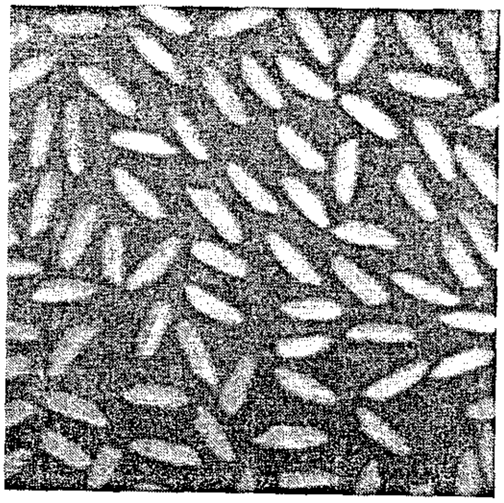
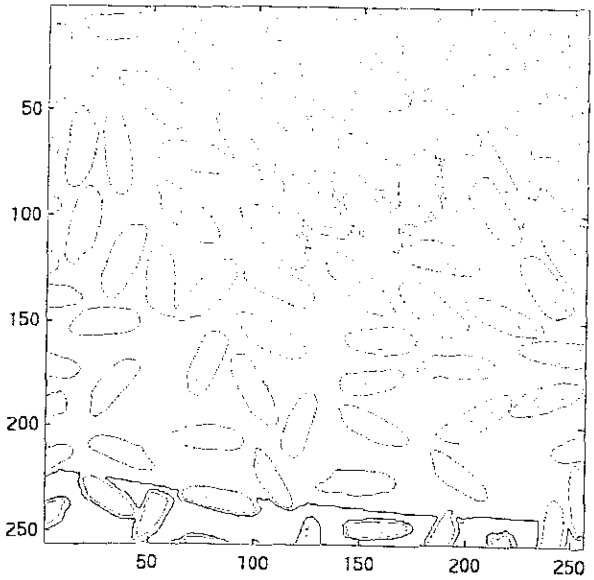
Image Contours

You can use the toolbox function imcontour to display a contour plot of the data in an intensity image. This function is similar to the contour function in MATLAB, but it automatically sets up the axes so their orientation and aspect ratio match the image.

This example displays a grayscale image of grains of rice and a contour plot of the image data.

```
I = imread('rice.tif');
imshow(I)
figure, imcontour(I)
```





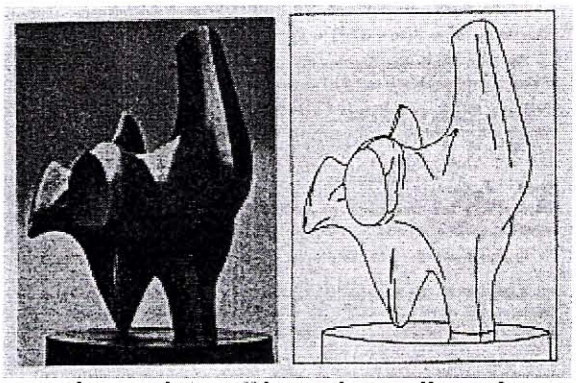
You can use the clabel function to label the levels of the contours. See the description of clabel in the online MATLAB Function Reference for details.

(a) (b)

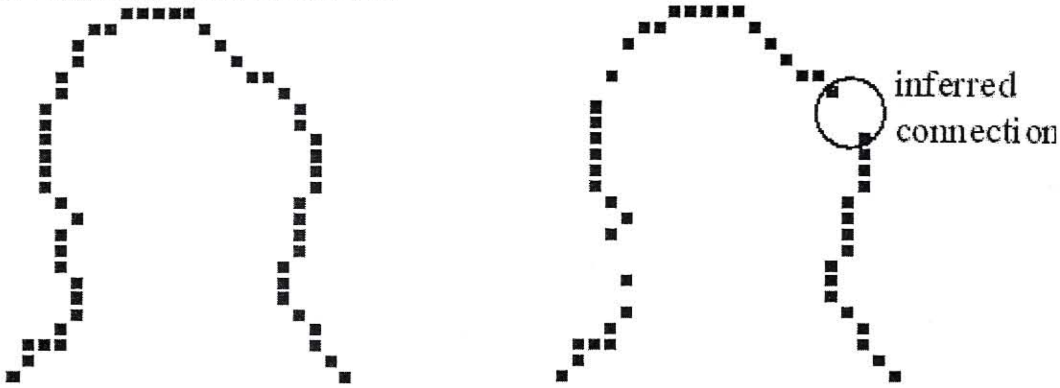
Figure 7.24 (a) Noisy image; (b) result of edge detection by using the heuristic graph search. (From Martelli [1976].)

Edge Linking: Difficulties

1. In general, it is difficult to determine precisely where a edge contour starts & where it stops



2. False negatives lead to "broken" edge segments that be connected somehow



Hough Transform

- Used to find lines
- Line
  - a collection of edge points that are adjacent and have the same direction
- Hough Transform
  - Takes edge points found by edge detector and find all lines on the edge points

16.5.2 Parameter Space; Hough Transform

The approach discussed here detects lines even if they are disrupted by noise or are only partially visible. We start by assuming that we have a segmented image that contains lines of this type. The fact that points lie on a straight line results in a powerful constraint that can be used to determine the parameters of the straight line. For all points $[x_n, y_n]^T$ on a straight line, the following condition must be met:

$$y_n = a_0 + a_1 x_n, \tag{16.7}$$

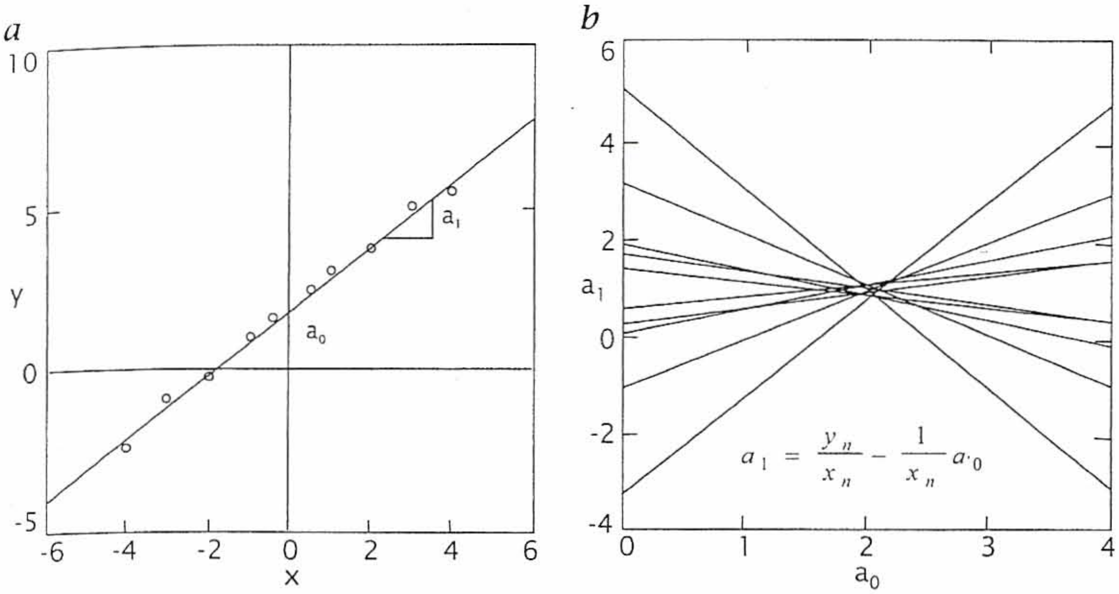


Figure 16.6: Hough transform for straight lines: the $[x, y]^T$ data space (a) is mapped onto the $[a_0, a_1]^T$ model space (b).

where a_0 and a_1 are the offset and slope of the line. We can read Eq. (16.7) also as a condition for the parameters a_0 and a_1:

$$a_1 = \frac{y_n}{x_n} - \frac{1}{x_n} a_0. \tag{16.8}$$

This is again the equation for a line in a new space spanned by the parameters a_0 and a_1. In this space, the line has the offset y_n/x_n and a slope of $-1/x_n$.

With one point given, we already cease to have a free choice of a_0 and a_1 as the parameters must satisfy Eq. (16.8).

The space spanned by the model parameters a_0 and a_1 is called the *model space*. Each point reduces the model space to a line. Thus, we can draw a line in the model space for each point in the data space, as illustrated in Fig. 16.6. If all points lie on a straight line in the data space, all lines in the model space meet in one point which gives the parameters a_0 and a_1 of the lines. As a line segment contains many points, we obtain a reliable estimate of the two parameters of the line. In this way, a line in the data space is mapped onto a point in the model space. This transformation from the data space to the model space via a model equation is called the *Hough transform*. It is a versatile instrument to detect lines even if they are disrupted or incomplete.

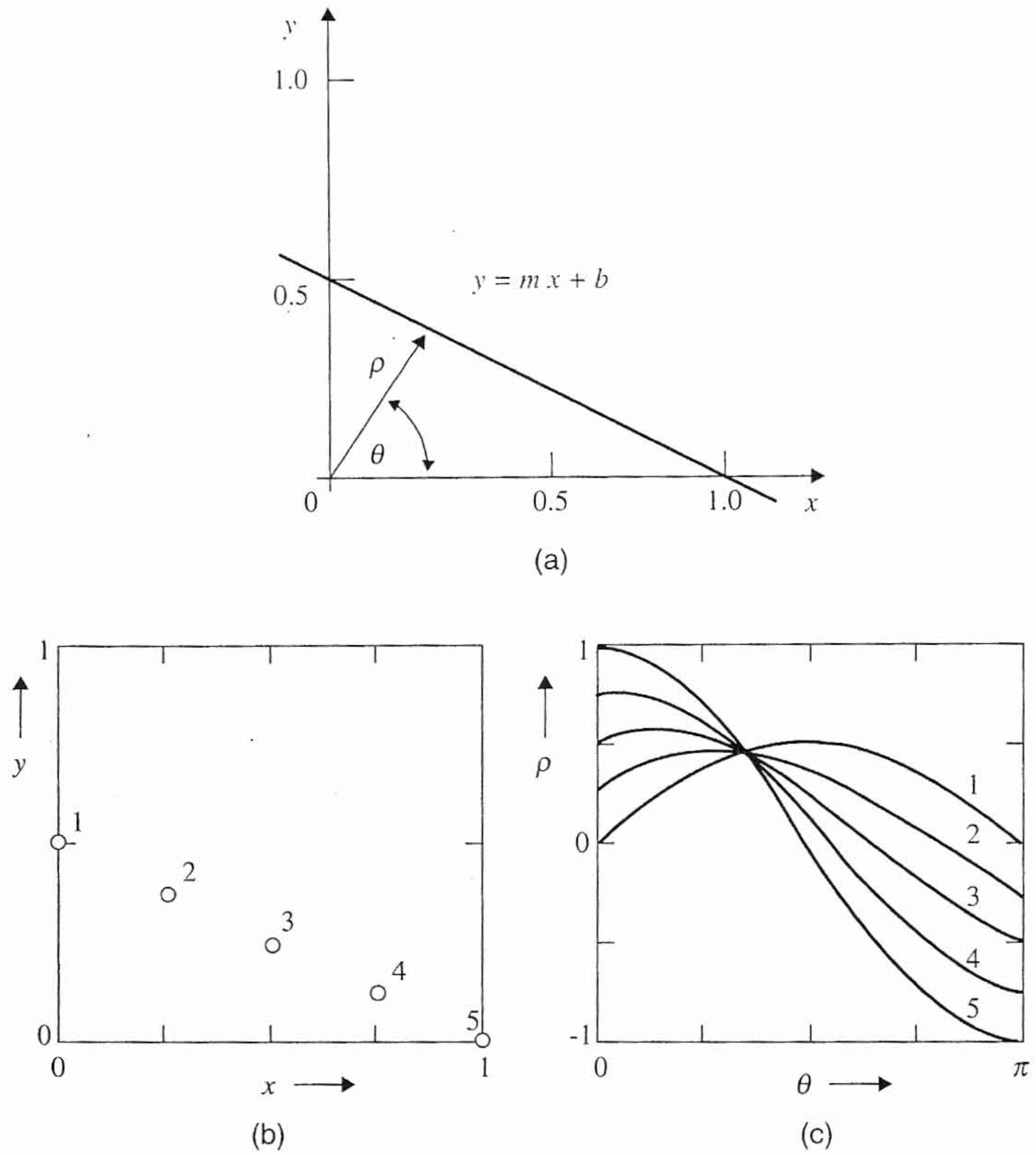


Figure 18–18 The Hough transform: (a) polar coordinate expression of a straight line; (b) x, y plane; (c) ρ, θ plane

Thus, to find the straight-line segment that the points fall upon, we can set up a two-dimensional histogram in ρ, θ space. For each edge point, (x_i, y_i), we increment all the histogram bins in ρ, θ space that correspond to the Hough transform (sinusoidal curve) for that point. When we have done this for all the edge points, the bin containing (ρ_0, θ_0) will be a local maximum. Thus, we search the ρ, θ space histogram for local maxima and obtain the parameters of linear boundary segments.

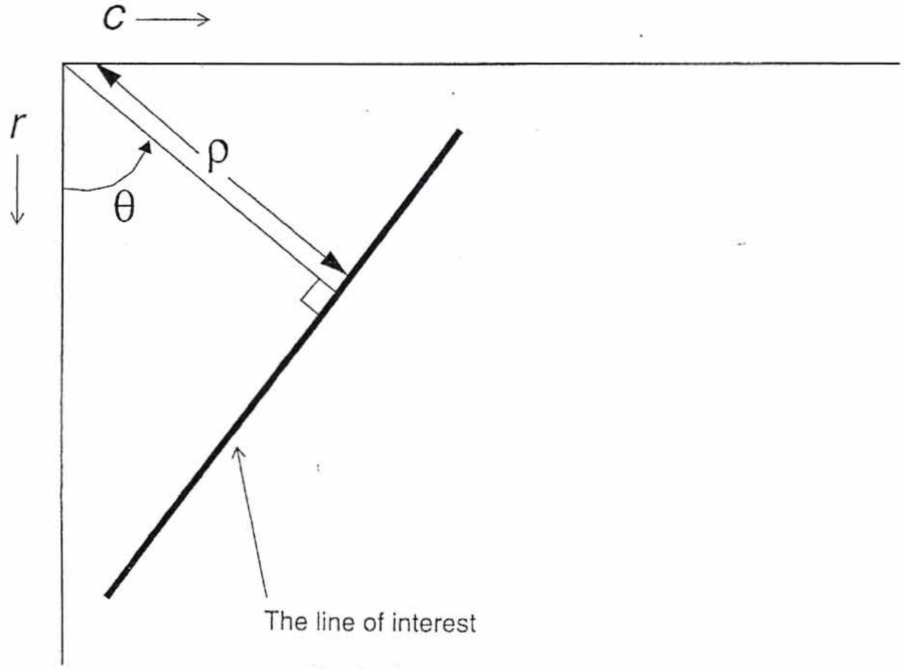
18.6 REGION GROWING

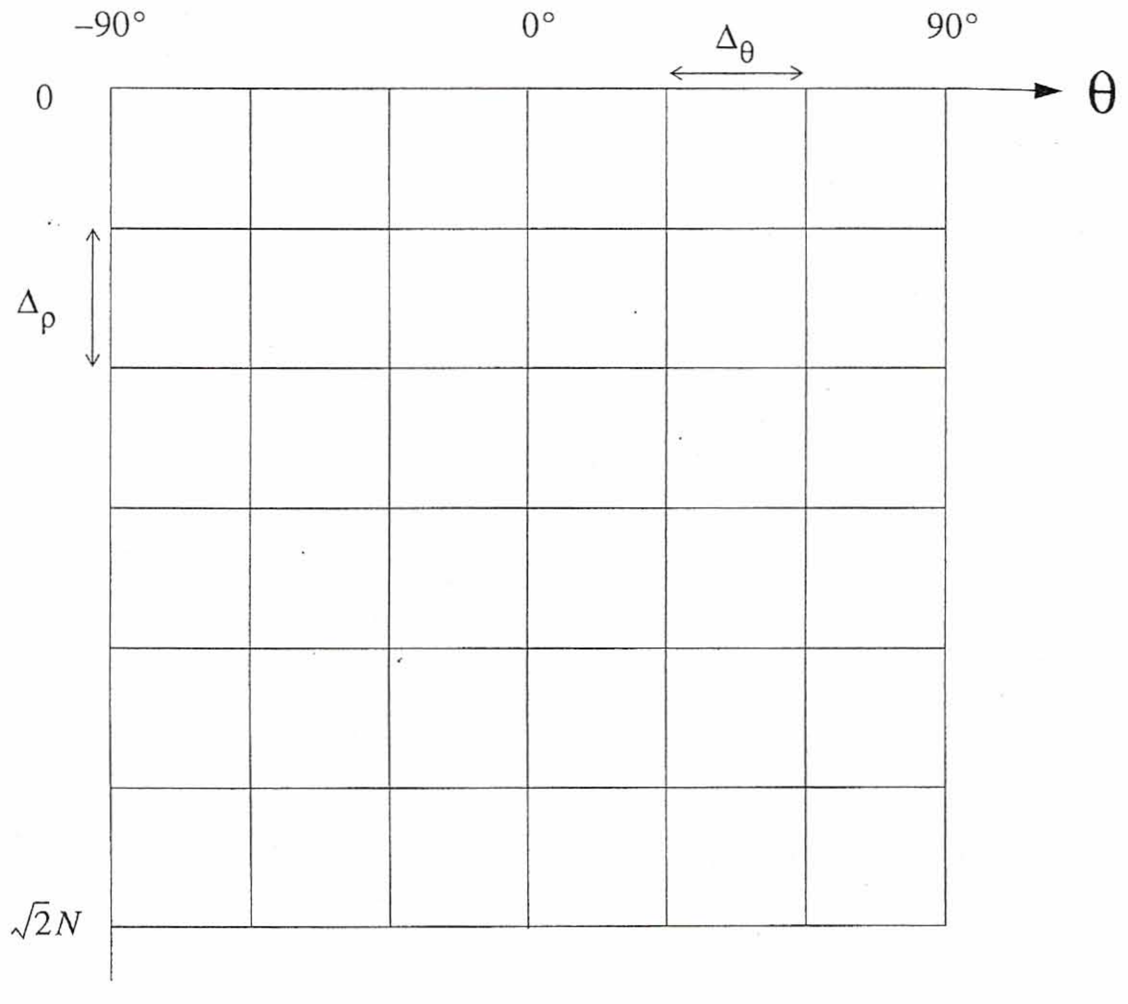
Region growing [24–27] is an approach to image segmentation that has received considerable attention in the computer vision segment of the artificial intelligence community. With this approach, one begins by dividing an image into many tiny regions. These initial regions may be small neighborhoods or even single pixels. In each region, suitably defined proper-

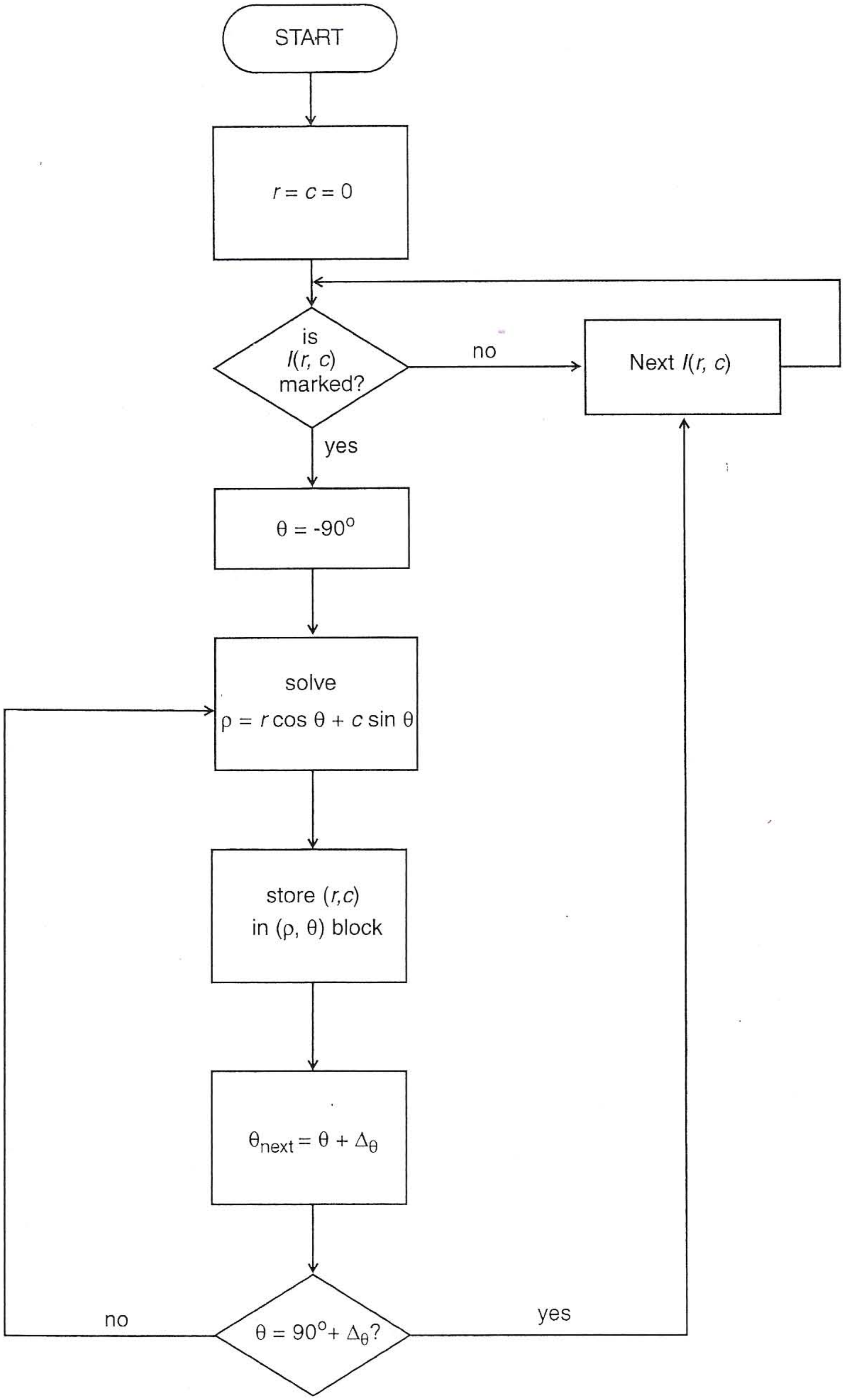
The algorithm used for the Hough transform (see Figure 2.3-9 for a flowchart of the process) will help understand what this means. The algorithm consists of three primary steps:

1. Define the desired increments on ρ and θ, Δ_ρ and Δ_θ, and quantize the space accordingly.
2. For every point of interest (typically points found by edge detectors that exceed some threshold value), plug the values for r and c into the line equation:

   Then, for each value of θ in the quantized space, solve for ρ.
3. For each ρθ pair from step 2, record the rc pair in the corresponding block in quantized space. This constitutes a hit for that particular block.





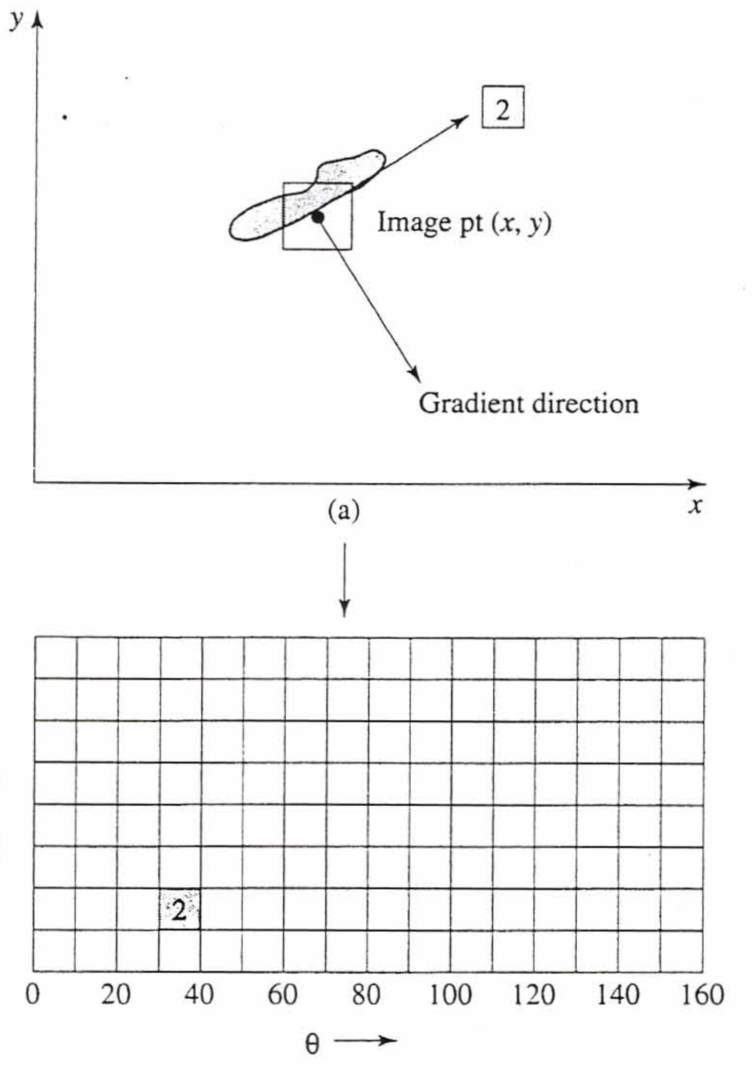


The flowchart is followed until all *I(r, c)* have been examined.

If a and b are quantized into Q values and n is the number of pixels in $I(x,y)$ then the brute force implementation of the Hough transform requires nQ invocations. This is somewhat expensive and an algorithm that requires just n invocations was developed by O'Gorman and Clowes. This tests for a straight line in a small neighbourhood of (x,y) using a measure of local edge strength and direction based on the gradient, and accumulates the appropriate $A(d,\theta)$ if an edge is found. The idea is illustrated in Figure 10.12. The algorithm can be summarized as:

for each point $I(x,y)$ in the image
  if strength_of_gradient > t **then**
    evaluate the direction of the line and increment appropriate $A(d,\theta)$

The Hough transform can be used as an edge linking scheme because it can locate the set of disjoint pixels that may be part of a fragmented line. Consider a structure in the image that exhibits gaps. This will map into a high value in the Hough transform. From such values and information on the structure that we are searching for, we can use cells in the Hough transform to direct us to potential gap pixels in the image. In other words, the Hough transform highlights the gaps in fragmented lines. The idea is shown in Figure 10.13 for a circular structure. A further examina-

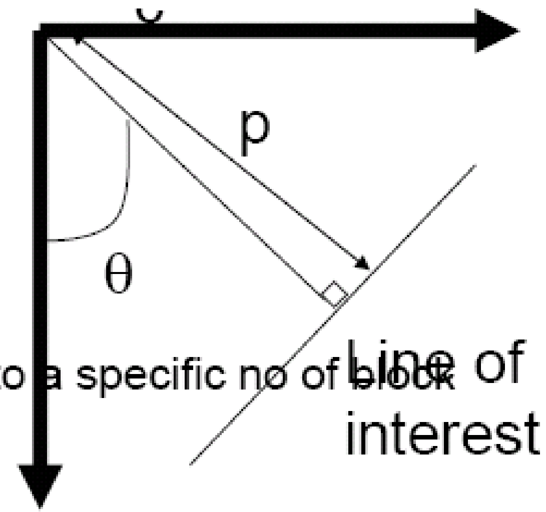


owes
f the
(a) The
and
ured
If the
than a
direction
he
ulators

(a) measure gradient strength + direction
(b) if gradient > T in direction 'd' add to accumulator

Hough Transform

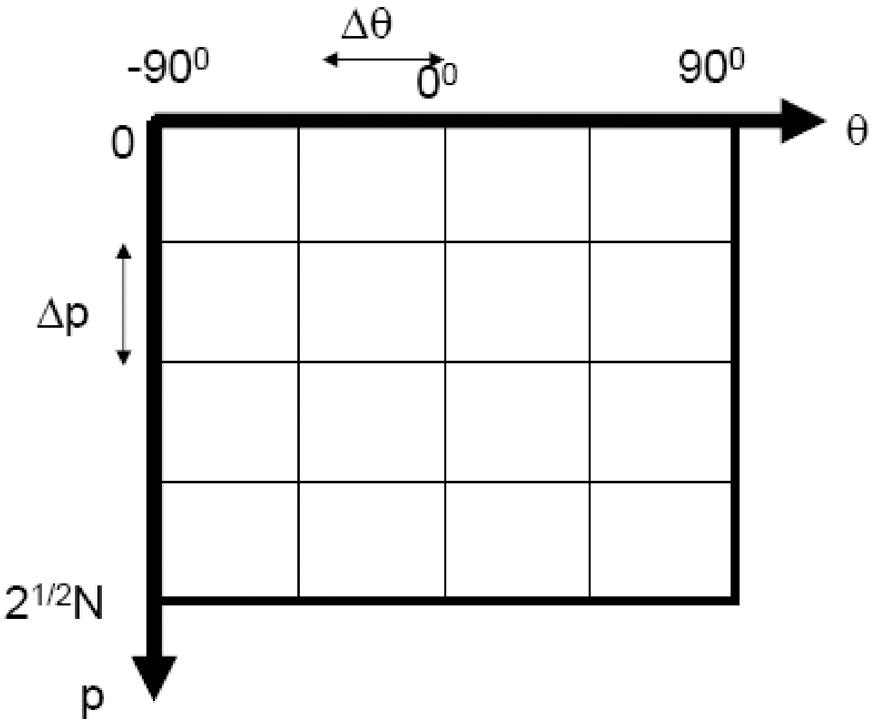
- $p = r \cos \theta + c \sin \theta$
- For each value of p and θ
  - Can define a particular line
    - Quantize $p\theta$ parameter by diving space into a specific no of block



- $-90^0 < \theta < 90^0$
- $0 < p < 2^{½} N$
- N = size of image

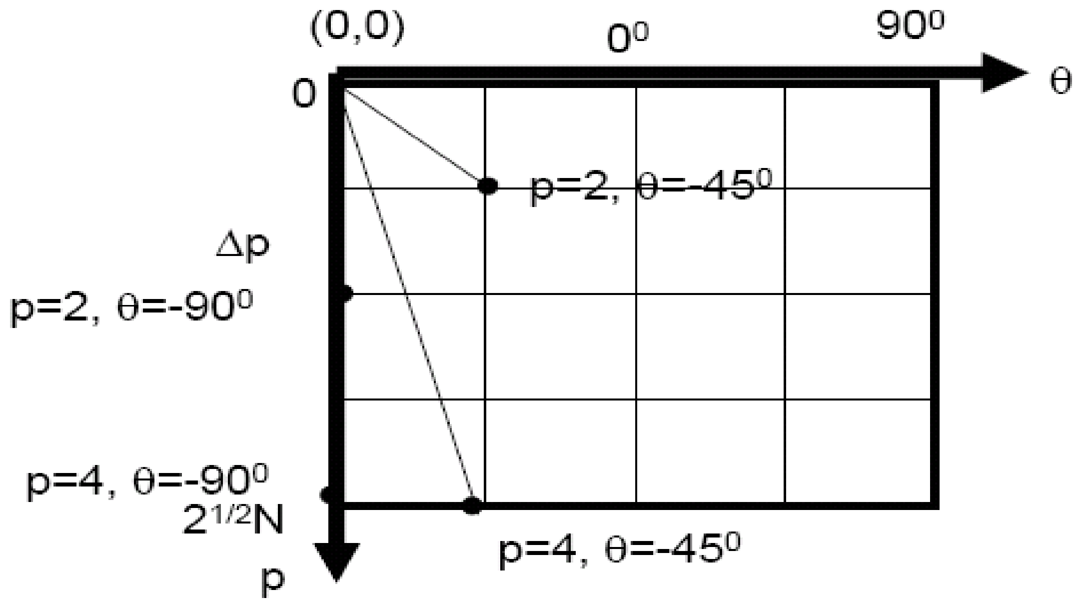
Hough Transform

- Quantize pθ parameter by diving space into a specific no of block



Steps in Hough Transform Step 1

- Define Δp and $\Delta\theta$
- Quantize the space accordingly
- $\Delta p = 2$
- $\Delta\theta = 45^0$

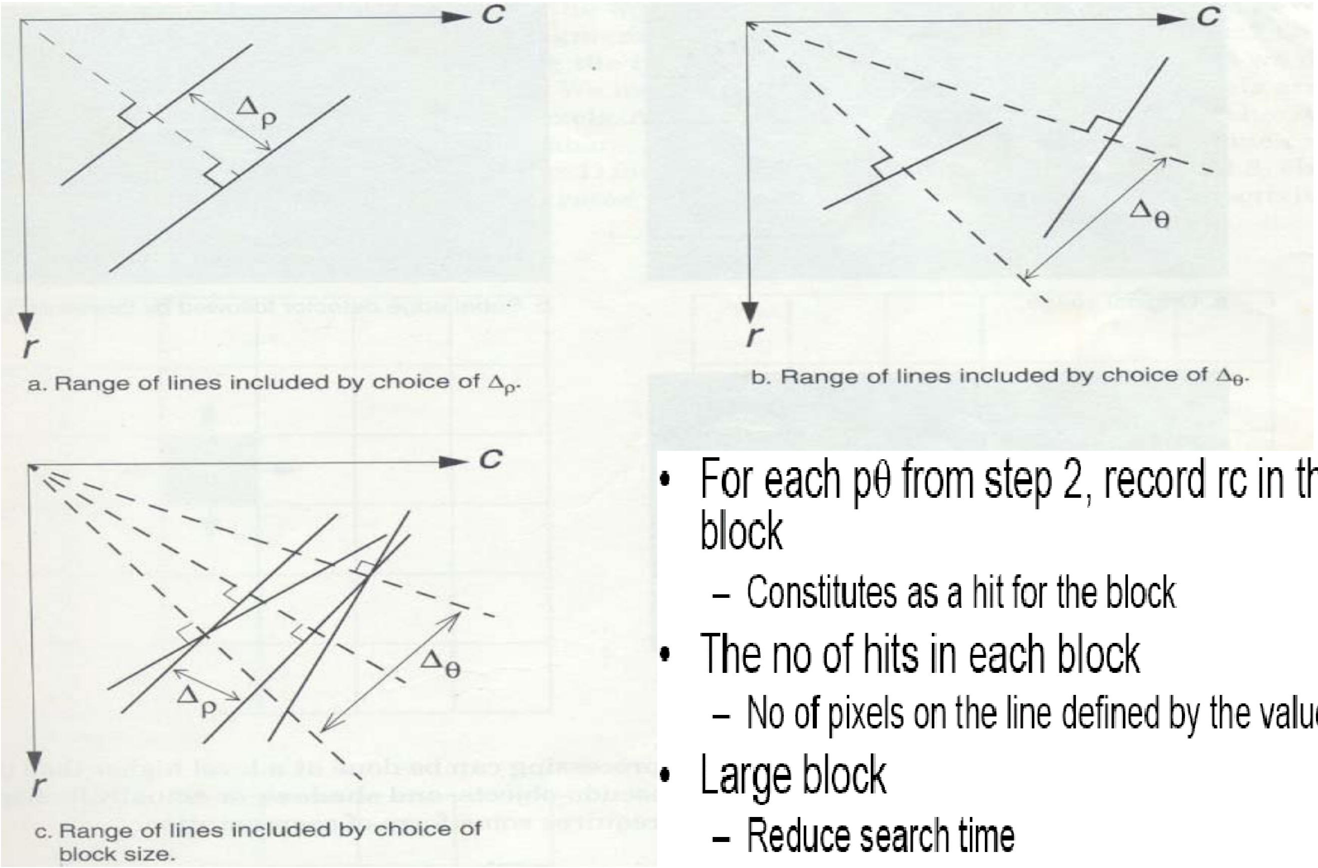


Steps in Hough Transform Step 2

- For every point of interest, put r and c into the equation
- (r,c) are from the edge detection process
  - $p = r \cos \theta + c \sin \theta$

- Example
- $\Delta\theta = 45^0$
- (r,c) = (1,1)
- $p = 1 \cos(-90^0) + 1 \sin(-90^0) = 0 + -1 = -1$
- $p = 1 \cos(-45^0) + 1 \sin(-45^0) = 0.707 + (-0.707) = 0$
- $p = 1 \cos(0^0) + 1 \sin(0^0) = 1 + 0 = 1$
- $p = 1 \cos(45^0) + 1 \sin(45^0) = 0.707 + 0.707 = 1.5$
- $p = 1 \cos(90^0) + 1 \sin(90^0) = 0 + 1 = 1$

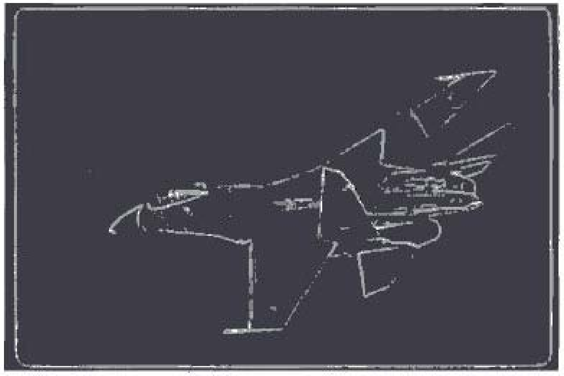
Steps in Hough Transform Step 3



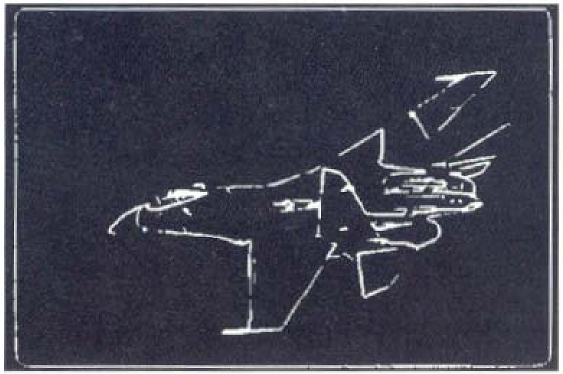
- For each pθ from step 2, record rc in the corresponding block
  - Constitutes as a hit for the block
- The no of hits in each block
  - No of pixels on the line defined by the value of p and θ
- Large block
  - Reduce search time
  - Low resolution

Steps in Hough Transform Step 4

a. Original image.



b. Sobel edge detector followed by thresholding.



c. Hough output—$\Delta_\theta = 1°$, $\Delta_\rho = 1$, threshold = 20.

- Select a threshold value and examine the quantization blocks more than the threshold value
- Look for continuity
  - Distance between the points on line
- Line marked as output image

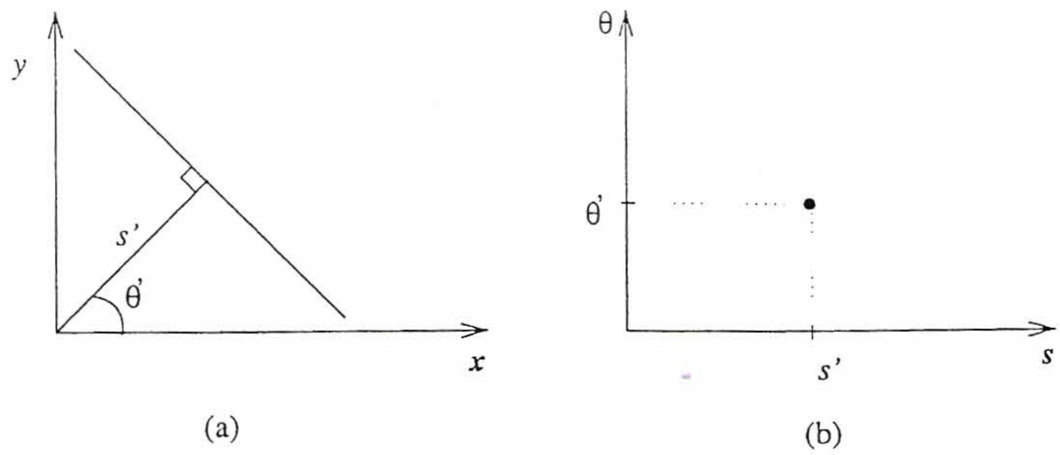


Figure 5.36: *Hough transform in s, θ space: (a) straight line in image space; (b) s, θ parameter space.*

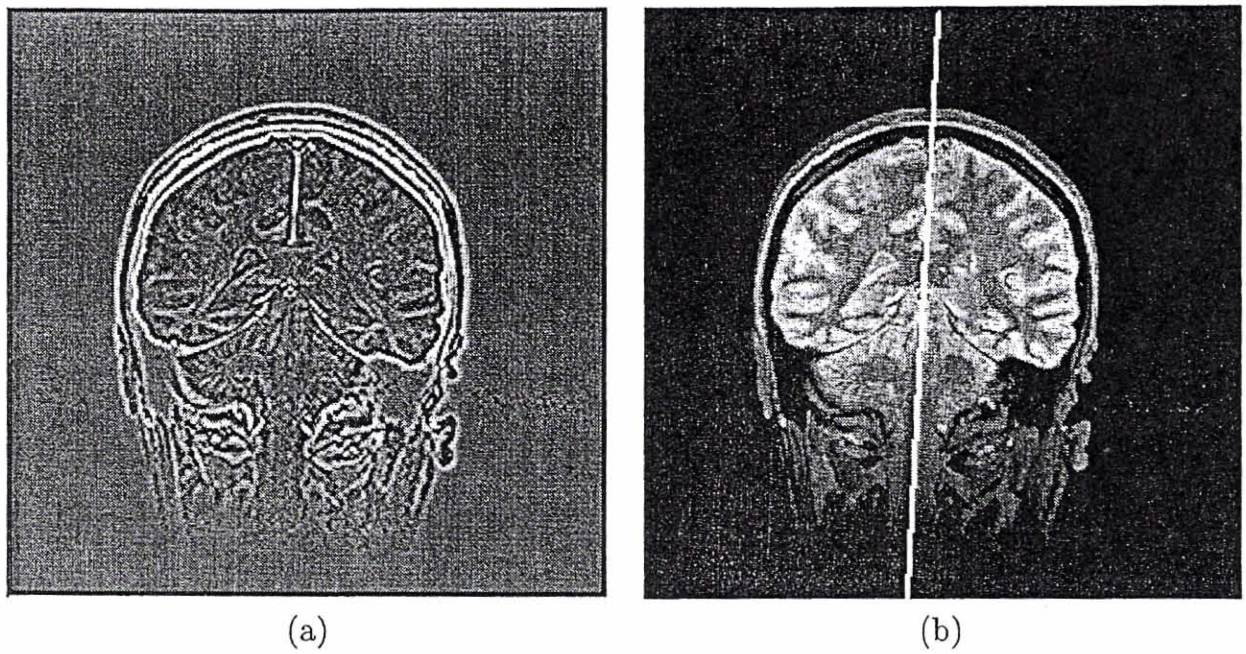


Figure 5.37: *Hough transform line detection used for MRI brain segmentation to the left and right hemispheres: (a) edge image; (b) segmentation line in original image data.*

If we are looking for circles, the analytic expression $f(\mathbf{x}, \mathbf{a})$ of the desired curve is

$$(x_1 - a)^2 + (x_2 - b)^2 = r^2 \tag{5.27}$$

where the circle has center (a, b) and radius r. Therefore, the accumulator data structure must be three-dimensional. For each pixel $\mathbf{x}$ whose edge magnitude exceeds a given threshold,

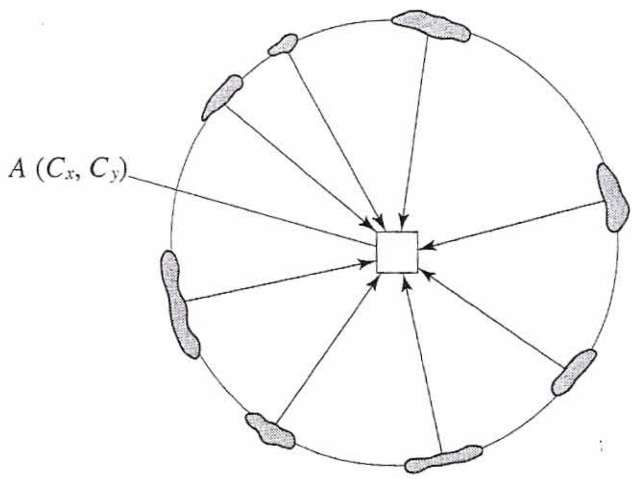


Figure 10.13
Hough transform technique for circles: any structures lying on the circumference of a circle with given radius have gradients at that point in the direction of the centre.

tion of the gaps can then be made in the image domain. Alternatively we can say that the value A gives us a direct numerical indication of the likelihood of a circular structure being present even if it appears in the image as a broken form.

10.3.3 Hough transform – circles

To extend the Hough transform to find circles or any other parametrized curve is straightforward. Consider finding the evidence for all circular structures in $I(x,y)$ of a fixed radius r. Look again at Figure 10.13. If the structures form themselves into a circle then the components will have gradient directions that point to the centre of the circle. Points (x,y) on the circumference of a circle centred at (Cx,Cy) satisfy

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

The algorithm can be summarized as:

for each point (x,y) in the image
  if strength_of_gradient > t **then**
    evaluate direction of gradient and
    increment $A(Cx,Cy)$ where

$$Cx = x - r\cos\theta$$
$$Cy = x - r\sin\theta$$

This approach is easily extended to detect circles of any radius by extending the dimension of A to 3 and adding an extra loop to the algorithm.

10.4 Edges and segmentation – explicit edge detection or boundary following

It would seem self-evident that a good way to segment an image is to use boundaries or edges. However, as we have seen, even an edge emphasis image that has points accumulated is equal to the number of edge pixels in the image*: this represents a significant saving in computational load. For this to be possible, the edge detection operator that is employed must be highly accurate. Fortunately, the Sobel operator is able to estimate edge orientation to 1° and is very simple to apply (Chapter 5). Thus the revised form of the transform is viable in practice.

As was seen in Chapter 5, once the Sobel convolution masks have been applied, the local components of intensity gradient g_x and g_y are available, and the magnitude and orientation of the local intensity gradient vector can be computed using the formulae:

$$g = (g_x^2 + g_y^2)^{1/2} \tag{9.1}$$

and

$$\theta = \arctan(g_y/g_x) \tag{9.2}$$

However, use of the arctan operation is not necessary when estimating centre location coordinates (x_c,y_c), since the trigonometric functions can be made to cancel out:

$$x_c = x - R(g_x/g) \tag{9.3}$$

$$y_c = y - R(g_y/g) \tag{9.4}$$

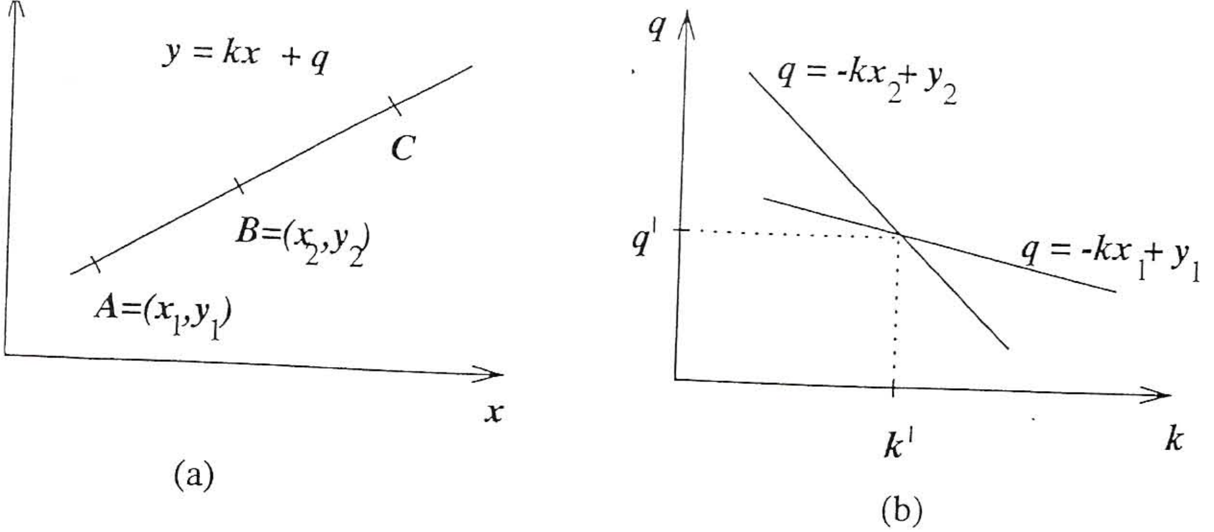
the values of $\cos\theta$ and $\sin\theta$ being given by:

$$\cos\theta = g_x/g \tag{9.5}$$

$$\sin\theta = g_y/g \tag{9.6}$$

In addition, the usual edge thinning and edge linking operations—which normally require considerable amounts of processing—can be avoided if a little extra smoothing of the cluster of candidate centre points can be tolerated (Davies, 1984c) (Table 9.2). Thus this Hough-based approach can be a very efficient one for locating the centres of circular objects, virtually all the superfluous operations having been eliminated, leaving only edge detection, location of candidate centre points and centre point averaging to be carried out. It is also relevant that the method is highly robust, in that if part of the boundary of an object is obscured or distorted, the object centre is

* We assume that objects are known to be *either* lighter *or* darker than the background, so that it is only necessary to move along the edge normal in one direction.



5.34: *Hough transform principles: (a) image space; (b) k, q parameter space.*

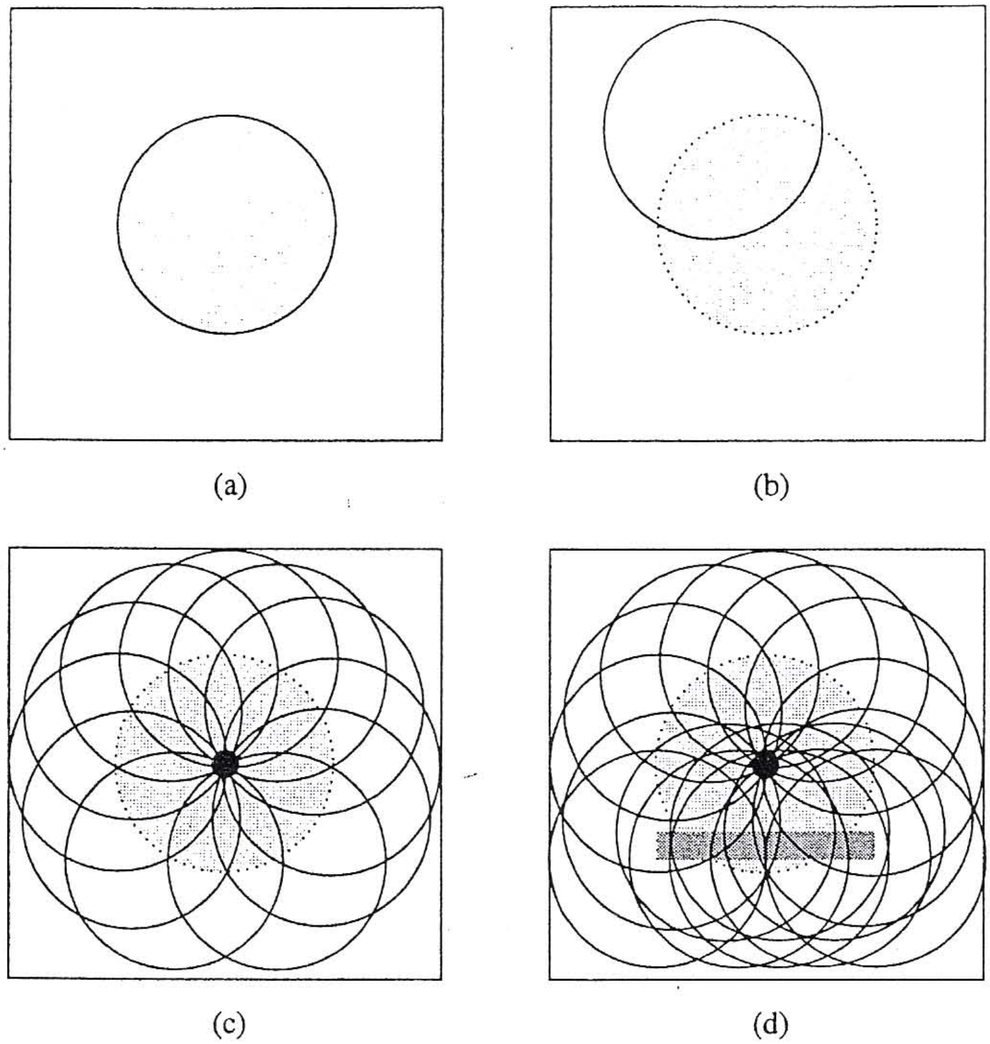


Figure 5.33: *Hough transform - example of circle detection: (a) original image of a dark circl (known radius r) on a bright background. (b) For each dark pixel, a potential circle-cente locus is defined by a circle with radius r and center at that pixel. (c) The frequency wit which image pixels occur in the circle-center loci is determined—the highest-frequency pixe represents the center of the circle (marked by •). (d) The Hough transform correctly detect the circle (marked by •) in the presence of incomplete circle information and overlappin structures. (See Figure 5.38 for a real-life example.)*

Table 9.2 A Hough-based procedure for locating circular objects.

1. Locate edges within the image
2. Link broken edges
3. Thin thick edges
4. For every edge pixel, find a candidate centre point
5. Locate all clusters of candidate centres
6. Average each cluster to find accurate centre locations

This procedure is particularly robust. It is largely unaffected by shadows, image noise, shape distortions and product defects. Note that stages 1–3 of the procedure are identical to stages 1–3 in Table 9.1. However, in the Hough-based method, computation can be saved, and accuracy actually increased, by omitting stages 2 and 3.

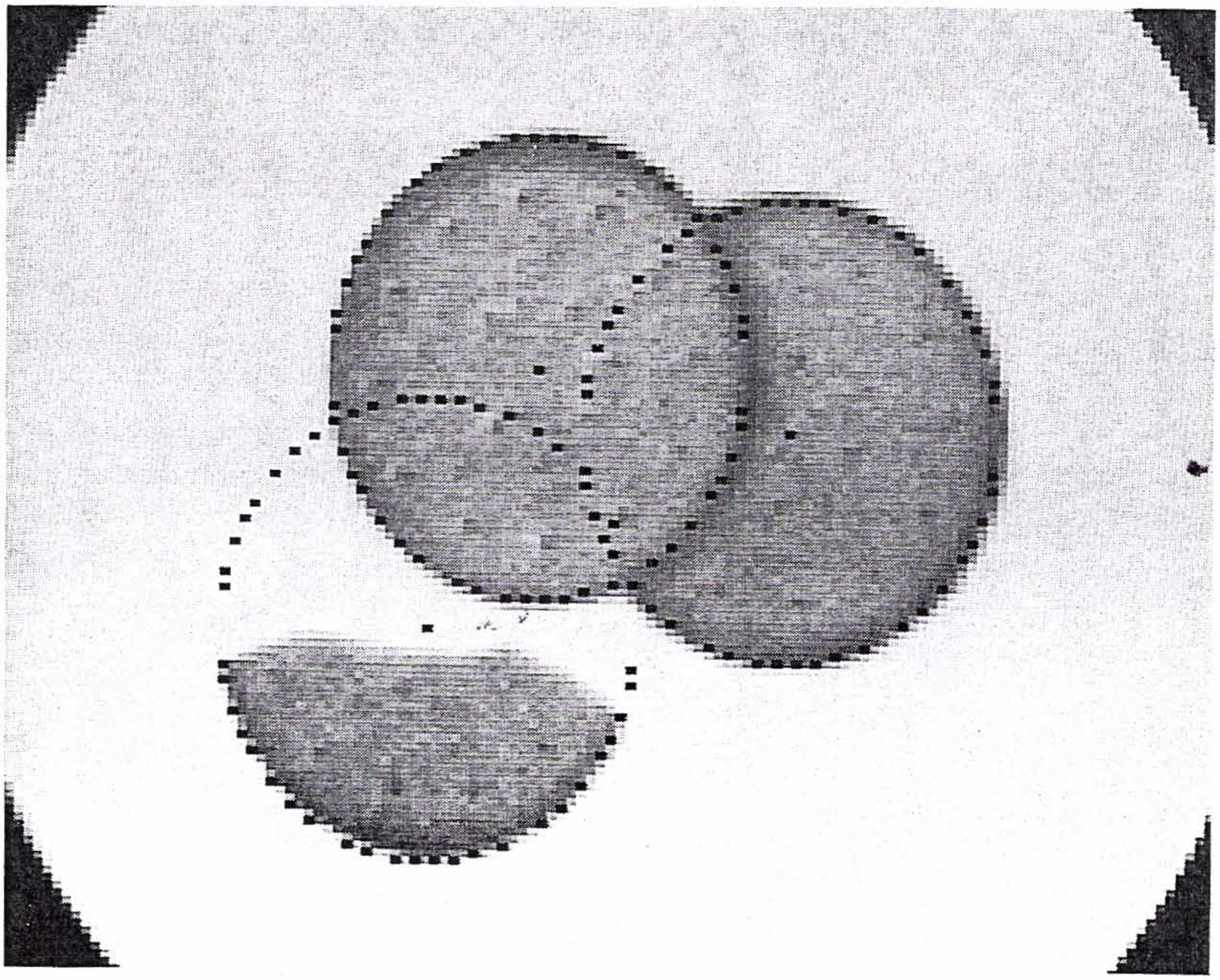


Fig. 9.1 Location of broken and overlapping biscuits, showing the robustness of the centre location technique. Accuracy is indicated by the black dots which are each within 1/2 pixel of the radial distance from the centre.

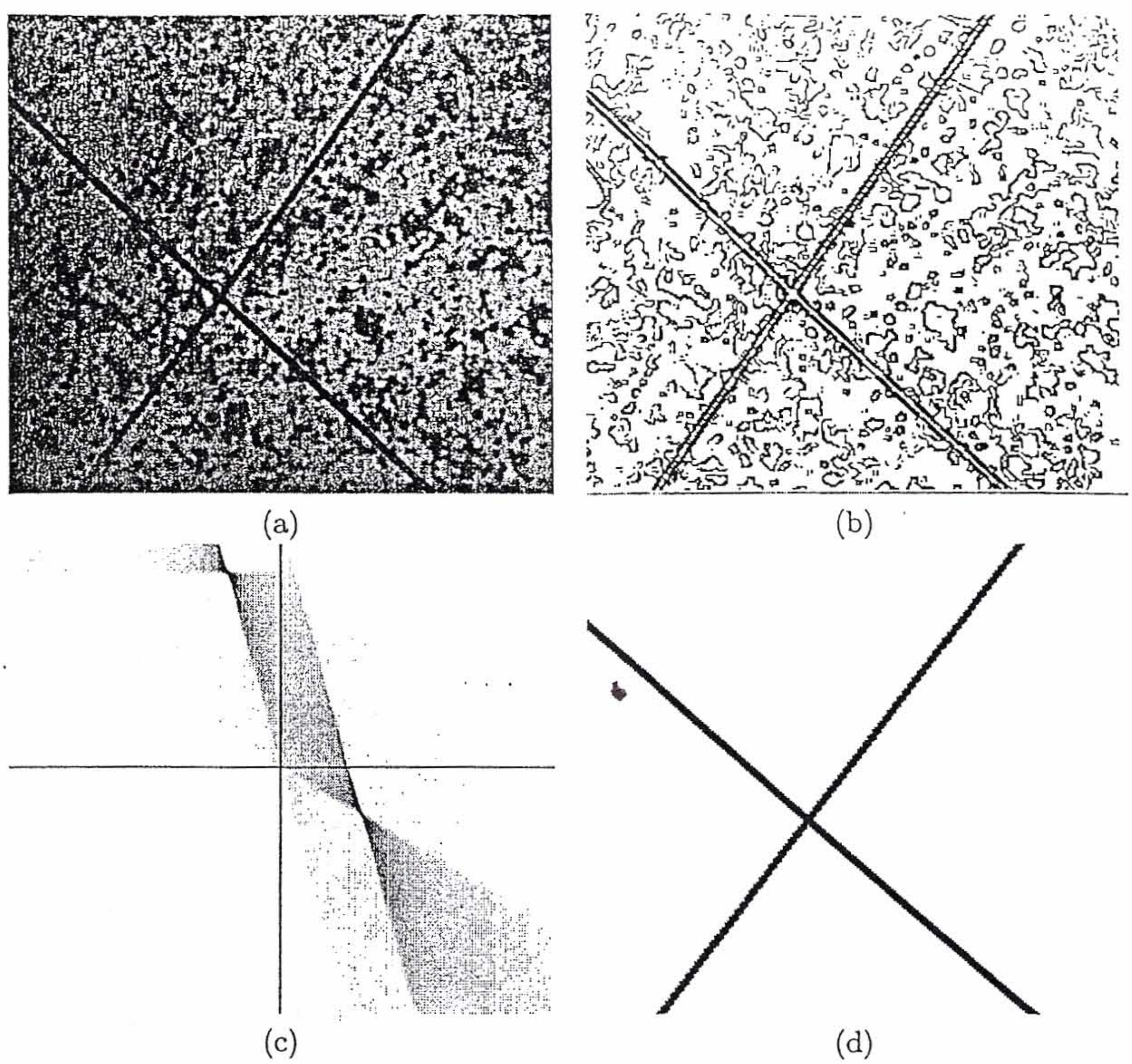


Figure 5.35: *Hough transform—line detection: (a) original image; (b) edge image (note many edges, which do not belong to the line); (c) parameter space; (d) detected lines.*

Algorithm 5.15: Curve detection using the Hough transform

1. Quantize parameter space within the limits of parameters $\mathbf{a}$. The dimensionality n of the parameter space is given by the number of parameters of the vector $\mathbf{a}$.

2. Form an n-dimensional accumulator array $A(\mathbf{a})$ with structure matching the quantization of parameter space; set all elements to zero.

3. For each image point (x_1, x_2) in the appropriately thresholded gradient image, increase all accumulator cells $A(\mathbf{a})$ if $f(\mathbf{x}, \mathbf{a}) = 0$

$$A(\mathbf{a}) = A(\mathbf{a}) + \Delta A$$

for all $\mathbf{a}$ inside the limits used in step 1.

4. Local maxima in the accumulator array $A(\mathbf{a})$ correspond to realizations of curves $f(\mathbf{x}, \mathbf{a})$ that are present in the original image.

gorithm 5.16: Generalized Hough transform

Construct an R-table description of the desired object.

Form a data structure A that represents the potential reference points

$$A(x_1, x_2, S, \tau)$$

Set all accumulator cell values $A(x_1, x_2, S, \tau)$ to zero.

For each pixel (x_1, x_2) in a thresholded gradient image, determine the edge direction $\Phi(\mathbf{x})$; find all potential reference points $\mathbf{x}^R$ and increase all $A(\mathbf{x}^R, S, \tau)$,

$$A(\mathbf{x}^R, S, \tau) = A(\mathbf{x}^R, S, \tau) + \Delta A$$

for all possible values of rotation and size change,

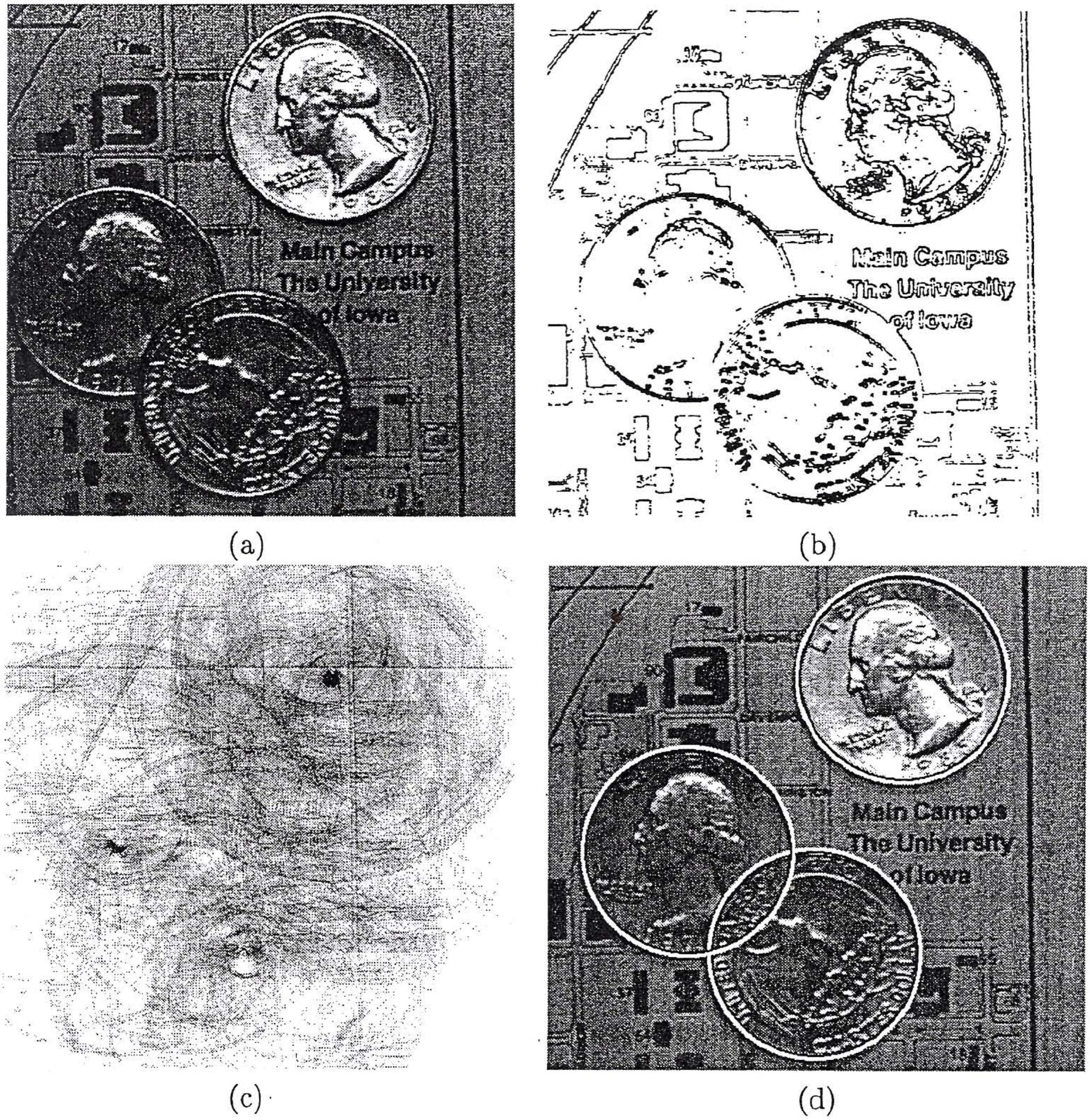
$$x_1^R = x_1 + r(\phi) S \cos(\alpha(\phi) + \tau)$$

$$x_2^R = x_2 + r(\phi) S \sin(\alpha(\phi) + \tau)$$

. The location of suitable regions is given by local maxima in the A data structure.

The Hough transform was initially developed to detect analytically defined shapes, such ines, circles, or ellipses in general images, and the generalized Hough transform can be d to detect arbitrary shapes. However, even the generalized Hough transform requires the nplete specification of the exact shape of the target object to achieve precise segmentation. erefore, it allows detection of objects with complex but pre-determined, shapes. Other ieties exist that allow detection of objects whose exact shape is unknown, assuming a ori knowledge can be used to form an approximate model of the object [Philip 91].

The Hough transform has many desirable features [Illingworth and Kittler 88]. It recog- es partial or slightly deformed shapes, therefore behaving extremely well in recognition of cluded objects. It may be also used to measure similarity between a model and a detected ject on the basis of size and spatial location of peaks in the parameter space. The Hough insform is very robust in the presence of additional structures in the image (other lines, rves, or objects) as well as being insensitive to image noise. Moreover, it may search for



5.38: *Hough transform—circle detection: (a) original image; (b) edge image (note that ge information is far from perfect); (c) parameter space; (d) detected circles.*