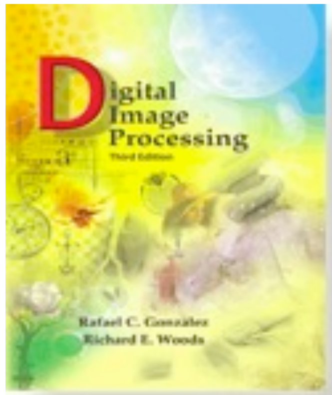


The Hough transform

This method is a global method in the sense that it uses *global* properties of an image, like shapes, curves,…

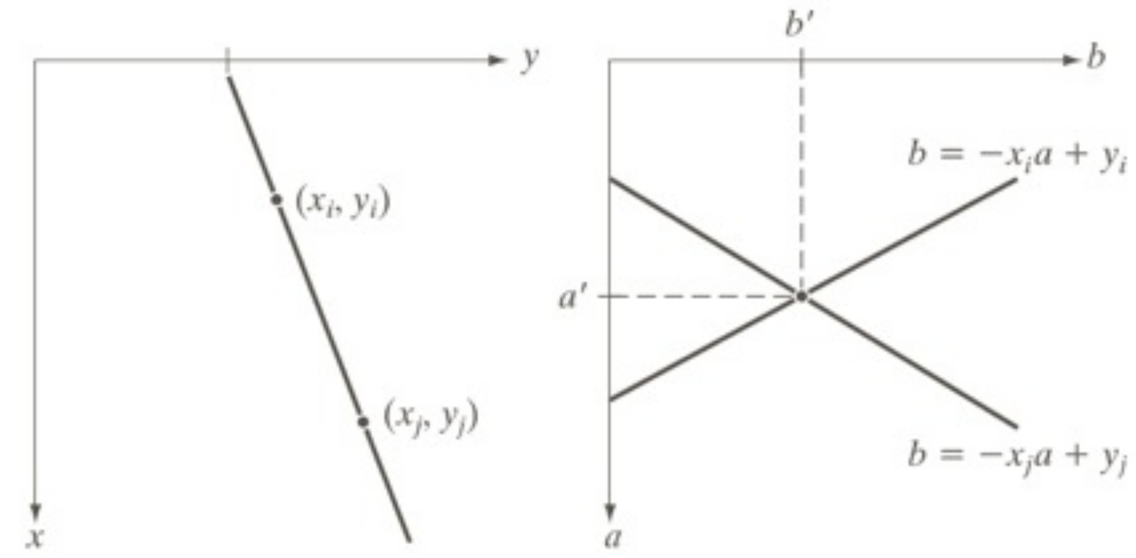
Problem: Given a set of *n* points, find out if they belong to the same straight line.

- find all the lines between any 2 such points ($n(n-1)/2 \sim n^2$ lines)
- compare all the points to all the lines ($n\, n(n-1)/2 \sim n^3$ comparisons)

If the number of points is large, the task can soon become prohibitive.

If a point belongs to a line, it will satisfy the generic line equation,

$$y_i = ax_i + b$$



a and b are the parameter (slope and intercept) of the line.
For any point *(xi,yi),* there are infinitely many lines through it.

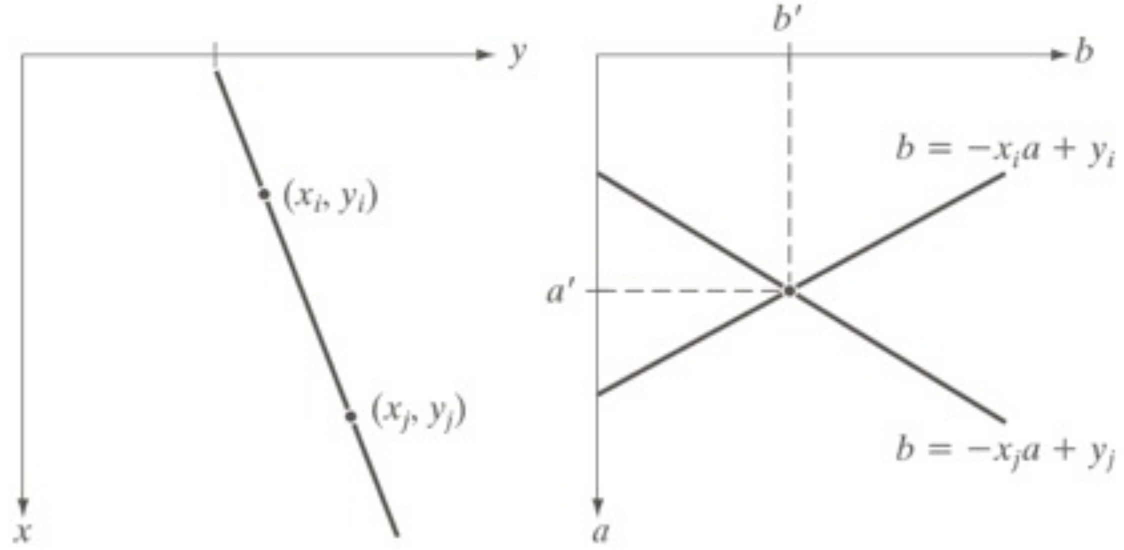
Let's look at the parameter space, *(a,b)*: in this space, the equation of the line becomes

$$b = -x_i a + y_i$$

All the lines through (*xi, yi*) have parameters *a,b,* satisfying the same equation.

$$b = -x_i a + y_i$$

Now, if *(xj, yj)* is another point, all the lines through this point will satisfy the equation



$$b = -x_j a + y_j$$

Thus, if the two points *(xi,yi)* and *(xj,yj)* are on the same line, the two lines in the parameter space will intersect in a point (*a'*, *b'*).

This suggests a procedure: for all the set of points, compute the lines in the parameter space. The principal lines in the image are then identified by finding the points where many lines in the parameter space intersect.

A problem is that when the line becomes vertical, $a \to \infty$, and it might be difficult to work with vertical lines.

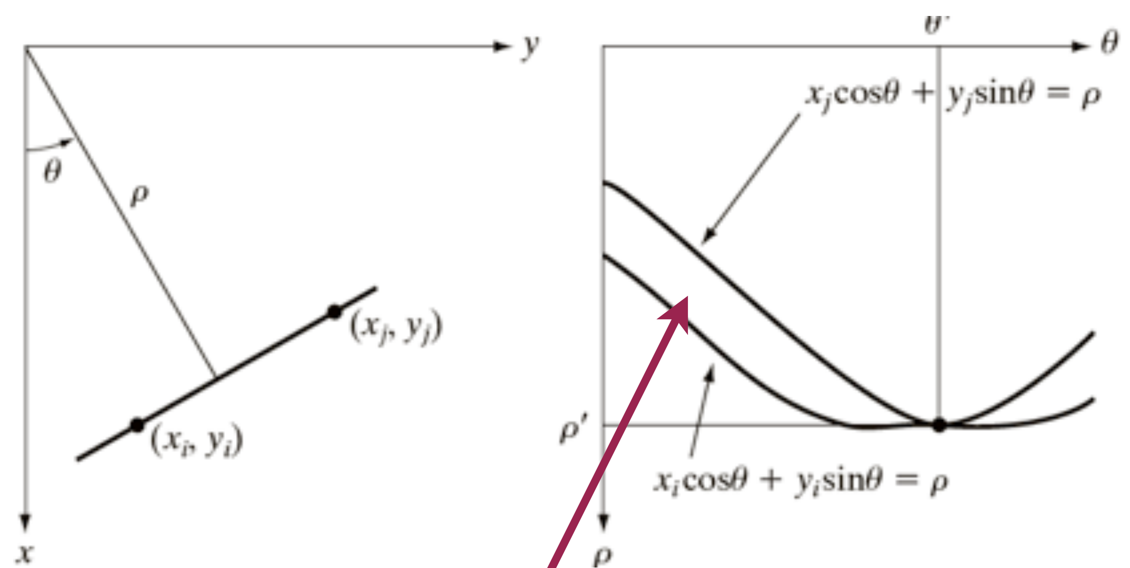
Solution: work with the equation of lines in polar coordinates.

If (x_i, y_i) is any point in the plane, any line passing through the point will have equation

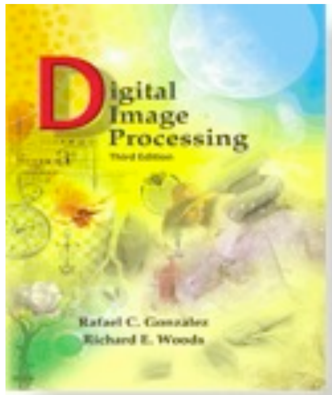
$$x_i \cos\theta + y_i \sin\theta = \rho$$

=> use ρ, θ as the parameter space

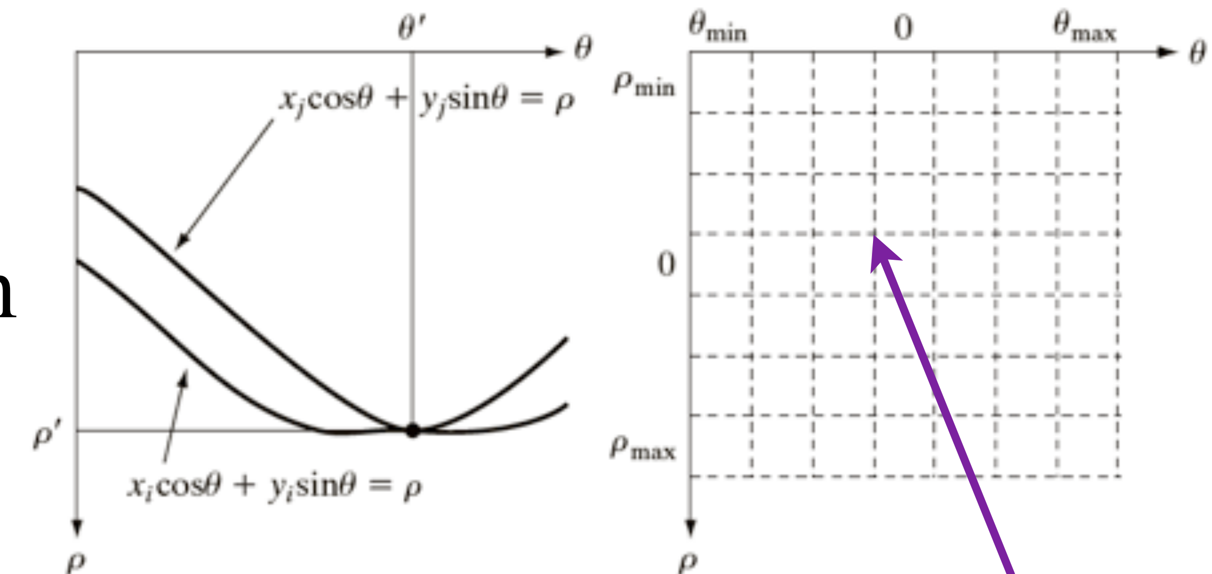
Hough Transform



Sinusoidal curves



The parameter space is divided in *accumulator cells* by discretizing ρ between -D and D and θ between -90º and 90º. Index by *p,q*.
Set *A(p,q) =0.*



For any point *(xi, yi)*, we take the discretized θ and solve for

$$\rho = x_i \cos\theta + y_i \sin\theta$$

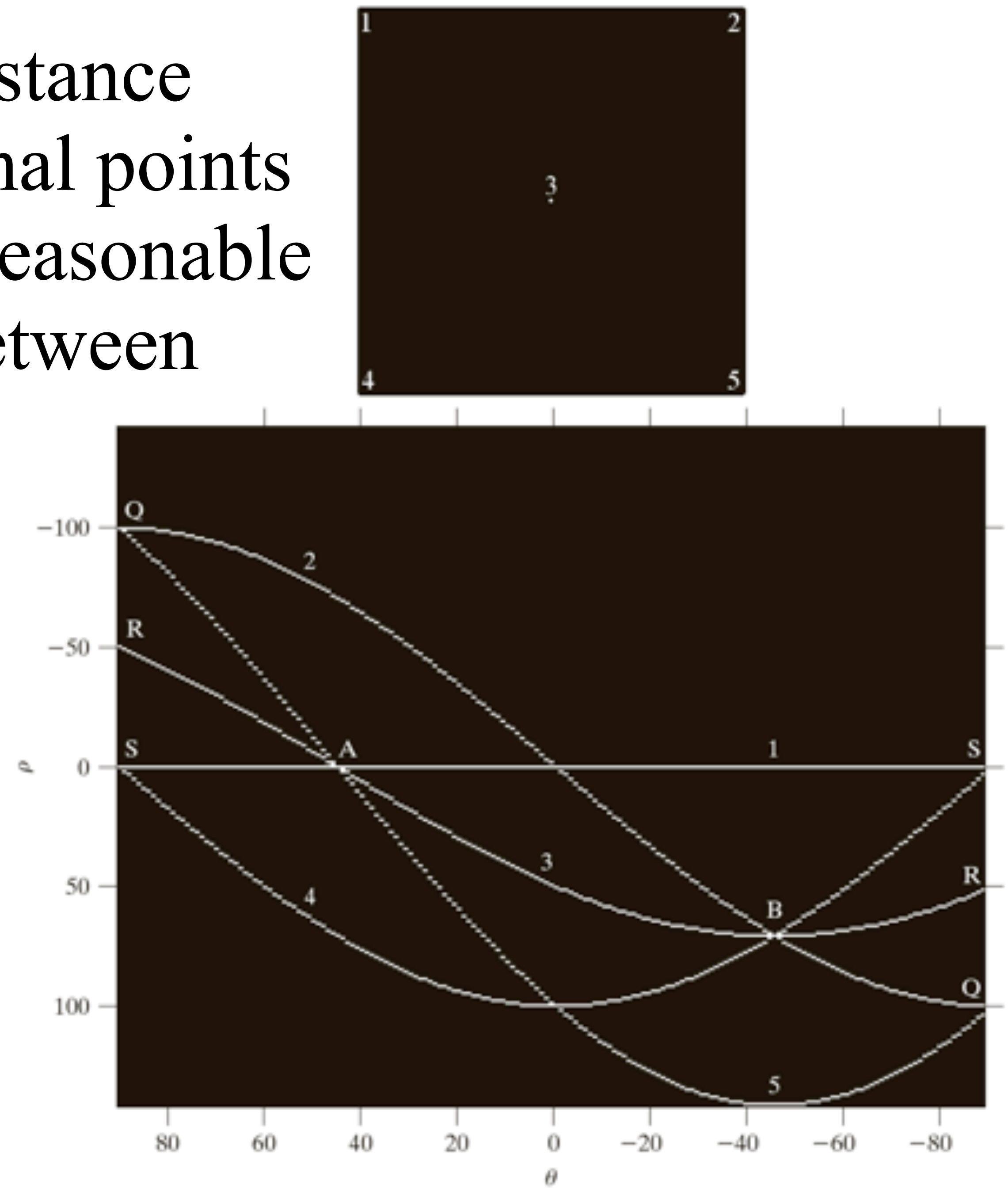
A(p,q)

The result is rounded to the nearest discrete rho value.

If two points are on the same line, they will have the same discrete parameter values: increment the index in the accumulation cell: *A(p,q) = A(p,q)+1.*
At the end, count the number of points in each cell.

The accuracy of the result will depend on the size of the accumulation cells.

Here, if D is the distance between the diagonal points in the image, it is reasonable to discretize rho between $-\sqrt{2}$ D and $+\sqrt{2}$ D.



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.)

For more general curves: assume that the curve (function) is defined implicitly as

$$g(\mathbf{v}, \mathbf{c}) = 0$$

vector of parameters (coefficients)

vector of coordinates

Then we do precisely the same thing by exchanging the role of **v** and **c**. Then we discretize the parameter space in accumulation cells, A(i,j,k, ….), with as many indices as the number of the parameters.

For each point, we need to solve for the parameters (now the curves can be pretty arbitrary in this multidimensional space), and look for intersection of many such curves.

How to use the Hough transform for edge linking?

1. Obtain a binary edge image using any of the techniques discussed earlier
2. Discretize the $\rho\theta$ plane in accumulator cells
3. Look for cells with high concentration of points (high accumulation number)
4. Examine further the pixels in the same accumulation cell (for continuity). If the distance is small, bridge a gap.

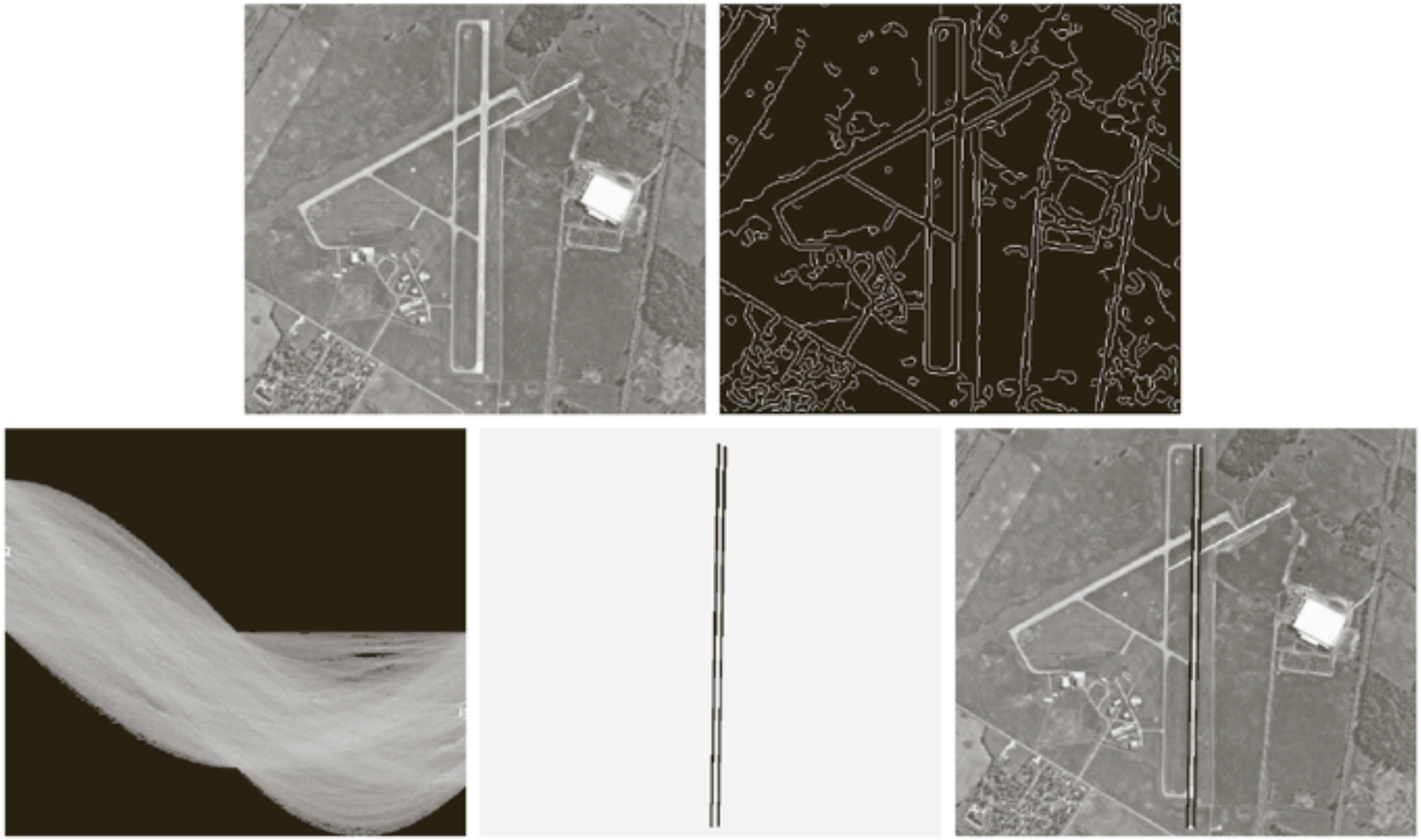
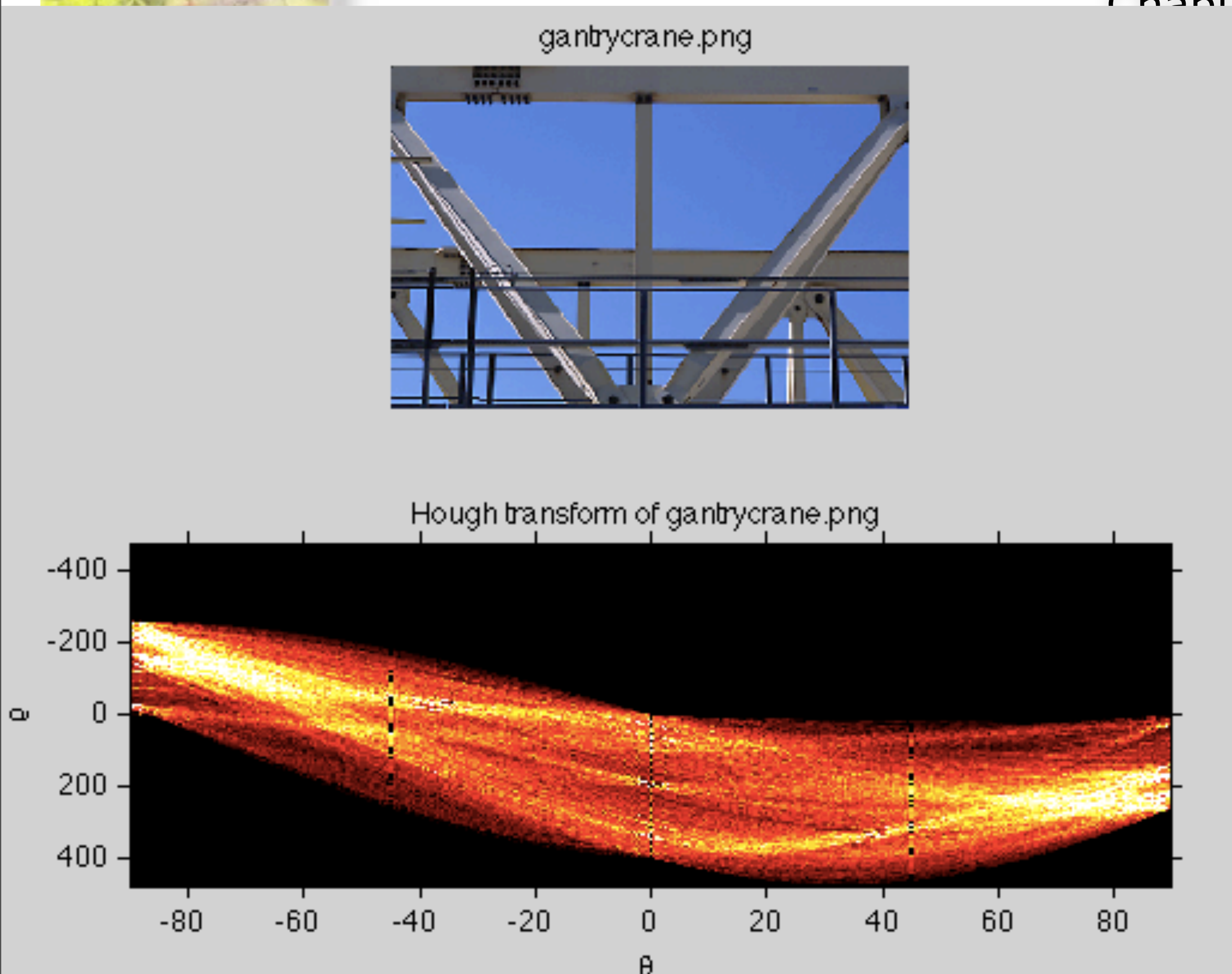
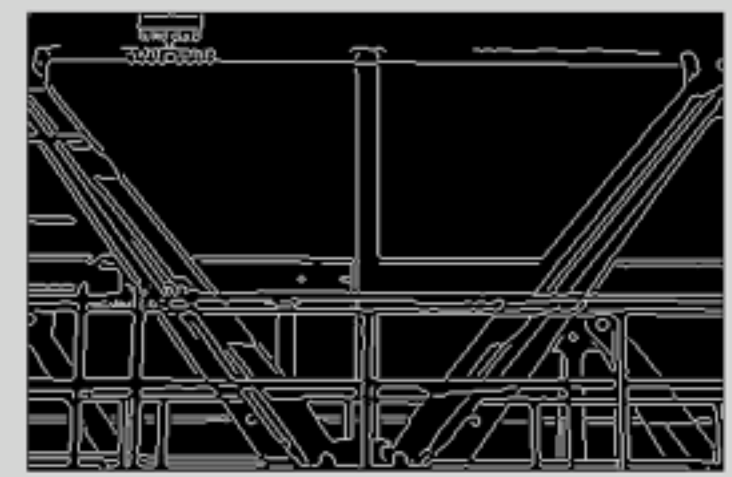
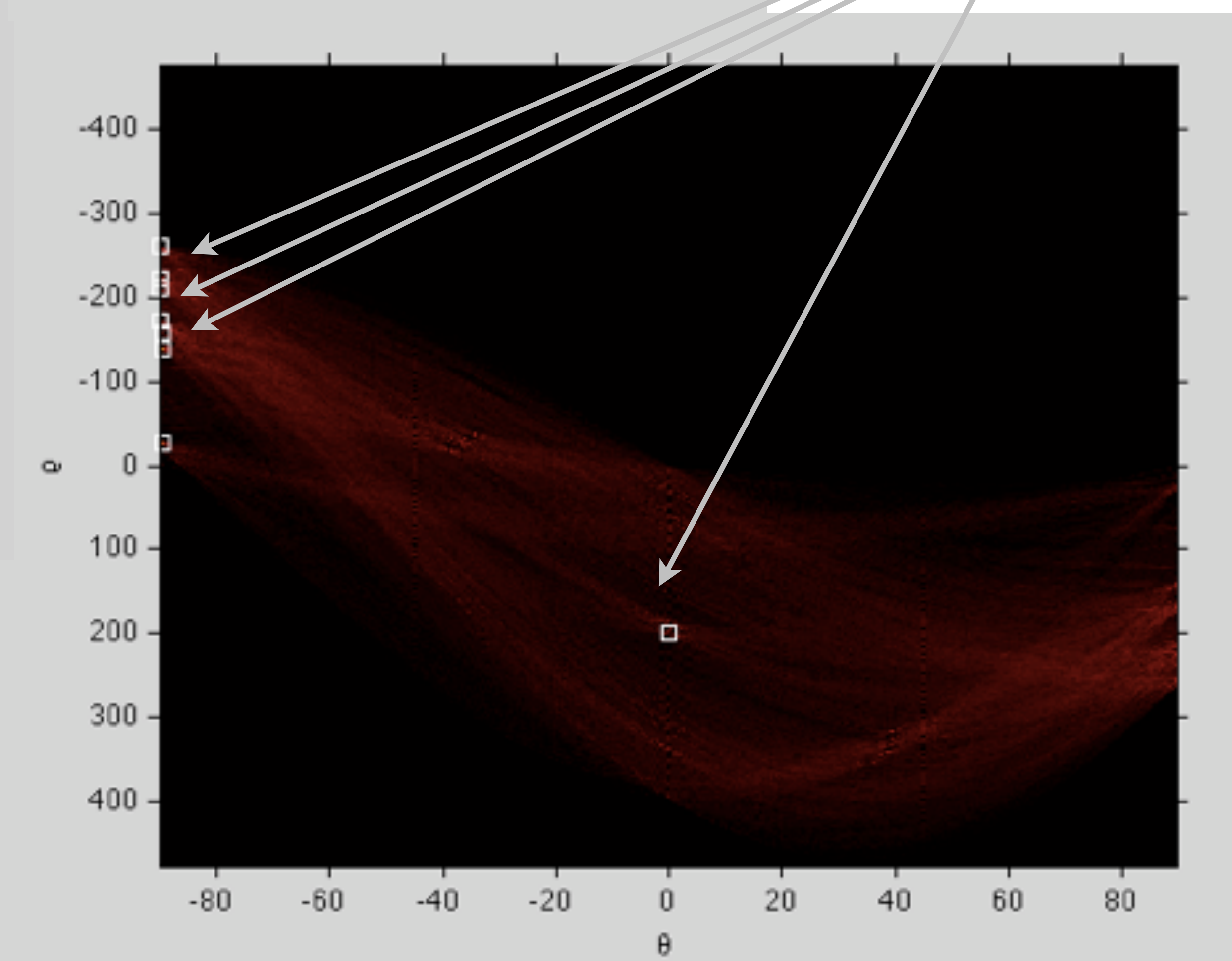


FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed on the original image.

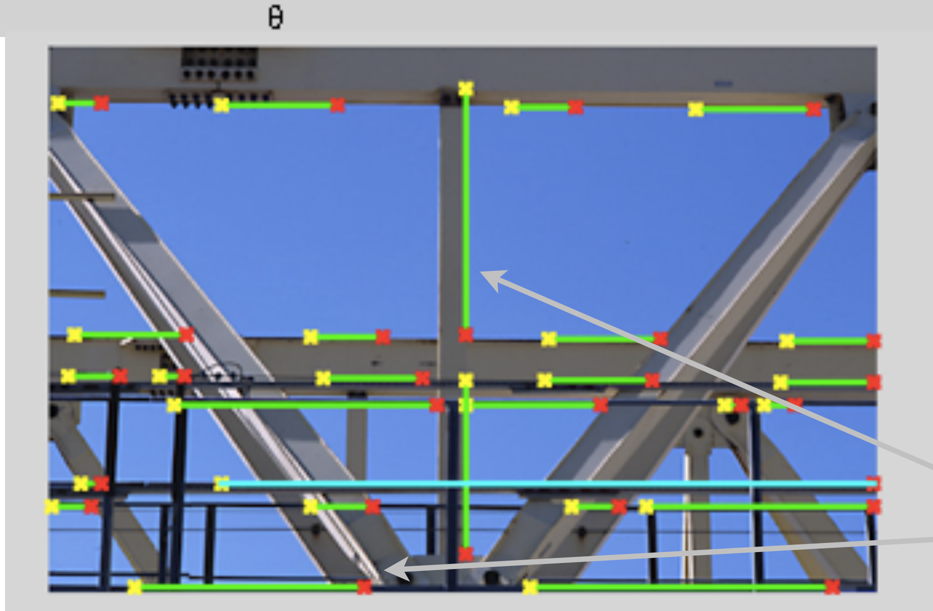
```
lookfor hough
HOUGH Hough transform.
HOUGHLINES Extract line segments based on Hough transform.
HOUGHPEAKS Identify peaks in Hough transform.


>> help hough
 HOUGH Hough transform.
    HOUGH implements the Standard Hough Transform (SHT). HOUGH is designed
    to detect lines. It uses the parametric representation of a line:

                        rho = x*cos(theta) + y*sin(theta).

    The variable rho is the distance from the origin to the line along a
    vector perpendicular to the line.  Theta is the angle between
    the x-axis and this vector. HOUGH generates parameter space matrix
    whose rows and columns correspond to rho and theta values respectively.
    Peak values in this space represent potential lines in the input image.

    [H, THETA, RHO] = HOUGH(BW) computes the SHT of the binary image BW.
    THETA (in degrees) and RHO are the arrays of rho and theta values over
    which the Hough transform matrix, H, was generated.

    [H, THETA, RHO] = HOUGH(BW,PARAM1,VAL1,PARAM2,VAL2) sets various
    parameters.  Parameter names can be abbreviated, and case does not
    matter. Each string parameter is followed by a value as indicated
    below: …
```



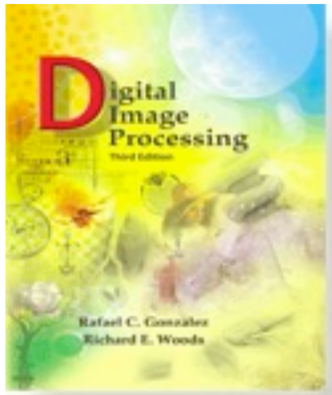




Pick 8 strongest lines



Plot 8 strongest lines



Methods based on intensity thresholding

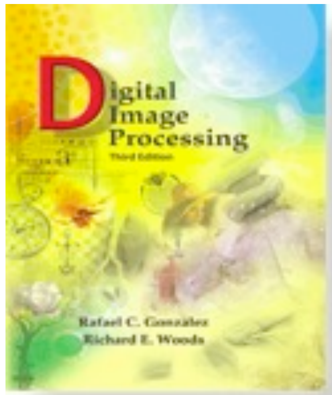
A simple and useful method for segmentation is that based on looking at the intensity statistics of the image:

image => histogram => look at the statistics of the pixel distribution

Simplest case:
-image consisting of an object and a background. The image will have two predominant tones. If we can separate the tones by a threshold *T*, segment the image:

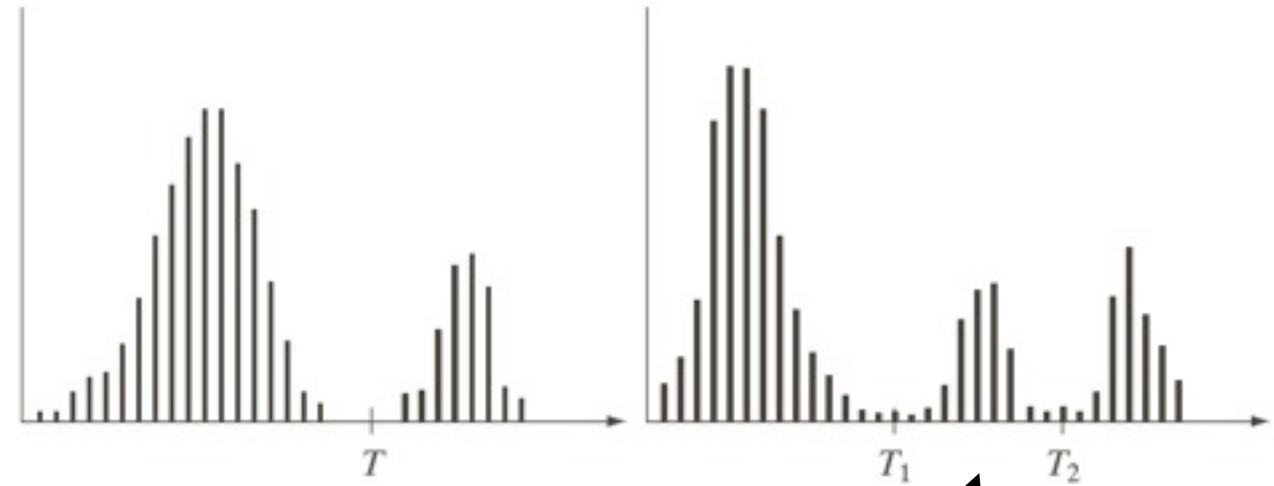
$g(x,y) = 1$ if $f(x,y)>T$

$g(x,y) = 0$ otherwise



The procedure is similar for 3 or more dominant modes:

$g(x,y) = a$ if $f(x,y) > T2$
$g(x,y) = b$ if $T1 < f(x,y) \leq T2$
$g(x,y) = 0$ otherwise



Histogram for 2 dominant modes. Note that two well separated peaks

Histogram for 3 dominant modes. Also here the peaks are well separated

In general, unless the image is very particular, images with more than 2 or 3 principal modes are extremely difficult to segment using intensity threshold.

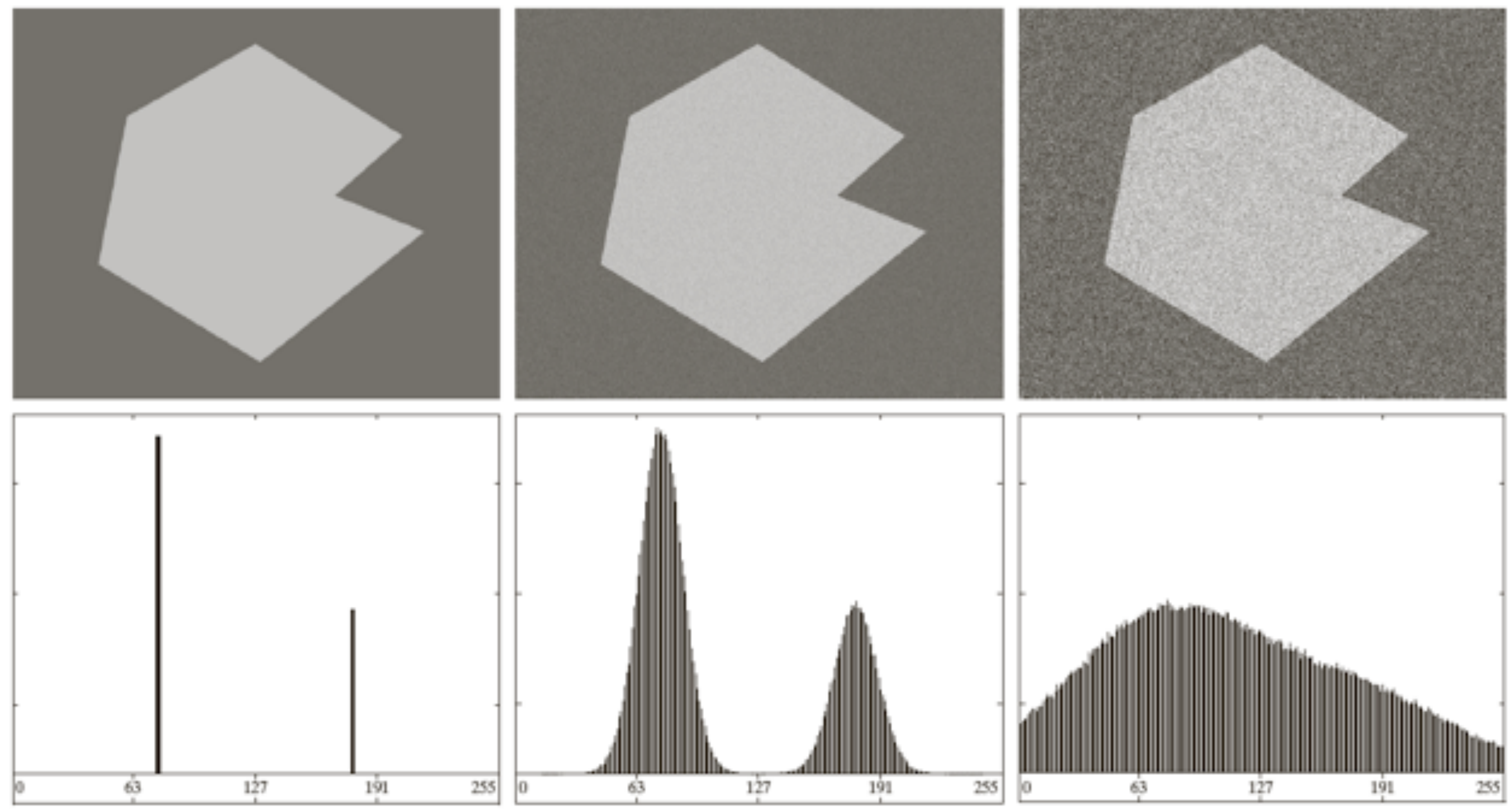
Key factors affecting the result of the intensity thresholding:

1) Separation between the peaks
2) The content of the image itself and the noise
3) The size of the objects and background
4) Illumination and reflectance properties of the object(s) in the image

Role of noise:

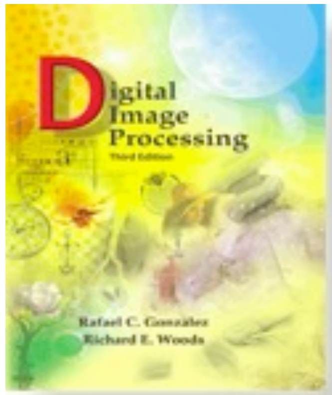
The noise broadens the modes of the histogram.

If there is too much noise, the mode merge and it is not possible to find a threshold for segmentation.

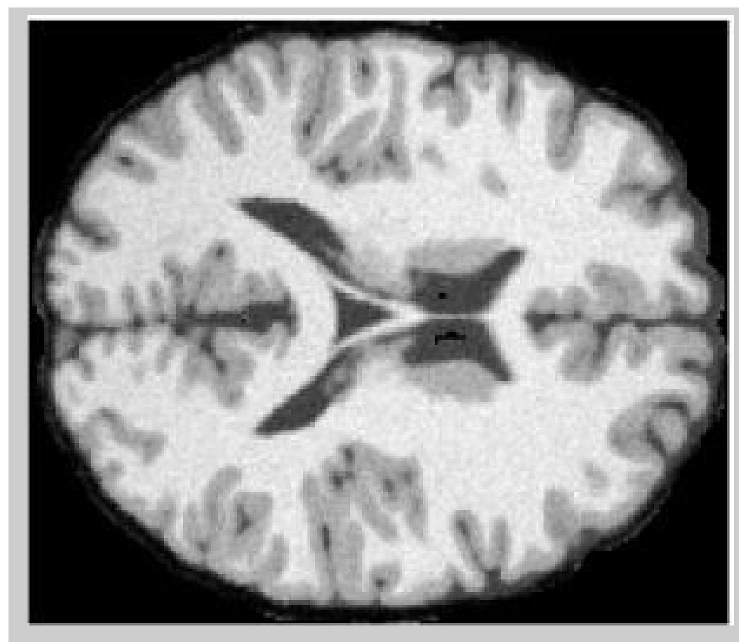


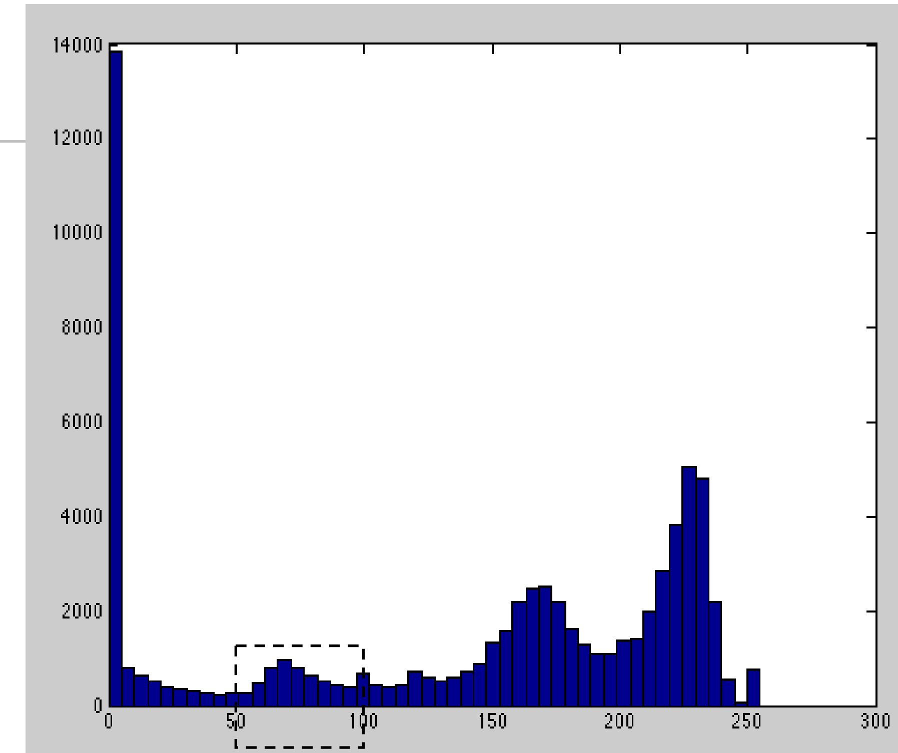
a b c
d e f

FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

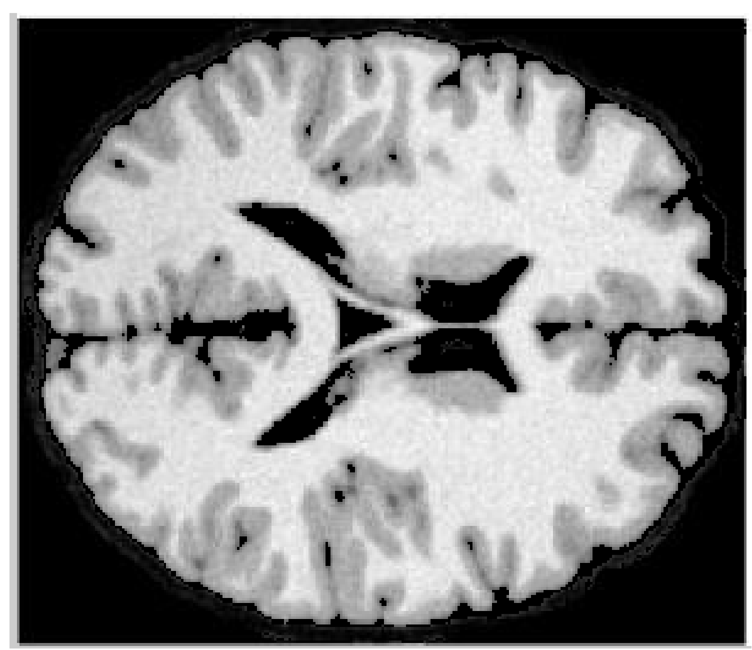


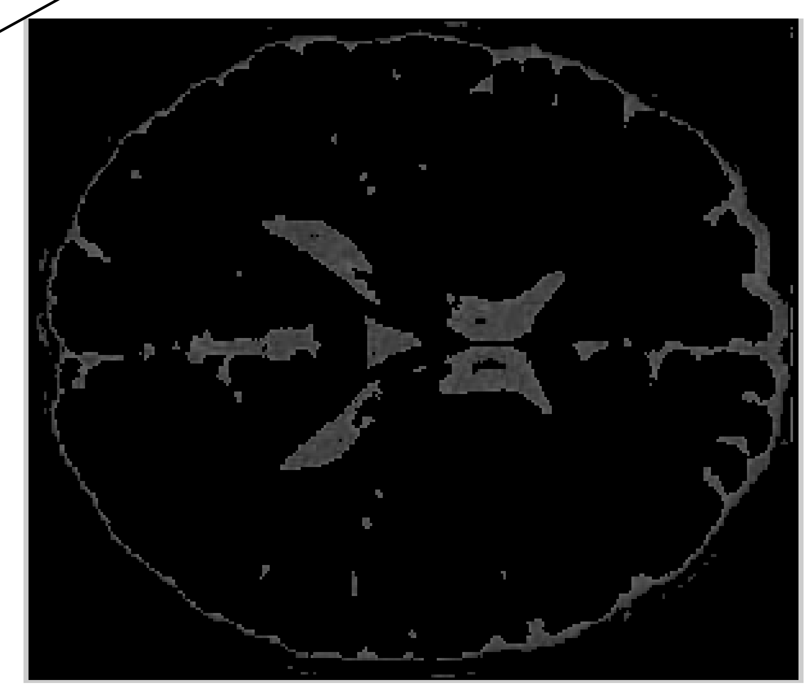
Digital
Image
Processing

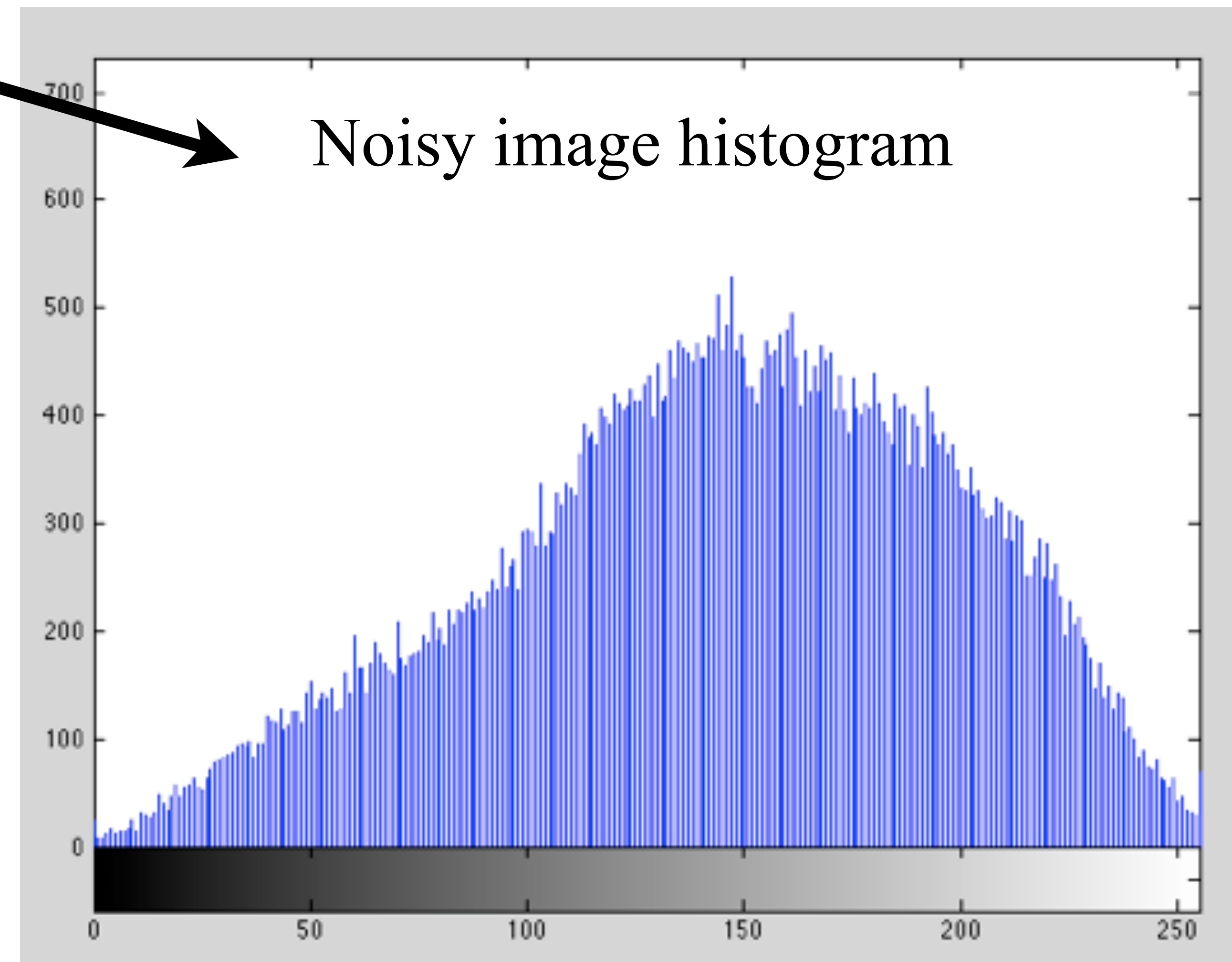




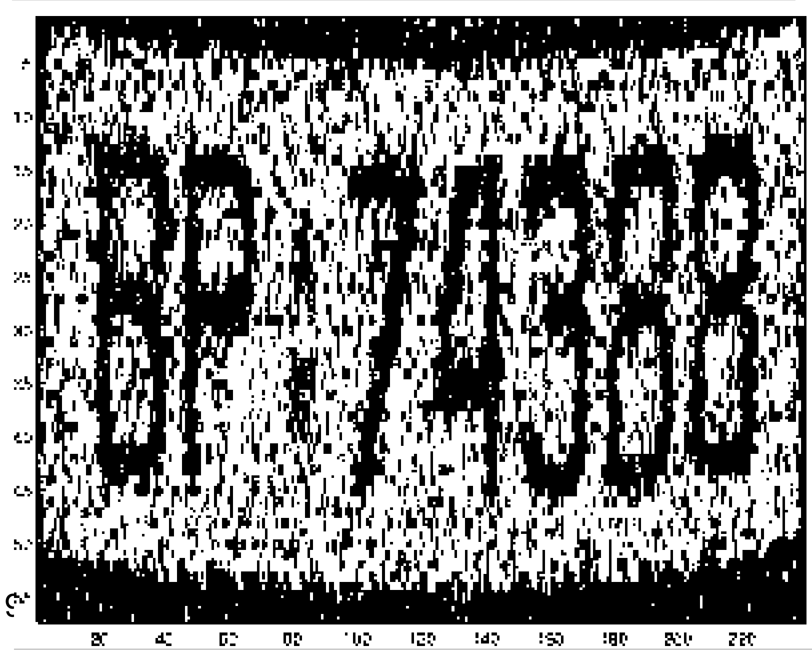
14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
300







Threshold=100



Threshold = 150

Tipstelefon: 02211 SMS/MMS: 2211 E-post: tips@bt.no Sentralbord: 05500

FINN Ann

bt.no

Bergen i dag: 4°C

tirsdag 27. mars 2007

KUNDESENTER DAGENS AVIS KONTAKT OSS

Nyheter
- Bergen
- Hordaland
- Sogn og Fjordane
- Innenriks
- Utenriks
- BTV - web-tv

Sport
- Fotball
- Håndball
- Basket
- Sting
- Ishockey
- Vintersport
- Ekstremsport

Bergenpuls
- Kulturnyheter
- Film
- Musikk
- Kjendiseri
- Litteratur
- TV og media
- Spill
- Scene
- Konsert

Fire av ti slipper unna

Fire av ti råkjørere som blir registrert i fotoboksene slipper unna forelegg på grunn av dårlige bilder. **Og verst er bildene i Hordaland.**

LES MER : TRAFIKK

Illumination and reflectance modify the histogram.

Here, the “valley” between the two modes disappears, and a third peak is generated.

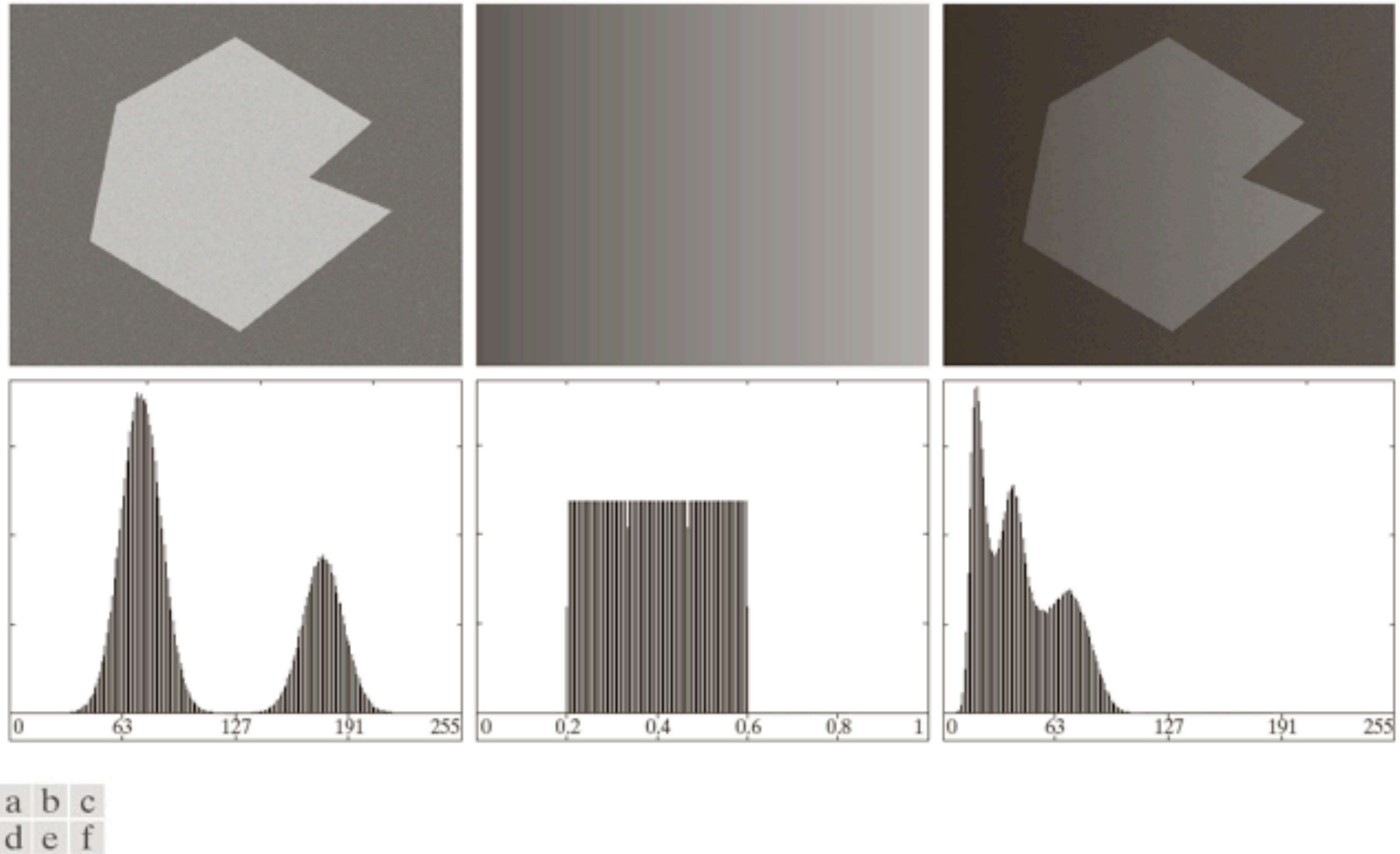
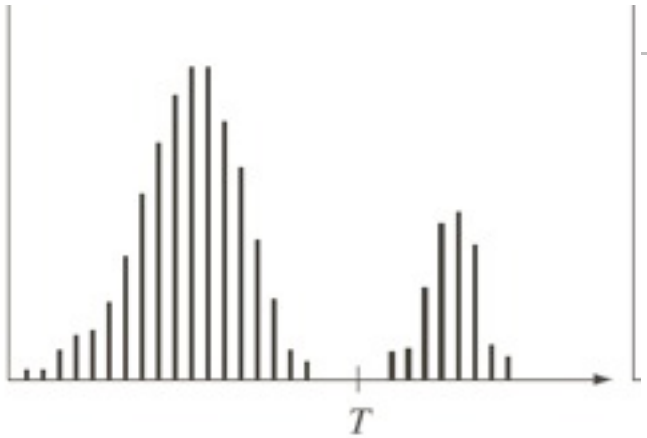


FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

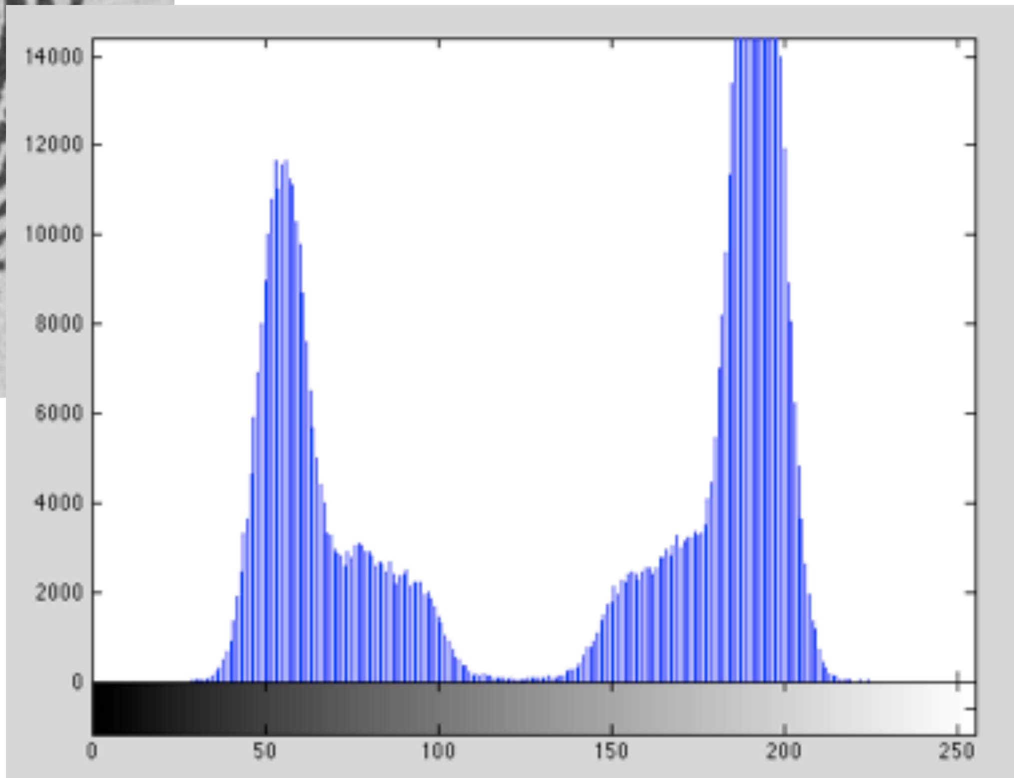


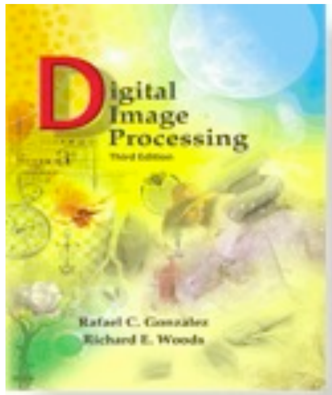
A basic single thresholding algorithm:
finds the “optimal” threshold value

1. Select an initial threshold *T*
2. Segment (by thresholding) using *T* in two groups of pixels, *G1, G2*
3. Compute the mean intensity values *m1, m2* in the two groups
4. Set *T=(m1+m2)/2*
5. Repeat 2-4 until the difference between two successive threshold values is less than a prescribed tolerance.

The initial threshold can be chosen as the average intensity of the image

T0=139
T=125
nit=3





Otsu's method: this method maximize the between-class variance

- well thresholded classes should be distinct wrt their intensity values
- the threshold giving the best separation in terms of intensity values is the optimal threshold.

Starting point: the image histogram:

0, 1, 2, … , L-1	intensity values
$n_0, n_1, n_2, \dots, n_{L-1}$	number of pixels

normalized histogram

Recall that *MN* is the number of pixels in the image,

$$\sum_k n_k = MN$$

$$p(r_k) = \frac{n_k}{MN} \qquad \sum_k p(r_k) = 1$$

Probability that a random pixels has intensity r_k.

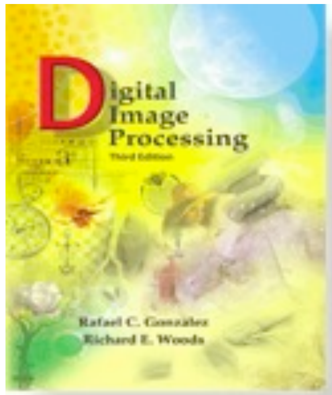
Now, assume a threshold *T(k)=k, 0≤k≤L-1,* is given: this threshold will divide the image in two classes, *C1, C2*.

C1: *[0,k]*
C2: *[k+1, L-1]*

The probability of a pixel to be assigned to the class *C1* is given by the cumulative sum

$$P_1(k) = \sum_{i=0}^{k} p_i \qquad P_2(k) = 1 - P_1(k)$$

Next, we compute the mean intensity value for the class *C1*.



mean intensity in *C1*:

for each intensity, multiply intensity times probability, and sum over all intensities

$$m_1(k) = \sum_{i=0}^{k} iP(i|C_1)$$

Conditional probability

$$= \sum_{i=0}^{k} iP(C_1|i)\frac{P(i)}{P(C_1)}$$

(follows by Bayes' rule for conditional probability)

$$P(A|B) = P(B|A)\frac{P(A)}{P(B)}$$

Since we are dealing only with *C1*, *P(C1|i)=1*.
P(i) = pi
P(C1) = P1

$$= \frac{1}{P_1(k)}\sum_{i=0}^{k} ip_i$$

mean intensity in *C2*:

$$m_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i p_i$$

In addition to *m1(k)*, *m2(k)*, we consider the average cumulative intensity (up to level *k*)

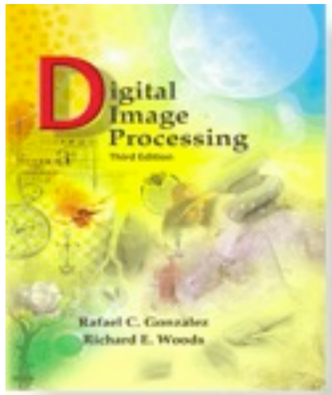
$$m(k) = \sum_{i=0}^{k} i p_i$$

and average intensity of the image

$$m_G = \sum_{i=0}^{L-1} i p_i$$

Note:

$$P_1 m_1 + P_2 m_2 = m_G$$



Now define the global variance: $$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_g)^2 p_i$$

and the *between-class* variance: $$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

A measure of the "goodness" of the segmentation is then

$$\eta = \frac{\sigma_B^2}{\sigma_G^2} = \frac{1}{\sigma_G^2} \cdot \frac{(m_G P_1(k) - m(k))^2}{P_1(k)(1 - P_1(k))}$$

and this is maximized when sigmaB^2 is maximized:

$$\sigma_B^2(k^*) = \max_{0 \le k \le L-1} \sigma_B^2(k)$$

Hence we compute

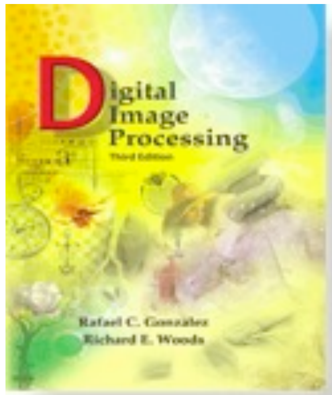
$$\sigma_B^2(k) = \frac{(m_G P_1(k) - m(k))^2}{P_1(k)(1 - P_1(k))}$$

for all values of k and select that k^* (those) that give a maximum. In case of ties, average them.

Note that $\eta(k^*)$ is then a measure of the separability of the classes

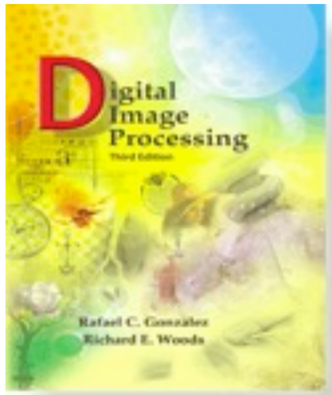
Summary, Otsu's method:

1. Compute normalized histogram
2. Compute cumulative sums
3. Compute cumulative means
4. Compute global intensity mean
5. Compute the between-class variance
6. Find the threshold k^*. If the maximum is not unique, average
7. Obtain the separability measure

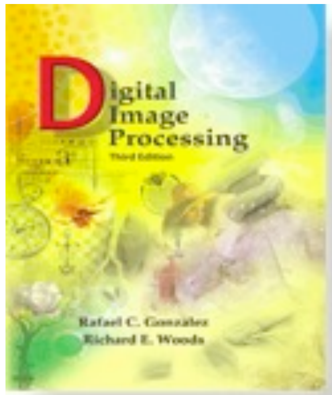


Some Matlab commands:

```
IM2BW Convert image to binary image by thresholding.
    IM2BW produces binary images from indexed, intensity, or RGB images. To do
    this, it converts the input image to grayscale format (if it is not already
    an intensity image), and then converts this grayscale image to binary by
    thresholding. The output binary image BW has values of 1 (white) for all
    pixels in the input image with luminance greater than LEVEL and 0 (black)
    for all other pixels. (Note that you specify LEVEL in the range [0,1],
    regardless of the class of the input image.)

    BW = IM2BW(I,LEVEL) converts the intensity image I to black and white.

    BW = IM2BW(X,MAP,LEVEL) converts the indexed image X with colormap MAP to
    black and white.

    BW = IM2BW(RGB,LEVEL) converts the RGB image RGB to black and white.

    Note that the function GRAYTHRESH can be used to compute LEVEL
    automatically.
```

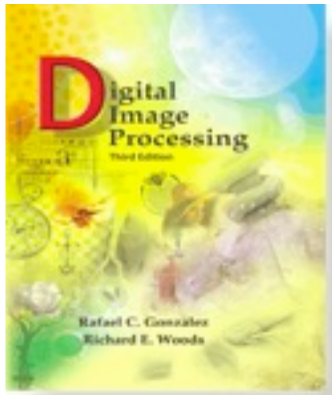



```
GRAYTHRESH Global image threshold using Otsu's method.
    LEVEL = GRAYTHRESH(I) computes a global threshold (LEVEL) that can be
    used to convert an intensity image to a binary image with IM2BW. LEVEL
    is a normalized intensity value that lies in the range [0, 1].
    GRAYTHRESH uses Otsu's method, which chooses the threshold to minimize
    the intraclass variance of the thresholded black and white pixels.

    [LEVEL EM] = GRAYTHRESH(I) returns effectiveness metric, EM, as the
    second output argument. It indicates the effectiveness of thresholding
    of the input image and it is in the range [0, 1]. The lower bound is
    attainable only by images having a single gray level, and the upper
    bound is attainable only by two-valued images.
```



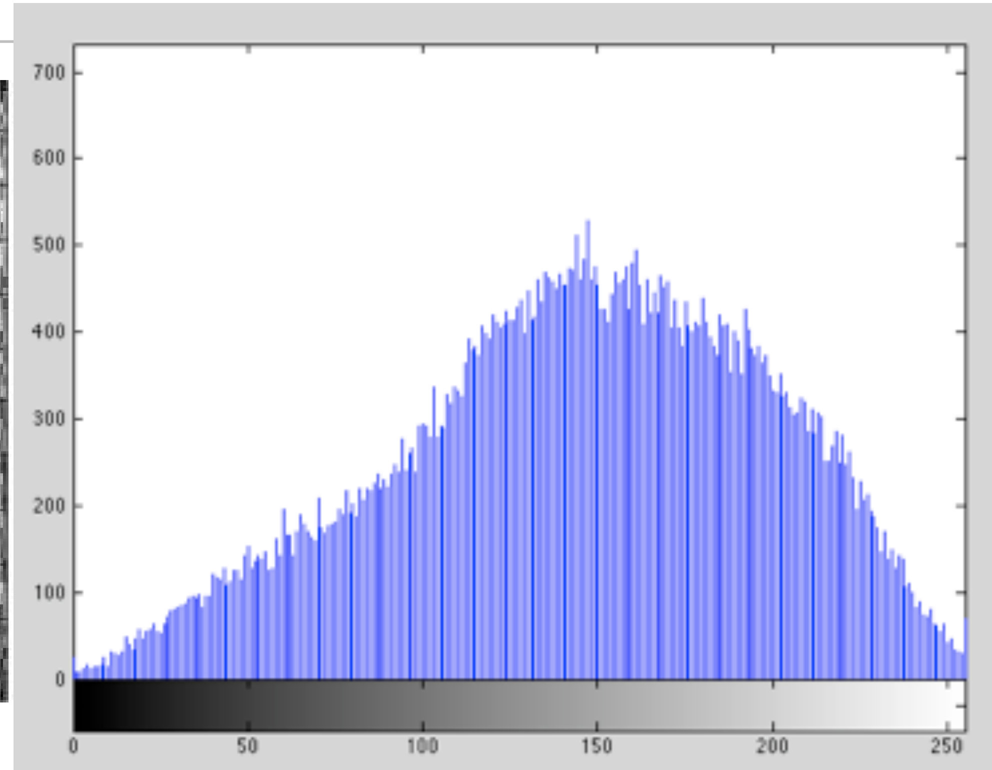
Using image smoothing for global thresholding:

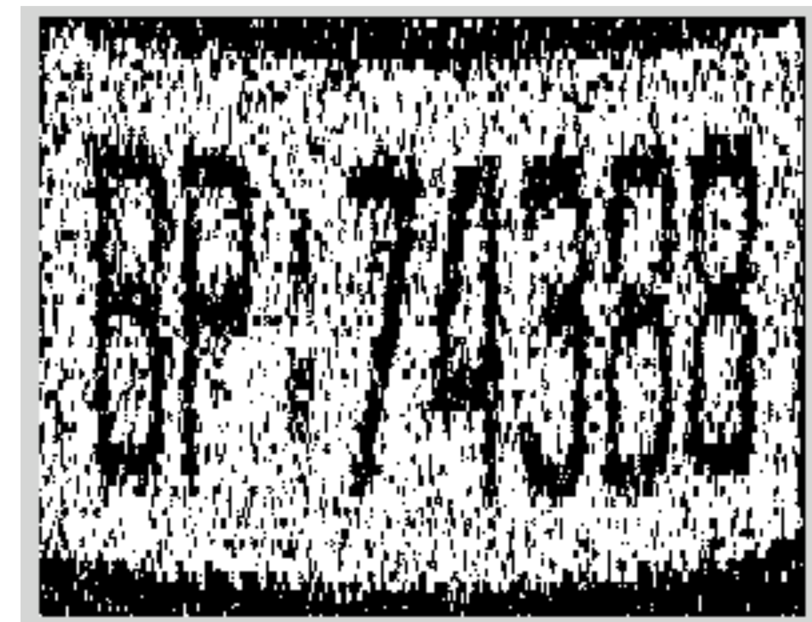
If the image contains noise, it is a good idea to smooth the image prior to segmentation.



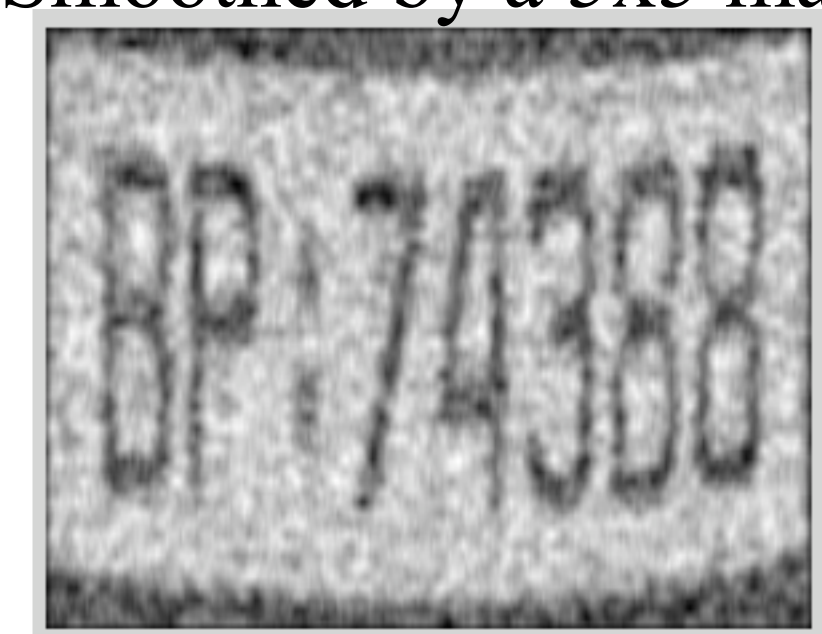
Segmentation by Otsu’s method

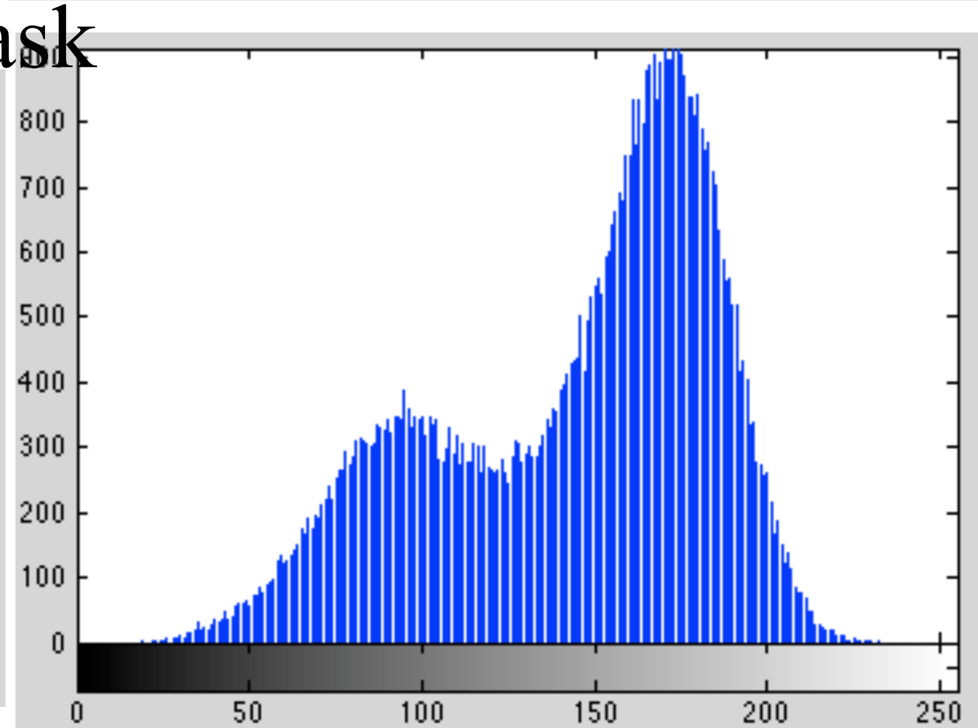
Original, noisy





Smoothed by a 5x5 mask



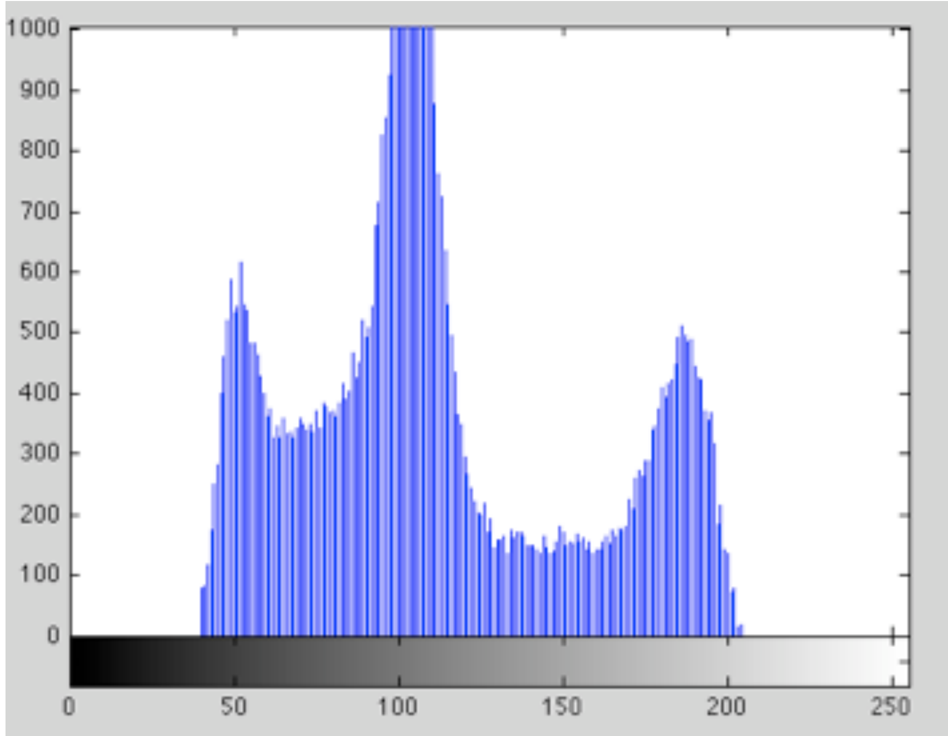


Similarly, to improve the result of segmentation, it is possible to combine the segmentation with edge detection and other filters.

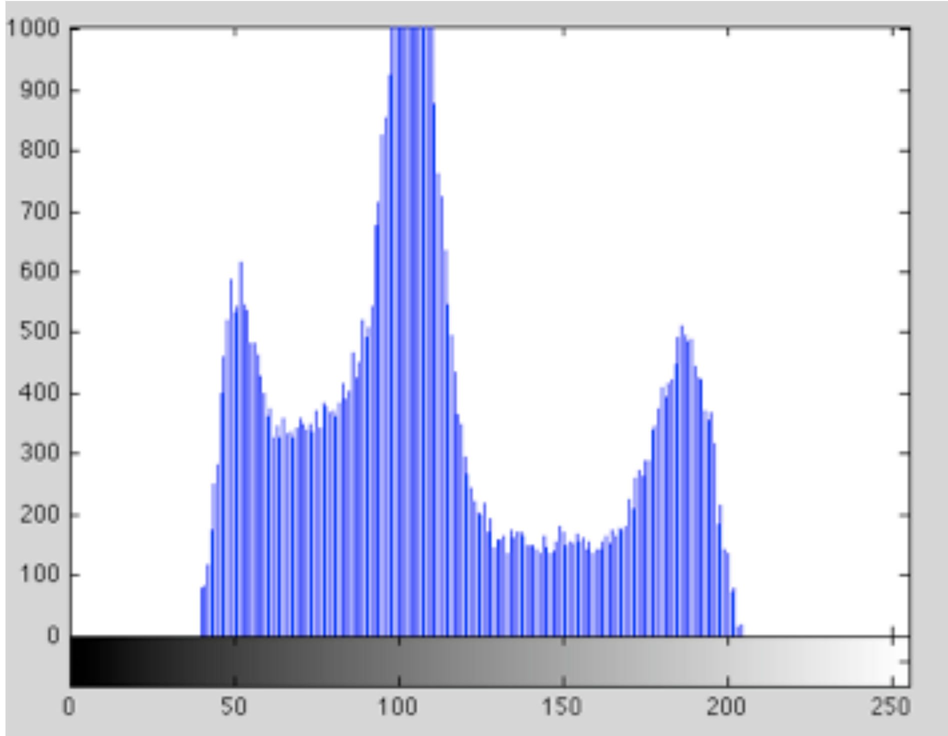
Otsu’s method can be extended in a very similar manner to multiple thresholds. The procedure is similar, see the textbook for details.

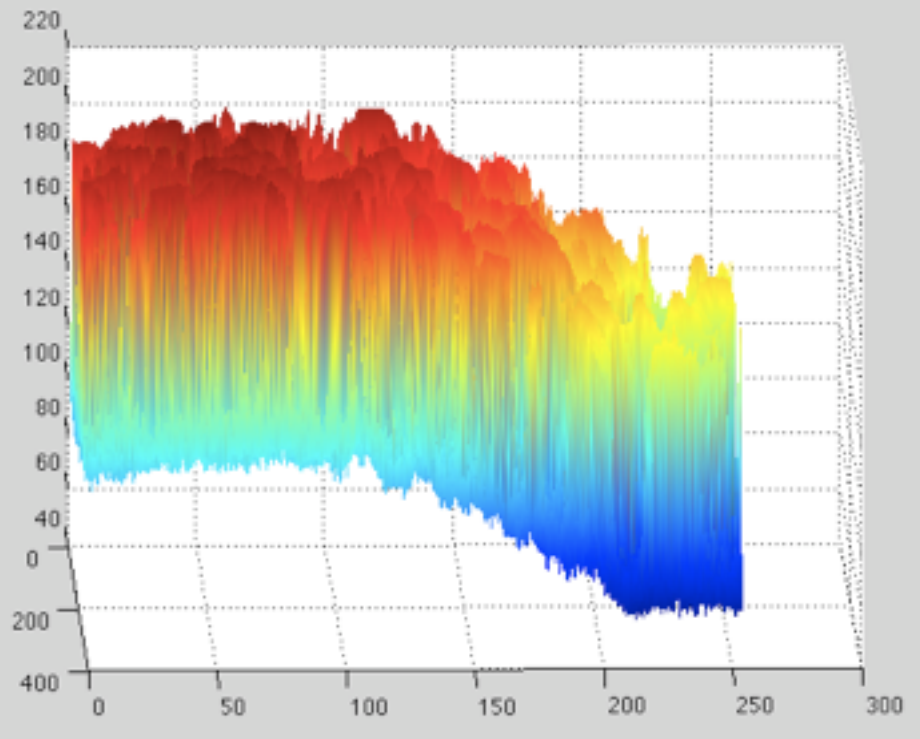
rice.png

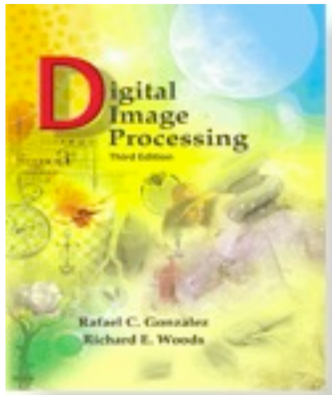
rice.png



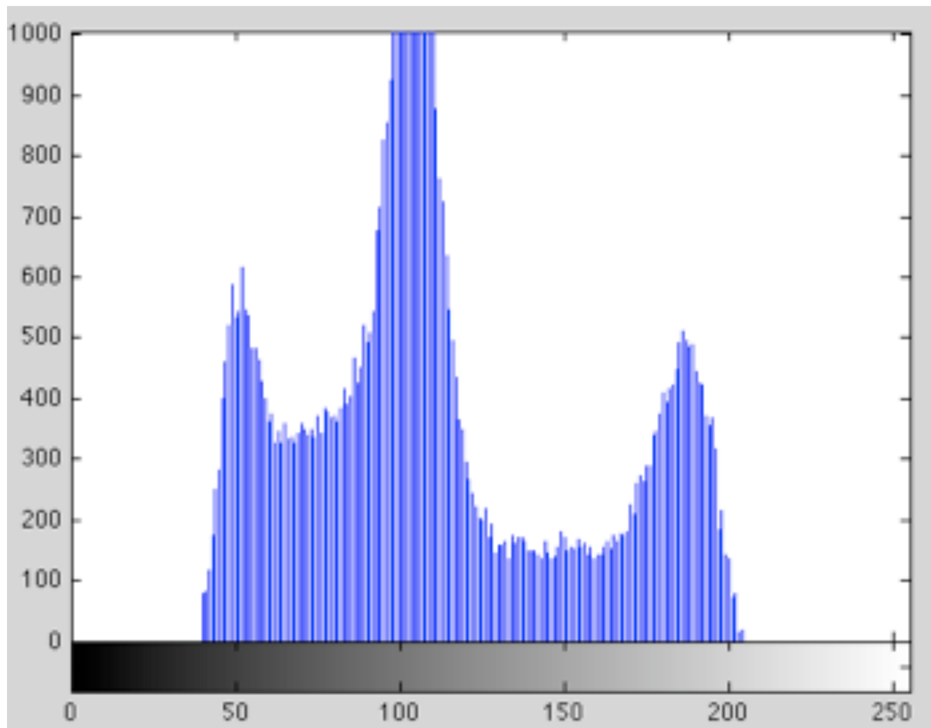
rice.png

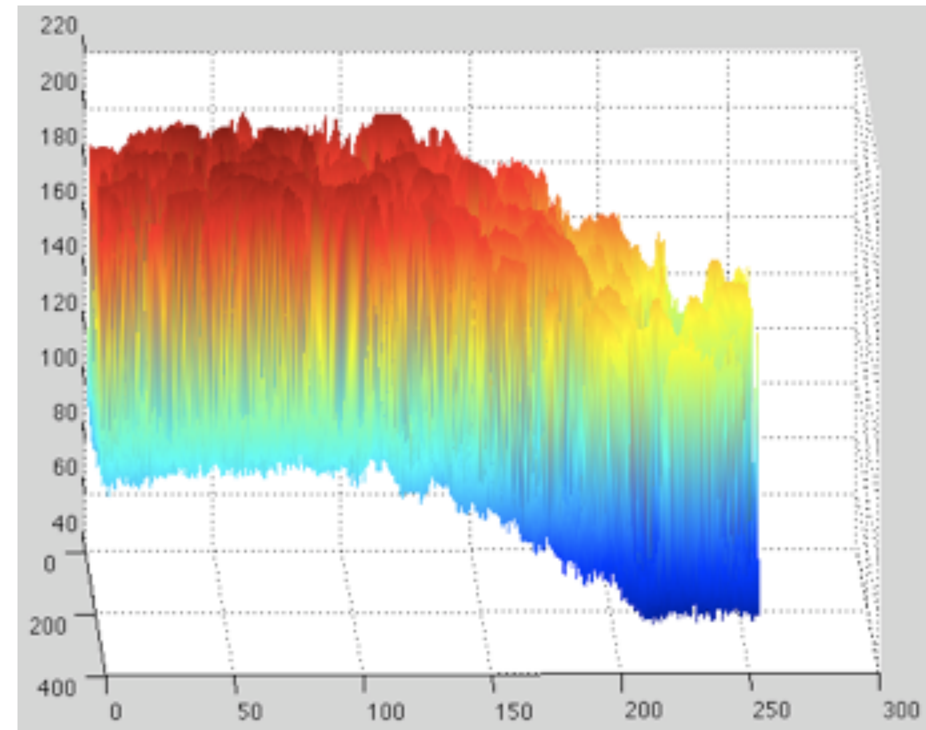


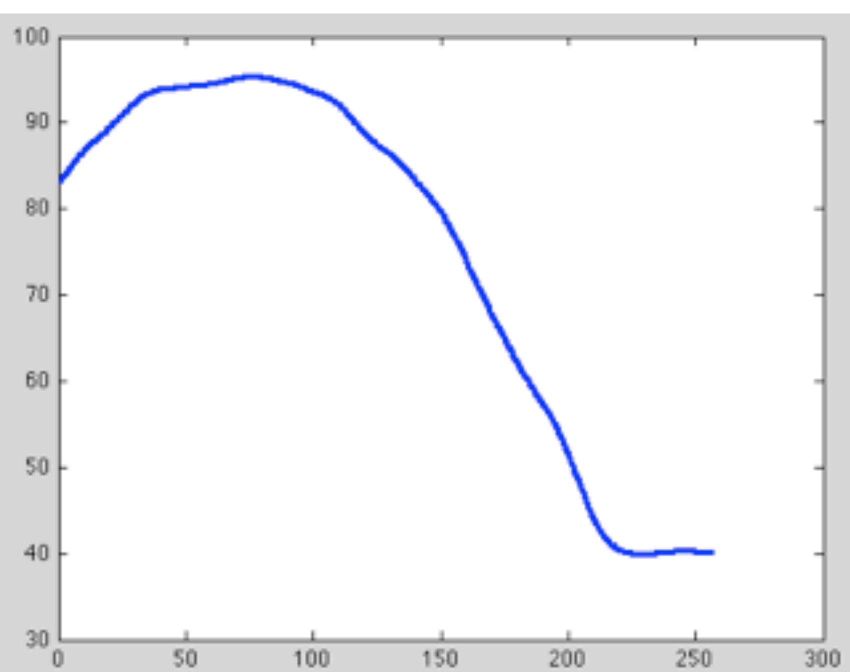


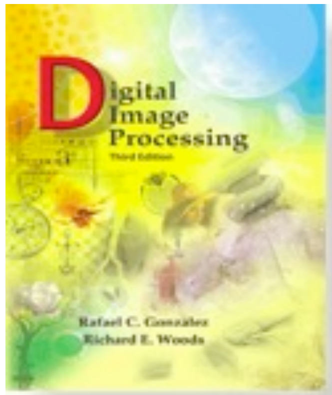

Digital
Image
Processing
Rafael C. Gonzalez
Richard E. Woods

rice.png

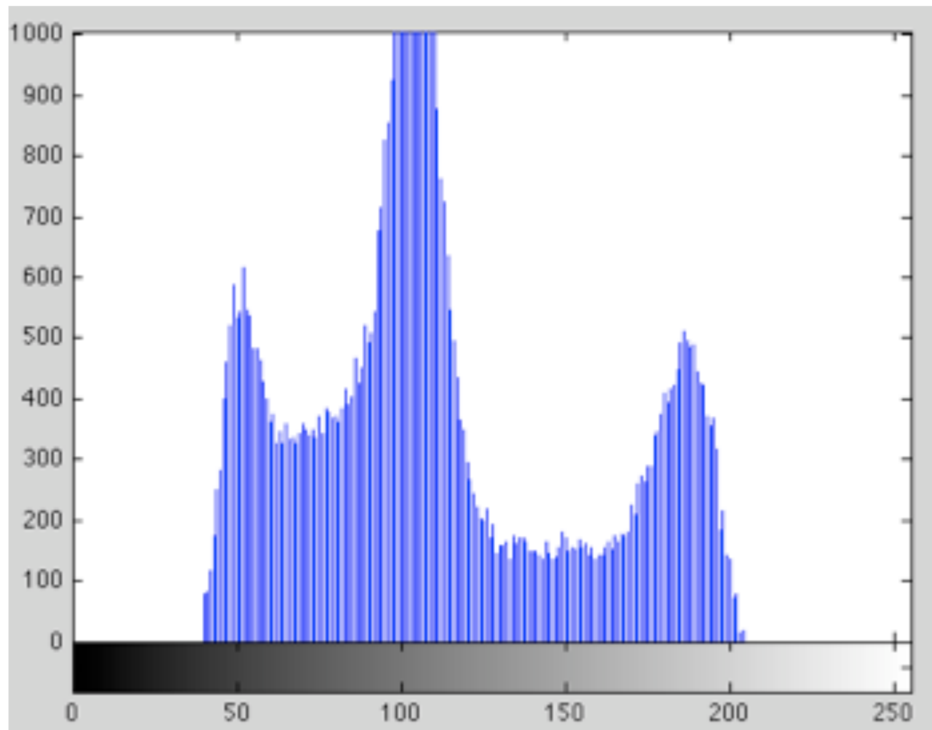

1000
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250

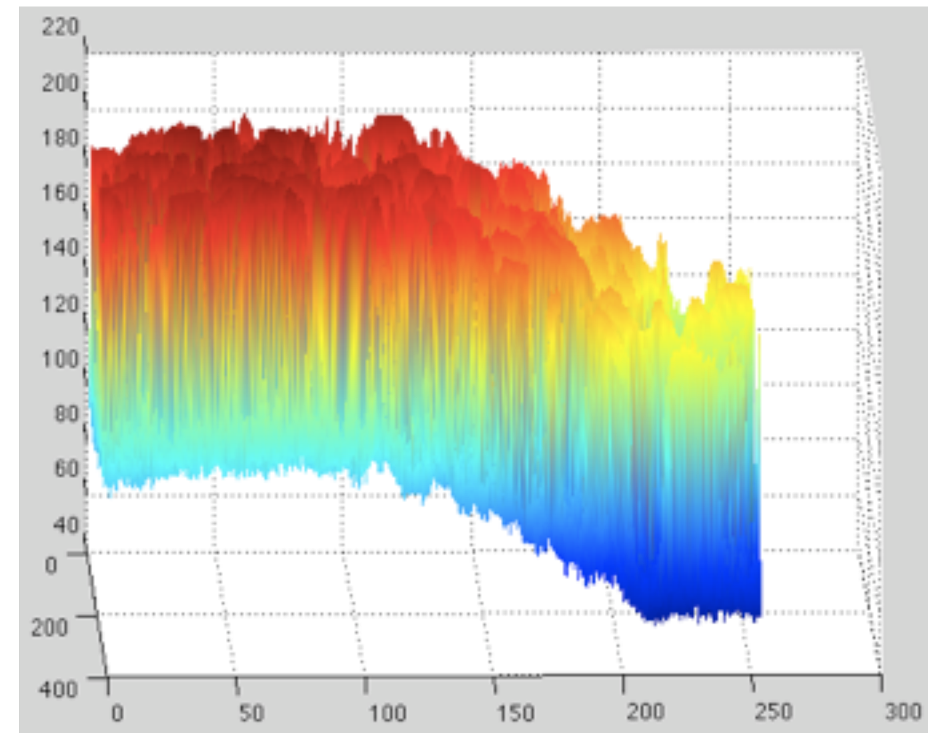

220
200
180
160
140
120
100
80
60
40
0
200
400
0
50
100
150
200
250
300

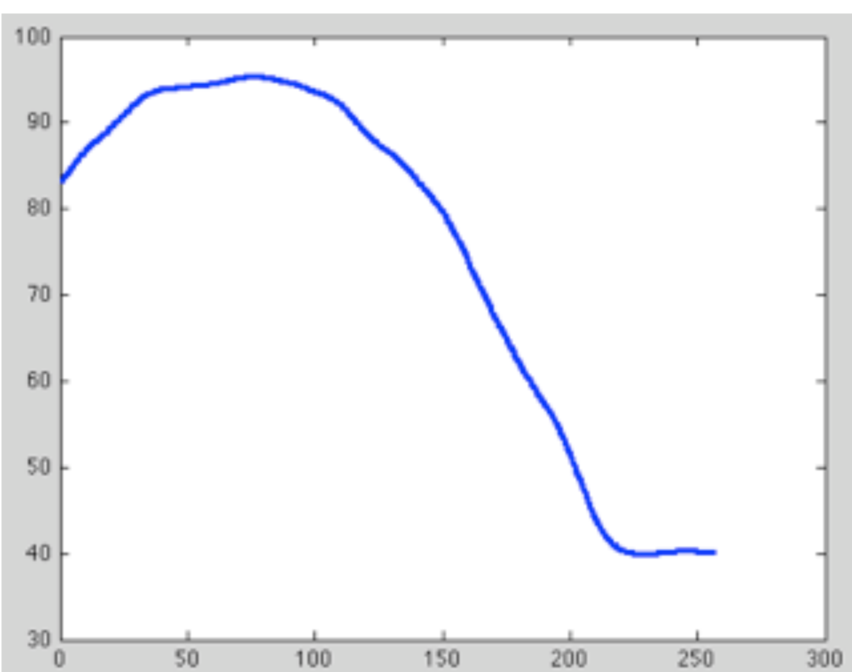

100
90
80
70
60
50
40
30
0
50
100
150
200
250
300

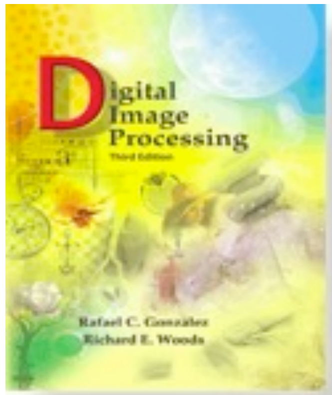


rice.png

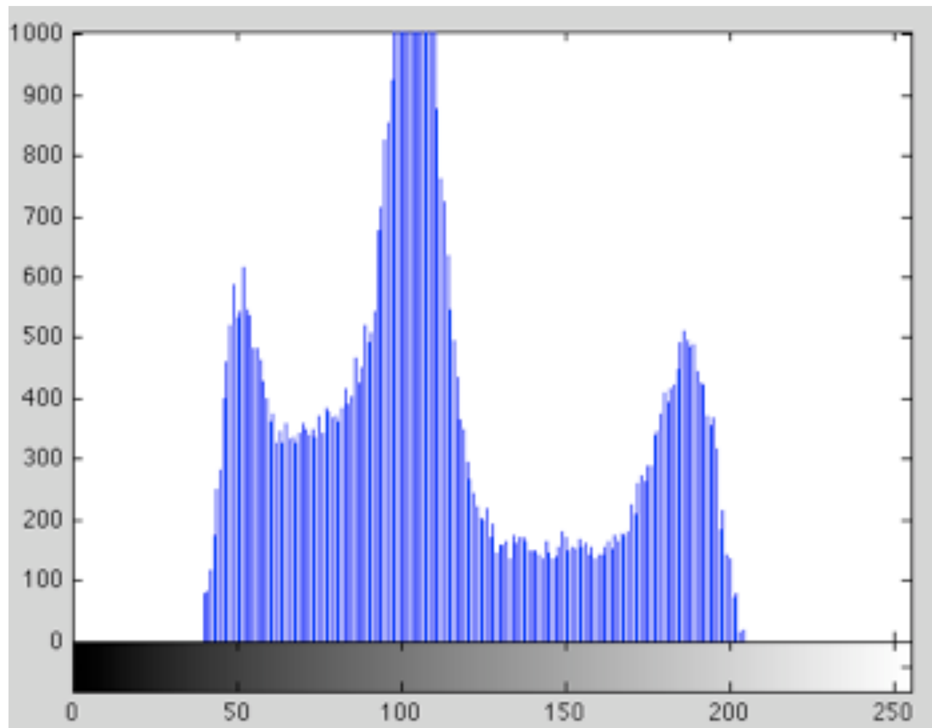

1000
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250

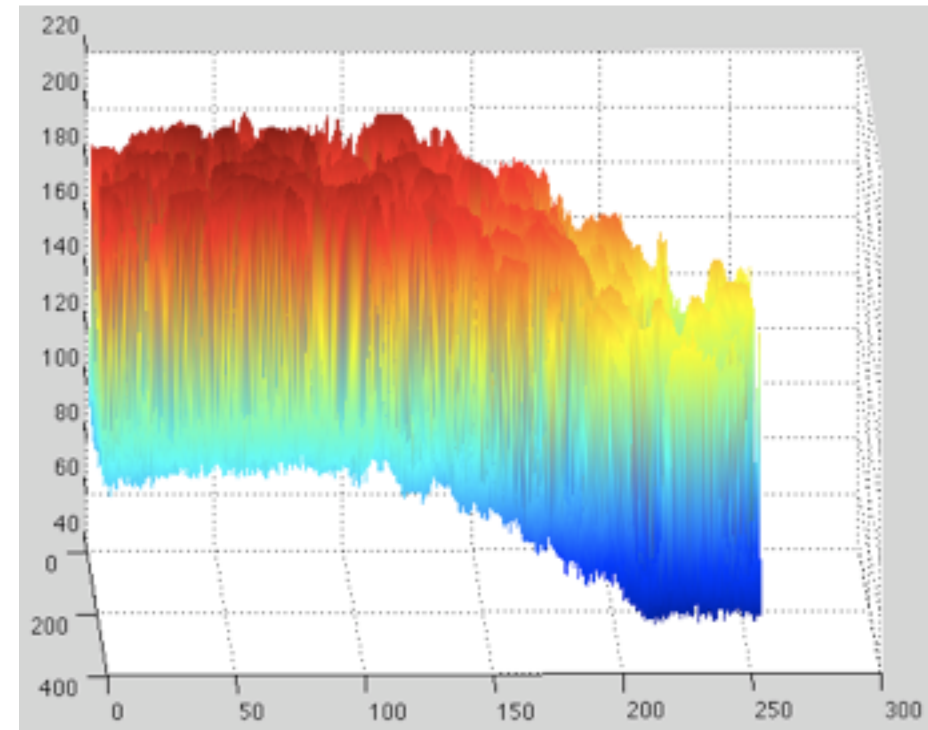

220
200
180
160
140
120
100
80
60
40
0
200
400
0
50
100
150
200
250
300

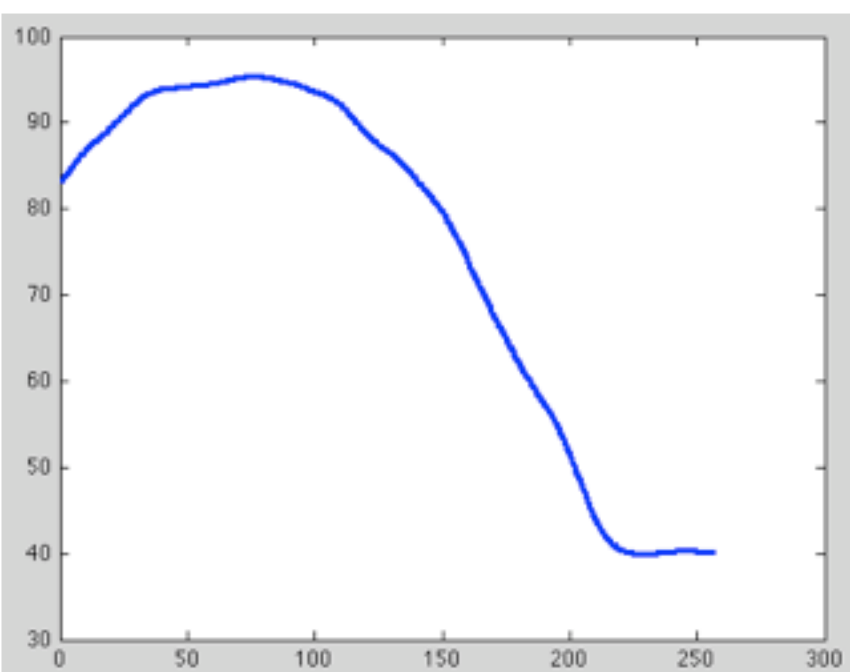

100
90
80
70
60
50
40
30
0
50
100
150
200
250
300

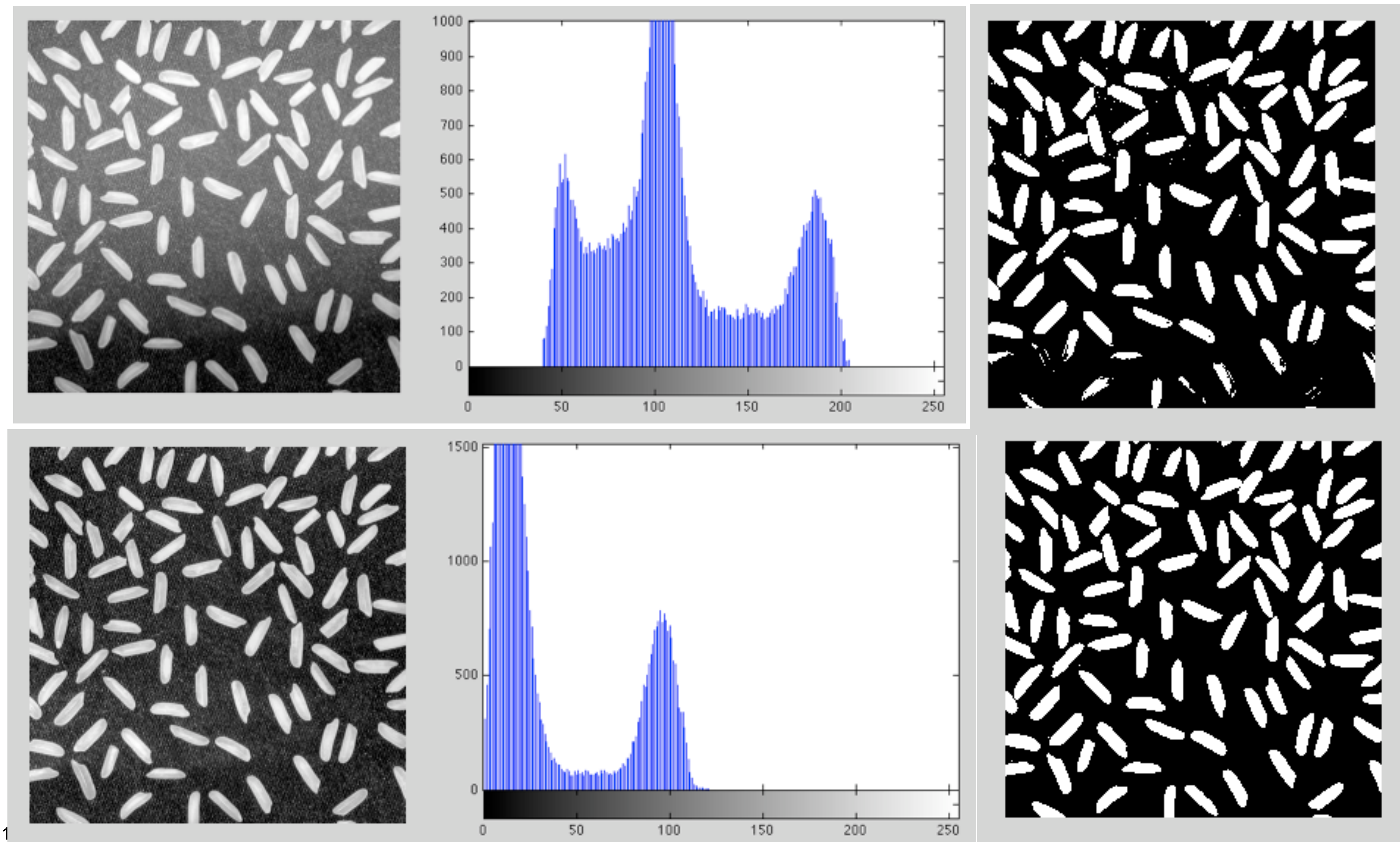

Digital
Image
Processing

rice.png

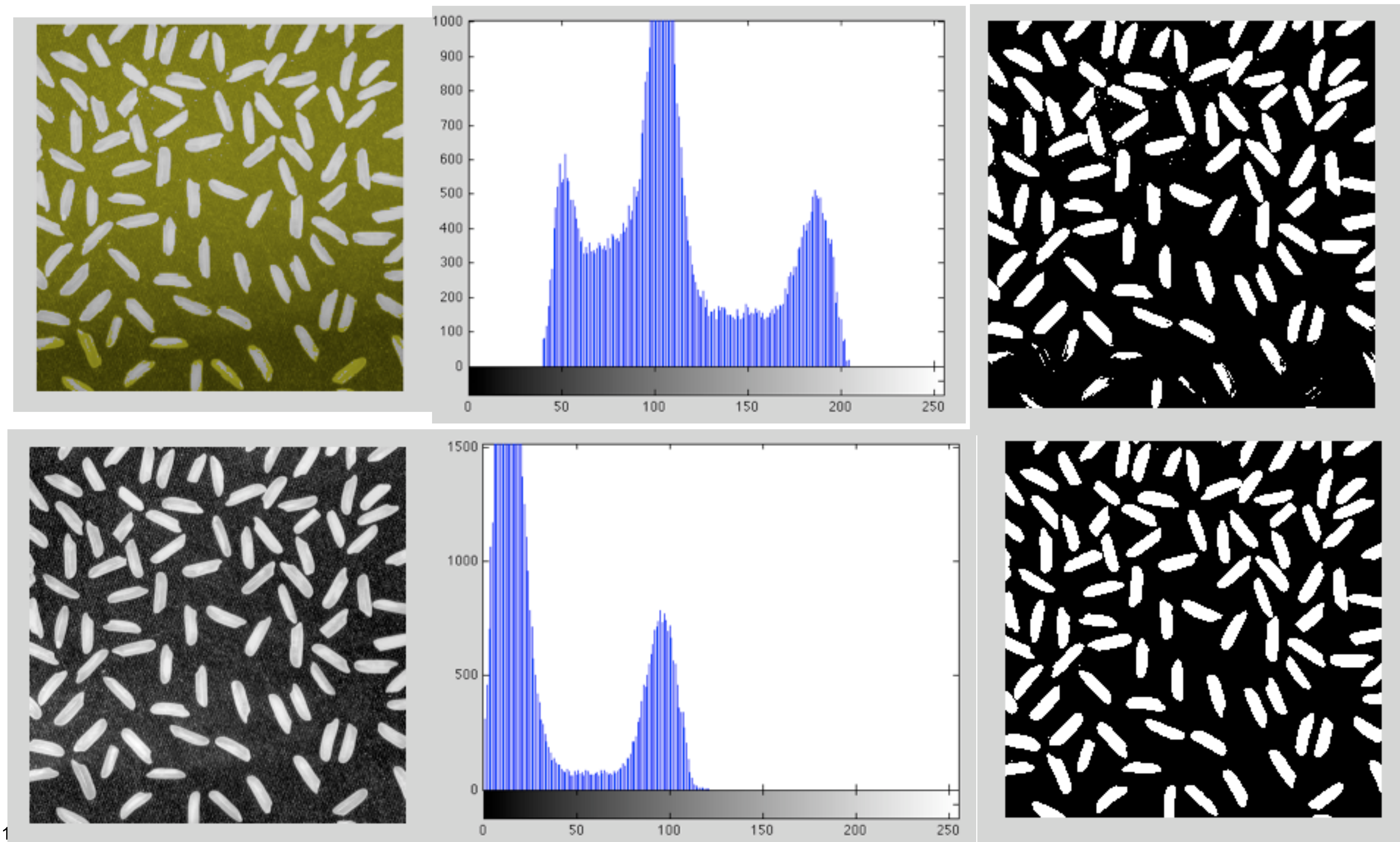

1000
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250

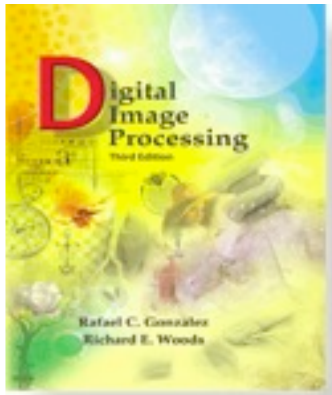

220
200
180
160
140
120
100
80
60
40
0
200
400
0
50
100
150
200
250
300


100
90
80
70
60
50
40
30
0
50
100
150
200
250
300



1000
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250
1500
1000
500
0
0
50
100
150
200
250


1000
900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250
1500
1000
500
0
0
50
100
150
200
250



Chapter 10

Segmentation