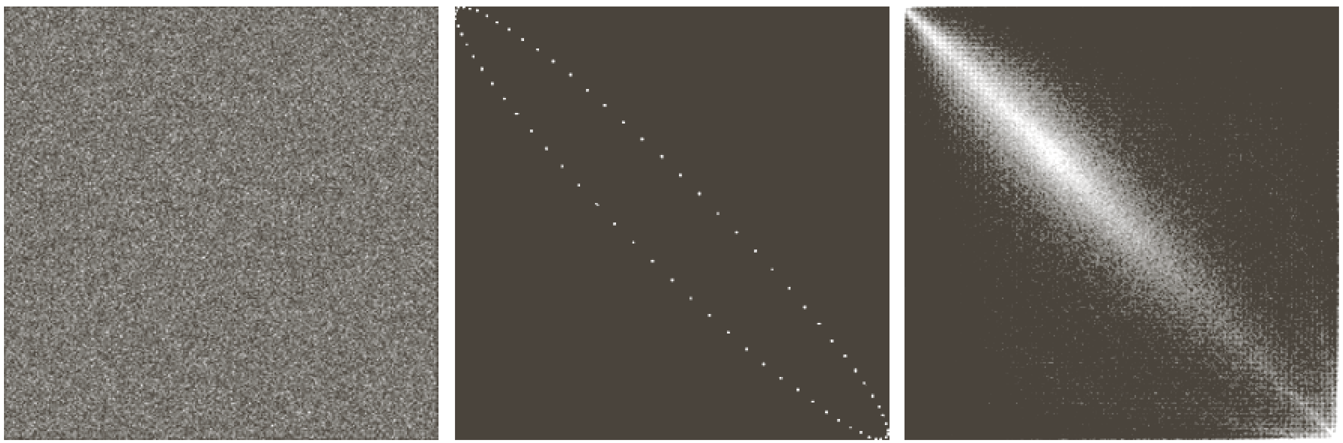


TABLE 11.4 Descriptors evaluated using the co-occurrence matrices displayed in Fig. 11.31.

Normalized Co-occurrence Matrix	Descriptor					
	Max Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	−0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	570	0.01230	0.0824	6.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	1356	0.00480	0.2048	13.58

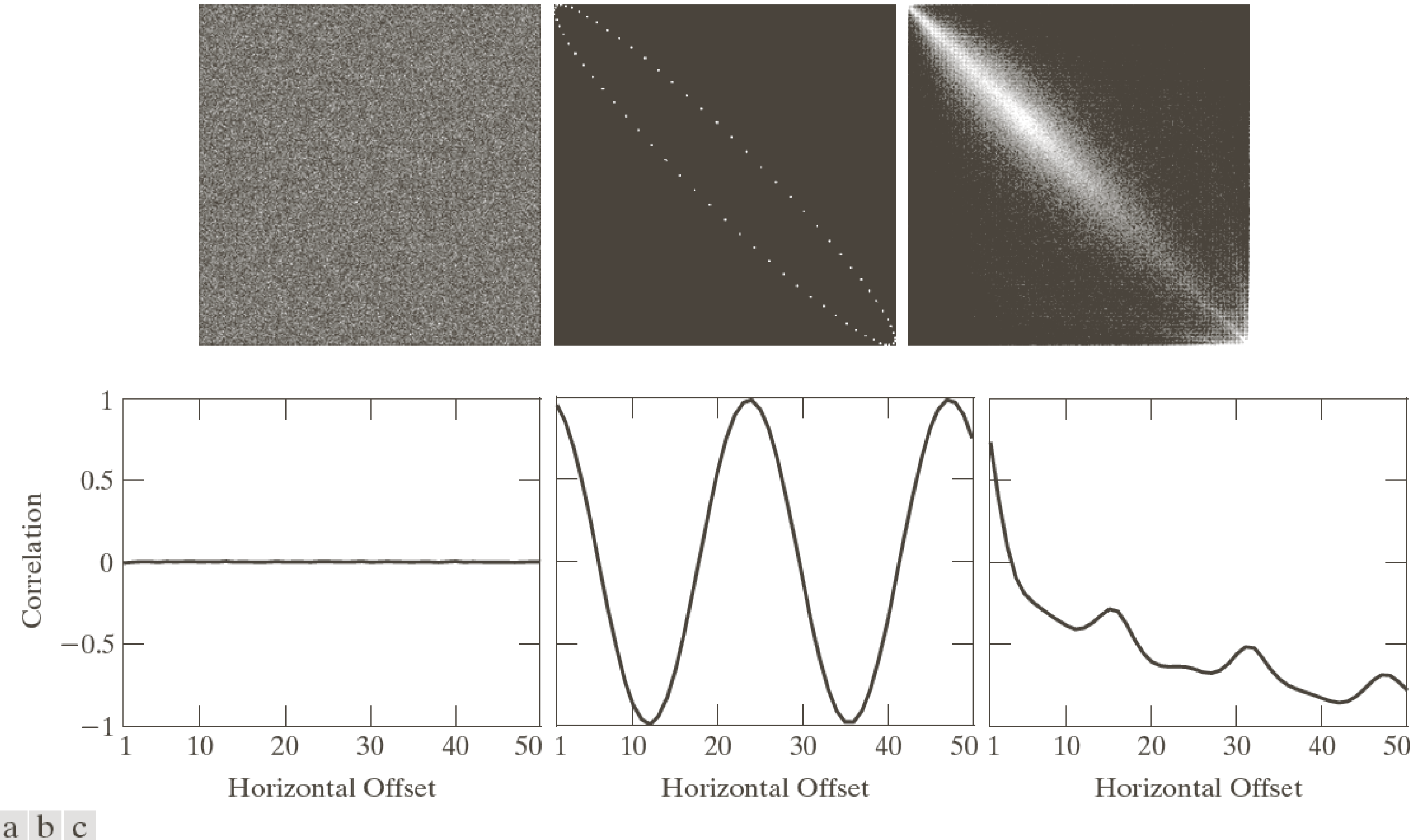


FIGURE 11.32 Values of the correlation descriptor as a function of offset (distance between "adjacent" pixels) corresponding to the (a) noisy, (b) sinusoidal, and (c) circuit board images in Fig. 11.30.

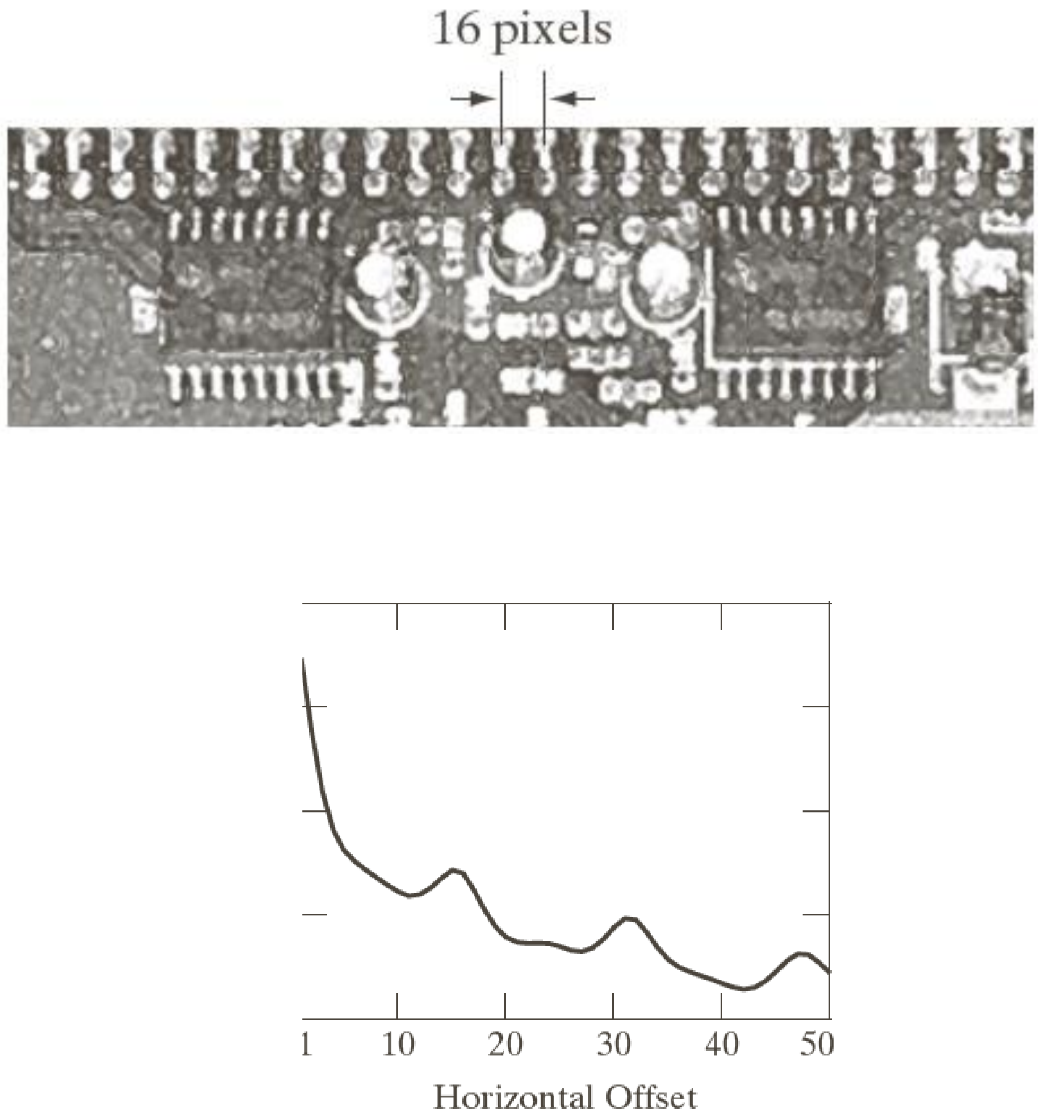


FIGURE 11.33
A zoomed section of the circuit board image showing periodicity of components.

Spectral techniques: Fourier transform

- Suitable to detect directionality of periodic and almost periodic 2-D patterns in an image
- Periodic texture patterns are easily detectable by concentration of high energy burst in the spectrum
- Features of Fourier spectrum for texture representation are:
  – Prominent peaks in the spectrum give the principal direction of texture patterns
  – The location of peaks give the frequency and thus the scale of repetition of a pattern
- Eliminating any periodic components via filtering leaves non-periodic image elements which can be described by statistical techniques

Spectral techniques: Fourier transform

- Simplified by expressing the spectrum in polar coordinates to yield a function $S(r, \theta)$ where S is the spectrum function and r and θ are the polar coordinates.

  For each direction θ, $S(r, \theta)$ = a 1-D function $S_\theta(r)$

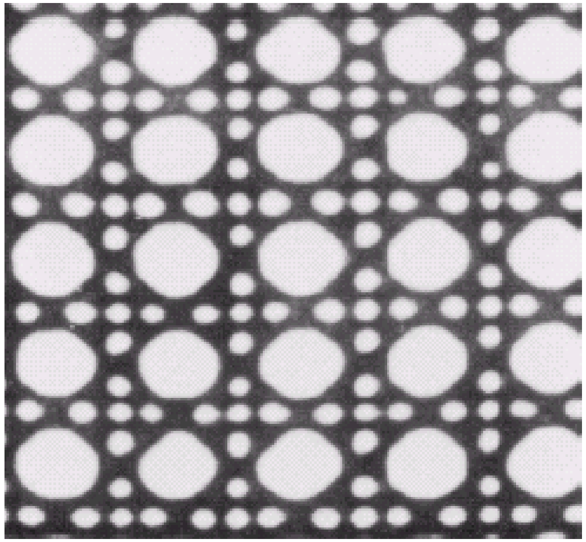
  For each frequency r, $S(r, \theta)$ = a 1-D function $S_r(\theta)$
- Analyzing $S_\theta(r)$ for a fixed θ, gives the distance from the origin and thus the scale of repetition of a texture pattern.
- Analyzing $S_r(\theta)$ for a fixed r, gives the direction and thus the orientation of the periodic texture pattern.
- To measure this analysis, we define two quantities

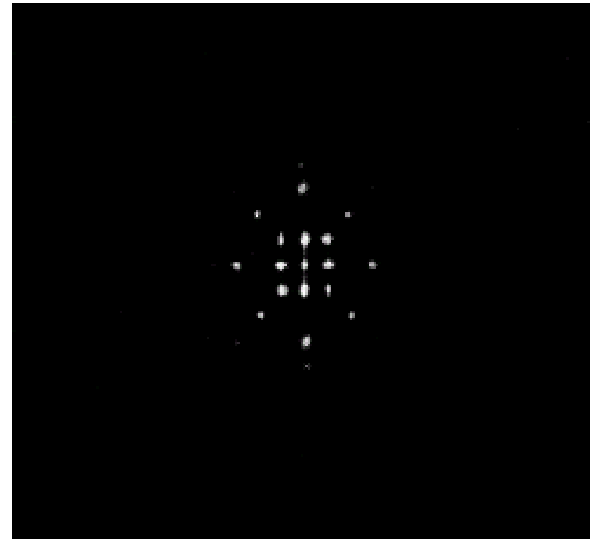
$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r), \qquad S(\theta) = \sum_{r=1}^{R_0} S_r(\theta).$$

These quantities measure the spectral response and give the dominant directions and scales of periodic texture patterns.

Spectral techniques: Fourier transform

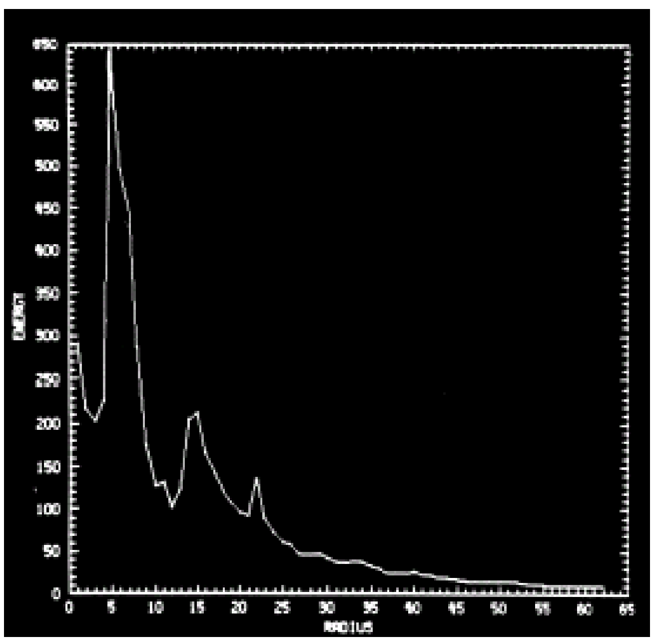
Image showing periodic texture

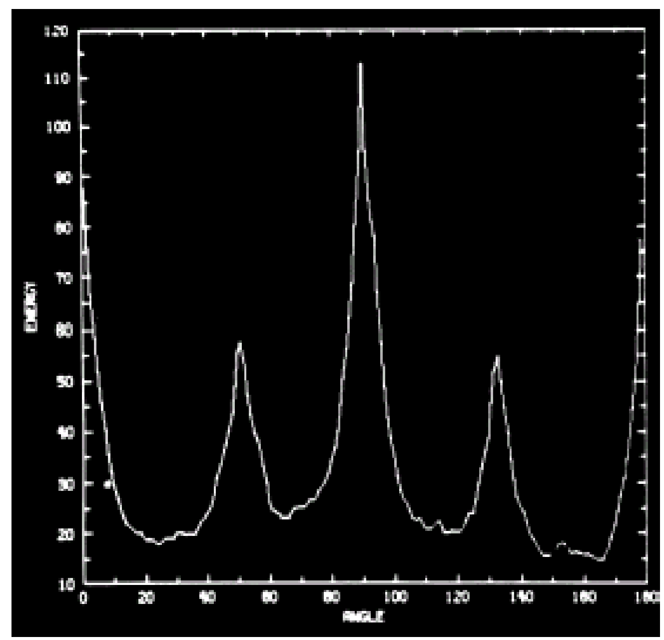




Spectrum

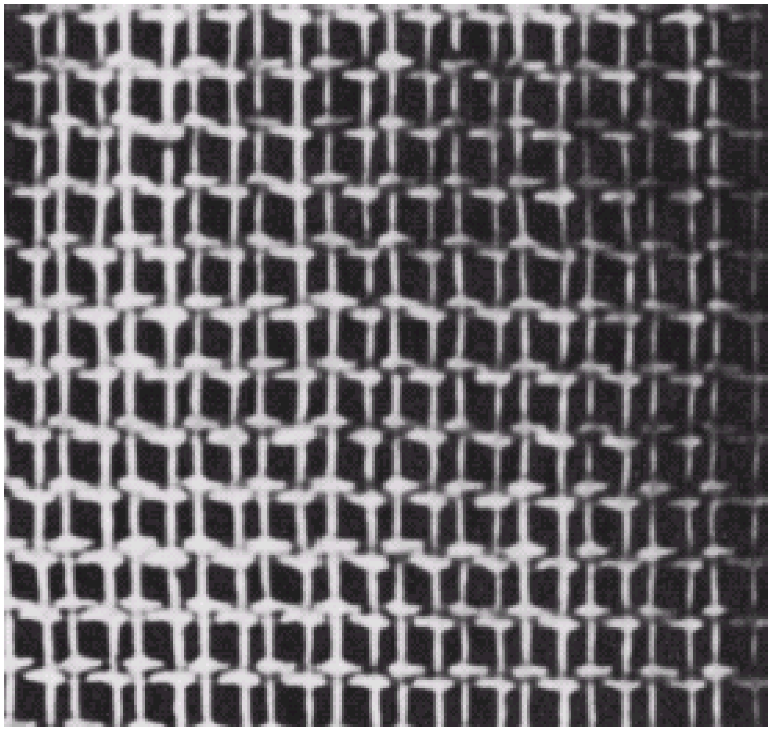
Plot of *S*(*r*)



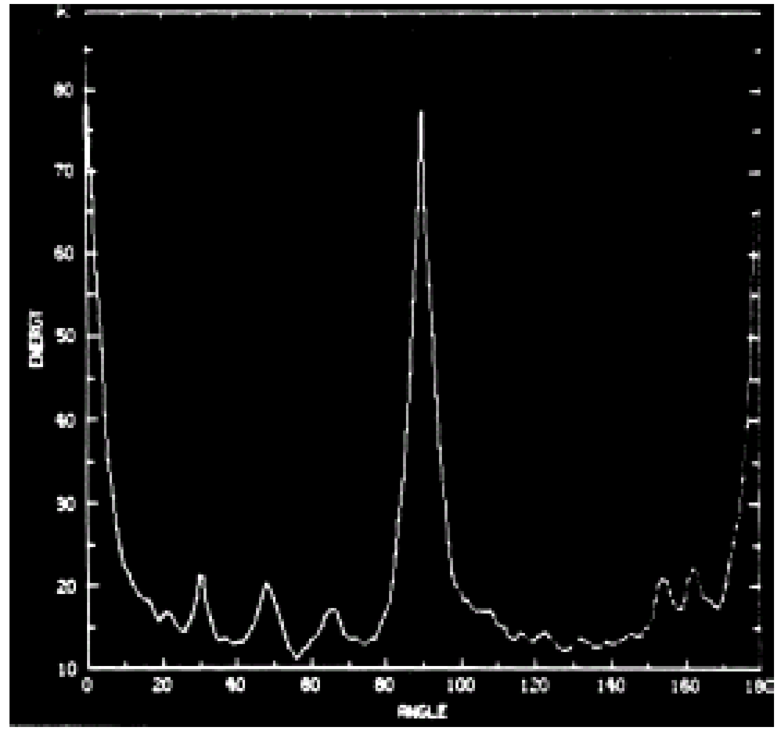


Plot of *S*(θ)

Spectral techniques: Fourier transform (example)



Another image showing periodic texture



Plot of $S(\theta)$

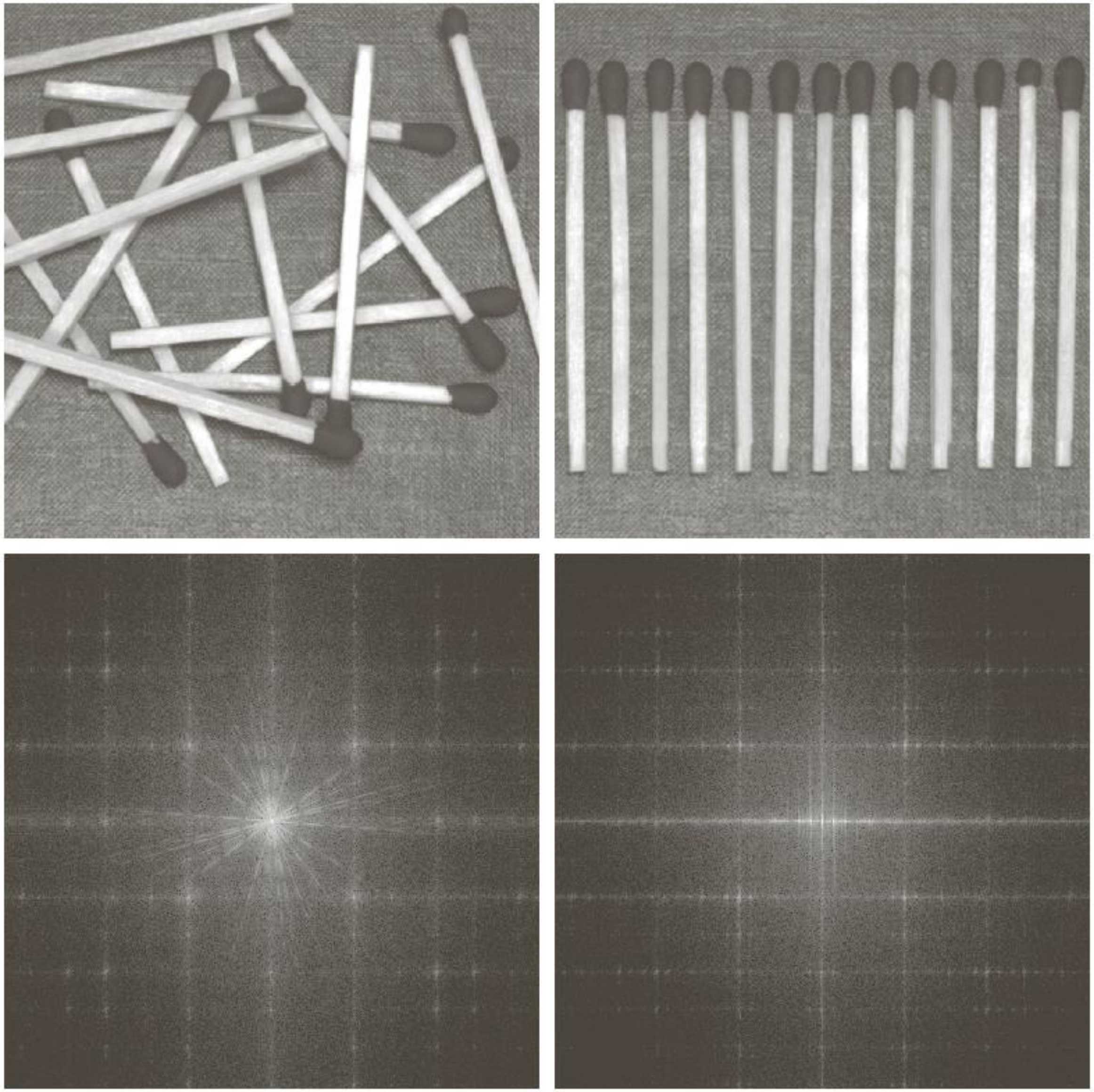
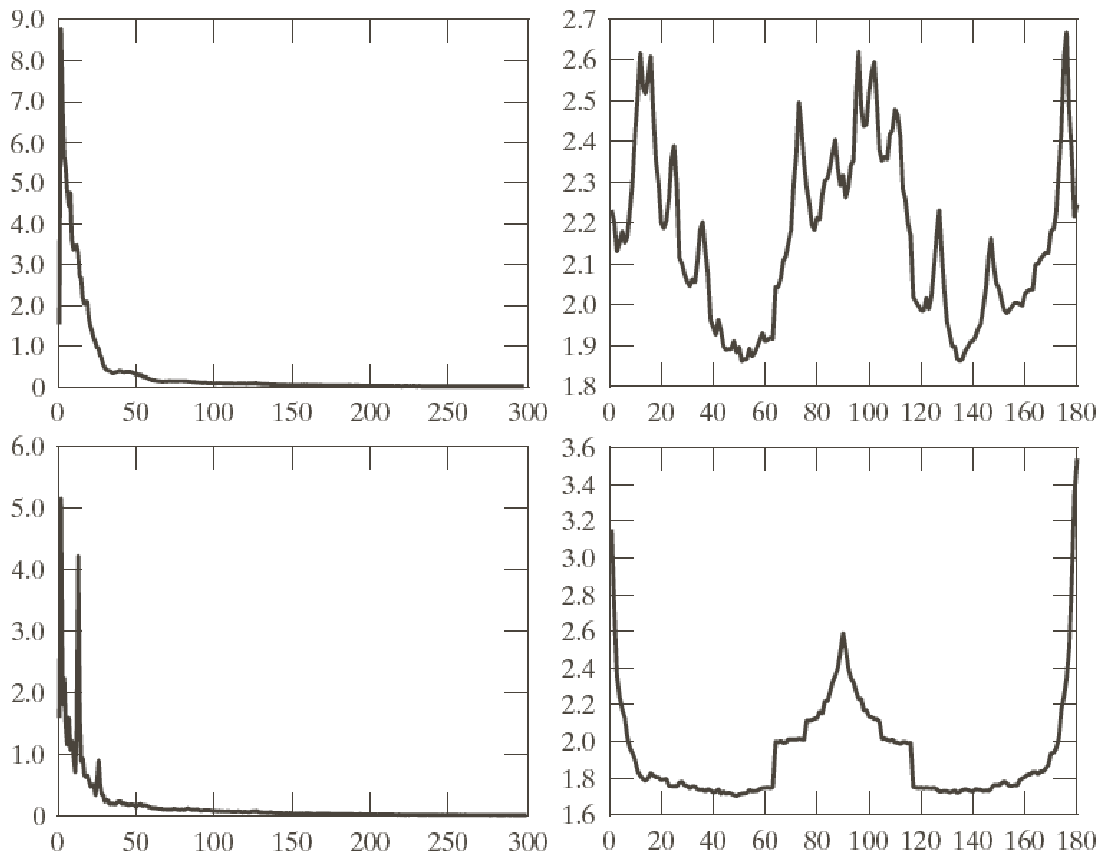


FIGURE 11.35 (a) and (b) Images of random and ordered objects. (c) and (d) Corresponding Fourier spectra. All images are of size 600×600 pixels.



a b
c d

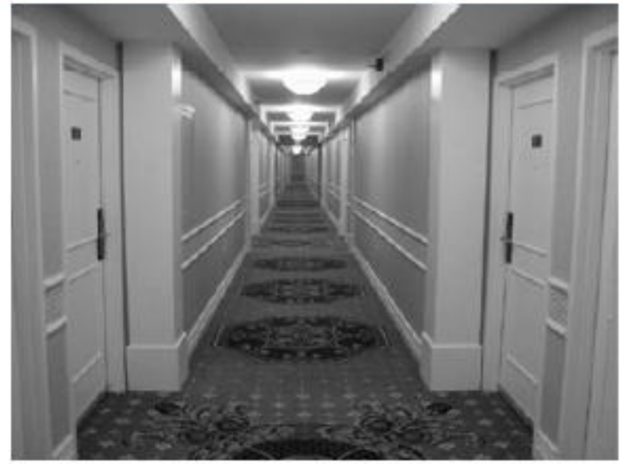
FIGURE 11.36 Plots of (a) $S(r)$ and (b) $S(\theta)$ for Fig. 11.35(a). (c) and (d) are plots of $S(r)$ and $S(\theta)$ for Fig. 11.35(b). All vertical axes are $\times 10^5$.

Introduction to Hough transform

- The Hough transform (HT) can be used to detect lines, circles or other parametric curves.
- It was introduced in 1962 (Hough 1962) and first used to find lines in images a decade later (Duda 1972).
- The goal is to find the location of lines in images.
- This problem could be solved by e.g. Morphology and a linear structuring element, or by correlation.
  - Then we would need to handle rotation, zoom, distortions etc.
- Hough transform can detect lines, circles and other structures if their parametric equation is known.
- It can give robust detection under noise and partial occlusion.

An image with linear structures

- Borders between the regions are straight lines.
- These lines separate regions with different grey levels.
- Edge detection is often used as preprocessing to Hough transform.

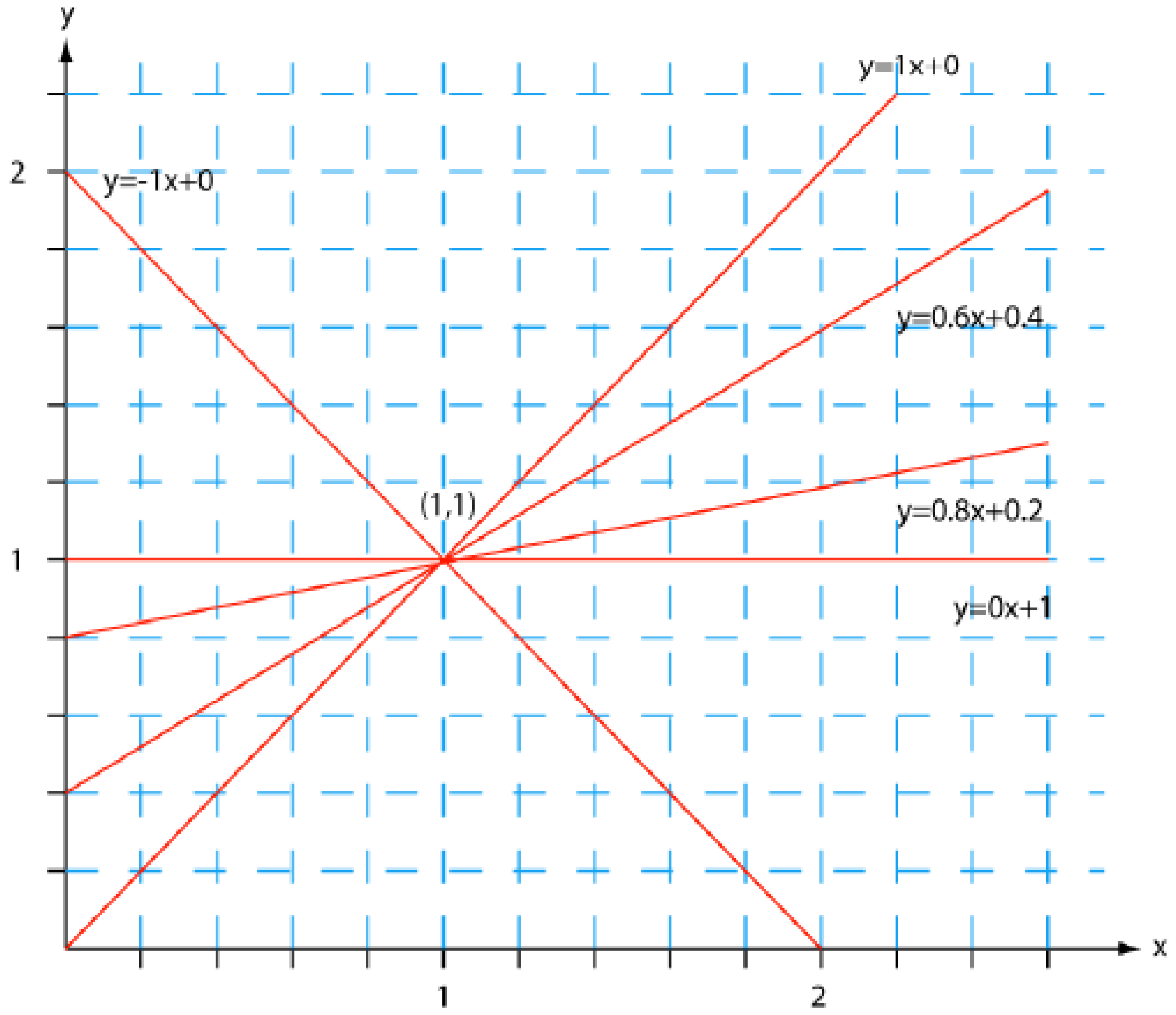


Hough-transform

- Assume that we have performed some edge detection, and a thresholding of the edge magnitude image.
- Thus, we have *n* pixels that may partially describe the boundary of some objects.
- We wish to find sets of pixels that make up straight lines.
- Regard a point $(x_i; y_i)$

  and a straight line $y_i = ax_i + b$
  - There are many lines passing through the point (x_i, y_i).
  - Common to them is that they satisfy the equation for some set of parameters (a,b).

Hough transform – basic idea



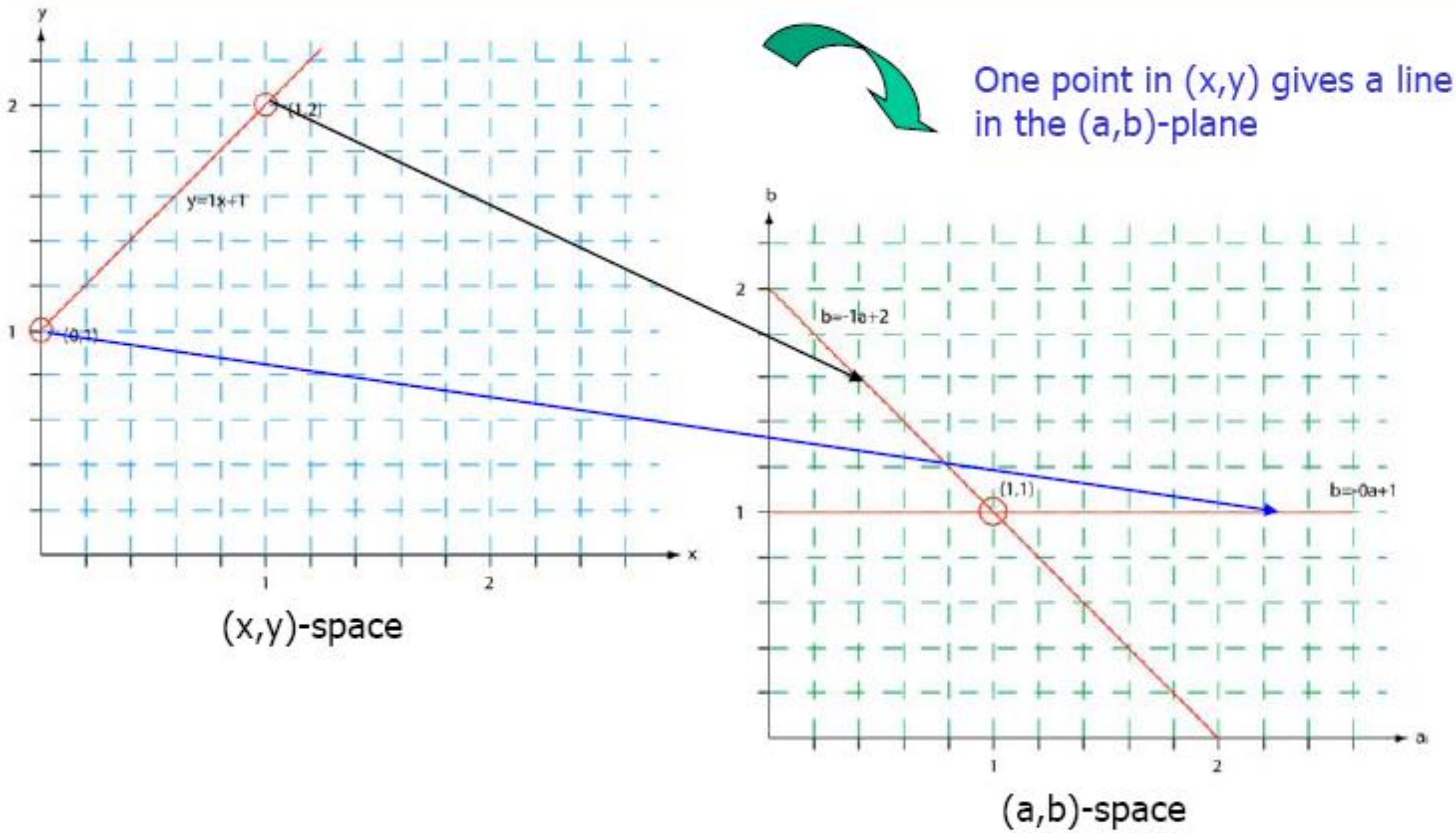
Hough transform – basic idea

- This equation can obviously be rewritten as follows:

$$b = -xa + y$$

- We now consider *x* and *y* as parameters and *a* and *b* as variables.
- This is a line in (*a,b*) space parameterized by *x* and *y*.
  - So: a single point in xy-space gives a line in (a,b) space.
- Another point (*x,y*) will give rise to another line in (*a,b*) space.

Hough transform – basic idea

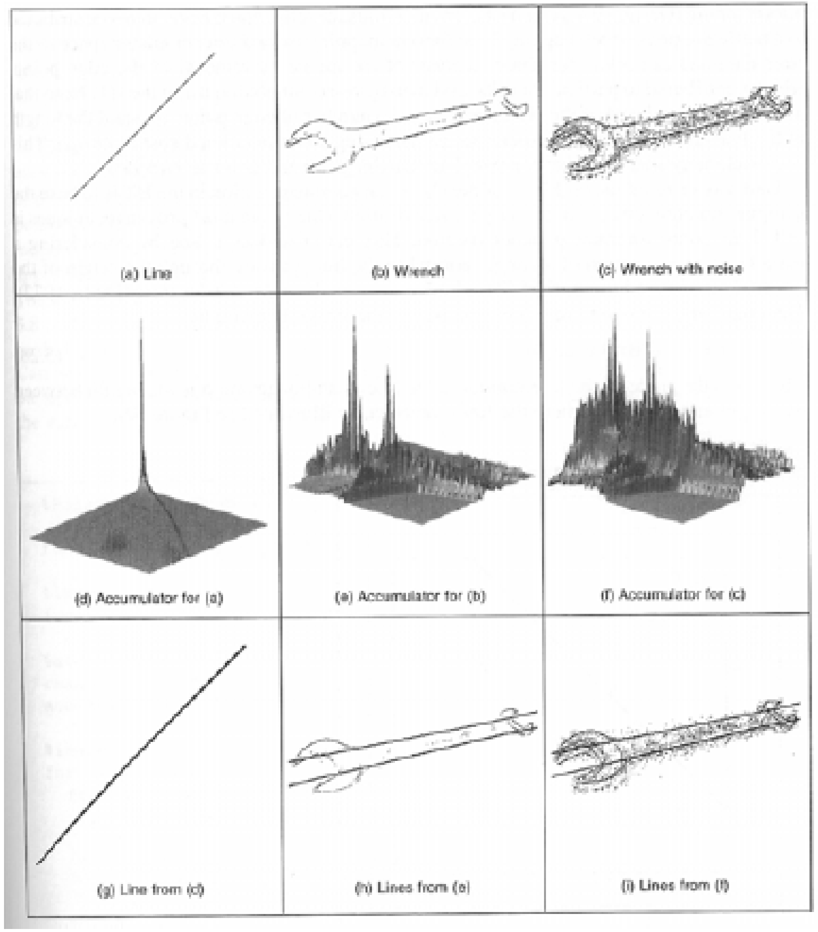


Example – images and accumulator space

Thresholded edge images

Visualizing the accumulator space
The height of the peak will be defined by the number of pixels in the line.

Thresholding the accumulator space and superimposing this onto the edge image

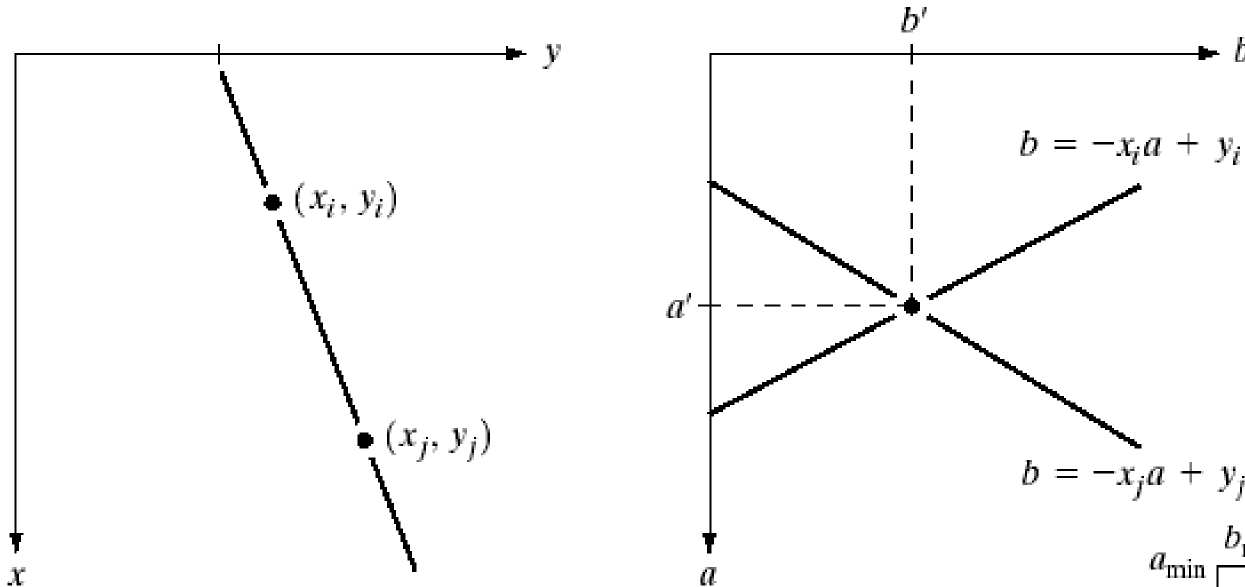


Note how noise effects the accumulator. Still with noise, the largest peaks correspond to the major lines.

Hough Transform

- The edge points are linked by determining if they lie on a curve of specified shape.
- Let us take the case of straight lines: we want to link points if they lie on a straight line.
- Consider a point (x_i,y_i), the equation of any line passing through this point is given by: $y_i = a x_i + b$, which can be written as: $b = - x_i a + y_i$. Therefore every point in *xy* plane corresponds to a straight line in *ab* plane.
- If there is a second point (x_j,y_j), another line with equation: $b = - x_j a + y_j$ is drawn in the *ab* plane.
- The intersection of the two lines in *ab* space give the values of $(a^{/},b^{/})$, which define a line passing through both points (x_i,y_i) and (x_j,y_j).

Hough Transform

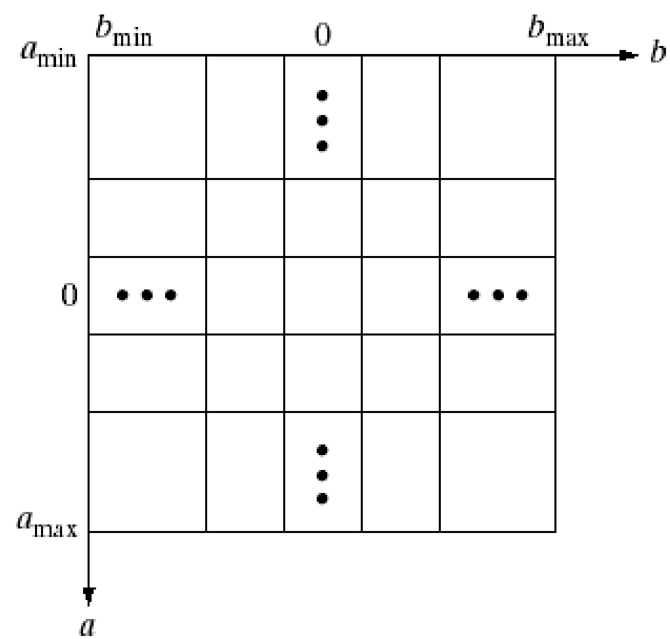


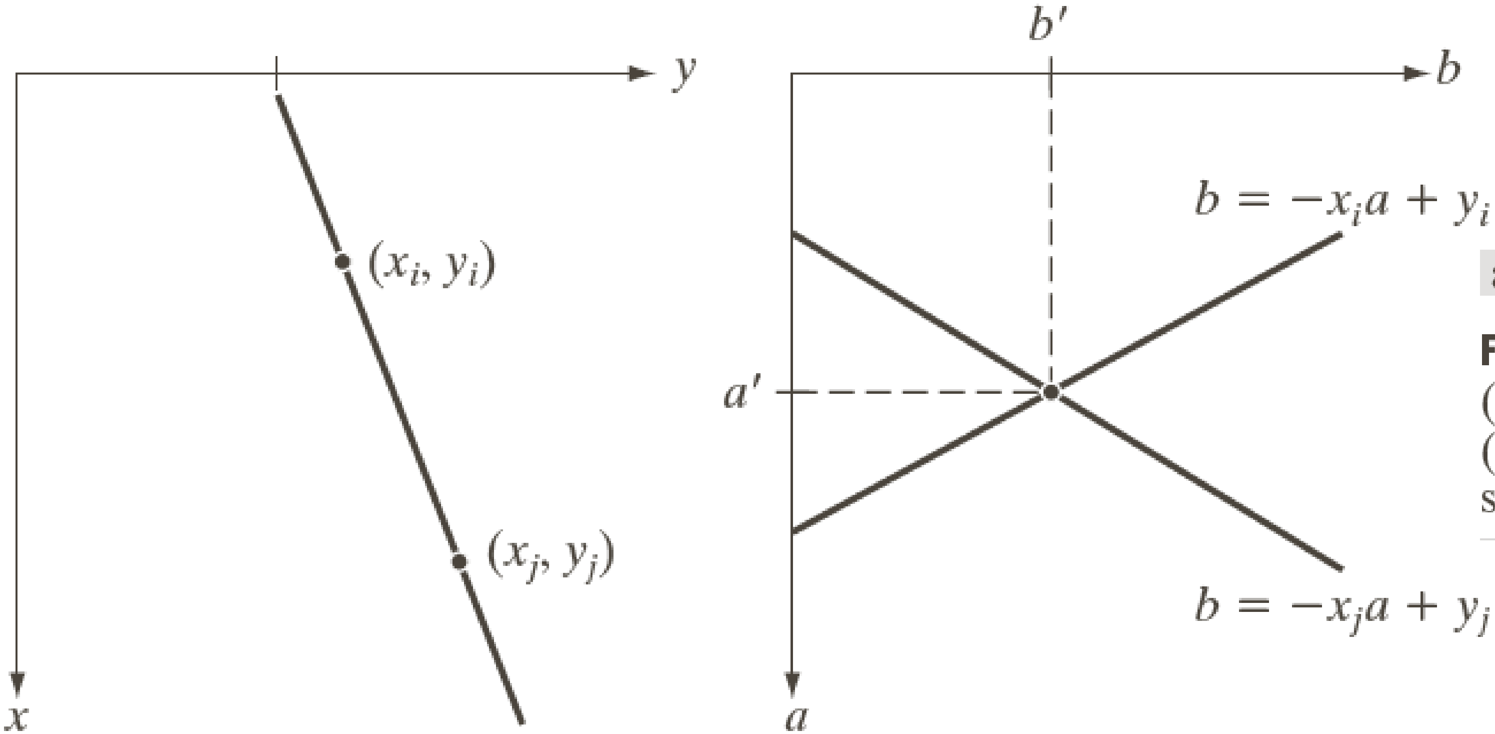
a b

FIGURE 10.17 (a) *xy*-plane. (b) Parameter space.

Implementation

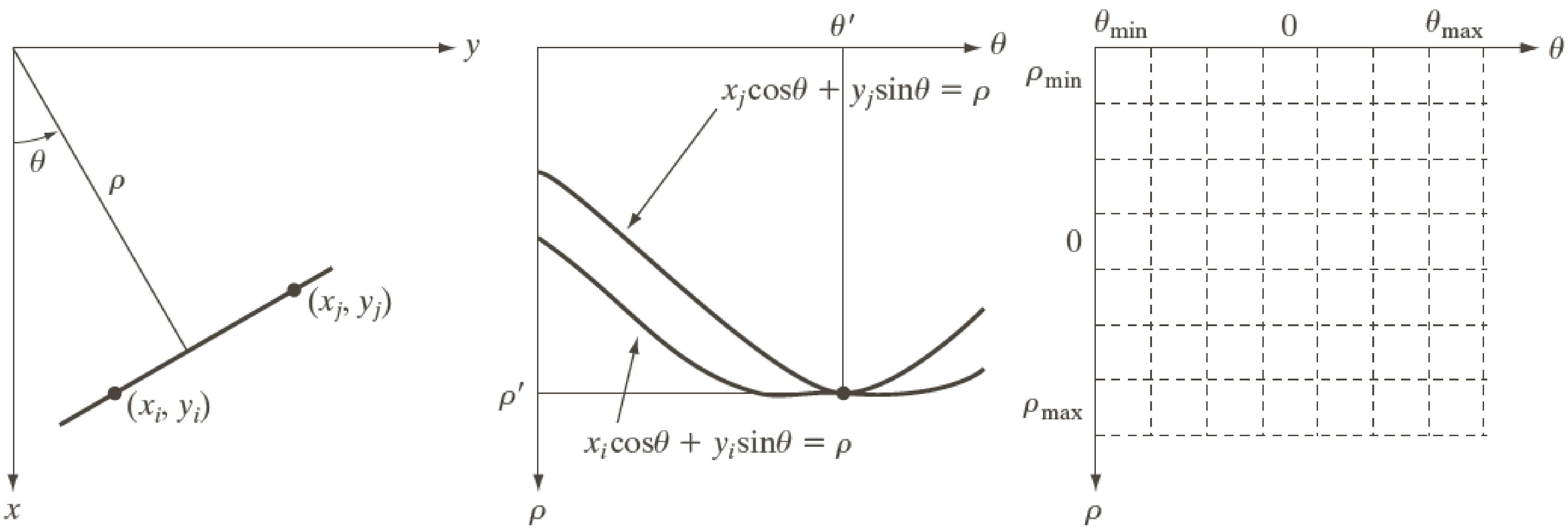
The parameter space *ab* is subdivided into the accumulator cells, where (a_{max},a_{min}) and (b_{max},b_{min}) are the expected ranges of slope and intercept values.





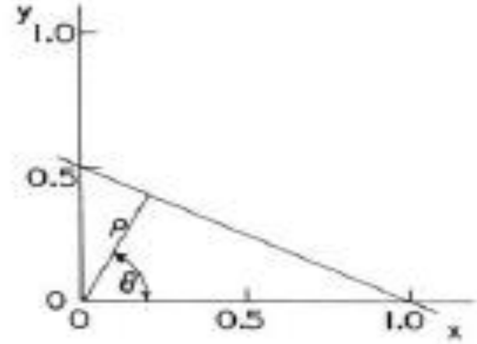
a b

FIGURE 10.31
(a) *xy*-plane.
(b) Parameter space.

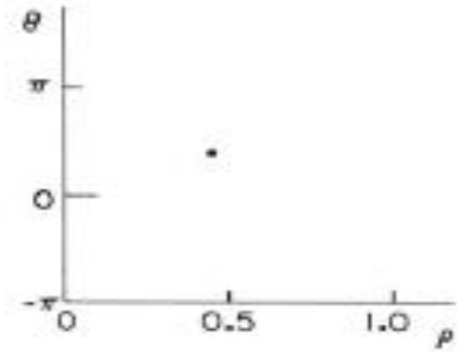


a b c

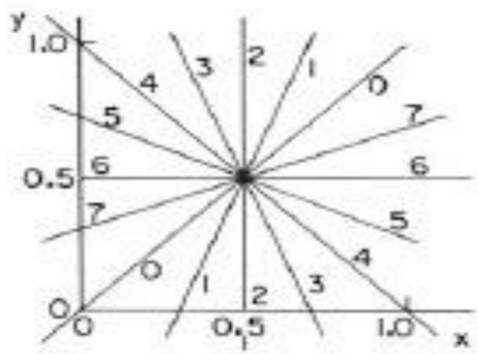
FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.



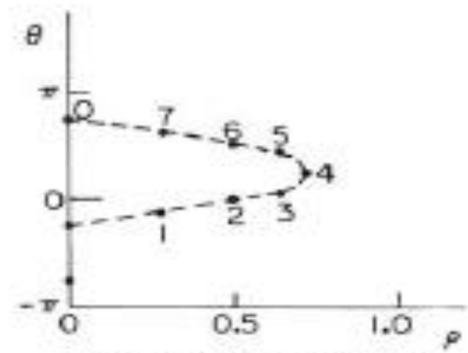
(*a*) Parametric line



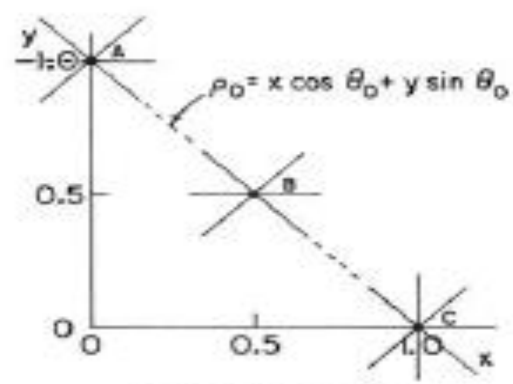
(*b*) Hough transform of (*a*)



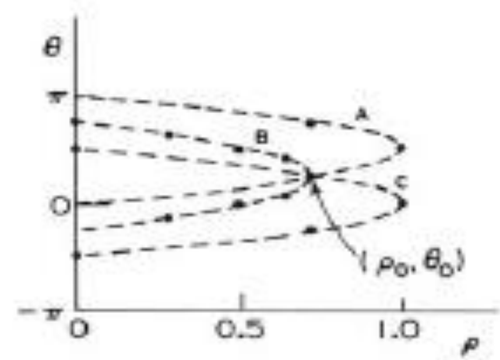
(*c*) Family of lines, common point



(*d*) Hough transform of (*c*)



(*e*) Colinear points



(*f*) Hough transform of (*e*)

Hough Transform

Implementation steps:

- Initially the accumulator cells are set to zero.
- Then for every point (x_k, y_k) in the image, a is varied over the allowed subdivision values and the corresponding b values are calculated using $b = -x_k a + y_k$.
- The resulting b are then rounded off to the allowed values of b.
- If a value of a_p results in solution b_q, we let the corresponding accumulator value $A(p,q) = A(p,q) + 1$.
- In the end the value of Q in $A(i,j)$ corresponds to Q points on the line $y = -a_i x + b_j$.

  Note: The number of subdivisions in the *ab* plane determines the accuracy of the colinearity of these points.

Hough Transform Implementation

1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	0

	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0

⇨

	-3	-2	-1	0	1	2	3
-3	0	0	0	1	0	0	0
-2	0	0	0	1	0	0	0
-1	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0
2	0	0	0	1	0	0	0
3	0	0	0	1	0	0	0

	-3	-2	-1	0	1	2	3
-3	0	0	0	1	0	0	0
-2	0	0	0	1	0	0	1
-1	0	0	0	1	0	1	0
0	0	0	0	1	1	0	0
1	0	0	0	2	0	0	0
2	0	0	1	1	0	0	0
3	0	1	0	1	0	0	0

⇨

	-3	-2	-1	0	1	2	3
-3	0	0	0	1	0	0	0
-2	0	0	0	1	0	0	1
-1	0	0	0	1	0	1	0
0	0	0	0	1	1	1	0
1	0	0	0	3	0	0	0
2	0	1	1	1	0	0	0
3	0	1	0	1	0	0	0

⇨ ....

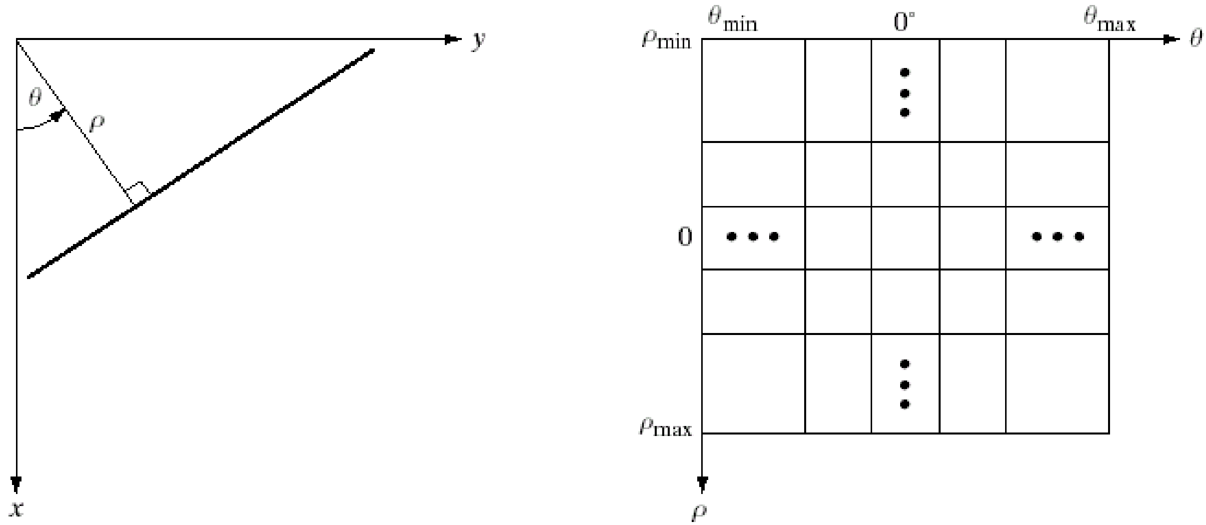
	-3	-2	-1	0	1	2	3
-3	0	0	0	1	0	0	0
-2	0	0	0	1	0	0	1
-1	0	0	0	1	0	1	0
0	0	0	0	1	1	1	2
1	0	0	1	5	0	0	0
2	1	1	1	1	0	0	0
3	0	1	0	1	0	0	0

Parameters a and b can take values between –infinity to +infinity

Hough Transform

- Limitation in using $y_i = a x_i + b$, as the representation of straight line is that slope approaches to infinity as the line approaches to be vertical.
- Therefore usually Hough Transform is implemented using the polar equation of straight line, i.e. $x\cos\theta + y\sin\theta = \rho$

  and instead of *ab*-plane $\rho\theta$-plane is used.
- Every point in image gives a sinusoidal curve in the $\rho\theta$-plane.

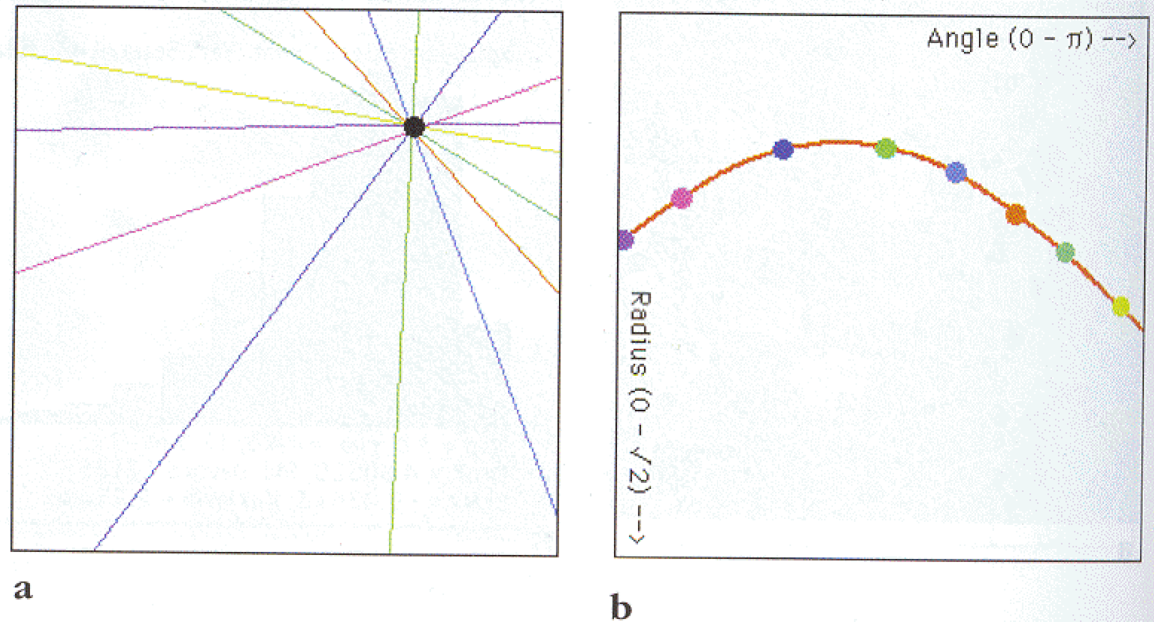


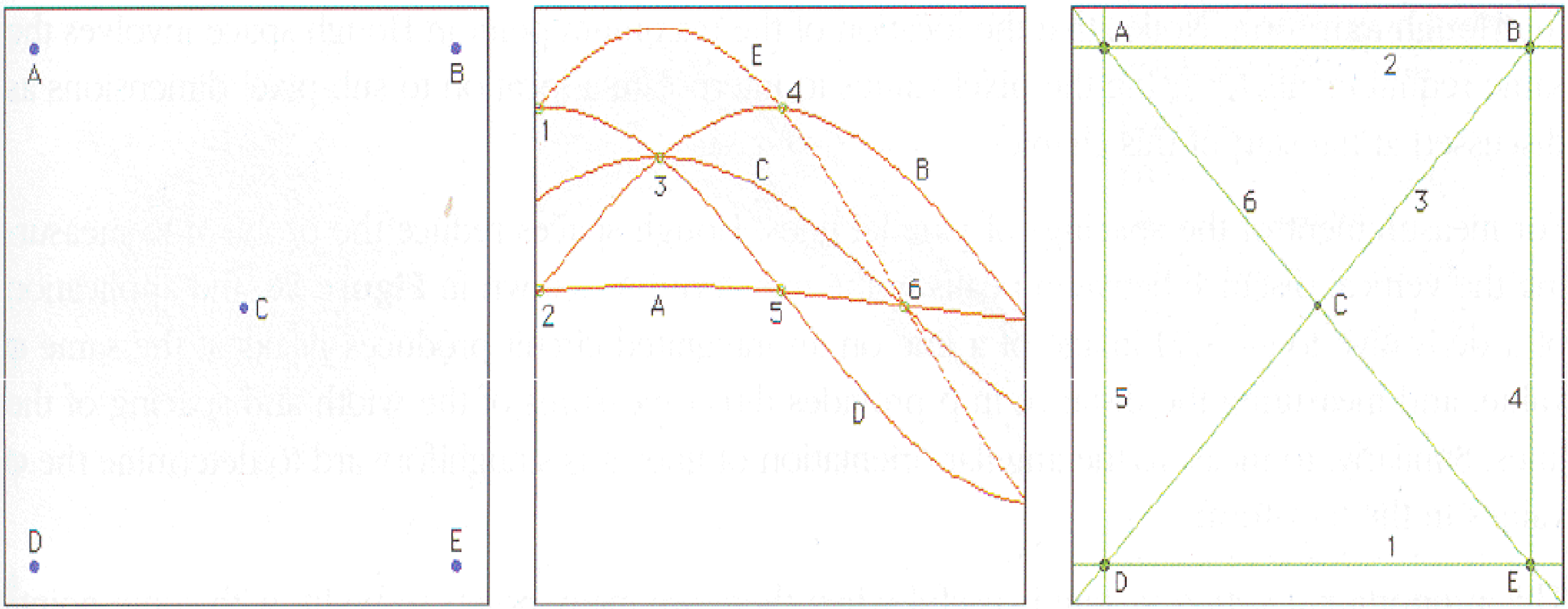
a b

FIGURE 10.19 (a) Normal representation of a line. (b) Subdivision of the $\rho\theta$-plane into cells.

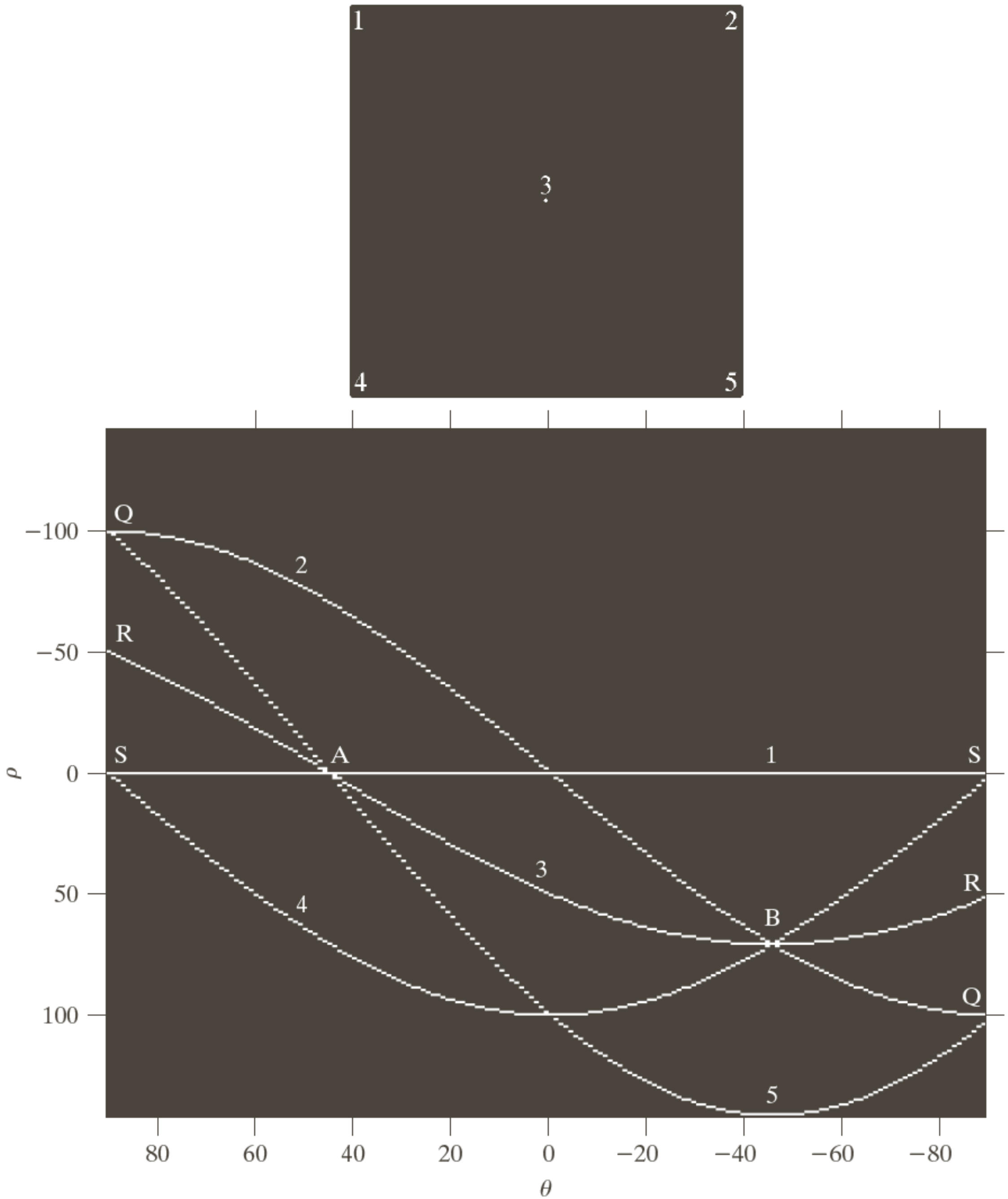
Hough Transform

Principle of the Hough transform. *Each point in the real-space image* ***(a)*** *produces a sinusoidal line in Hough space* ***(b)*** *representing all possible lines that can be drawn through it. Each point in Hough space corresponds to a line in real space. The real-space lines corresponding to a few of the points along the sinusoid are shown, with color coding to match them to the points.*





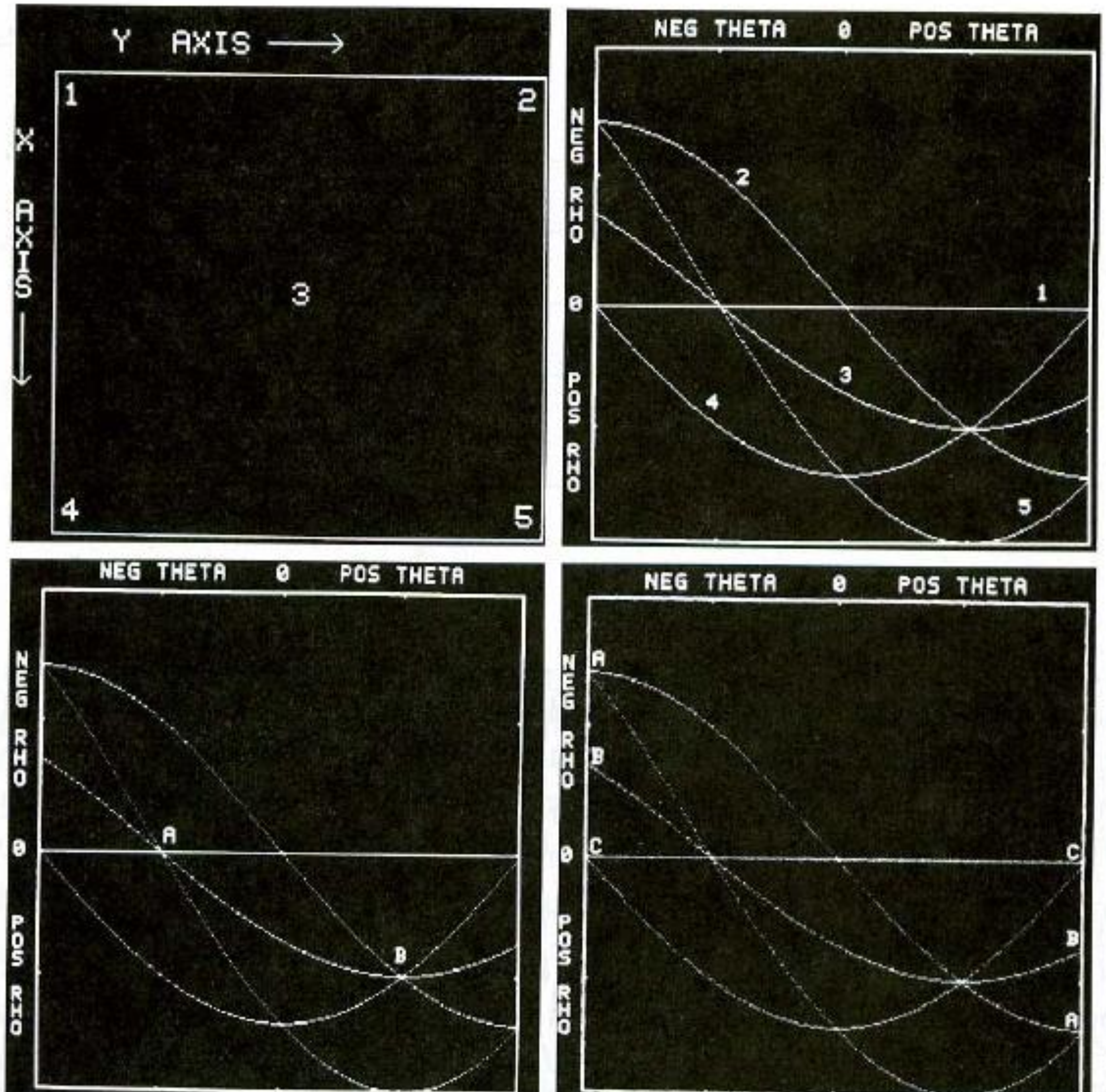
a b c



a
b

FIGURE 10.33 (a) Image of size 101 × 101 pixels, containing five points. (b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

Hough Transform



Implementation of the Hough transform

- Construct an array representing θ, ρ
- For each point, render the curve (θ, ρ) into this array, adding one at each cell
- Difficulties
  - how big should the cells be? (too big, and we cannot distinguish between quite different lines; too small, and noise causes lines to be missed)
- How many lines?
  - count the peaks in the Hough array
- This method can be extended to other type of curves also by using the equation for the desired curve, e.g. circle:

$$(x-c_1)^2+(y-c_2)^2=c_3^2$$

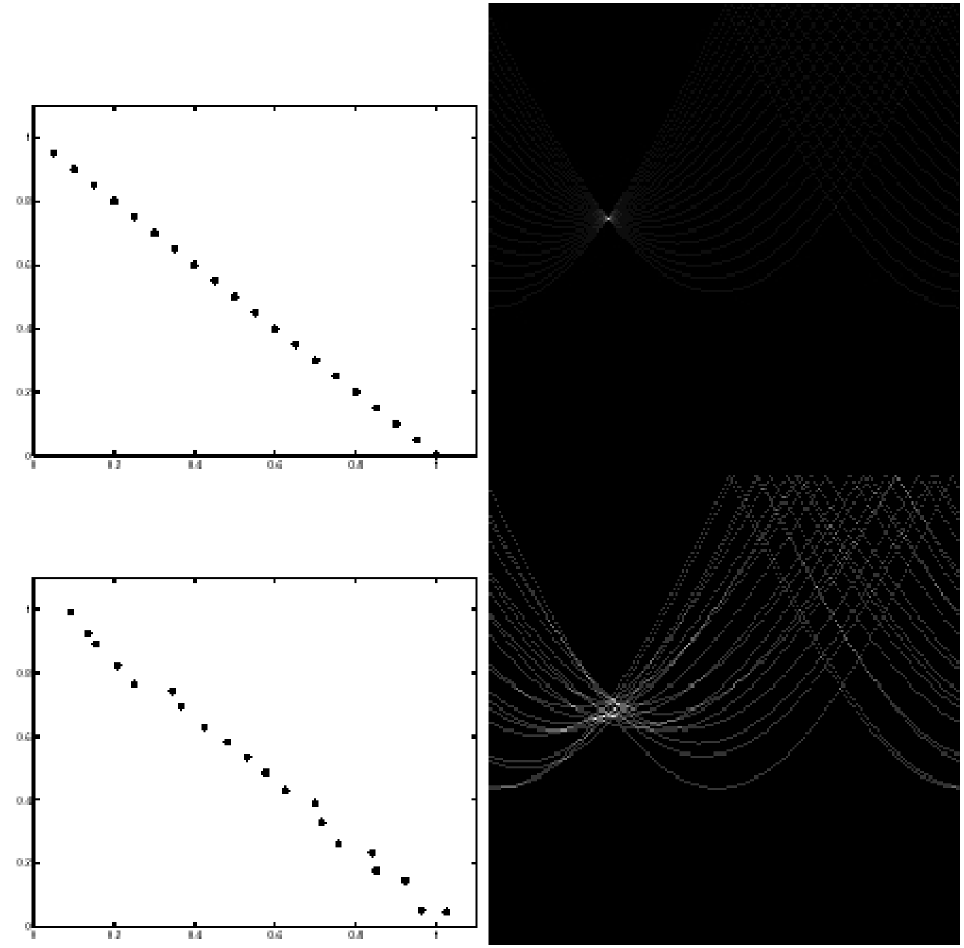


Figure 16.2. The Hough transform maps each point like token to a curve of possible lines (or other parametric curves) through that point. These figures illustrate the Hough transform for lines. The left hand column shows points, and the right hand column shows the corresponding accumulator arrays (the number of votes is indicated by the grey level, with a large number of votes being indicated by bright points). The top shows a set of 20 points drawn from a line next to the accumulator array for the Hough transform of these points. Corresponding to each point is a curve of votes in the accumulator array; the largest set of votes is 20. The horizontal variable in the accumulator array is θ and the vertical variable is r; there are 200 steps in each direction, and r lies in the range $[0, 1.55]$. In the center, these points have been offset by a random vector each element of which is uniform in the range $[0, 0.05]$; note that this offsets the curves in the accumulator array shown next to the points; the maximum vote is now 6.

Hough Transform

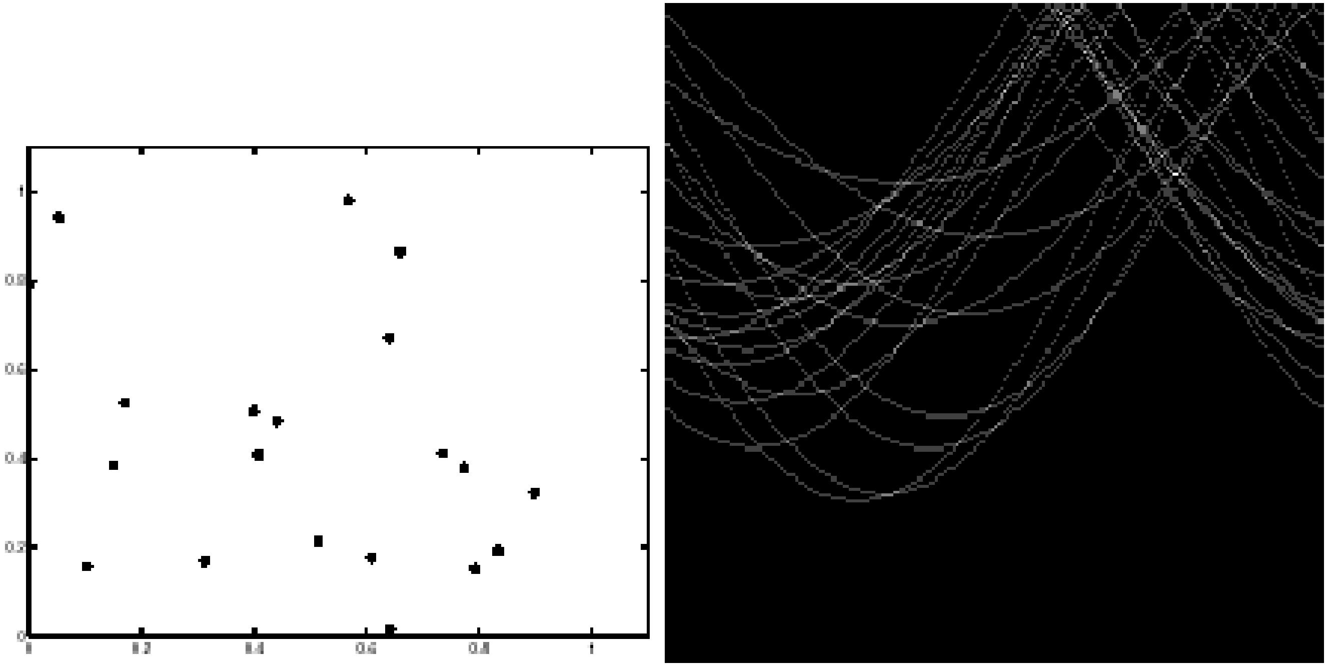
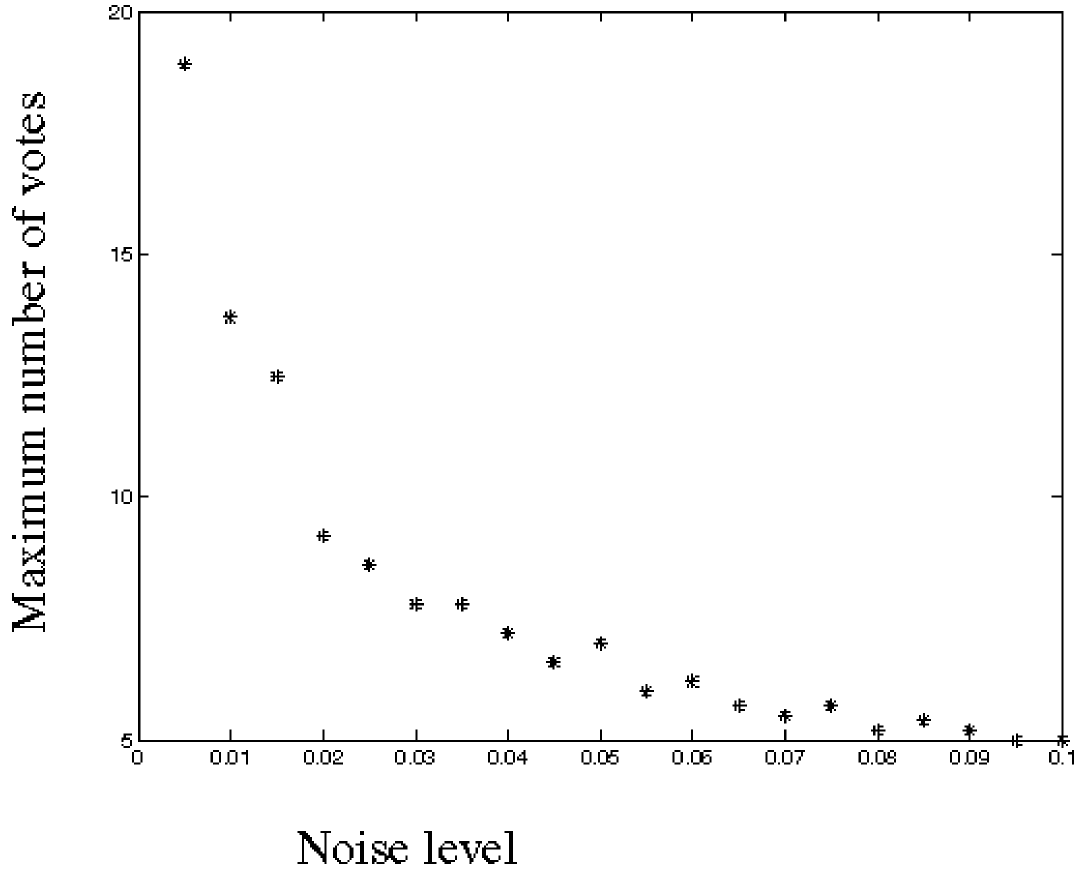


Figure 16.3. The Hough transform for a set of random points can lead to quite large sets of votes in the accumulator array. As in figure 16.2, the left hand column shows points, and the right hand column shows the corresponding accumulator arrays (the number of votes is indicated by the grey level, with a large number of votes being indicated by bright points). In this case, the data points are noise points (both coordinates are uniform random numbers in the range [0, 1]); the accumulator array in this case contains many points of overlap, and the maximum vote is now 4. Figures 16.4 and explore noise issues somewhat further.

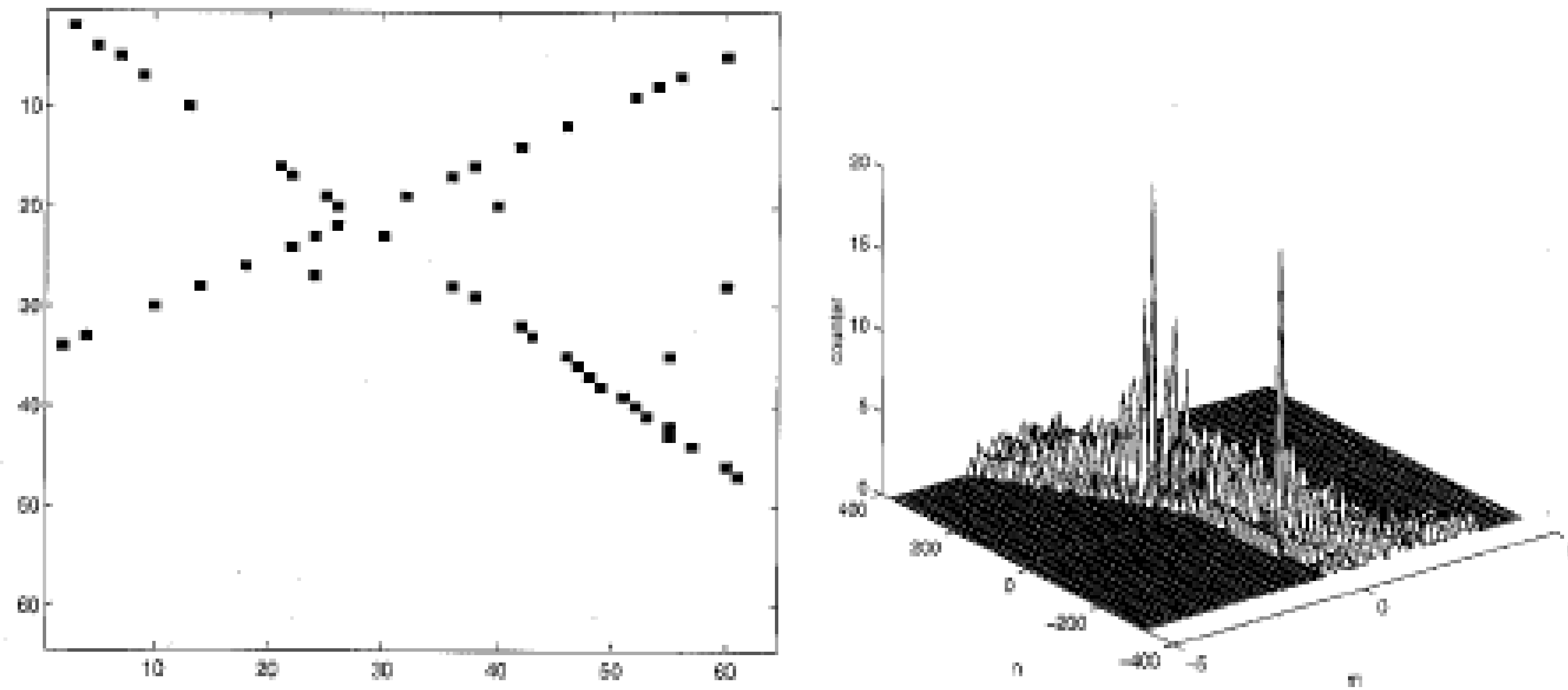
Noise Limitations of Hough Transform

- Two main limitations: noise and cell size



Results for a specimen of line with 20 points, with different amounts of noise

Hough Transform



(a) (b)

Figure 5.2 (a) An image containing two lines, sampled irregularly, and several random points. (b) Plot of the counters in the corresponding parameter space (how many points contribute to each cell (m, n)). Notice that the main peaks are obvious, but there are many secondary peaks.