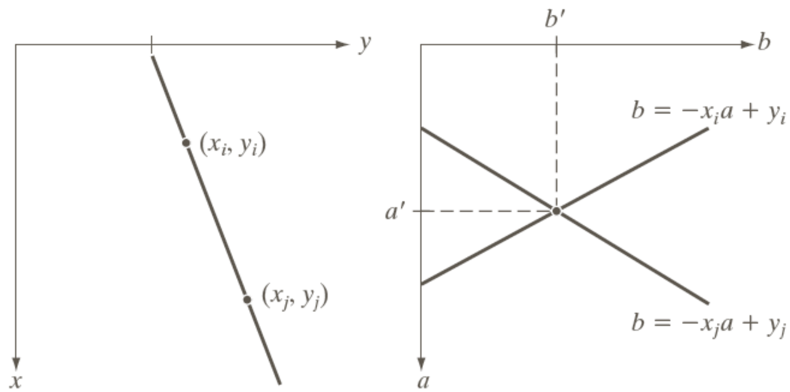
Lecture #21

- Hough transform
- Graph searching
- Area based segmentation
- Thresholding, automatic thresholding
- Local thresholding
- Region segmentation

Hough Transform



a b

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

Points (x_i,y_i) and (x_j,y_j)

Define a line $y=ax+b$ in x-y space

Rearrange equation as $b=-ax+y$

For every point (x_i,y_i) there is a corresponding line describing all the possible slopes and y-intercepts. Their intersection is the original line in x-y space.

© 2002 R. C. Gonzalez & R. E. Woods

Hough Transform

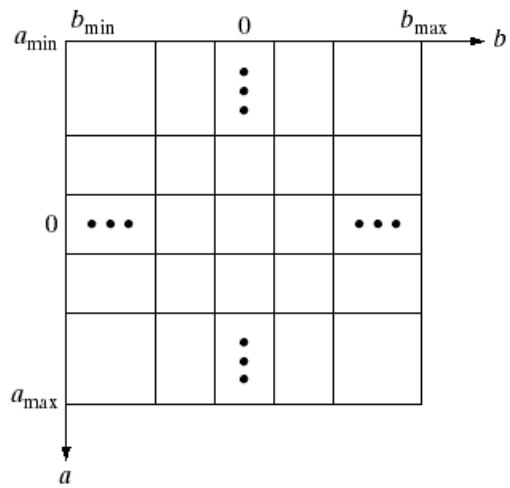
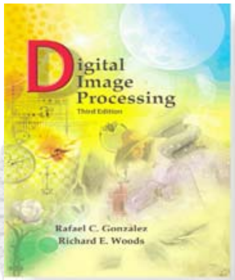


FIGURE 10.18 Subdivision of the parameter plane for use in the Hough transform.



Hough Transform

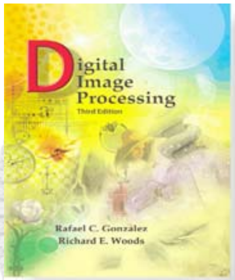
1. Quantize parameter space between appropriate maxima and minima for y-intercept b and slope a
2. Form an accumulator array A[b,a]:=0
3. For each **point** (x,y) in an edge-enhanced image such that E(x,y)>T, increment all points in A[b,a] along the appropriate **line** in a-b space, i.e., A[b,a]:=A[b,a]+1 for b=-ax+y
4. **Local maxima** in A[b,a] space correspond to collinear points (i.e., lines) in the image array. Values in A[b,a] correspond to how many points exist on that line.

Hough Transform

- Problem: a-b space is unbounded as for near-vertical lines a->±∞
- Solution: convert to polar coordinates before transforming

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

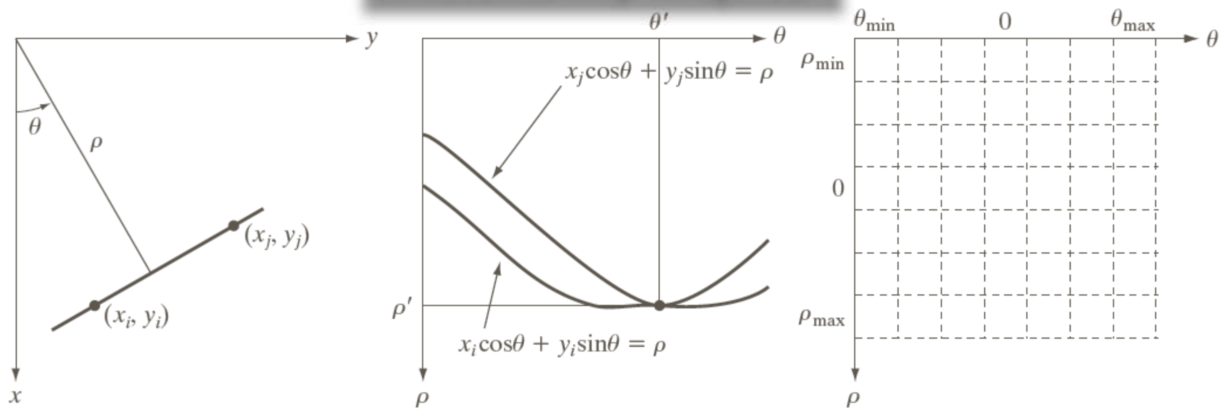
This will give sinusoids instead of straight lines



Hough Transform

The possible lines are sinusoids in ρ-θ space

Points (x_i,y_i) and (x_j,y_j) define a line in polar coordinates



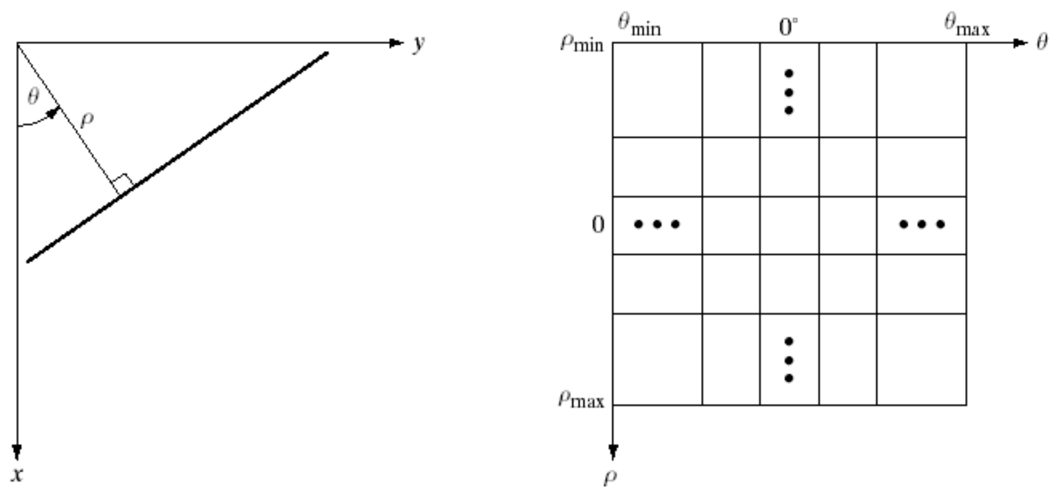
Map possible sinusoids into accumulator cells in discrete ρ-θ space

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.

© 2002 R. C. Gonzalez & R. E. Woods

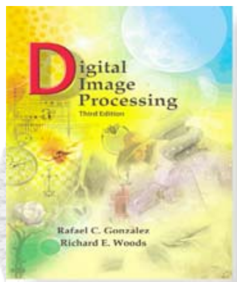
Chapter 10
Image Segmentation



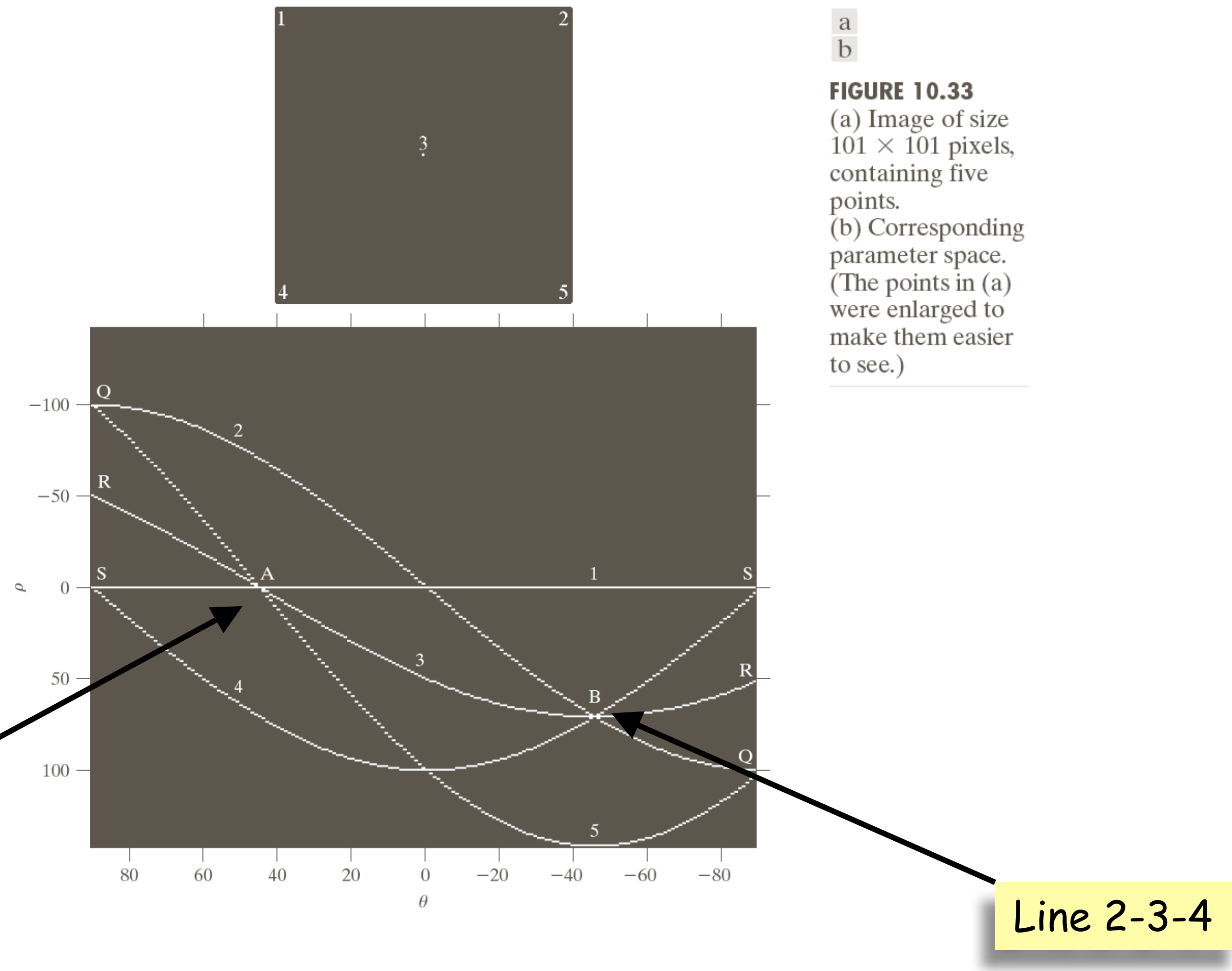
a b

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

Map possible sinusoids into accumulator cells in discrete ρ-θ space



Hough Transform



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.)

© 2002 R. C. Gonzalez & R. E. Woods

Hough Transform

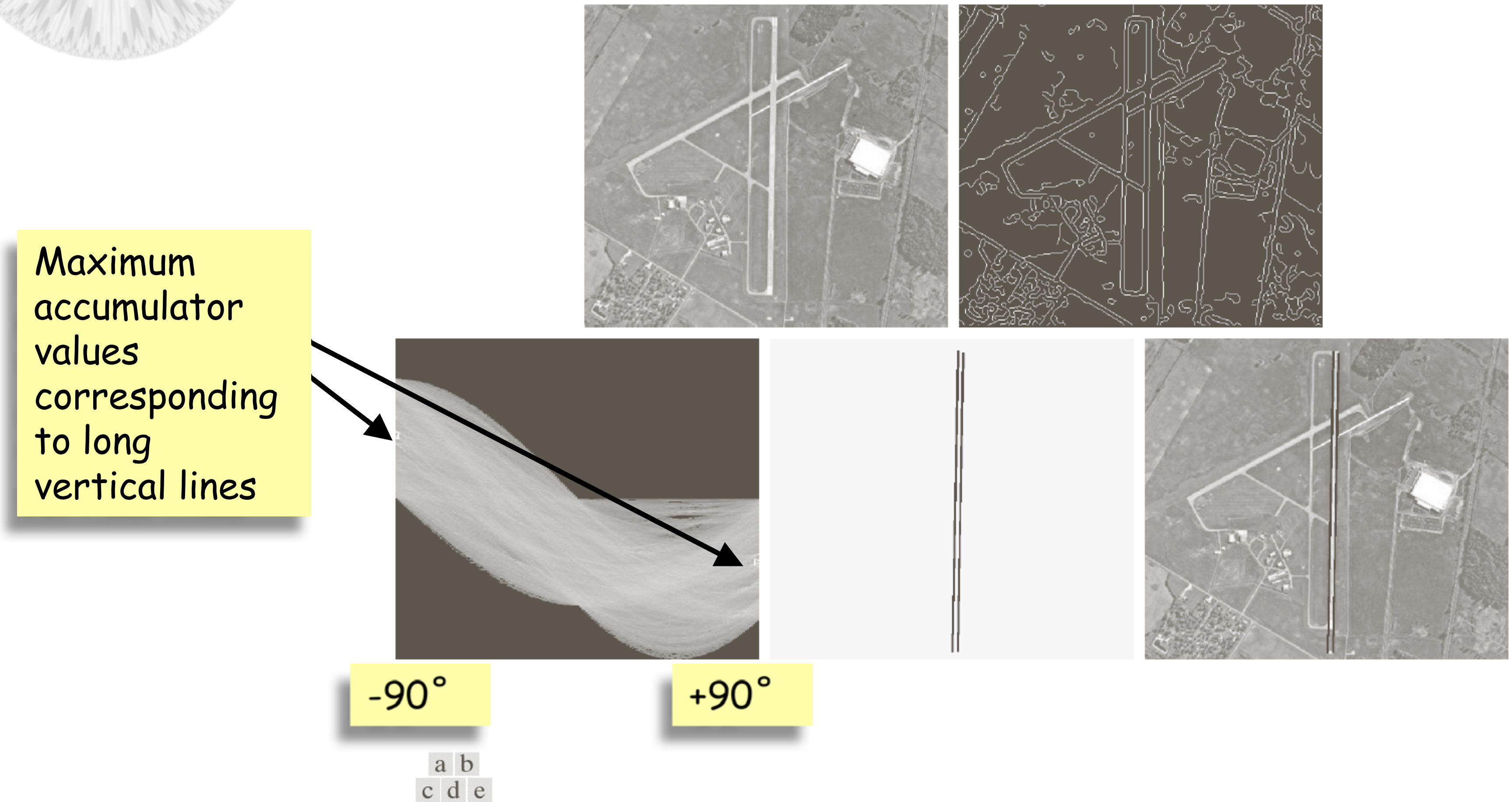
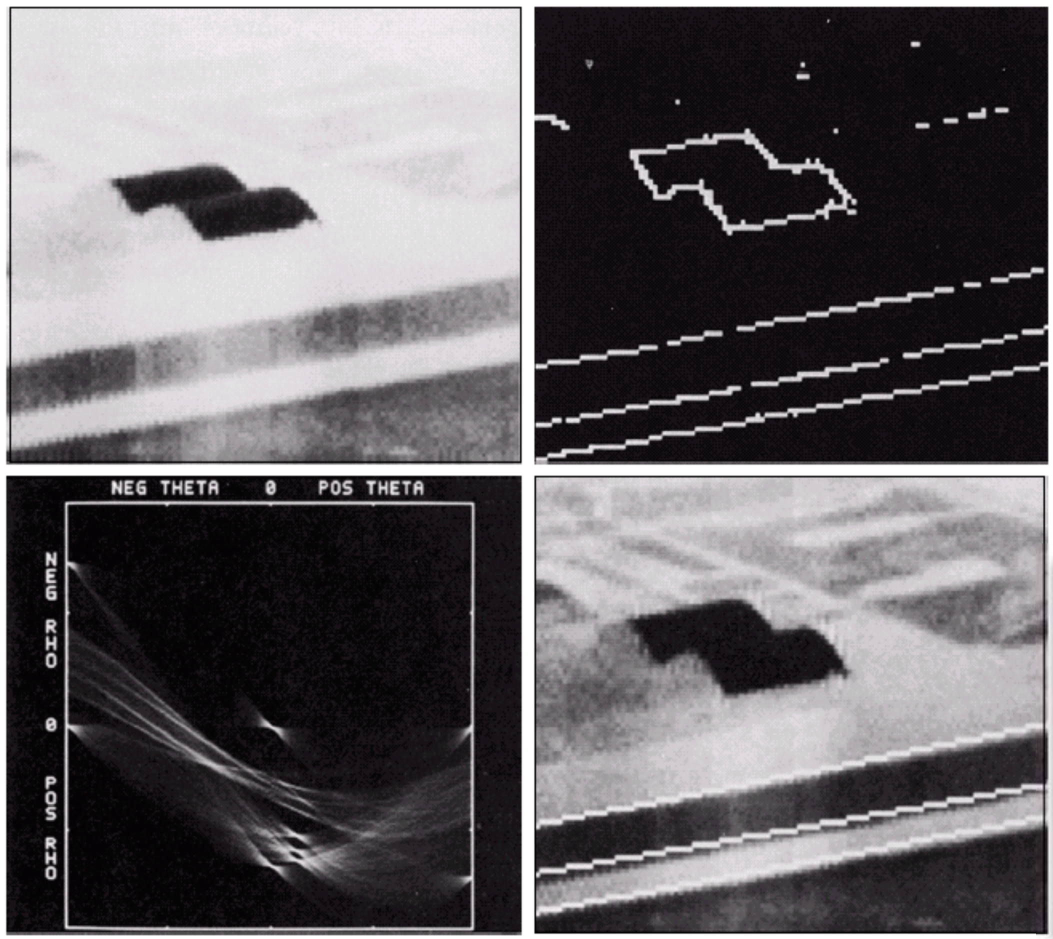


FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed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

© 2002 R. C. Gonzalez & R. E. Woods

Hough Transform

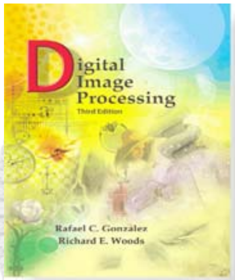


a b
c d

FIGURE 10.21 (a) Infrared image. (b) Thresholded gradient image. (c) Hough transform. (d) Linked pixels. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

Linked pixels from strongest points in accumulator. No gaps in linked image.

© 2002 R. C. Gonzalez & R. E. Woods



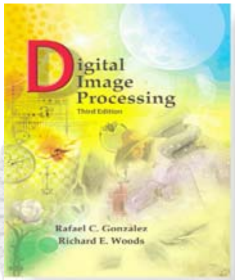
MATLAB/Image Processing Toolbox

MATLAB's Hough Transform routine

```
Function [h, theta, rho] = hough (f, dtheta, dpho)
% HOUGH Hough transform
%   [H, THETA, RHO] =HOUGH(F, DTHETA, DRHO) computes the Hough transform of the image F.
%   DTHETA specifies the spacing [in degrees] of the Hough transform bins along the theta axis.
%   DRHO specifies the spacing of the Hough transform bins along the rho axis.
%   H is the Hough transform matrix.  It is NRHO-by-NTHETA where NRHO=2*ceil(norm(size(F))/DRHO-1 and
%   NTHETA=2*ceil(90/DTHETA).  If 90/DTHETA is not an integer, the actual angle will be 90/ceil(90/DTHETA).

%   THETA is an NTHETA-element vector containing the angle (in degrees) corresponding to each column of H
%   RHO is an NRHO-element vector containing the value of rho corresponding to each row of H.

%   [H, THETA, RHO] =HOUGH(F) computes the Hough transform using DTHETA=1 and DRHO=1
```

```
>>f=zeros(101,101);
>>f(1,1)=1; f(101,1)=1; f(1,101)=1; f(101,101)=1; f(51,51)=1;
>>H=hough(f);
>>imshow(H,[ ])
```

SEE GWE, Section 10.2 Line Detection Using the Hough Transform



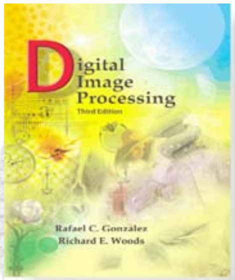
MATLAB/Image Processing Toolbox

MATLAB's Hough Peaks routine

```
PEAKS = HOUGHPEAKS(H,NUMPEAKS) locates peaks in the Hough
   transform matrix, H, generated by the HOUGH function.
NUMPEAKS
   specifies the maximum number of peaks to identify.
PEAKS is
   a Q-by-2 matrix, where Q can range from 0 to NUMPEAKS. Q holds
   the row and column coordinates of the peaks. If NUMPEAKS is
   omitted, it defaults to 1.

PEAKS = HOUGHPEAKS(...,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:

   'Threshold' Nonnegative scalar.
            Values of H below 'Threshold' will not be considered
            to be peaks. Threshold can vary from 0 to Inf.

            Default: 0.5*max(H(:))

   'NHoodSize' Two-element vector of positive odd integers: [M N].
            'NHoodSize' specifies the size of the suppression
            neighborhood. This is the neighborhood around each
            peak that is set to zero after the peak is identified.

            Default: smallest odd values greater than or equal to
                   size(H)/50.
```

SEE GWE, Section 10.2 Line Detection Using the Hough Transform

Because peaks are not necessarily confined to one cell you want to suppress (set to 0) the Hough transform cells in the immediate neighborhood.

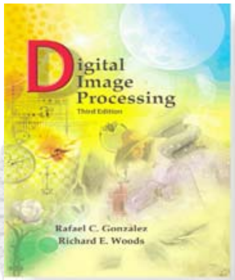


MATLAB/Image Processing Toolbox

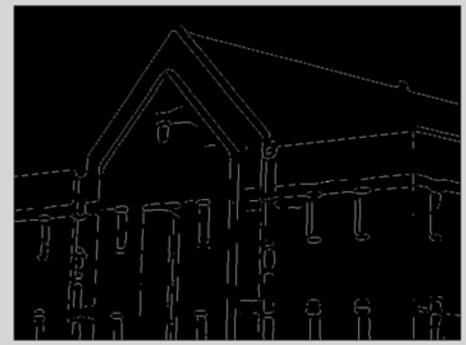
MATLAB’s Hough lines routine

```
Function lines = houghlines (f,theta, rho, rr, cc, fillgap, minlength)
% HOUGHLINES Extract line segments based on the Hough transform
%   LINES =HOUGHLINES(F, THETA, RHO, RR, CC, FILLGAP, MINLENGTH)
%   Extracts line segments in the image f associated with particular bins in a Hough transform.
%   THETA and RHO are vectors returned by the function HOUGH.
%   Vectors RR and CC specify the rows and columns of the Hough transform bins to use
%   in searching for line segments.  If HOUGHLINES finds two lines segments that are associated
%   with the same Hough transform bin that are separated by less than FILLGAP pixels,
%   HOUGHLINES merges them into a single line segment. FILLGAP defaults to 20 if omitted.
%   Merged line segments less than MINLENGTH pixels long are discarded.
%   MINLENGTH defaults to 40 if omitted.

>> f=imread(‘fig10.10(a).jpg’);        %load in building figure
>>[H, theta, rho]=hough(f,0.5);       % Hough transform with finer angular spacing

>>imshow(theta, rho, H, [ ], ‘nottruesize’), axis on, axis normal
>>xlabel(‘\theta’), ylabel(‘\rho’)

>>P=houghpeaks[H,5];                %find five Hough transform peaks
>>hold on
>>plot(theta(P(:,2),rho(P(:,1)), 's','color','red')

>>lines=houghlines(f,theta,rho,r,c)
>>figure, imshow(f), hold on
>>for k=1:length(lines)
Xy=[lines(k), point1; lines(k),point2];
Plot(xy:2),xy(:1), “LineWidth’,4,’Color’,[.6 .6 .6]);
end
```

SEE GWE, Section 10.2 Line Detection Using the Hough Transform



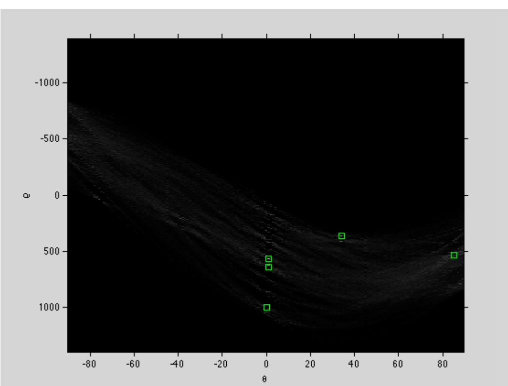
MATLAB/Image Processing Toolbo

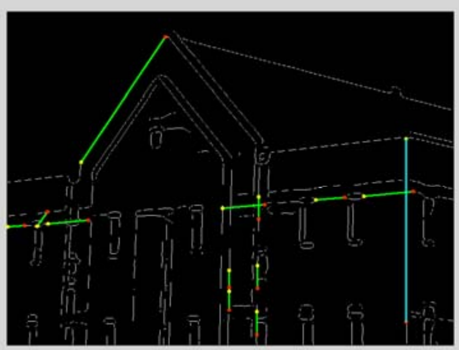


```
f=imread('Fig1016(a).tif');

imshow(f)
figure
BW=edge(f,'canny',[0.04 0.1],4);
imshow(BW)

[H,theta,rho] = hough(BW);
figure
imshow(H,[],'XData',theta,'YData',rho,'InitialMagnification','fit');
xlabel('¥theta'), ylabel('¥rho');
axis on, axis normal, hold on;
P  = houghpeaks(H,5);
x = theta(P(:,2)); y = rho(P(:,1));
plot(x,y,'s','color','green');

% Find lines and plot them
lines = houghlines(BW,theta,rho,P,'FillGap',30,'MinLength',40);
```

Using the same parameters as listed in GW [Example 10.8] produced a much poorer edge image than that shown in Figure 10.25

SEE GWE, Section 10.2 Line Detection Using the Hough Transform





Hough Transform

The Hough Transform can be applied to any curve of the form $f(\underline{x},\underline{a})=0$

where $\underline{x}$ is the position vector; $\underline{a}$ is the parameter vector

For example, $(x-a)^2+(y-b)^2=r^2$ is a three-parameter space (a,b,r)

This approach is impractical for too many parameters. Effectively, it is a matched filtering process.

Consider looking for a circle of 1's (edge only)

A(a,b,r) is the correlation with that circle template where (a,b) is the possible center and r are the possible radii

SPECIAL TECHNIQUES are needed to find the end points of a line segment

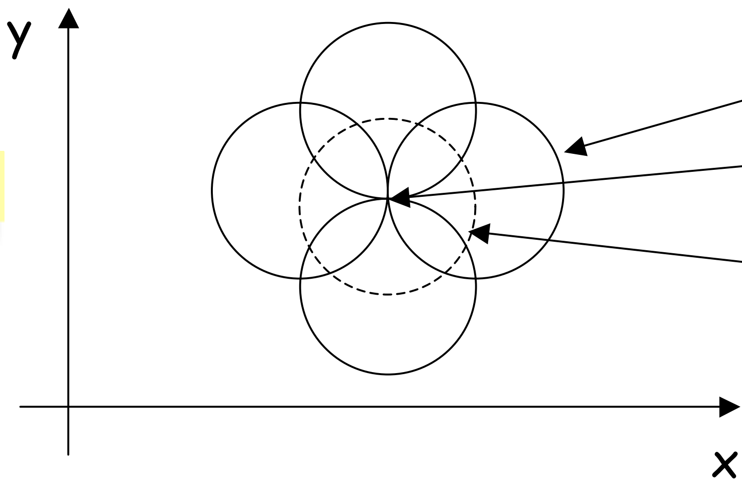
Hough Transform

The Hough Transform is typically slow and uses a lot of memory so we use gradient direction to reduce computations

Hough Transform

Example: find a circle of fixed radius r

Image space

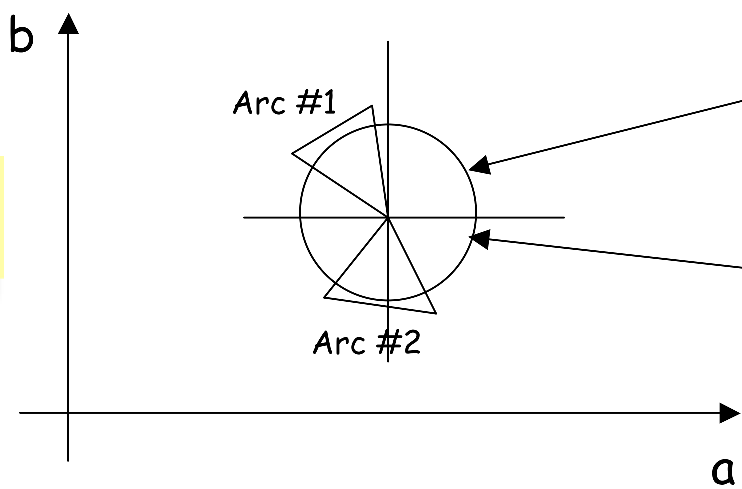


All possible circles of radius r going through the edge point

The edge point

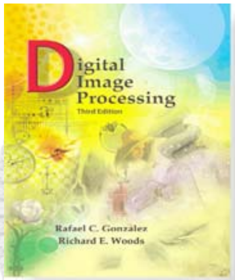
Locus of all possible centers (a,b). These would be the (a,b) incremented in Hough space

Parameter space



Locus of possible a,b's in Hough space

Would only need to accumulate some a,b's if we knew additional information such as the local gradient in image space. This would localize the regions to accumulate in Hough space.



Hough Transform

Original chest X-ray

Hough accumulator array

Identified circular tumors

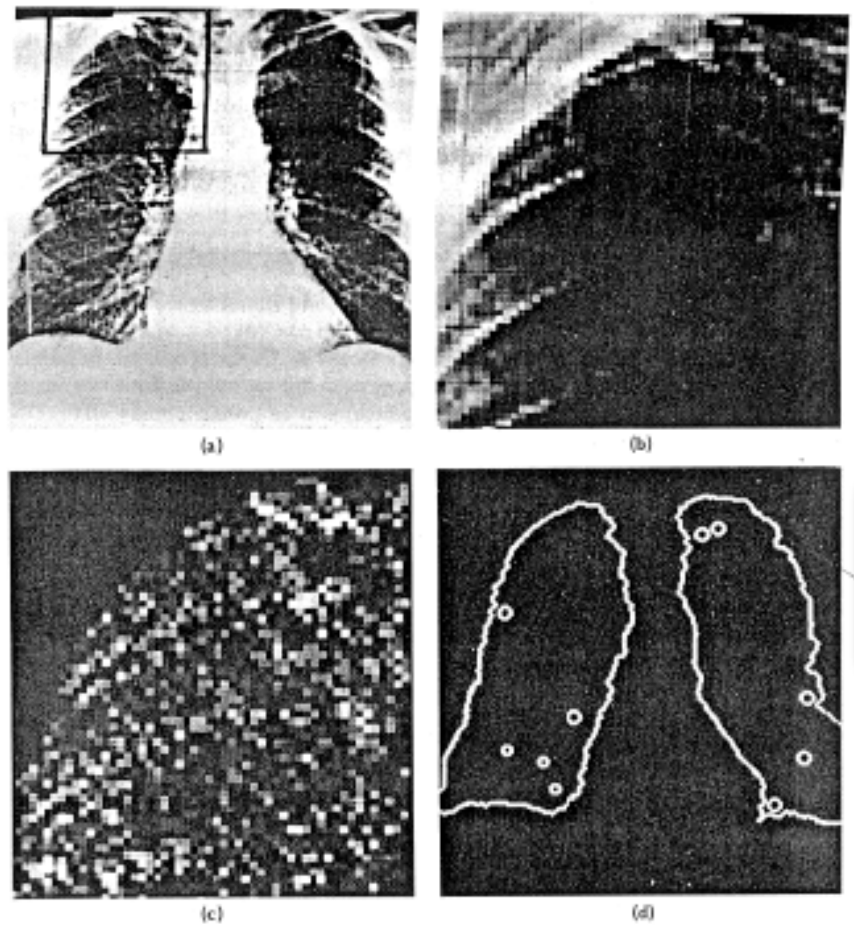
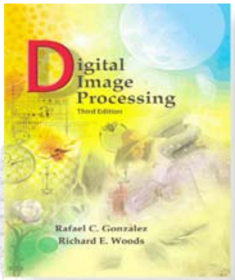


Fig. 4.7 Using the Hough technique for circular shapes. (a) Radiograph. (b) Window. (c) Accumulator array for $r = 3$. (d) Results of maxima detection.

© Ballard & Brown



Hough Transform

Consider equation of circle $(x-a)^2-(y-b)^2=R^2$

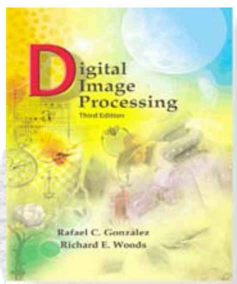
Differentiating $2(x-a)dx-2(y-b)dy=0$

$$\frac{dy}{dx}=\frac{x-a}{y-b}$$

Define the gradient of the edge in image space as

$$\frac{dy}{dx}=\tan\phi$$

Now solve for a and b in terms of x,y,R <u>and</u> ϕ



Hough Transform

Substituting
$$\left[\frac{dy}{dx}(y-b)\right]^2-(y-b)^2=R^2$$

$$(y-b)^2\left[\left(\frac{dy}{dx}\right)^2+1\right]^2=R^2$$

$$(y-b)^2\left[\tan^2\phi+1\right]^2=R^2=(y-b)^2\frac{1}{\cos^2\phi}$$

$$(y-b)^2=R^2\cos^2\phi$$

$$y-b=\pm R\cos\phi$$

$$b=y\mp R\cos\phi$$

Hough Transform

- Do the same for b

$$(x-a)^2 - \left[\frac{x-a}{\dfrac{dy}{dx}}\right]^2 = R^2$$

$$(x-a)^2\left[1+\frac{1}{\dfrac{dy}{dx}}\right]^2 = R^2$$

$$(x-a)^2\left[1+\cot^2\phi\right]^2 = R^2 = (x-a)^2\frac{1}{\sin^2\phi}$$

$$(x-a)^2 = R^2\sin^2\phi$$

$$x-a = \pm R\sin\phi$$

$$a = x \mp R\sin\phi$$

Hough Transform

- Use the results where R is known in this example and ϕ comes from known gradient. (x,y) is location of edge candidate. Then,

$$a = x \mp R\sin\phi \qquad b = y \mp R\cos\phi$$

Remember $\phi=\text{Tan}^{-1}(dy/dx)$

- This gives a single point in (a,b) space. Do an arc through (a,b) to cover errors in ϕ due to quantization, noise, etc.

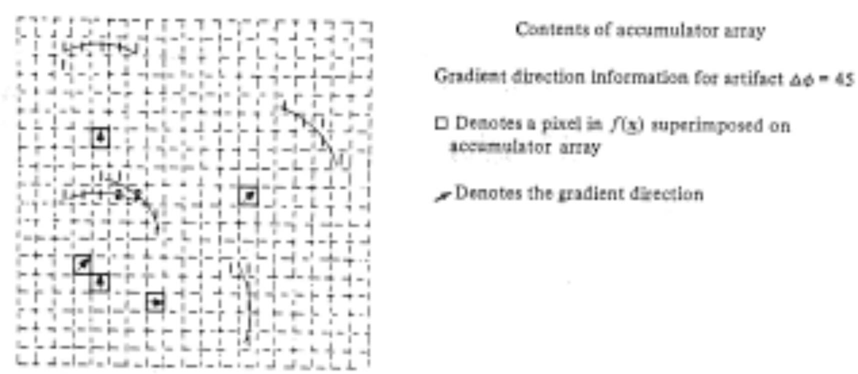
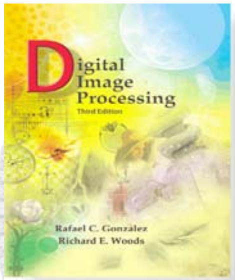


Fig 4.6 Reduction in computation with gradient information



Generalized Hough Transform

1. Construct an R-table (next slide) for the shape to be localized
2. Initialize accumulator array $A(x_{cmin}, x_{cmax}, y_{cmin}: y_{cmax})$ for all possible reference points.
3. For <u>each</u> edge point
   a) Compute $\phi(\underline{x})$, direction of gradient
   b) Calculate all possible centers
   $x_c := x + r(\phi)\cos(\alpha(\phi))$; $y_c := y + r(\phi)\sin(\alpha(\phi))$
   a) Increment the accumulator array
   $A(x_c, y_c) := A(x_c, y_c) + 1$
4. Possible locations for shape are local maxima of A

(Can handle scale S and rotation θ by increasing dimensionality of accumulator)

Generalized Hough Transform

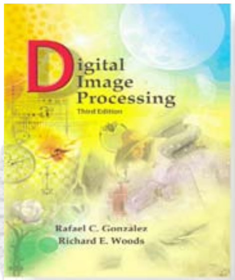
Basic premise of Hough

Edge point data in image space corresponds to loci of possible points in parameter space

R-table

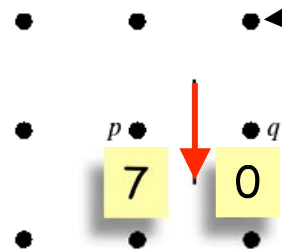
Gradient angle	Location of reference points from boundary points
ϕ_1	$\underline{r}_1', \underline{r}_2', \underline{r}_3', \ldots, \underline{r}_n'$ or $r=(r,\alpha)$
ϕ_2	
ϕ_m	

Basically the possible center is specified by r and α for each identified edge point (and gradient direction)



Graph Search

FIGURE 10.22 Edge element between pixels *p* and *q*.



These correspond to the center of the pixels

Goal nodes

Each edge element has a cost for the edge

c(p,q)=H-[f(p)-f(q)]

H is the highest gray scale value in the edge image, f(p) and f(q) are gray scale edge values

Cost is low (high edge gradient) traveling along this edge from top to bottom

NOTE: the cost function has direction

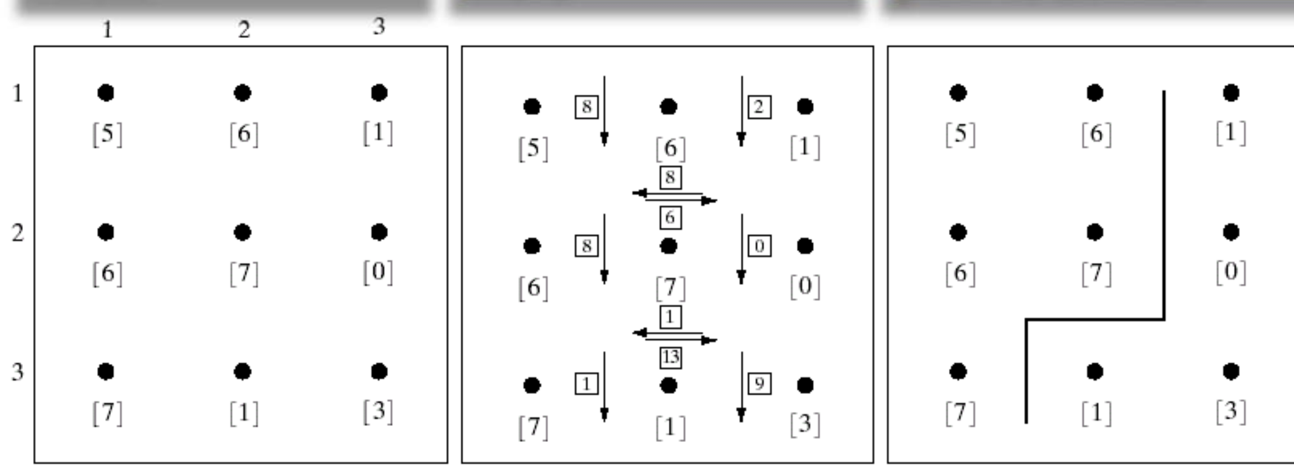
Let f(p)=7, f(q)=0, H=7
c(p,q)=H-[f(p)-f(q)]=7-[7-0]=0

Graph Search

Edge image with computed gradient values

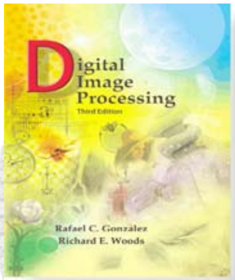
Edge segments and all computed costs

Pick lowest starting cost and traverse lowest cost path to bottom



a b c

FIGURE 10.23 (a) A 3 × 3 image region. (b) Edge segments and their costs. (c) Edge corresponding to the lowest-cost path in the graph shown in Fig. 10.24.



Graph Search

Graph of all nodes from previous figure shows all possible paths. Nodes correspond to edge pixels in figure. Arcs are potential linked edges.

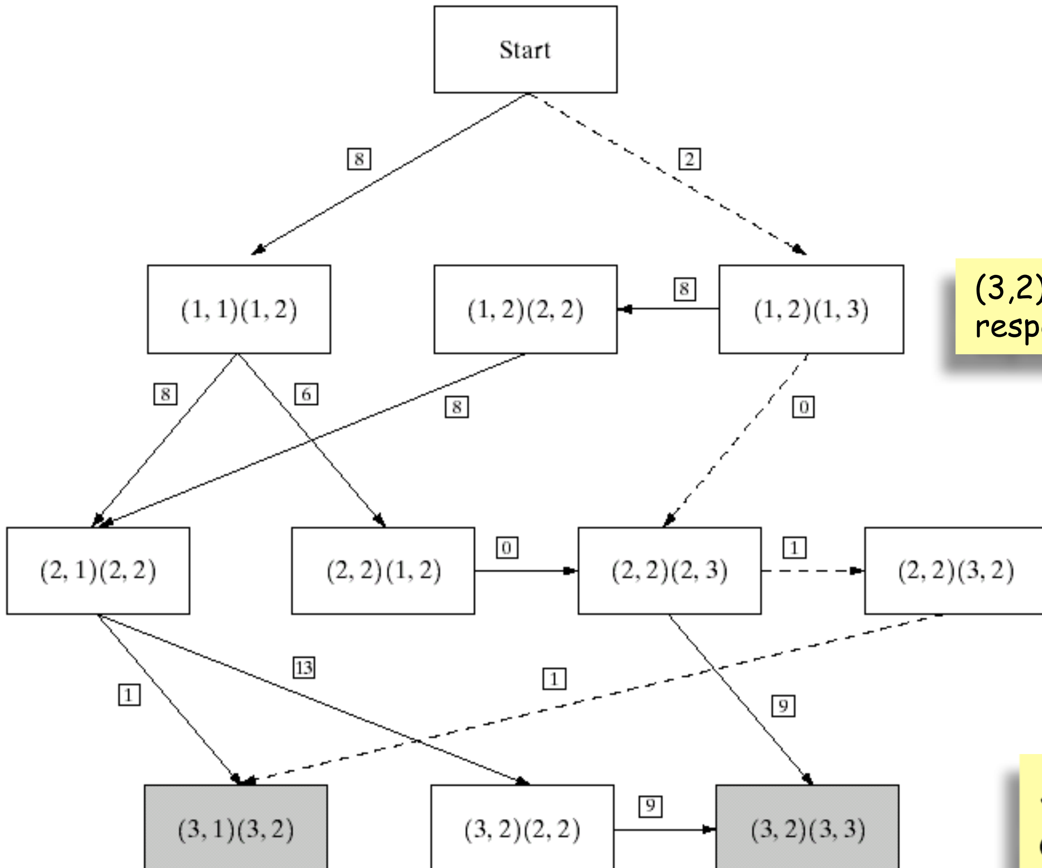


FIGURE 10.24 Graph for the image in Fig. 10.23(a). The lowest-cost path is shown dashed.

(3,2)(3,3) specify p and q respectively for each edge

Shaded pixels are end nodes

Graph Search

Improved algorithm will estimate the cost to the ends as well.

FIGURE 10.25 Image of noisy chromosome silhouette and edge boundary (in white) determined by graph search.

$$r(n) = g(n) + h(n)$$

r(n) is the estimate of the minimum cost path
g(n) is the cost of the lowest cost path to n
h(n) estimated cost from n to goal node using some heuristic

Heuristic used here to to pick lowest cost edge 5 levels down.

Image Thresholding

Thresholding is a very important technique for segmenting images. It can be applied to intensity, color components, etc.

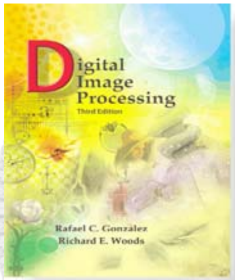
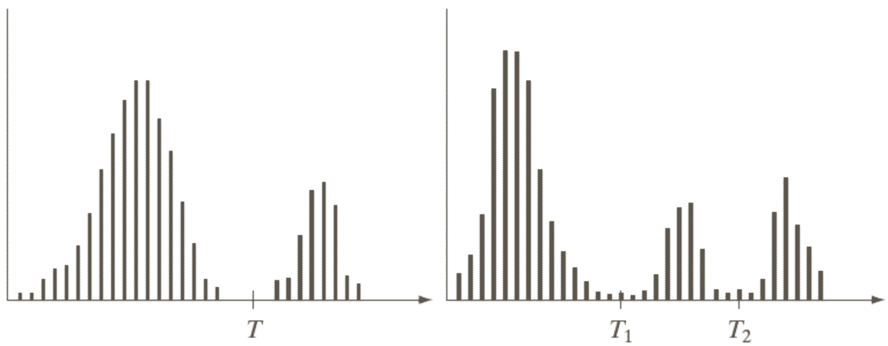


Image Thresholding



a b

FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

Image segmentation can be done by classifying pixels which are connected and have similar characteristics, e.g., gray scale distribution.

© 2002 R. C. Gonzalez & R. E. Woods

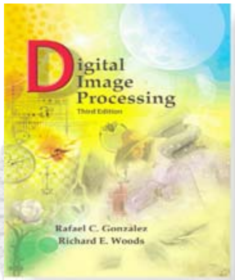
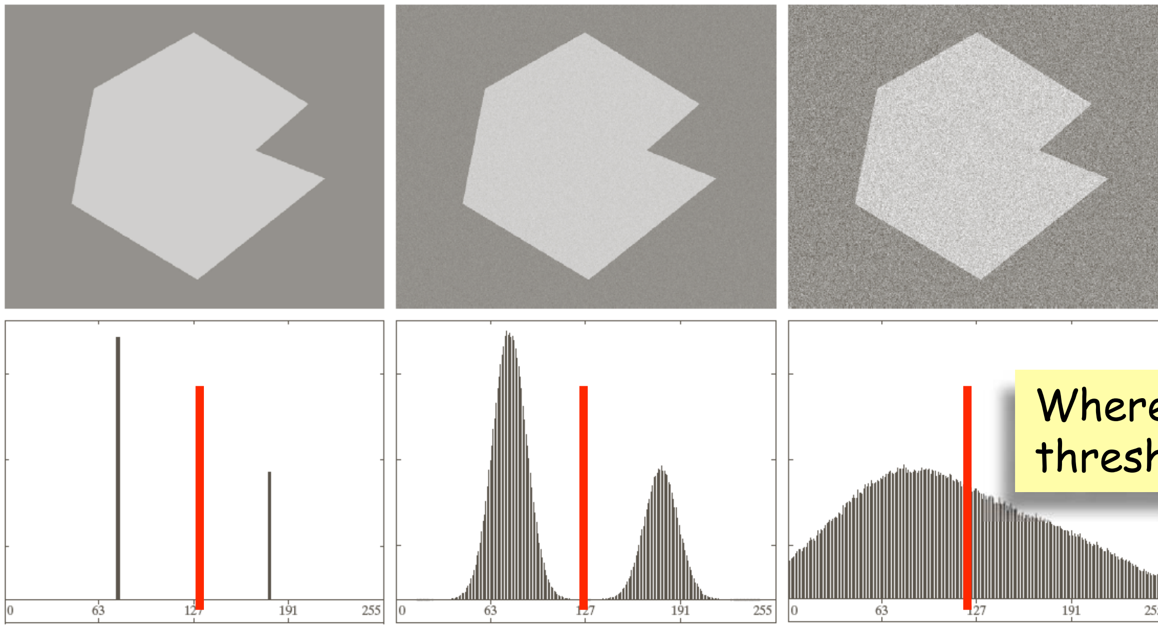


Image Thresholding



Where should the threshold be placed?

Basic two gray level image which can be readily separated by a single threshold

Same image with additive Gaussian noise mean=0, σ=10

Same image with additive Gaussian noise mean=0, σ=50

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.

© 2002 R. C. Gonzalez & R. E. Woods

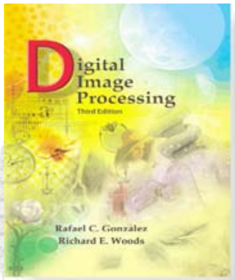
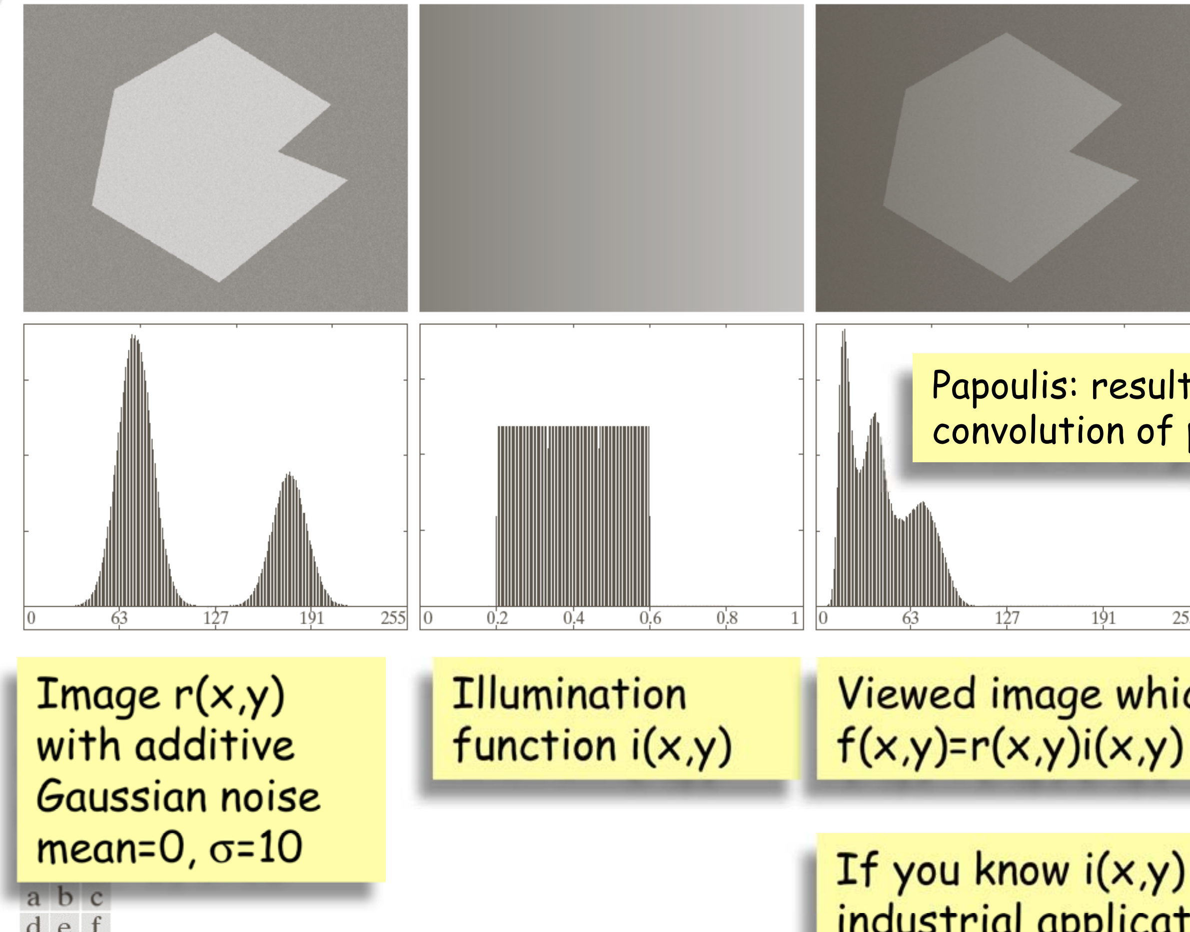


Image Thresholding



Papoulis: resultant pdf will be convolution of pdf's for iid rv's

Image r(x,y) with additive Gaussian noise mean=0, σ=10

Illumination function i(x,y)

Viewed image which is product f(x,y)=r(x,y)i(x,y)

If you know i(x,y) as in some industrial applications you can recover r(x,y)=f(x,y)/i(x,y)

FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2
(d)–(f) Corresponding histograms.

© 2002 R. C. Gonzalez & R. E. Woods

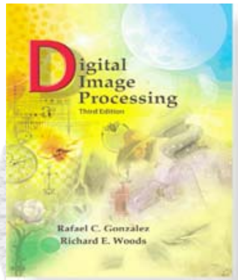
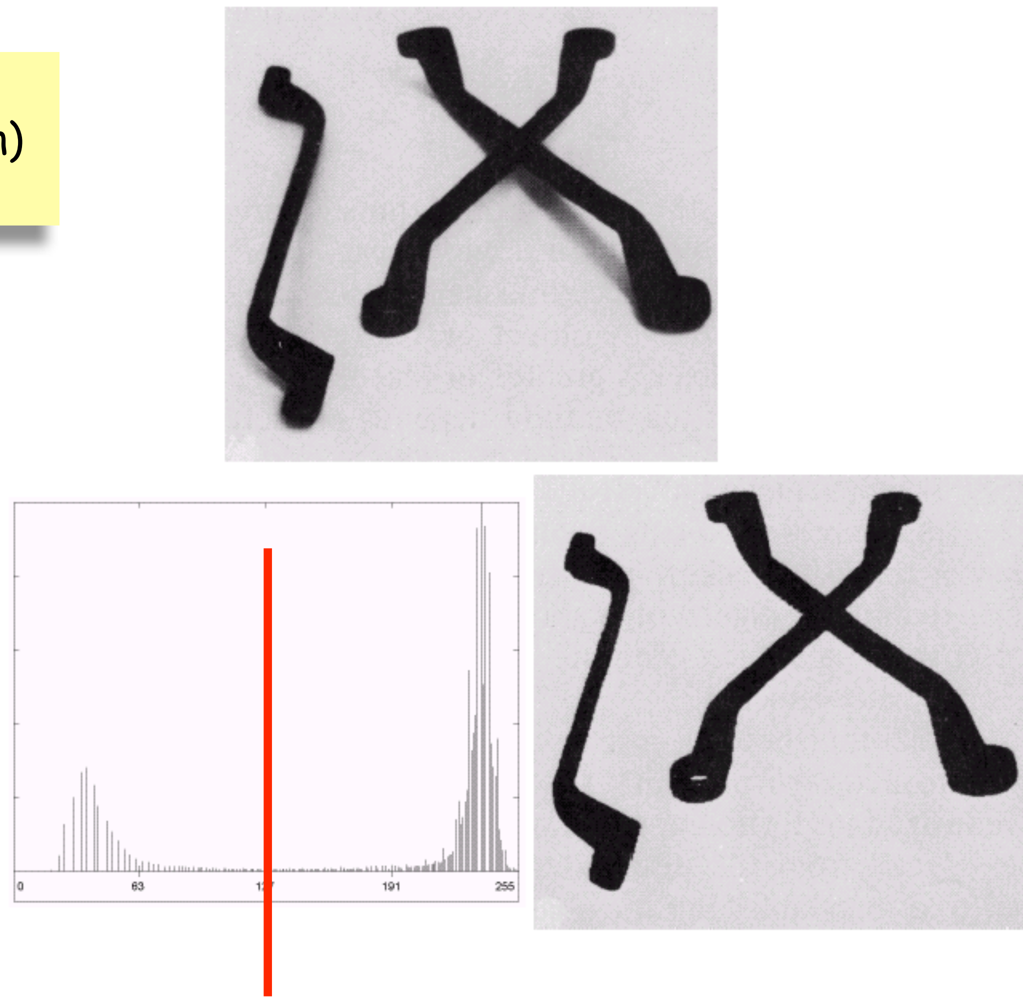


Image Thresholding

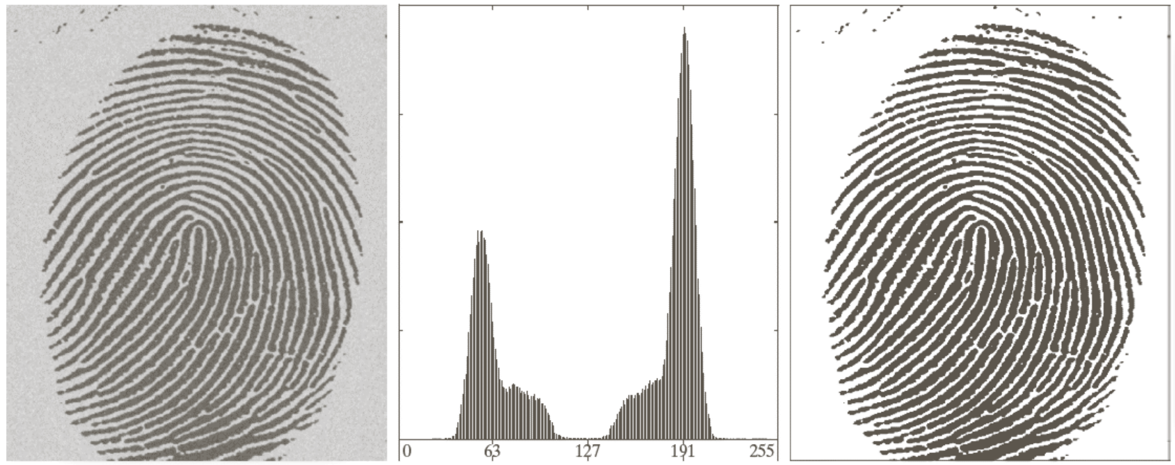
Simple example of global (single common) threshold.



a
b c

FIGURE 10.28 (a) Original image. (b) Image histogram. (c) Result of global thresholding with *T* midway between the maximum and minimum gray levels.

Image Thresholding



Works well for histograms (pdf's) with clear valleys between the classes.

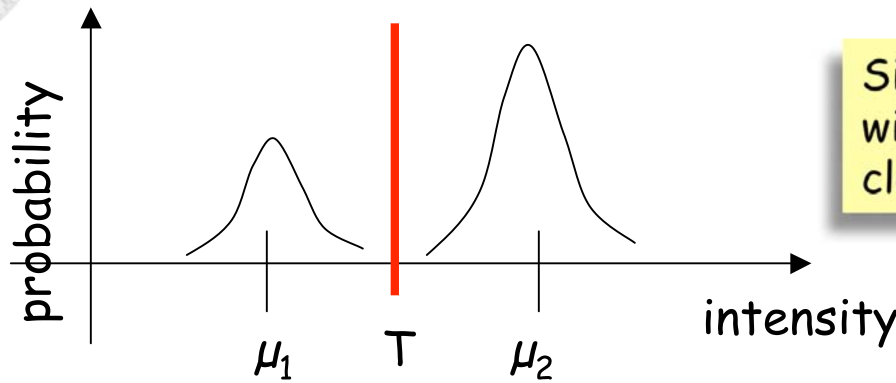
Automated threshold calculation:
1. Select an initial estimate for threshold T
2. Segment the image, i.e., threshold
3. Compute the average gray scale value μ_1 for pixels that are classified as 0's. Compute the average gray scale value μ_2 for pixels classified as 1's.
4. Compute new threshold $T=0.5*(\mu_1+\mu_2)$
5. Repeat until ΔT less than some threshold

a b c

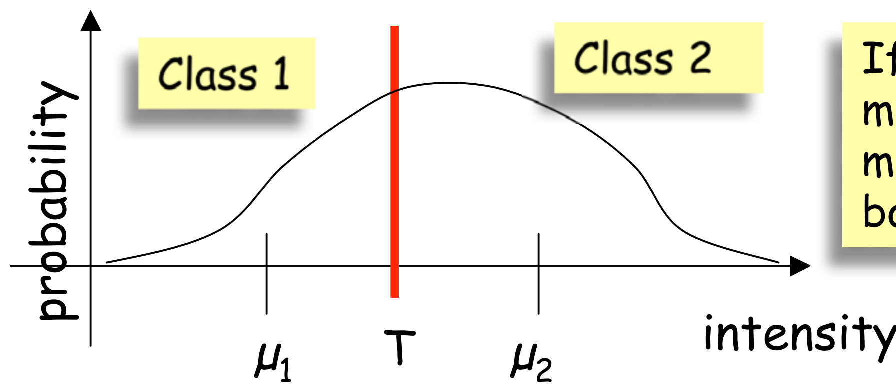
FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

© 2002 R. C. Gonzalez & R. E. Woods

Optimum Thresholding



Simple bimodal distribution with two homogeneous classes



If there is no valley one method of determining T is to minimize the variance within both classes.

© 2002 R. C. Gonzalez & R. E. Woods

1999-2007 by Richard Alan Peters II

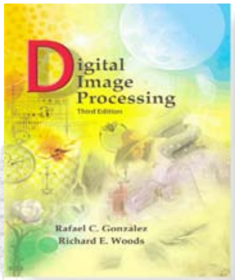


Image Thresholding

- Otsu method minimizes the overall within-class variance by minimizing the weighted sum of class variances

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Class 1

$$q_1(t) = \sum_{i=0}^{t} P(i) \qquad \sigma_1^2(t) = \frac{1}{q_1(t)} \sum_{i=0}^{t} \left[i - \mu_1(t)\right]^2 P(i)$$

Class 2

$$q_2(t) = \sum_{i=t}^{L-1} P(i) \qquad \sigma_2^2(t) = \frac{1}{q_2(t)} \sum_{i=t+1}^{L-1} \left[i - \mu_2(t)\right]^2 P(i)$$

CDF for each class

Standard deviation of the intensities within each class normalized by the probability of that class

© 2002 R. C. Gonzalez & R. E. Woods

1999-2007 by Richard Alan Peters II

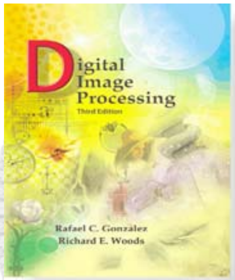


Image Thresholding

http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=16205

1. Compute the histogram of the image. Let each gray level have probability p_i.
2. Compute the cumulative sum $P_1(k)$ for k=0,…,L-1 $$P_1(k) = \sum_{i=0}^{k} p_i$$
3. Compute the cumulative mean m(k) for k=0,…,L-1 $$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^{k} i p_i$$
4. Compute the global intensity mean m_G $$m_G = \sum_{i=0}^{L-1} i p_i$$
5. Compute the between class variance $\sigma_B^2(k)$ for k=0,…,L $$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$
6. Compute the Otsu threshold k* as the value of k for which $\sigma_B^2(k)$ is maximum
7. Obtain the separability measure η^* $$\eta^* = \frac{\sigma_B^2(k^*)}{\sigma_G^2}$$

The farther apart the means the larger will be $\sigma_B^2(k)$

This is a measure of how easily separable the classes are. Uniform distribution is 0 and a clear, bimodal is 1

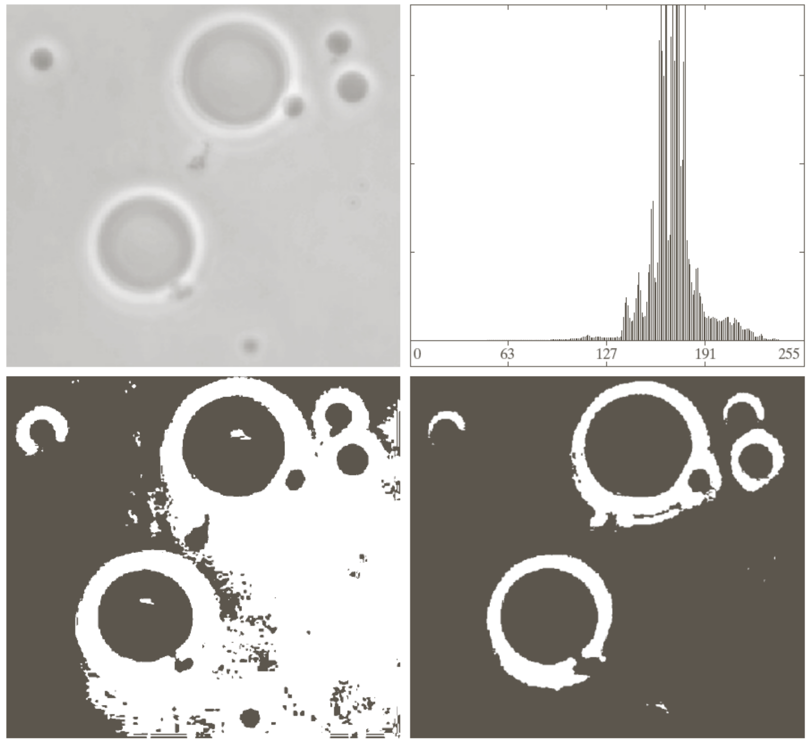
© 2002 R. C. Gonzalez & R. E. Woods

1999-2007 by Richard Alan Peters II

Image Thresholding

Original image.

Global thresholding calculating $T=0.5*(\mu_1+\mu_2)$ until ΔT less than some ε



a b
c d

FIGURE 10.39 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

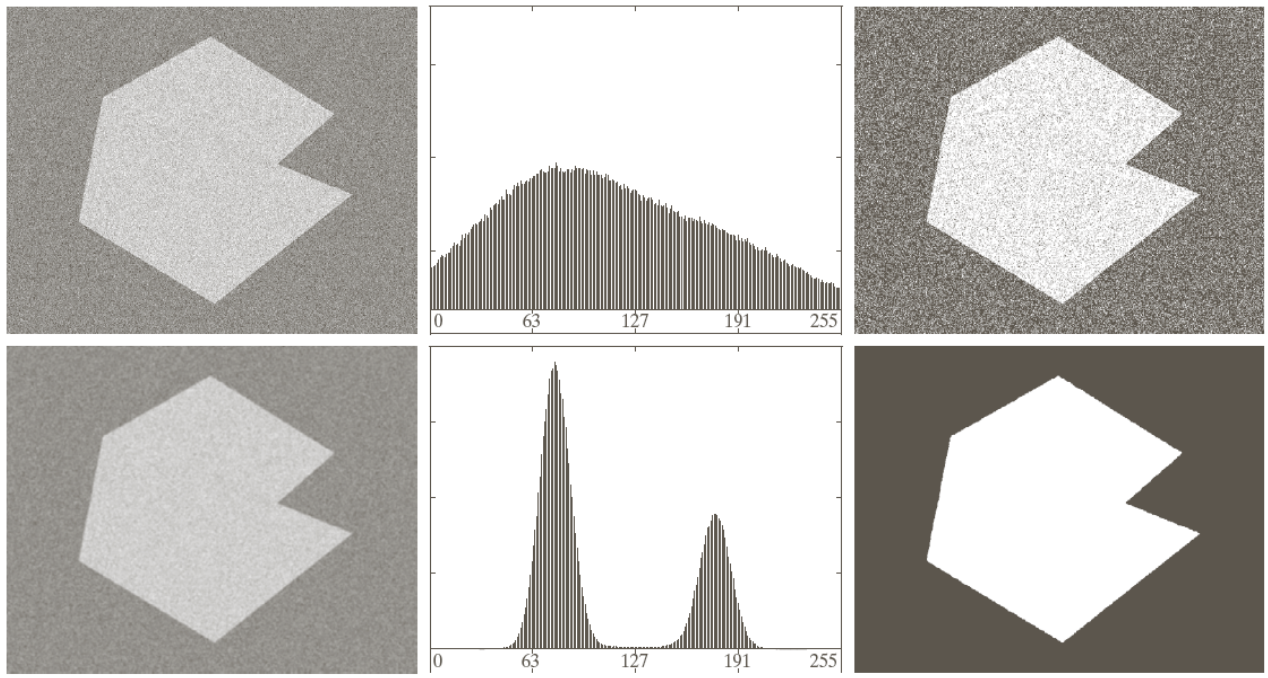
Global thresholding using Otsu algorithm

© 2002 R. C. Gonzalez & R. E. Woods

Image Thresholding

Noisy image

Smoothed noisy image



Averaging greatly improves separability

Global thresholding using Otsu algorithm

a b c
d e f

FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

© 2002 R. C. Gonzalez & R. E. Woods

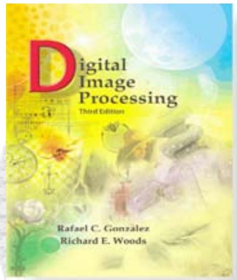
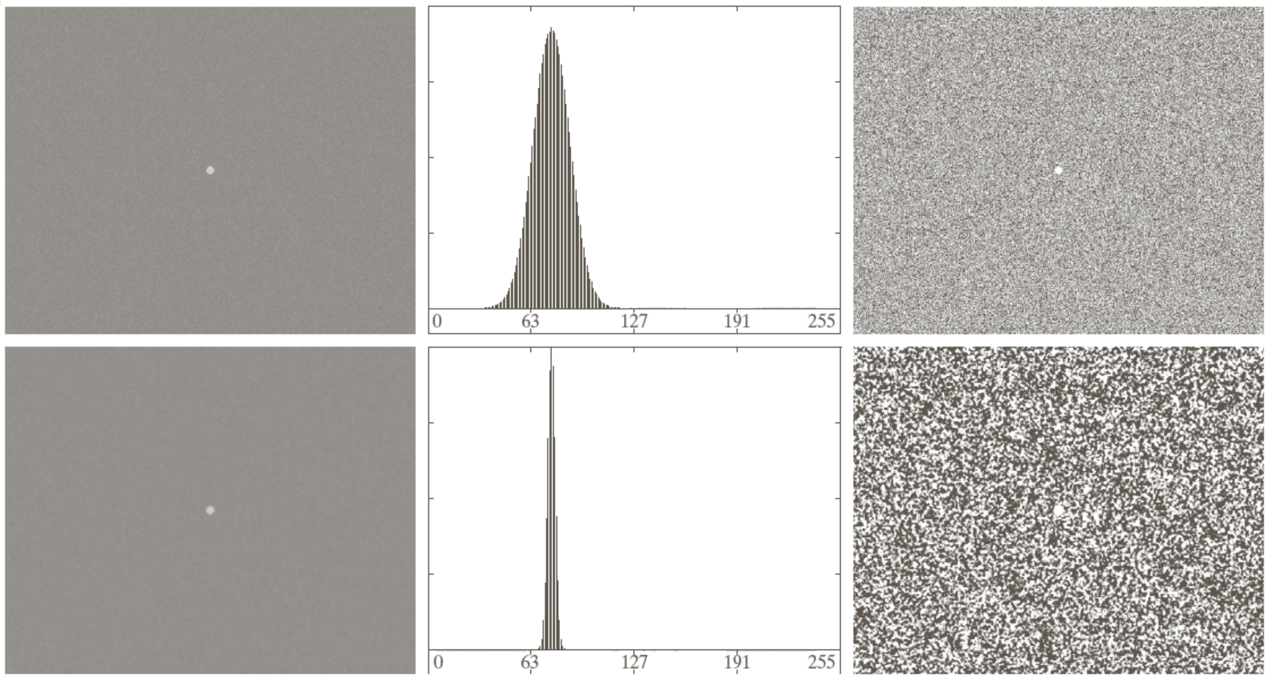


Image Thresholding

Noisy image

Smoothed noisy image



Averaging does not improve separability

Global thresholding using Otsu algorithm

a b c
d e f

FIGURE 10.41 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

© 2002 R. C. Gonzalez & R. E. Woods

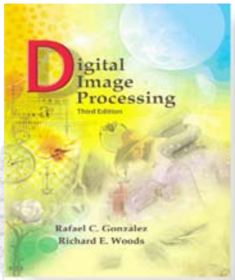


Image Thresholding

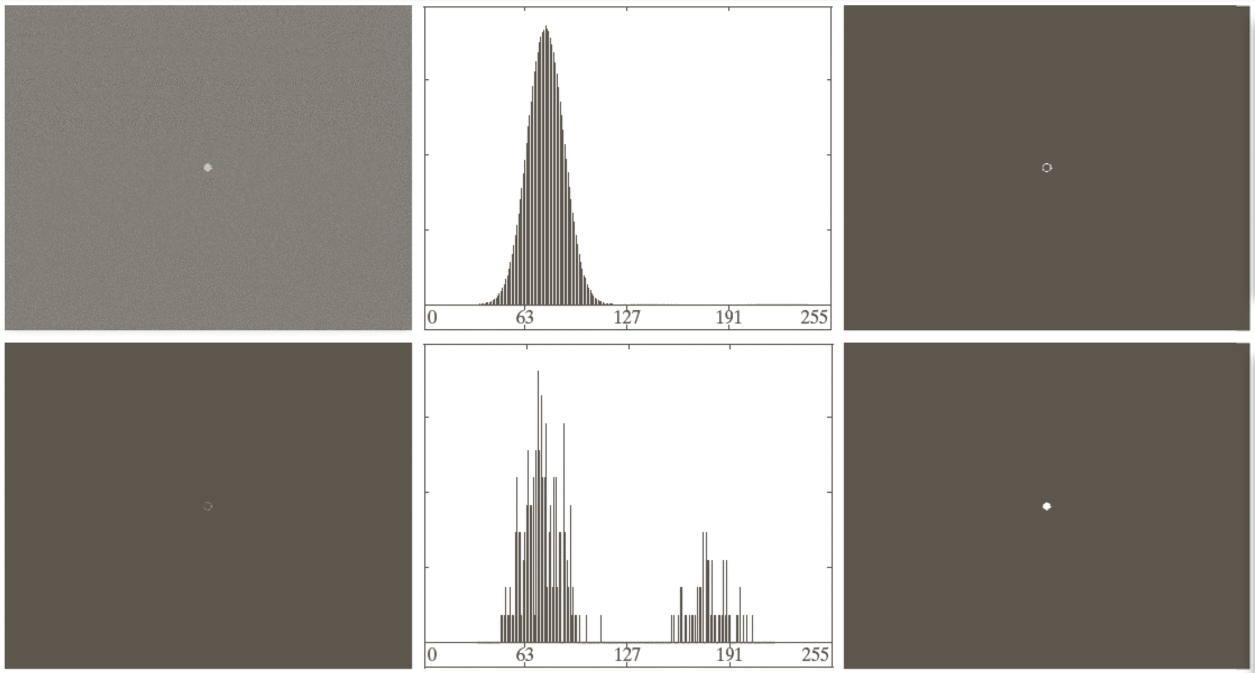
Basic idea is to look at the histogram of pixels only near edges

Noisy image

Thresholded gradient image (very high threshold)

Produce of thresholded gradient image and original image

Global thresholding of ORIGINAL image using Otsu algorithm on histogram of masked image



Edge masking uses only pixels near edges to form histogram

a b c
d e f

FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.

© 2002 R. C. Gonzalez & R. E. Woods

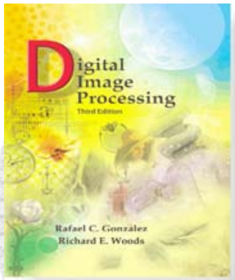


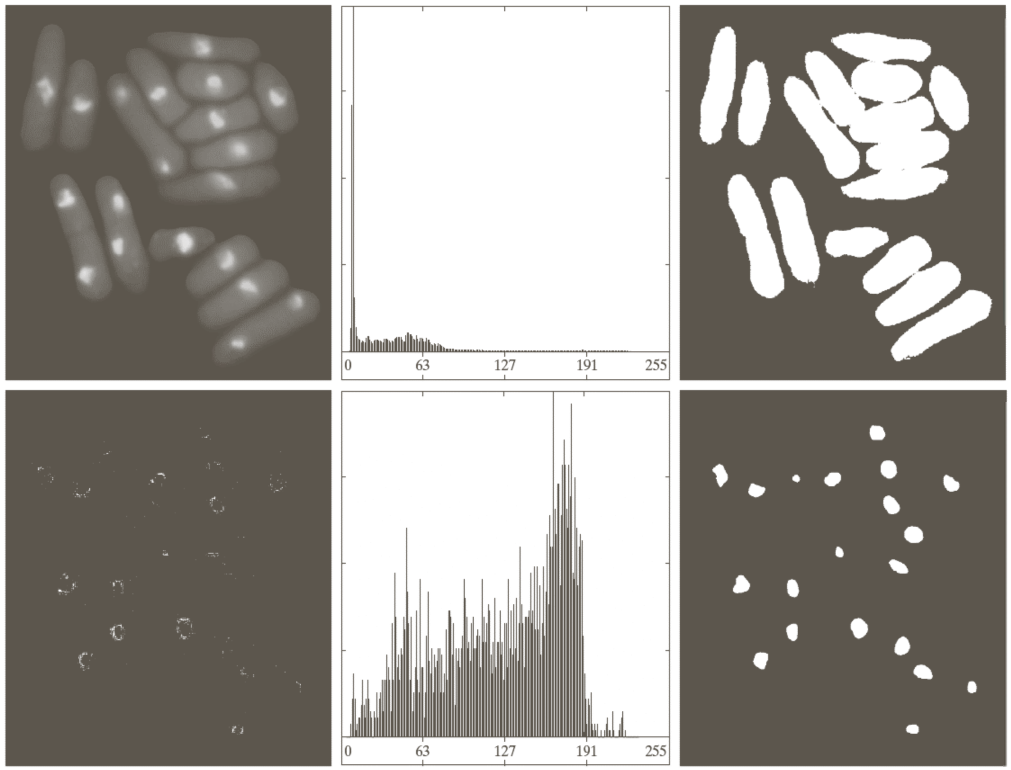
Image Thresholding

Noisy image of yeast cells — want to find bright spots

Thresholded absolute Laplacian image (high threshold)

Product of thresholded gradient image and original image

Global thresholding of ORIGINAL image using Otsu algorithm on histogram of masked image



Edge masking uses only pixels near edges to form histogram

a b c
d e f

FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

© 2002 R. C. Gonzalez & R. E. Woods