7 – SEGMENTATION

Segmentation is the process that partitions a scene into its constituent parts or objects. These components are then subsequently analyzed. There are two main approaches:

Based on discontinuity:

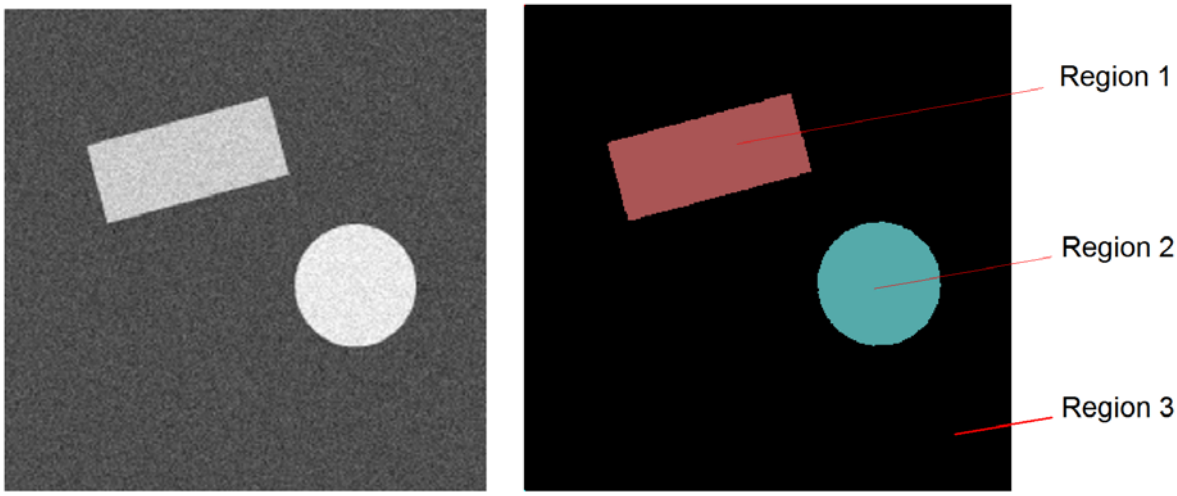
- Contour tracking
- Local analysis – edge linking
- Global analysis – Hough transform

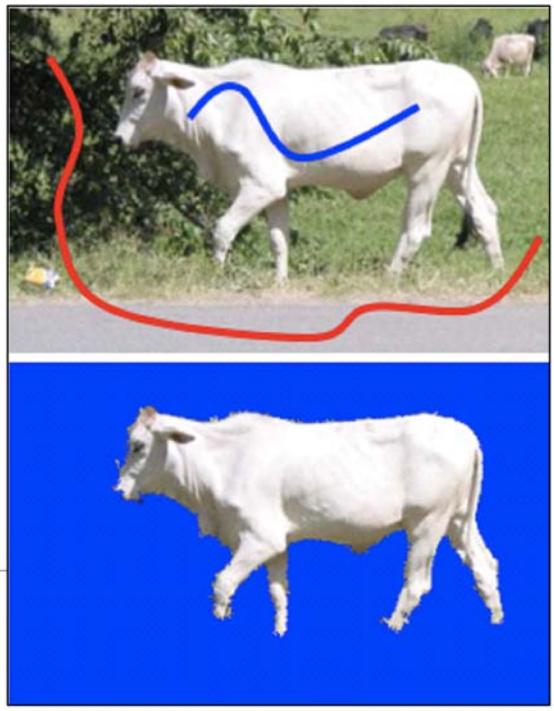
Based on similarity:

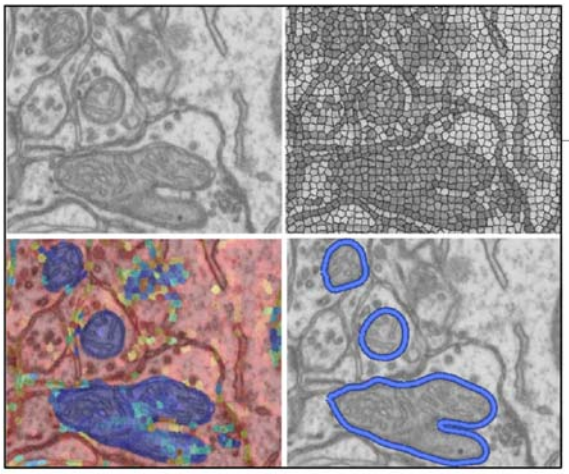
- Thresholding
- Region-oriented methods (e.g., region growing and region splitting)

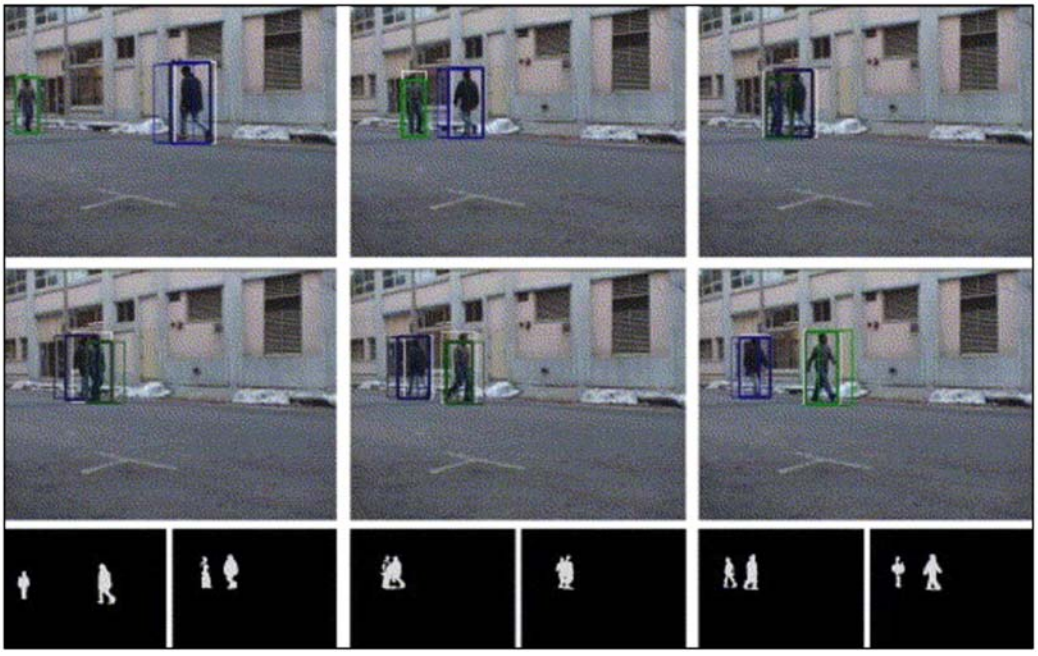
Some kind of segmentation technique will be found in applications involving the detection, recognition and measurement of objects/regions in images. Examples of such applications include

- Industrial inspection
- Optical character recognition
- Detecton and characterisation of tissues in medical images







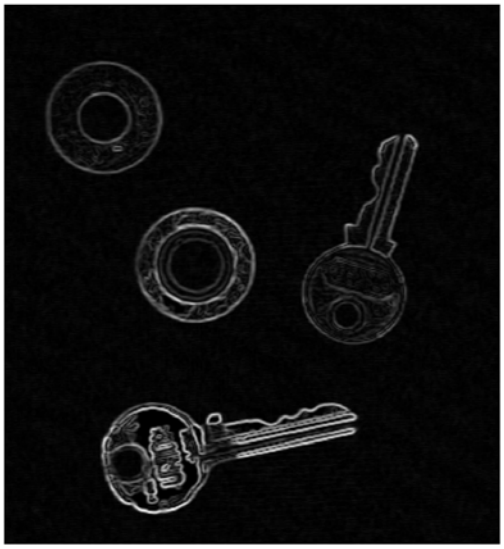


BOUNDARY DETECTION

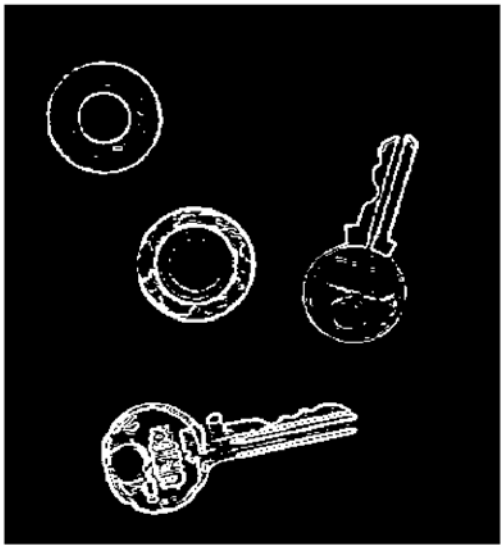
To obtain complete, one-pixel thick boundaries from detected edge pixels, further processing is required because

1. The edge pixels form boundaries several pixels thick.
2. Detected edge pixels do not, in general, form a complete boundary between objects and background because of noise and non-uniform illumination.

Original



Gradient magnitude



Edge map

Edge map (detail)

Contour Tracking

The edge image obtained in the edge detection process generally contains edges which are several pixels thick. Subsequent feature extraction may require boundaries only one pixel thick. A procedure to track the outer elements of a contour is as follows:

1. The first boundary pixel is located by searching the edge image from top-to-bottom and from left-to-right until an edge pixel is found. The location of this pixel is stored and N, the number indicating the direction of this pixel relative to the preceding one, is set to 1.

2. The checking of boundary pixels begins with pixel M, where

   $$M = (N - 2) \text{ modulo } 8.$$

   For example, if $N = 1$, $M = 7$.

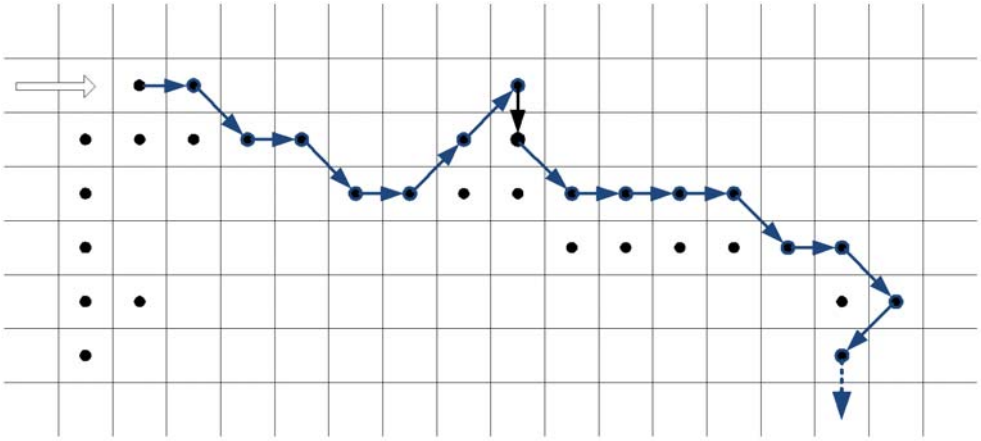
3. If a boundary pixel is found, its location is stored. N is set to M and Step 2 repeated; otherwise set

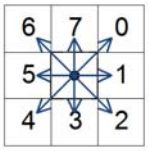
   $$M = (M + 1) \text{ modulo } 8$$

   and check neighbour M for a boundary pixel.

4. The search is terminated when the starting point is again encountered, i.e., the complete boundary has been traversed.

Note:
Given two numbers, a (the dividend) and n (the divisor), a modulo n is the remainder obtained the division of a by n.

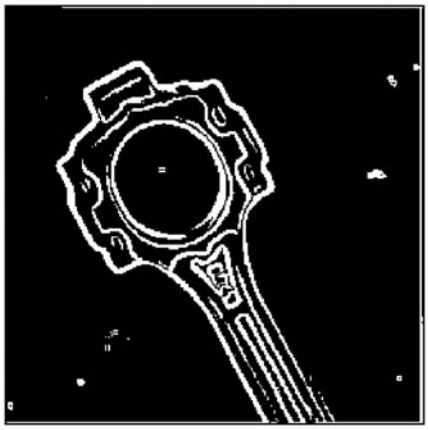




Directions

Original

Gradient magnitude



Edge map

After contour tracking
and noise removal

Local Analysis

Procedure: Analyze the pixels in a small neighbourhood about every point (x, y). All points that are defined as "similar" are linked.

Consider two neighbouring pixels at (x_1, y_1), (x_2, y_2). Similarity of edge pixels can be established by the magnitude (M) and angle (θ) of the gradient vector.

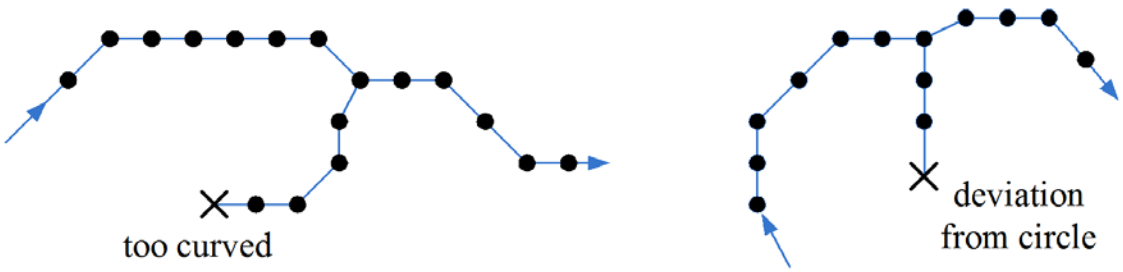
Magnitude criterion: $|M(x_1, y_1) - M(x_2, y_2)| \leq T_M$

Direction criterion: $|\theta(x_1, y_1) - \theta(x_2, y_2)| \leq T_\theta$

We link (x_1, y_1) and (x_2, y_2) if both the magnitude and direction criteria are satisfied. The process is repeated for every location in the image, keeping a record of linked points as the centre of the neighbourhood is moved from pixel to pixel.

Other criteria may be used:

- *Curvature of segment.* Constraints on the maximum or minimum curvature of the contours may be known.
- *Closeness to a known contour.* We may know that the boundary is circular or elliptic, for example.
- *Local property.* Local property values, such as local contrast, may be used.

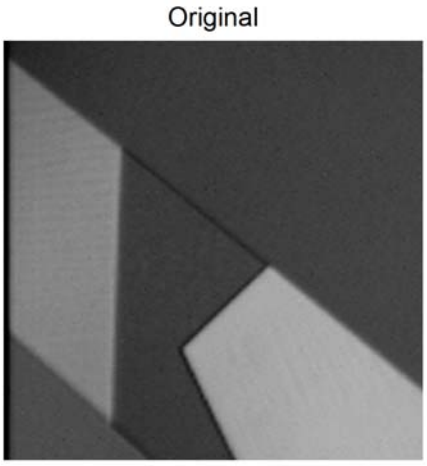


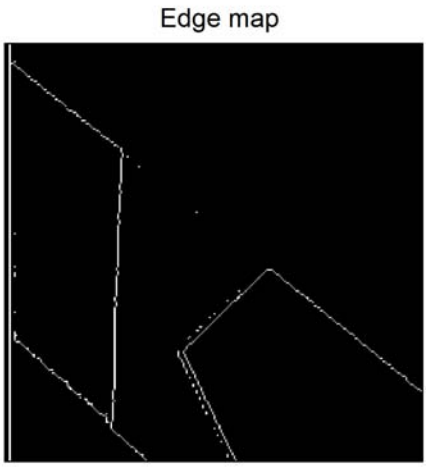
Global Analysis – Hough Transform

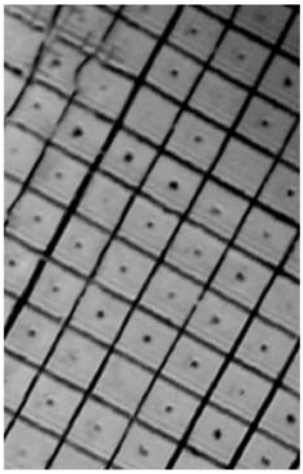
Detection of Straight Lines

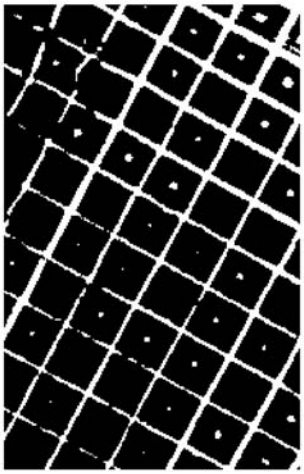
The Hough transform is used to link edge points by determining whether or not they lie on a curve of specified shape. It is particularly useful in cases when the number of solutions is not known.

Here we study how the Hough transform can be used for the detection of straight lines, which is a problem that is often encountered in digital image processing. The input image for the application of the Hough transform is the edge map of the image.









Consider a point (x_i, y_i). Using the *slope-intercept* representation of a straight line ($y = ax + b$), there is an infinite number of lines that pass through (x_i, y_i):

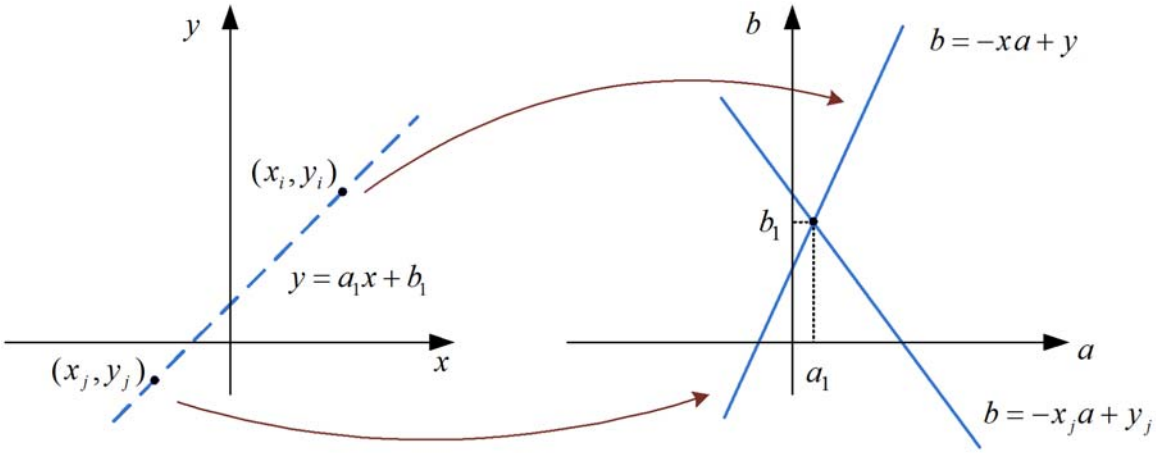
$$y_i = ax_i + b \tag{1}$$

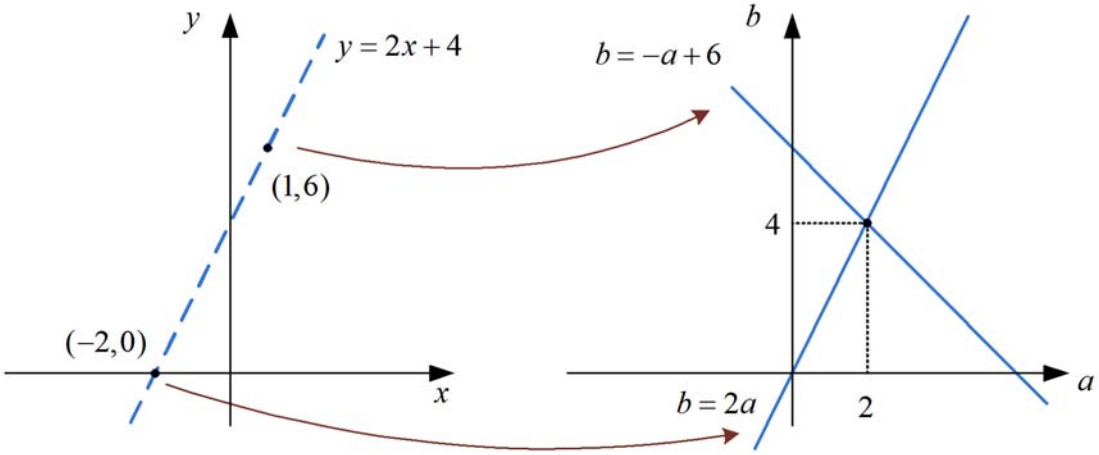
If we write this equation as

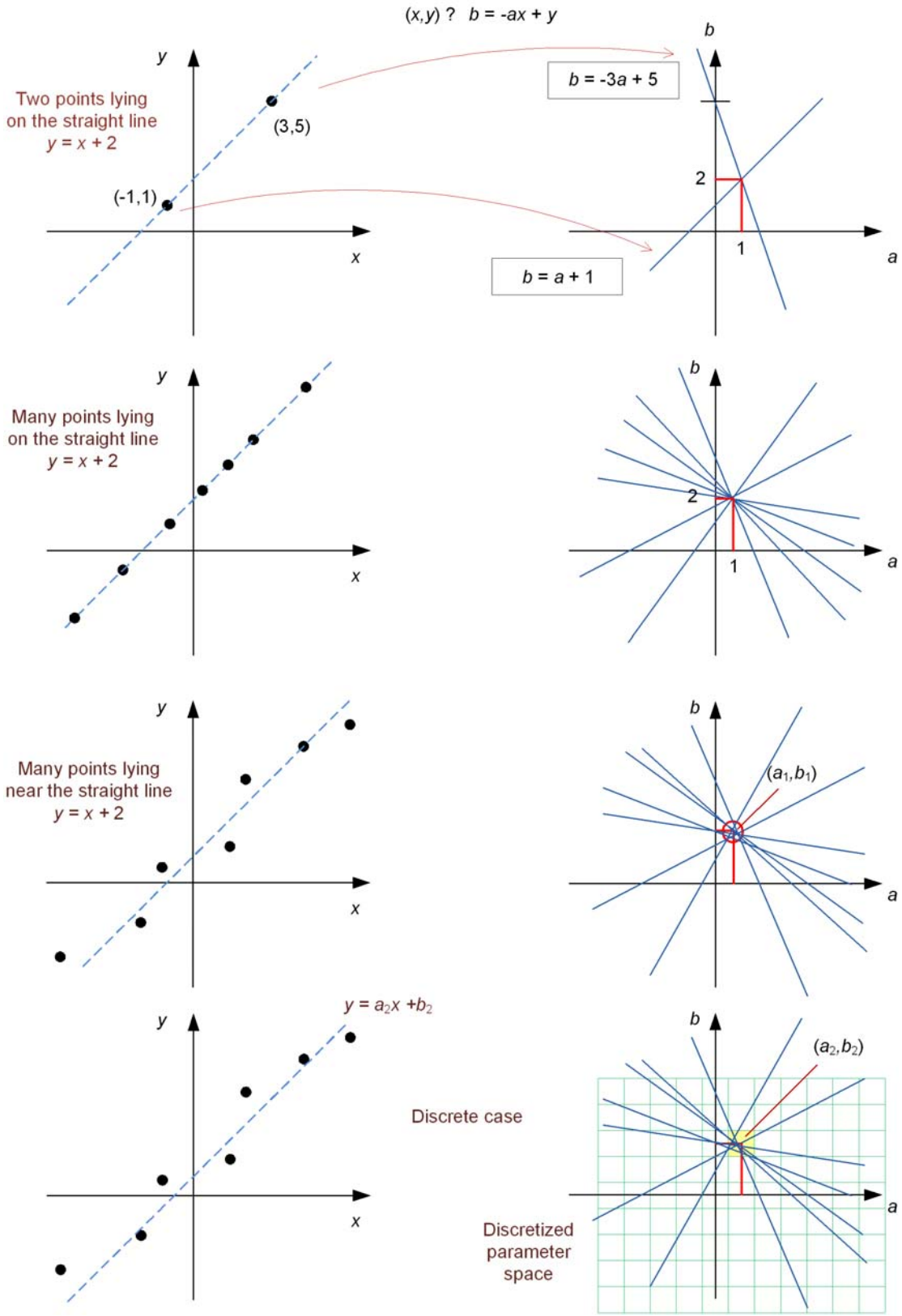
$$b = -x_i a + y_i \tag{2}$$

and consider the ab plane (also called *parameter* or *transform* space), then we have the equation of a *single* line for a fixed pair (x_i, y_i).

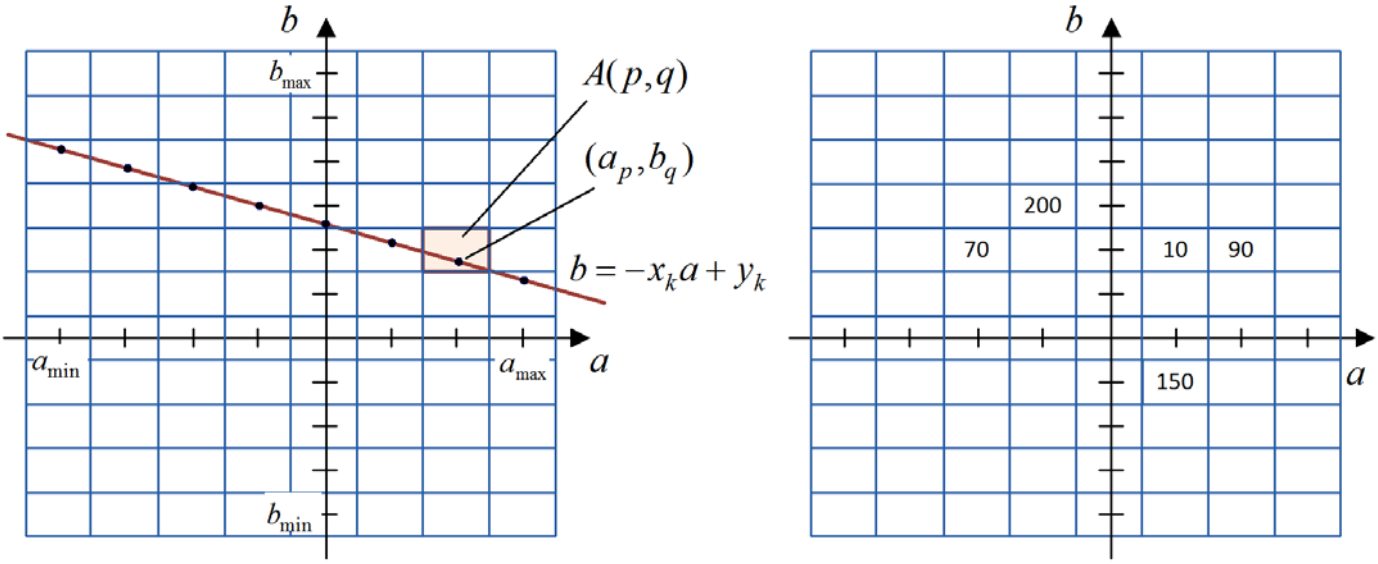
A second point (x_j, y_j) will also have a line in parameter space associated with it, and this line will intersect the line associated with (x_i, y_i) at (a_1, b_1), where a_1 is the slope and b_1 the intercept of the line containing both (x_i, y_i) and (x_j, y_j) in the xy plane. (We can say that a line in xy space transforms to a point in ab space.)





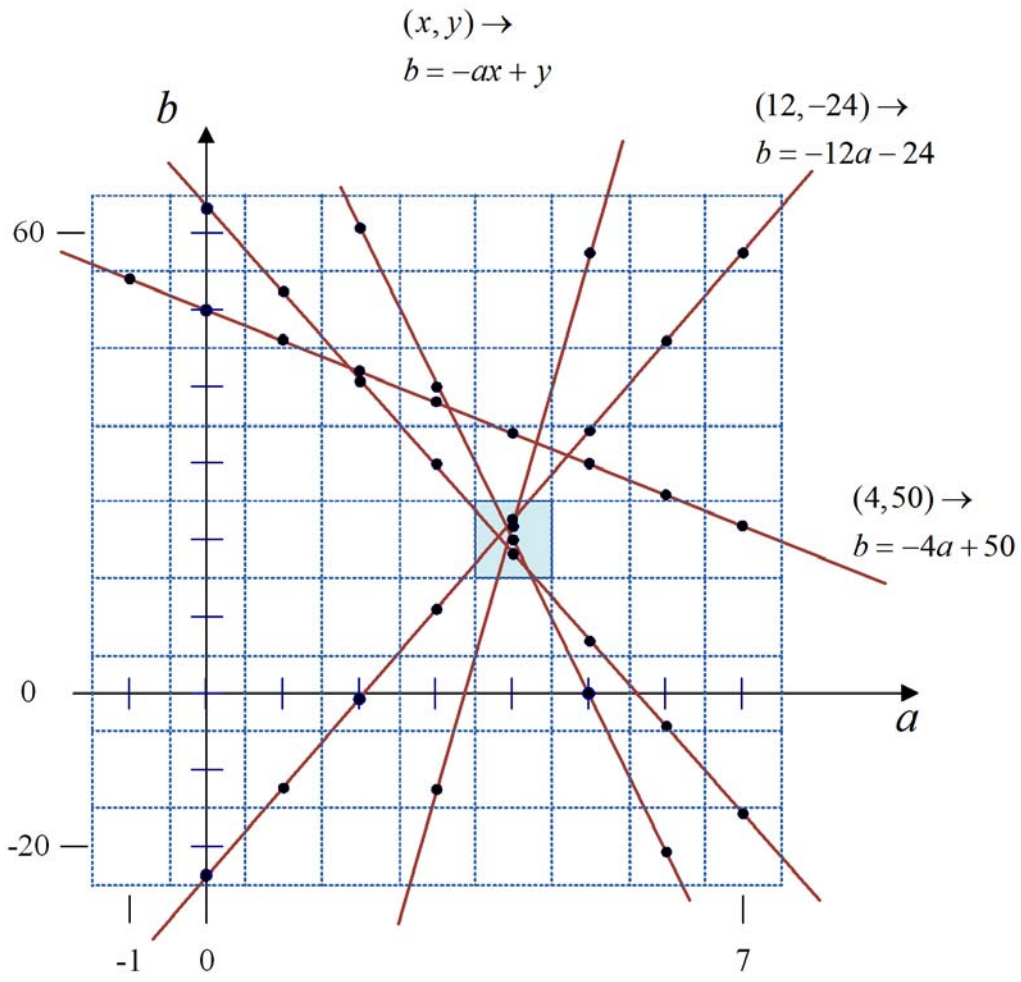

(x,y) ? b = -ax + y
Two points lying on the straight line y = x + 2
(3,5)
(-1,1)
b = -3a + 5
b = a + 1
Many points lying on the straight line y = x + 2
Many points lying near the straight line y = x + 2
y = a2x +b2
Discrete case
Discretized parameter space
(a1,b1)
(a2,b2)
x
y
a
b
1
2

The parameter space can be divided into *accumulator cells*, where $(a_{\max}, a_{\min})$ and $(b_{\max}, b_{\min})$ are the expected ranges of slope and intercept values. Accumulator cell $A(i, j)$ corresponds to the square associated with parameter space coordinates (a_i, b_j).



Procedure:

1. Set the cells to zero.
2. For every point (x_k, y_k) in the image plane, let the parameter a equal each of the allowed subdivision values on the a axis, and solve for the corresponding b using $b = -x_k a + y_k$.
3. Round off the resulting b's to the nearest allowed value.
4. If a choice of a_p results in solution b_q, let $A(p, q) = A(p, q) + 1$.
5. A final value of M of cell $A(i, j)$ corresponds to M points in the xy plane lying on the line $y = a_i x + b_j$.
6. Search the transform space for peaks, which could be
   - the cells with the n largest values, or
   - the cells with values exceeding a set threshold
7. Note the parameters at a peak, and use these parameters to find the pixels in the source image which are loacated on the specified line.


$(x, y) \rightarrow$
$b = -ax + y$
$(12, -24) \rightarrow$
$b = -12a - 24$
$(4, 50) \rightarrow$
$b = -4a + 50$
b
a
60
0
-20
-1
0
7

Example

In this example, we define the accumulator space to lie in the range

$$0 \leq a \leq 6, \qquad 0 \leq b \leq 60$$

The cells are of size 1×10.

Consider the edge point $(x, y) = (8, 60)$, which gives rise to

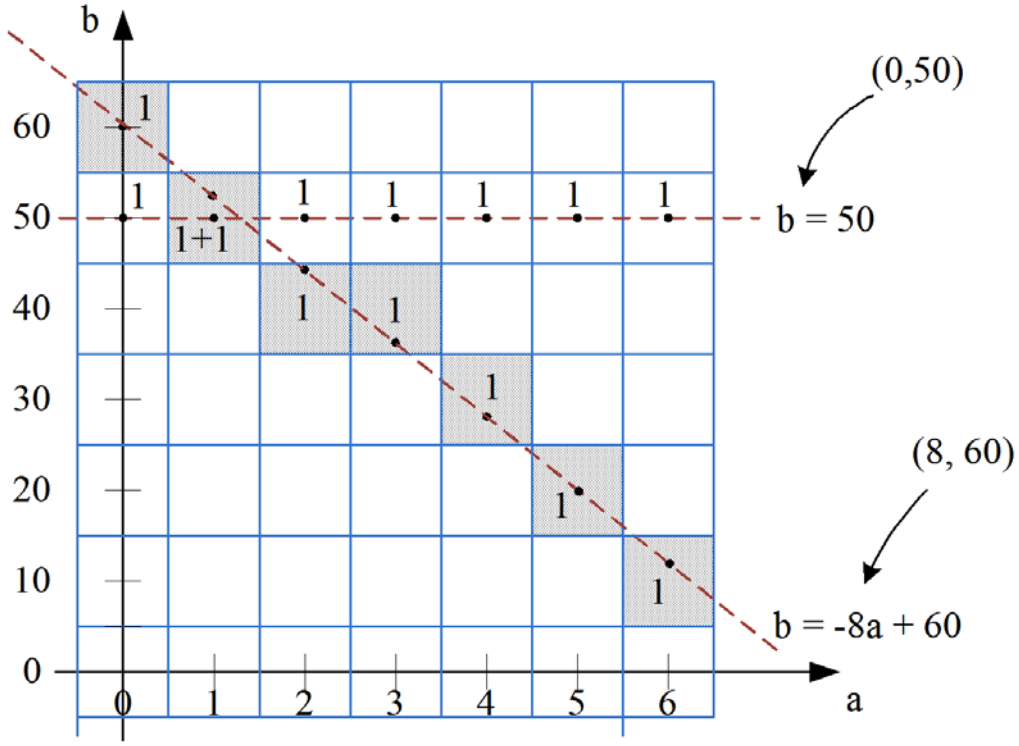
$$b = -8a + 60$$

The corresponding values of a and b are

a	0	1	2	3	4	5	6
b	60	52	44	36	28	20	12
round(b)	60	50	40	40	30	20	10

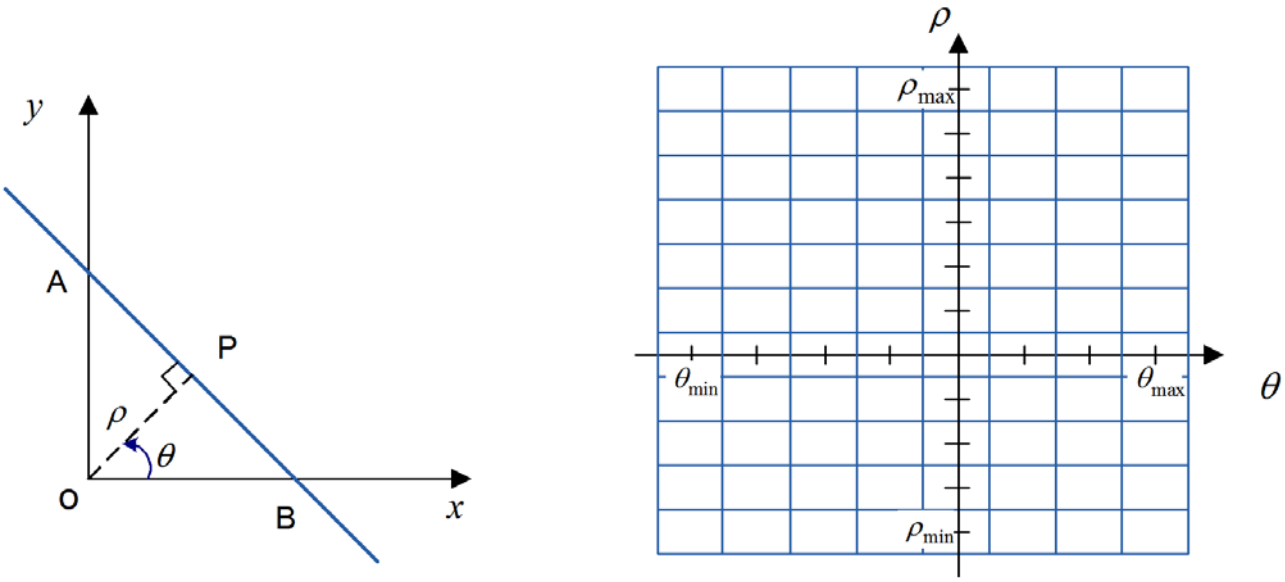
For another edge point $(0, 50)$, we have

$$b = 50$$



With $y = ax + b$, the slope and intercept approach infinity as the line tends to a vertical orientation, i.e., the accumulator array becomes unmanageably large. Hence we use the normal representation of a line:

$$x \cos\theta + y \sin\theta = \rho. \tag{3}$$



Now,

$$\begin{aligned} \rho &= x \cos\theta + y \sin\theta & (4) \\ &\equiv \sqrt{x^2 + y^2} \cos(\theta - \tan^{-1} \frac{y}{x}) & (5) \end{aligned}$$

Therefore, a point in image space is represented by a sinusoid in $\theta\rho$ transform space. (θ may be restricted to the range $-90° < \theta \leq 90°$.)

The procedure for detecting straight-line segments is similar to that used for ab parameter space.

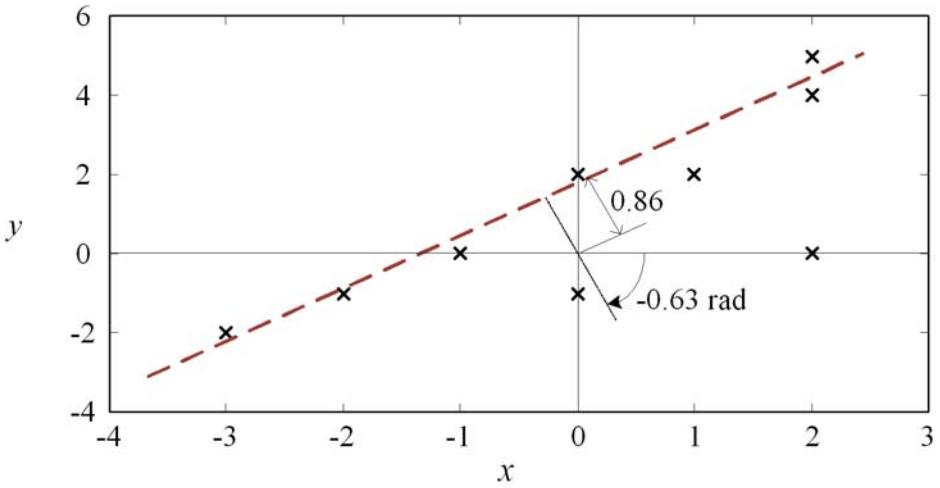
Example

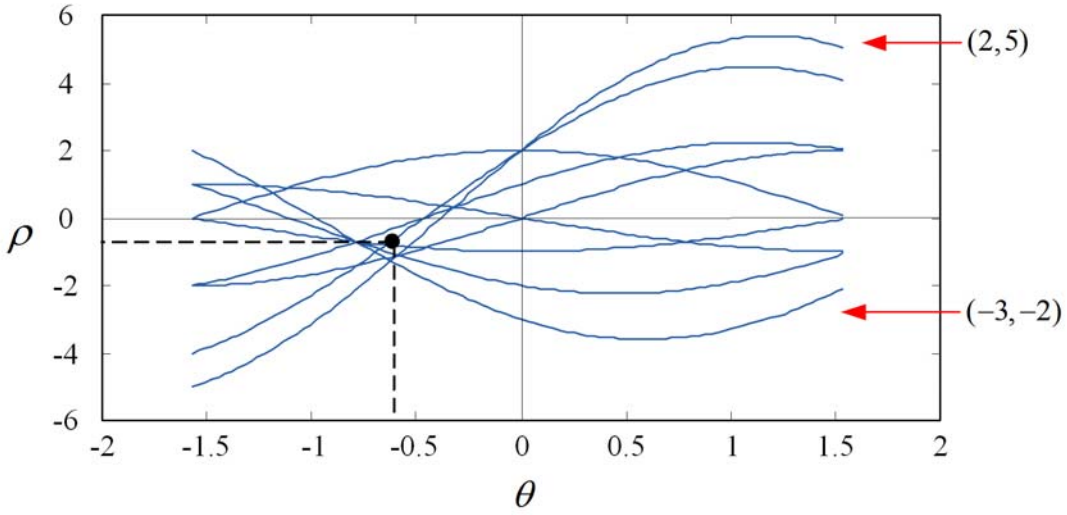
Consider detected edge points at

$$(-3,-2),(-2,-1),(-1,0),(0,-1),(0,2),(1,2),(2,0,),(2,4),(2,5)$$

For example,

$$\begin{aligned}(2,5) &\rightarrow \rho = 5.4\cos(\theta - 68^\circ) = 5.4\cos(\theta - 1.19 \text{ rad})\\ (-3,-2) &\rightarrow \rho = 3.6\cos(\theta - 146^\circ) = 3.6\cos(\theta - 2.55 \text{ rad})\end{aligned}$$





From the graphs, we obtain

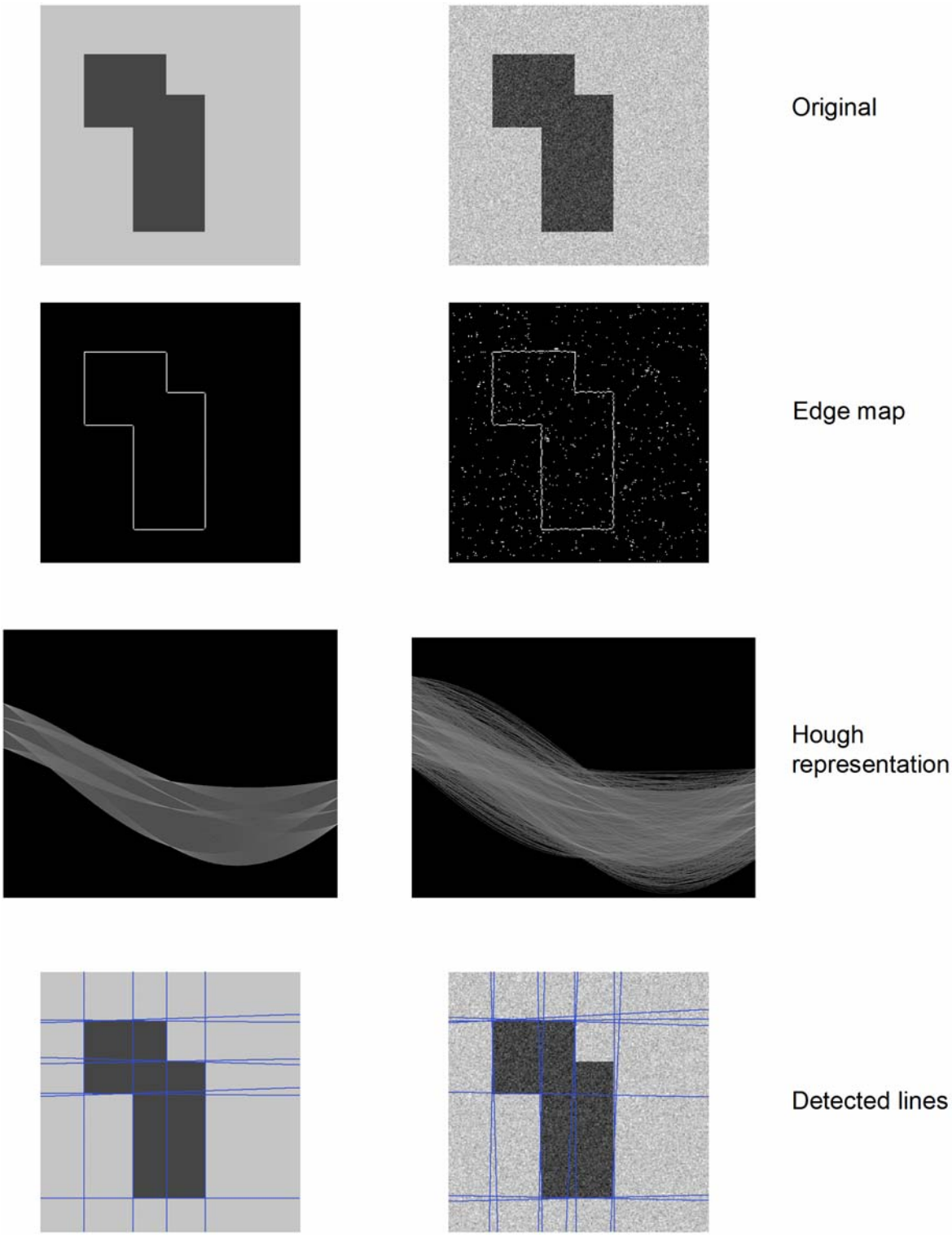
$$\begin{aligned}\theta &= -0.63 \text{ rad}\\ \rho &= -0.86\end{aligned}$$

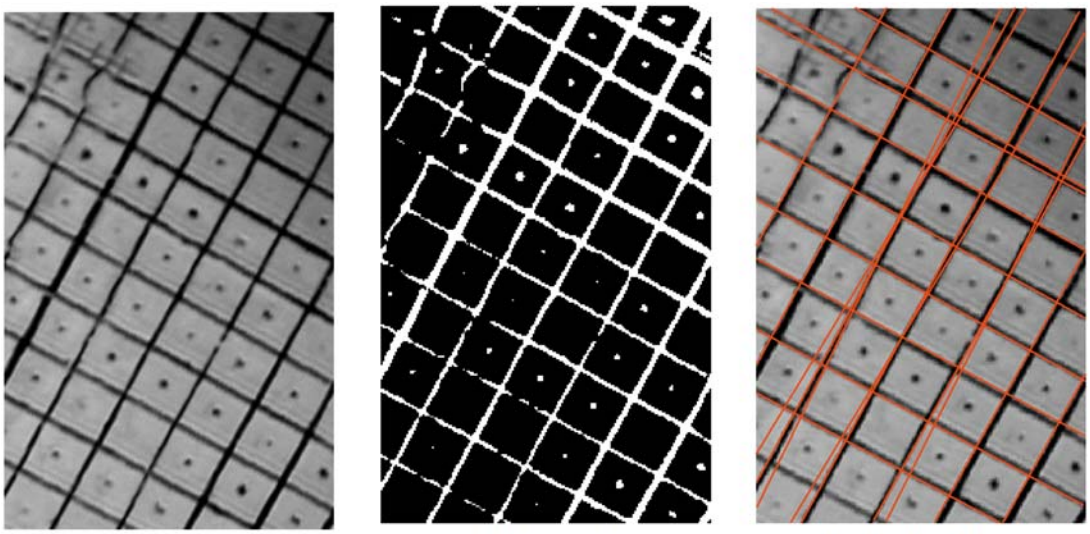
Substituting into Eq. (3) gives us

$$0.81x - 0.59y = -0.86$$

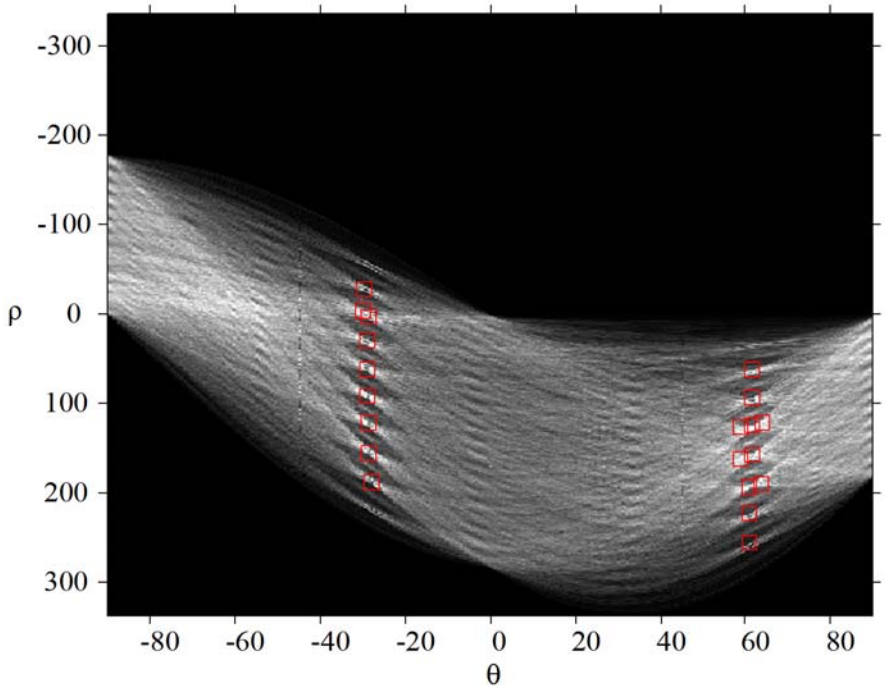
which is the dashed line in the upper figure.

Examples





Original Edge map Detected lines



Hough representation

Extensions of the Hough Transform

The Hough-based feature extraction method may be extended to geometric shapes more complex than lines. Consider a circle of radius r and centered at (a_1, a_2). If it contains point (x_i, y_i), its equation is given by

$$(x_i - a_1)^2 + (y_i - a_2)^2 = r^2 \tag{6}$$

Three parameters are thus necessary to characterize a circle: the centre coordinates (a_1, a_2) and the radius r, resulting in a three-dimensional parameter space.

The Hough transform may be further generalized for arbitrary shapes. For example, consider an ellipse centered at (x_0, y_0), with major and minor axes a and b. If the major axis is parallel to the x axis, any point (x_i, y_i) on this ellipse may be described by

$$\frac{(x_i - x_0)^2}{a^2} + \frac{(y_i - y_0)^2}{b^2} = 1 \tag{7}$$

Adding a fifth parameter, an angle ϕ, to handle orientation allows the application of the Hough transform. This requires a 5-D parameter solution space.

THRESHOLDING

Gray level (or intensity) thresholding may be used for segmentation when the object and background regions have gray levels grouped into distinct modes. For images with a bimodal histogram (i.e., with two major modes), we select a threshold, T, which separates the two modes. The thresholded image is then

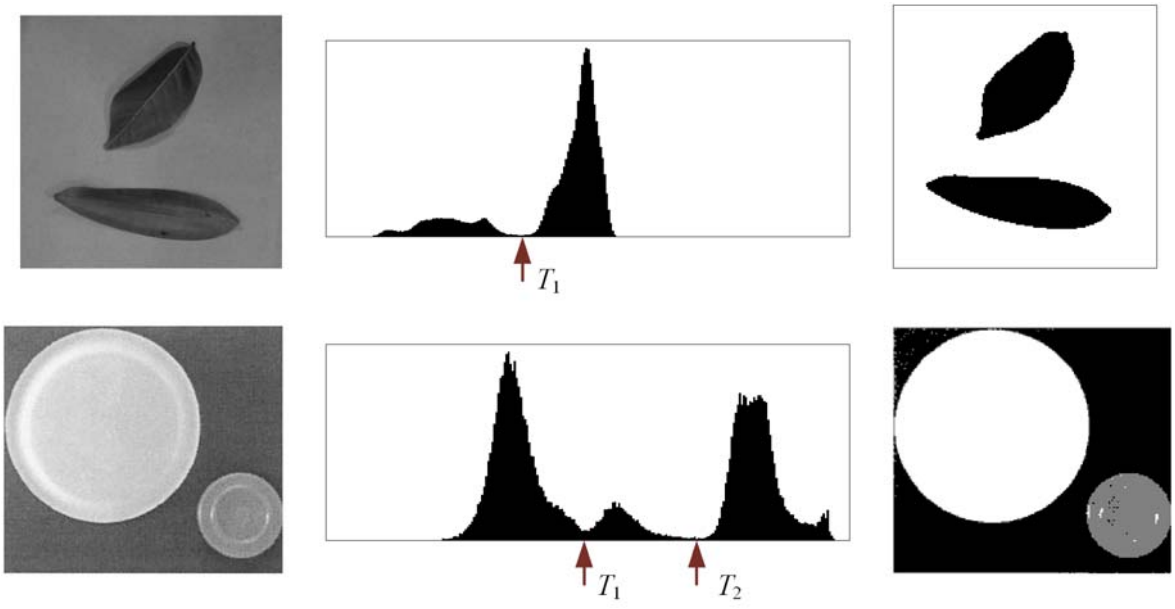
$$g(x,y) = \begin{cases} 1 & \text{if} \quad f(x,y) > T \\ 0 & \text{if} \quad f(x,y) \leq T. \end{cases} \tag{8}$$

If the image consists of light objects against a dark background, the pixels labelled 1 (or any convenient intensity level) will correspond to the objects, while pixels labelled 0 will correspond to the background.

When there are more than two modes, *multilevel thresholding* may be employed. For example, with three modes, two thresholds, T_1 and T_2 are selected. The thresholded image is then

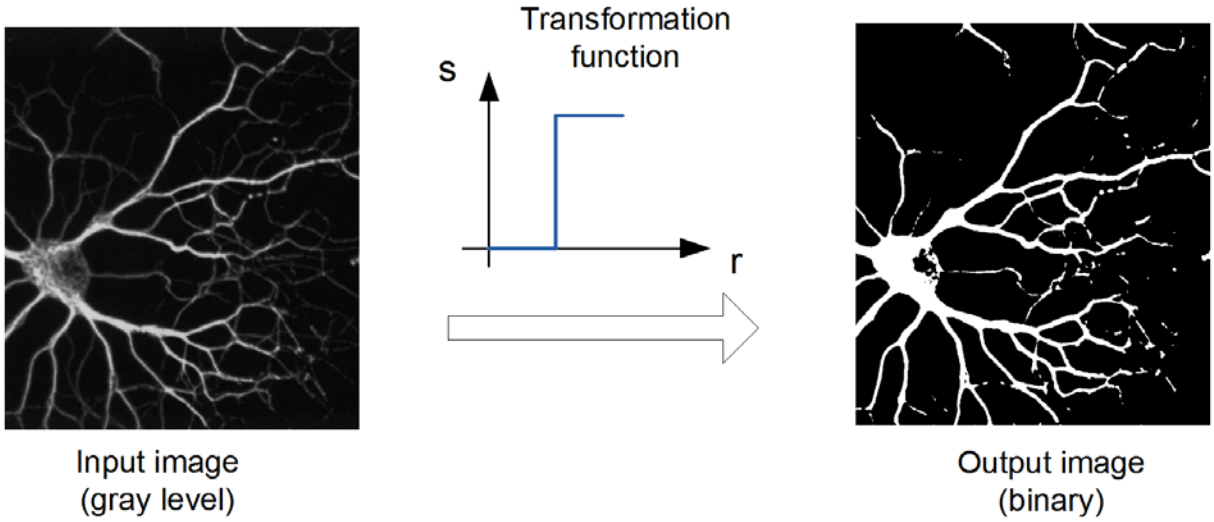
$$g(x,y) = \begin{cases} L_2 & \text{if} \quad f(x,y) > T_2 \\ L_1 & \text{if} \quad T_1 < f(x,y) \leq T_2 \\ 0 & \text{if} \quad f(x,y) \leq T_1 \end{cases} \tag{9}$$

where L_1 and L_2 are different gray levels.



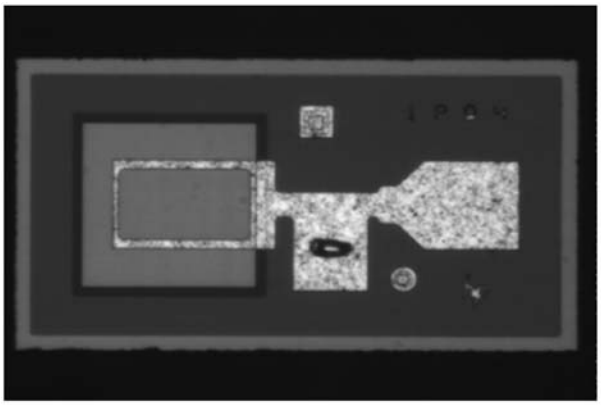
Global Thresholding

When the threshold is constant over the entire image, the process is called *global thresholding*.

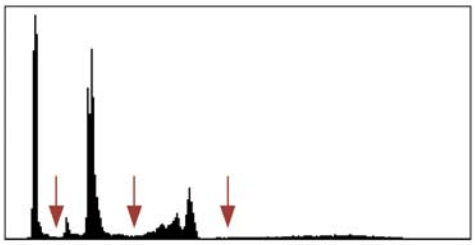


When the threshold value changes over the image, we use the term *variable* or *local thresholding*. For example, the threshold value may depend on the average intensity of the pixels in the neighbourhood. Such an approach may be useful when the illumination varies across the scene.

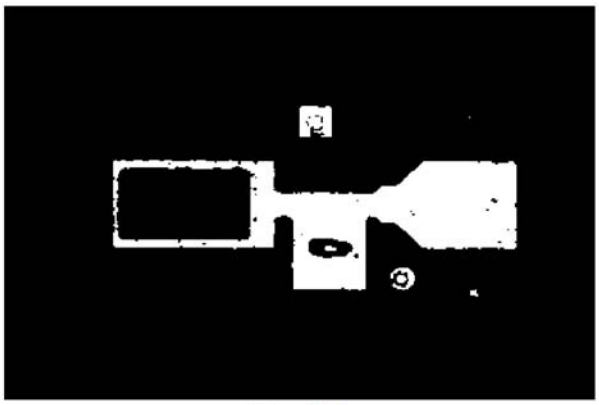
Global thresholding is fastest, and is suitable when (a) the reflectance of the object is distinct from that of the background and (b) the illumination is uniform. More sophisticated thresholding is required to compensate for the presence of complicating factors such as non-uniform illumination, shadows, texture and specular reflection.



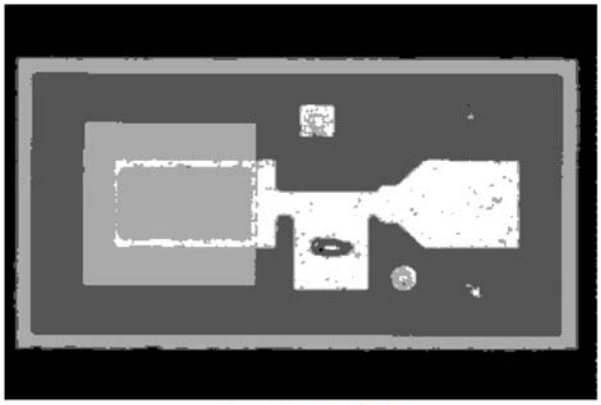
Original



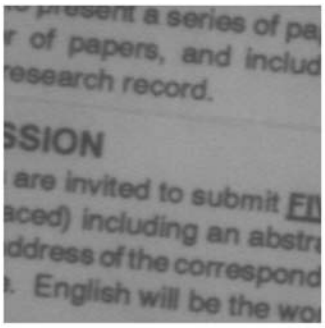
Histogram



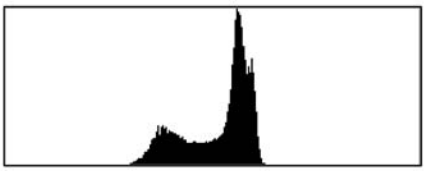
T = 110

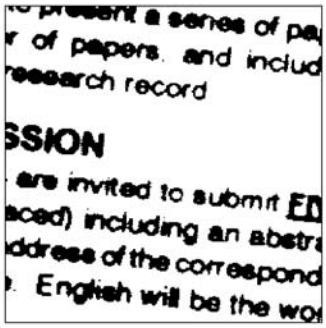


T = 25, 70, 110

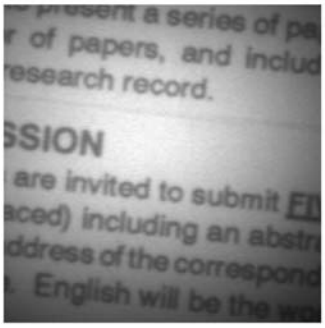


Under uniform illumination

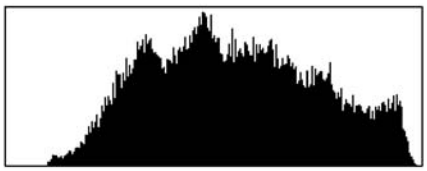




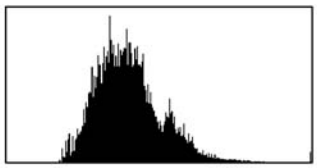
After thresholding



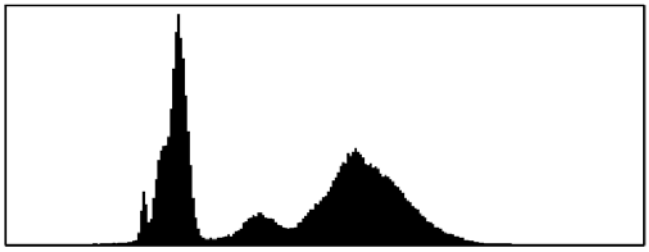
Under non-uniform illumination



After thresholding

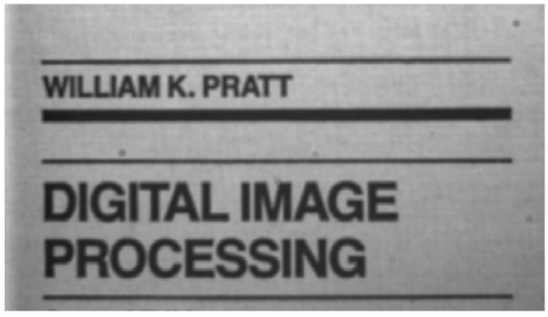


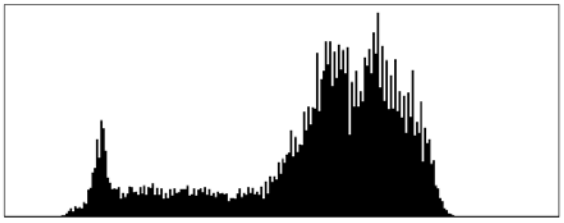
Suitable thresholds may be obtained by finding the minimum points (valleys) between modes.

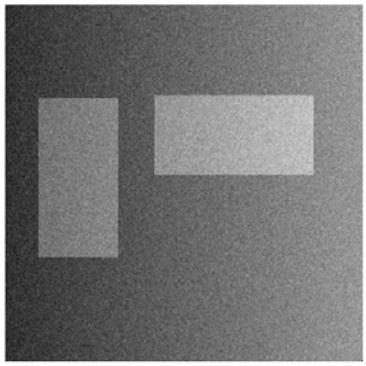


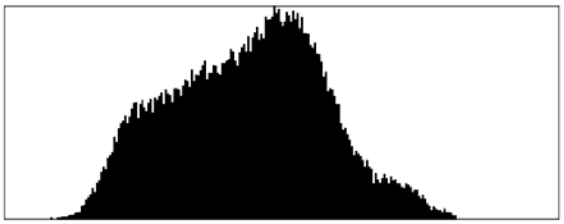
However, the following difficulties may arise:

1. The valley may be so broad that it is difficult to locate a significant minimum.
2. Noise within the valley may inhibit location of the optimum position.
3. There may be no clearly visible valley in the distribution because noise may be excessive or because the background lighting may vary appreciably over the image.









Histogram smoothing

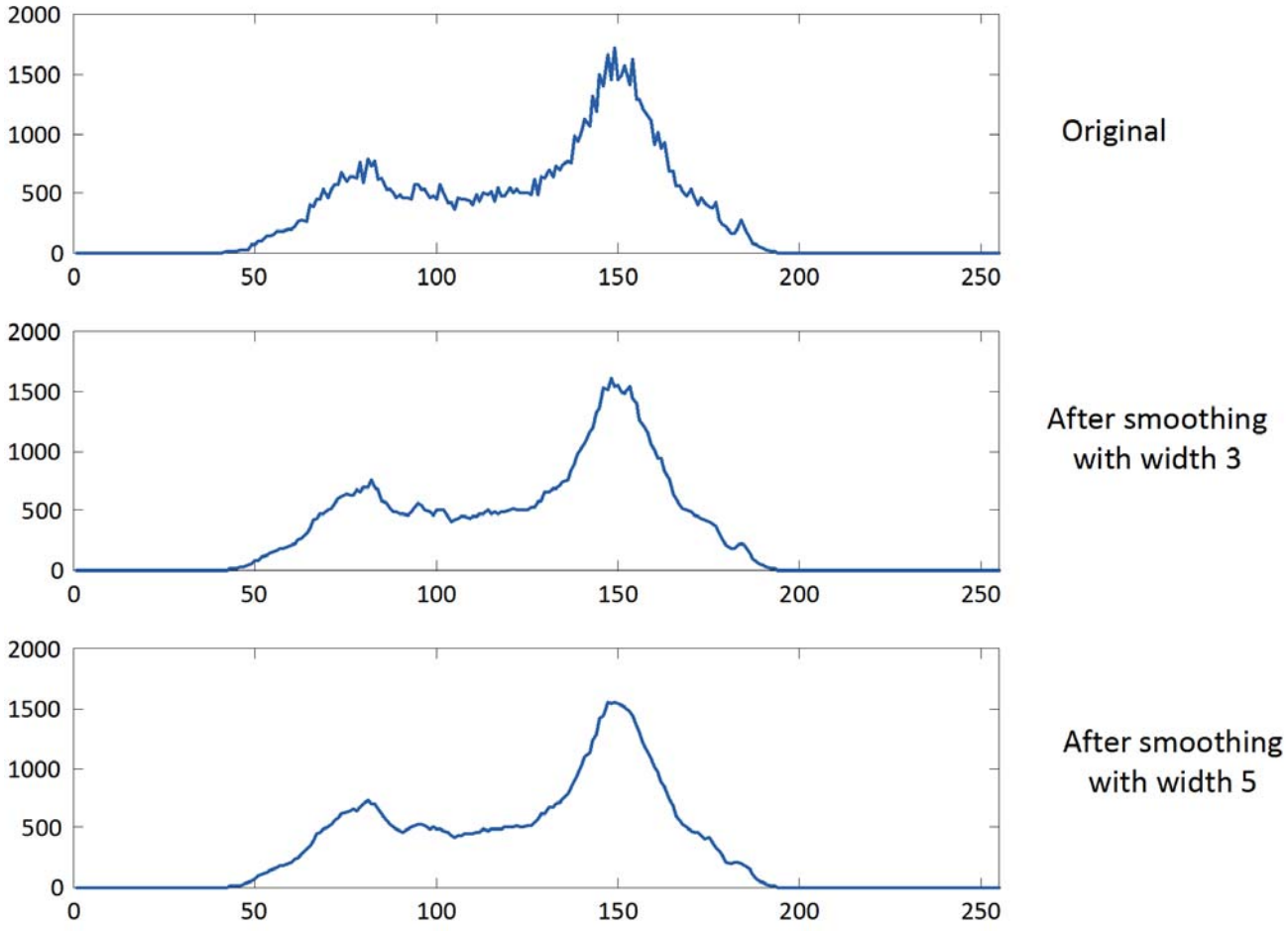
The details in a histogram may be smoothed to enable the peaks and valleys to be more easily seen. This is done by replacing a bin value by the average of its neighbours.

Smoothing with a width of three would involve replacing each bin value with the average of that value and its immediate neighbours:

$$h(i) \leftarrow (h(i-1) + h(i) + h(i+1))/3$$

If greater smoothing is required, we use a width of five:

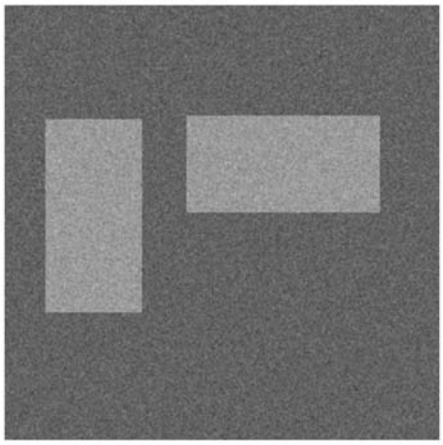
$$h(i) \leftarrow (h(i-2) + h(i-1) + h(i) + h(i+1) + h(i+2))/5$$

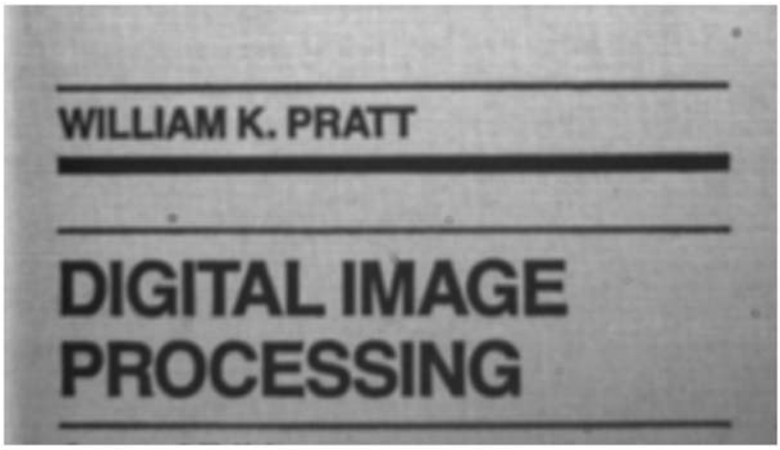


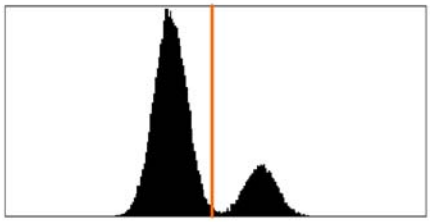
Intermeans Algorithm

An iterative threshold selection method starts with an approximate threshold and then successively refines this estimate.

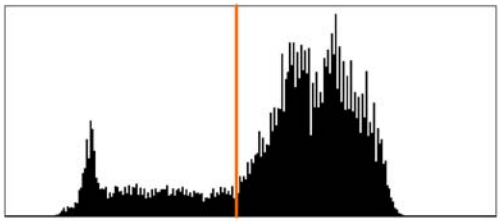
1. Select an initial estimate of the threshold, T. A good initial value is the average intensity of the image.
2. Partition the image into two groups, R_1 and R_2, using the threshold T.
3. Calculate the mean gray values μ_1 and μ_2 of the partitions R_1 and R_2.
4. Select a new threshold
$$T = \frac{1}{2}(\mu_1 + \mu_2)$$
5. Repeat steps 2-4 until the mean values μ_1 and μ_2 do not change significantly in successive iterations.



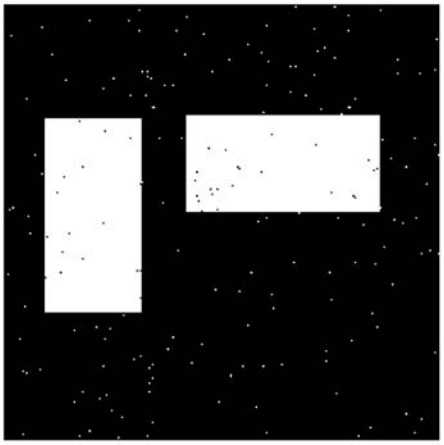

WILLIAM K. PRATT
DIGITAL IMAGE
PROCESSING

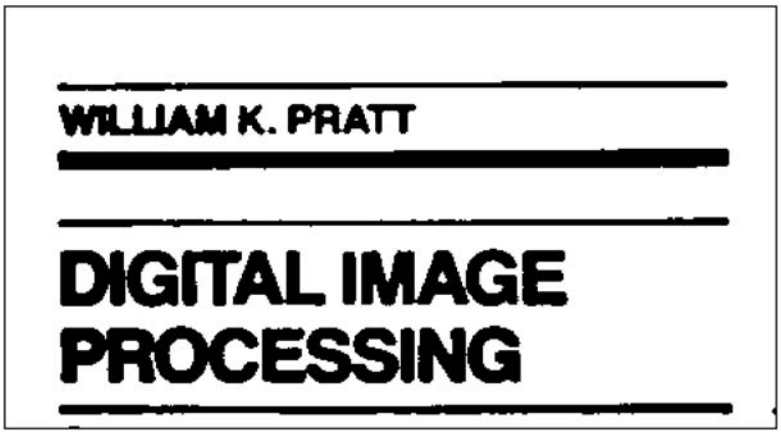


T = 127



T = 116




WILLIAM K. PRATT
DIGITAL IMAGE
PROCESSING

Thresholding of Colour Images

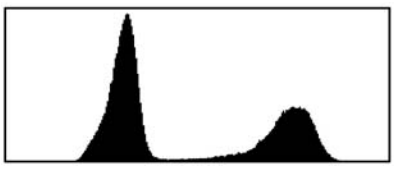
For colour images, it may be possible to use one of the three colour planes for segmentation.

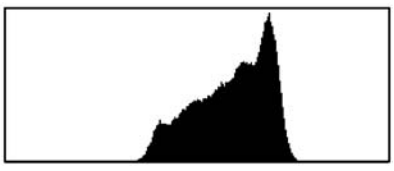
Colour image

Red plane

Green plane

Blue plane

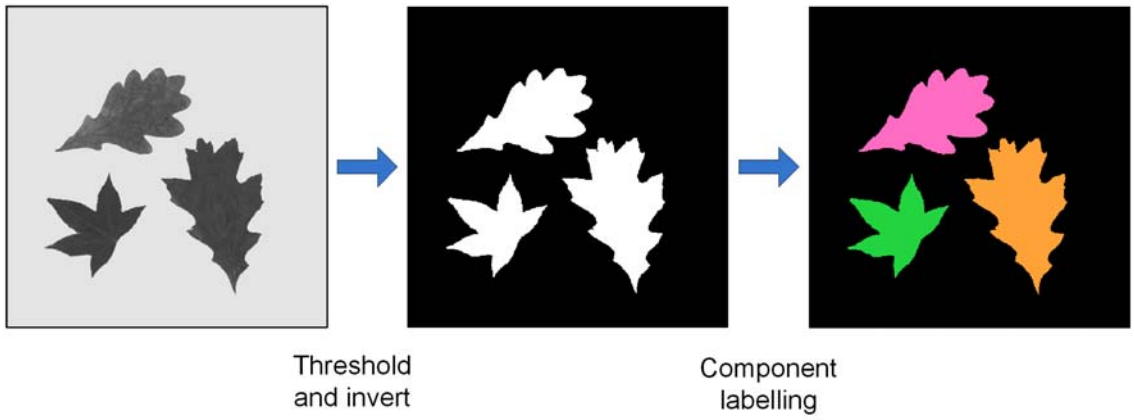




LABELLING OF CONNECTED COMPONENTS

In this procedure, an image is scanned and its pixels grouped into connected components (or regions). The aim is to identify connected components that share the same set of intensity values.

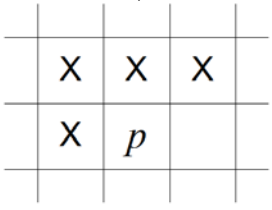
The procedure is typically performed on binary images that contain distinct object regions, with object pixel value = "1" and background pixel value = "0" (obtained, for example, after thresholding). Once all the groups have been determined, each pixel is labelled with a gray level or colour according to the component it was asigned to. Once we have the labelled components, we can proceed with analyzing each of them, e.g., by measuring object features such as area, perimeter, and so on.



With 8-connectivity, the procedure is as follows:

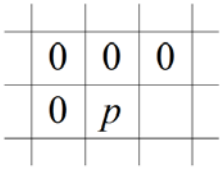
Step 1: Scan the image moving along a row until it comes to a pixel p for which $V = \{1\}$.

Step 2: Examine the four neighbours of p which which have already been encountered in the scan (i.e., the neighbours to the left of p, above it, and the two upper diagonal ones).



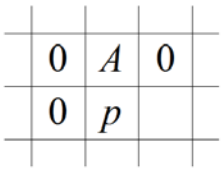
Step 3: p is labelled (as A, B, C, etc.) following these rules:

(a) If all four neighbours are 0, assign a new label to p.



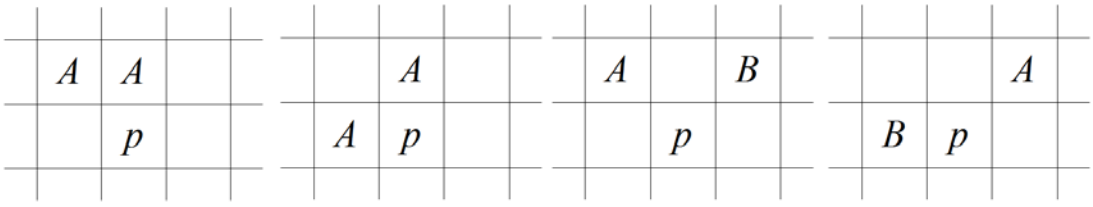
New label ← p

(b) If only one neighbour is 1, assign its label to p.



$A \leftarrow p$

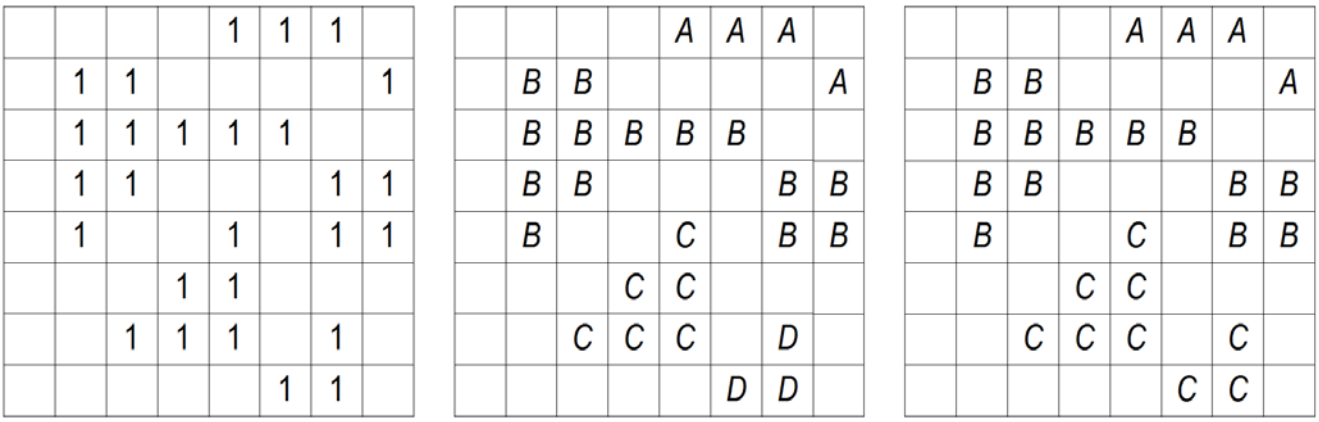
(c) If more than one of the neighbours have a value of 1, assign one of the labels to p and make a note of the equivalences,



Step 4: Scan to the end of this and repeat with the next row.

Step 5: After completing the scan of the image, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class.

Step 6: As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes. For display, the labels might be different gray levels or colors.



REGION-ORIENTED METHODS

The aim is to group pixels having similar "properties" into regions. This measure of similarity or uniformity may be based on intensity, colour, local texture, or some other image attribute. Region-growing methods are generally better than edge-based methods in noisy images, where borders may be difficult to detect.

The uniformity or otherwise of a group of pixels may be indicated by a *uniformity predicate*, which is a logical statement that is true only if pixels in the region are sufficiently in terms of a specified attribute.

A common uniformity predicate is:

$$P(R) = \begin{cases} \text{TRUE} & \text{if } |f(x_1, y_1) - f(x_2, y_2)| \leq \Delta \\ \text{FALSE} & \text{otherwise} \end{cases} \tag{10}$$

where (x_1, y_1) and (x_2, y_2) are coordinate of neighbouring pixels in region R. This predicate states that a region R is uniform if and only if any two neighbouring pixels differ in gray level by no more than Δ.

A similar predicate is:

$$P(R) = \begin{cases} \text{TRUE} & \text{if } |f(x_1, y_1) - \mu_R| \leq \Delta \\ \text{FALSE} & \text{otherwise} \end{cases} \tag{11}$$

where $f(x_1, y_1)$ is the gray level of a pixel from region R with coordinates (x_1, y_1) and μ_R is the mean gray level of all pixels in R (excluding the pixel at (x_1, y_1)).

A more stringent uniformity predicate is:

All pixels within the region must have exactly the same gray level.

Region Growing

In region growing, pixels or subregions are grouped into larger regions. It is a bottom-up procedure that starts with a set of seed pixels. The aim is to grow a uniform, connected region from each seed. A pixel is added to a region if and only if

- It has not been assigned to any other region.
- It is a neighbour of that region.
- The new region created by the addition of the pixel satisfies the uniformity predicate.

Example

0	0	5	6	7
1	1	5	8	7
0	$\underline{1}$	6	$\underline{7}$	7
2	0	7	6	6
0	1	5	6	5

Original

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

Result for $T = 3$

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

Result for $T = 8$

- Numbers represent intensity values.
- Seeds are indicated by the underscore (_); the pixel with gray level 1 (for region R_1, and the pixel with gray level 7 for region R_2.
- $P(R_i)$ = TRUE if the absolute difference between the intensity of the pixel and the seed is less than a threshold T. A pixel that satisfies P simultaneously for the two seeds is assigned to R_1.

Problems:

1. *Selection of initial seeds.* The seed points must properly represent regions of interest. This depends on the nature of the problem. For example,

   - The most prominent points may be chosen.
   - Compute at every pixel the same set of properties that will be used as the basis for region growing. If the computation shows clusters, then the pixels whose values place them near the centroids of the clusters may be used as seeds.

2. *Selection of suitable properties.* This depends on the problem under consideration as well as the image data. Typically, region analysis is carried out with a set of descriptors based on intensity and spatial properties (such as moments or texture).

3. *Formulation of a stopping rule.* A region should stop growing when no more pixels satisfy the criteria for inclusion in that region. In addition to local criteria (e.g. intensity, texture, colour) other factors such as size and the region shape should also be considered.

4. *Incomplete segmentation.* The number of seeds defined by the user may not be sufficient to create a region for every pixel. Hence some pixels in the image will be left untouched.

Region Splitting/Merging

An alternative to region growing is to subdivide an image initially into a set of disjointed regions and then merge and/or split the regions in an attempt to satisfy the four conditions stated earlier.

1. Select a predicate P.
2. Split into four disjoint quadrants any region R_i for which

$$P(R_i) = \text{FALSE}$$

.

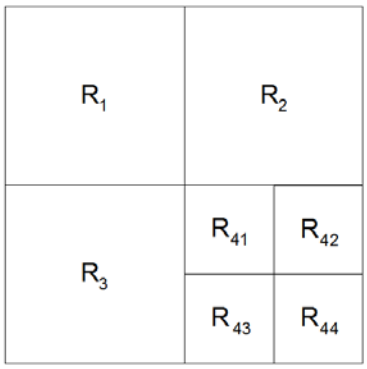
3. Merge any adjacent regions R_j and R_k for which

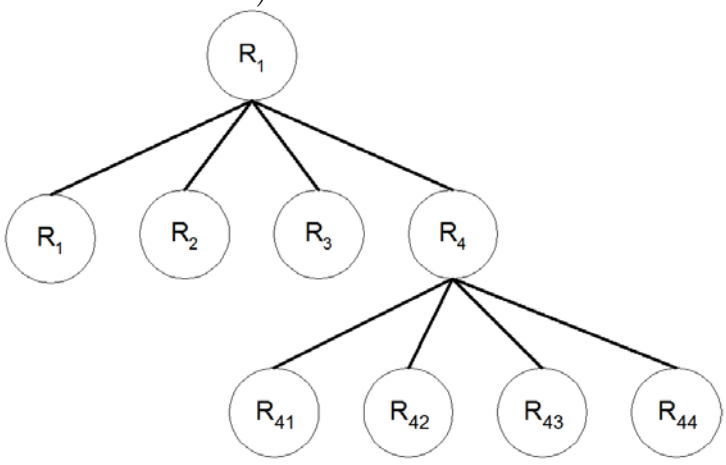
$$P(R_j \cup R_k) = \text{TRUE}$$

.

4. Stop when no further merging or splitting is possible.

This technique can be represented in the form of a quadtree (a tree in which each node has exactly four descendants.)





Example:

- Both object and background have constant intensities.
- $P(R_i)$ = TRUE if all pixels in R_i have the same intensity.