پردازش تصاویر دیجیتال
(بخش سوم)

تشخیص لبه

دانشگاه شهید بهشتی
پاییز ۱۴۰۲
احمد محمودی ازناوه

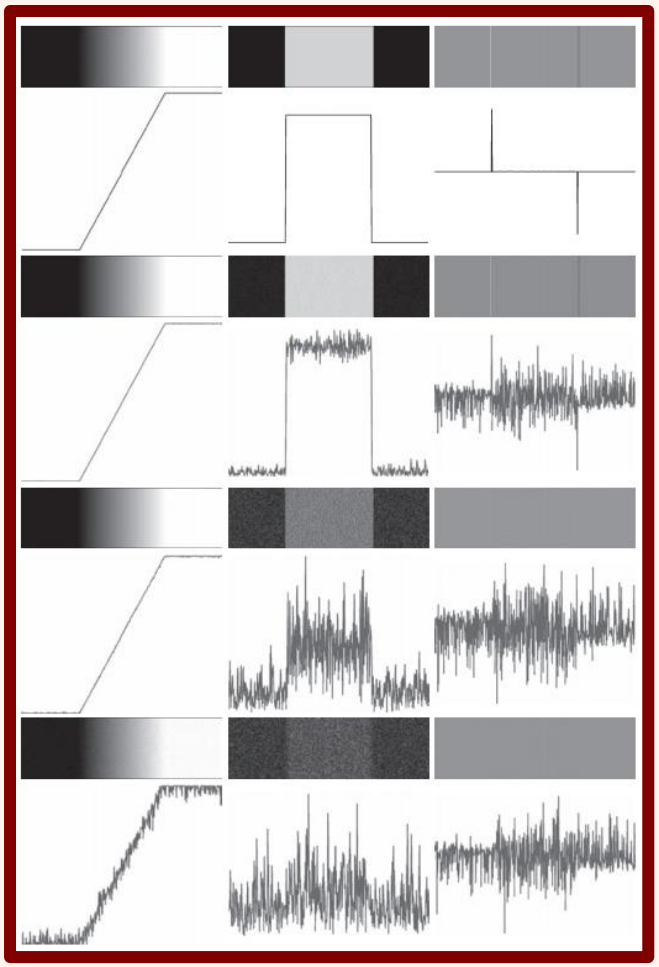
فهرست مطالب

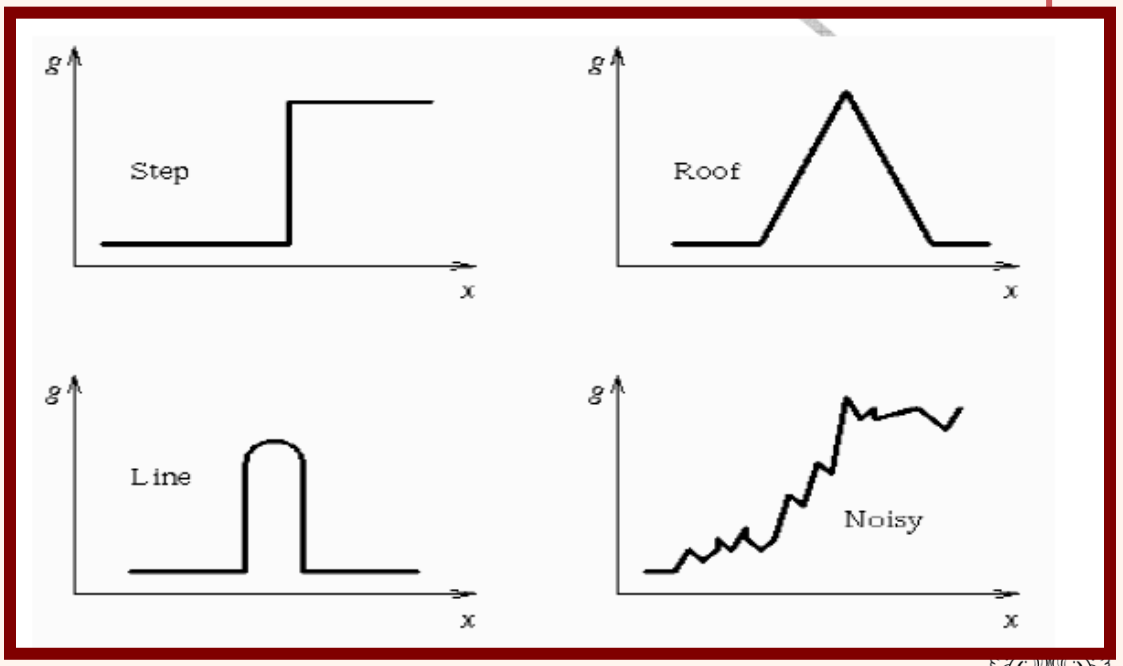
- الگوریتم‌های تشخیص لبه
- مشتق اول
- مشتق دوم
- تاثیر نویز
- Roberts، Prewitt و Sobel
- Canny
- فیلتر لاپلاسین و Marr-Hildreath

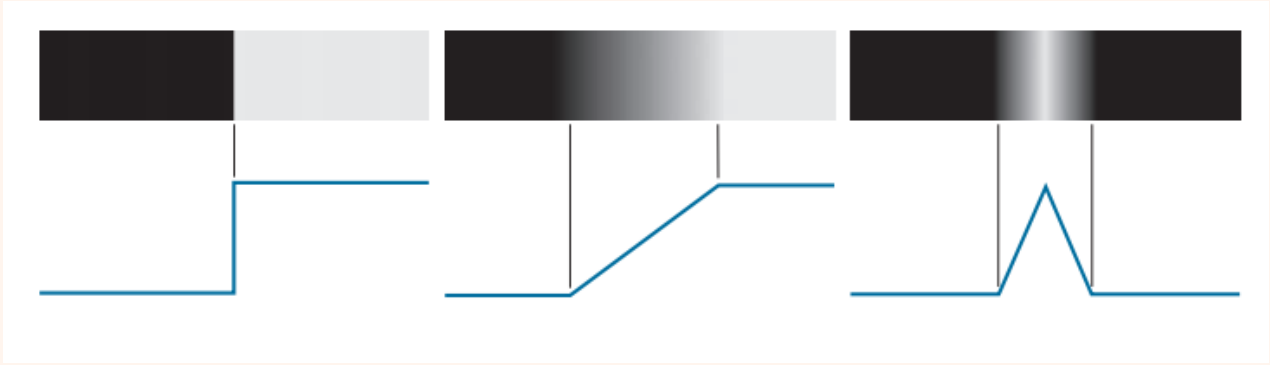
تشخیص لبه

لبهها مجموعه پیکسل‌های به هم پیوسته هستند که مرز نواحی را مشخص می‌کنند.

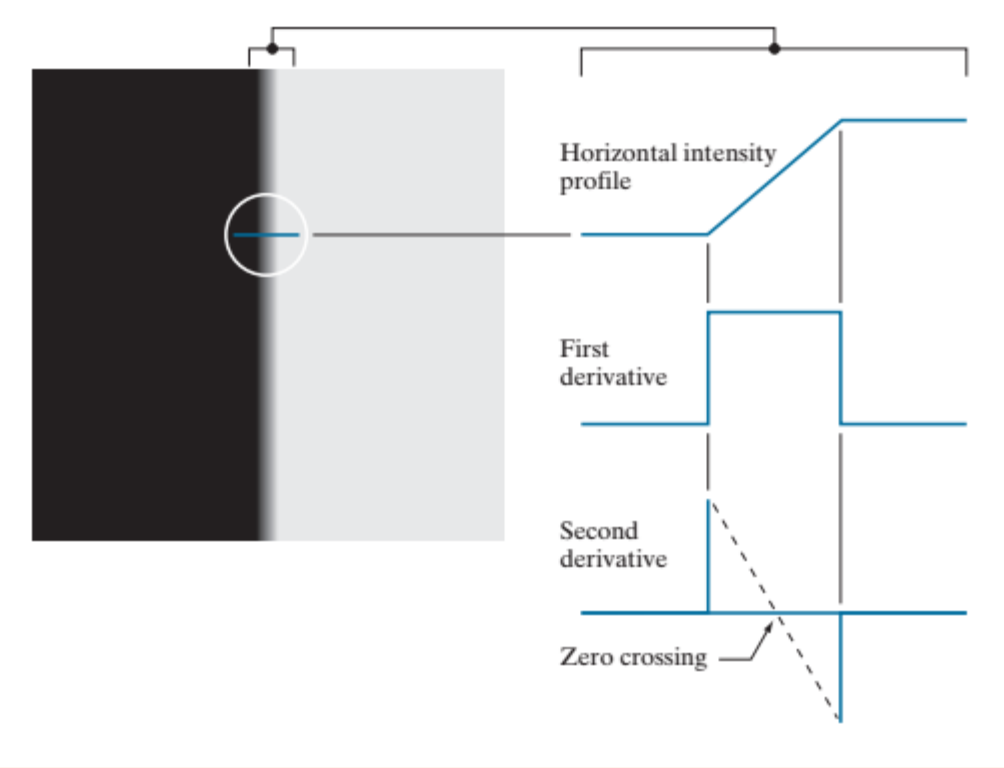
تشخیص‌دهنده‌ی لبه





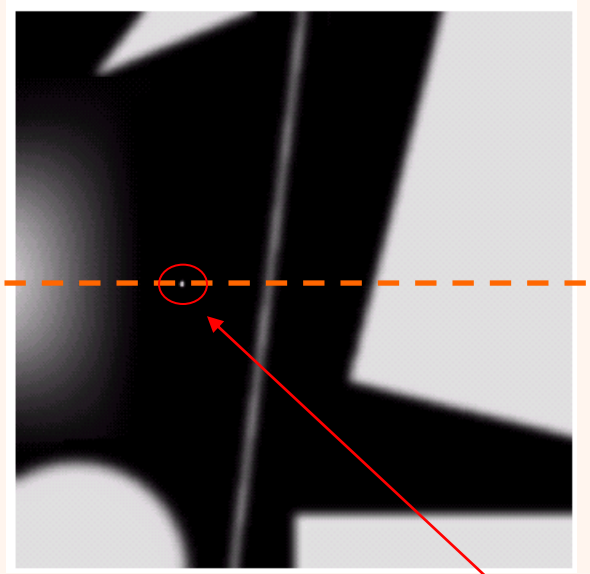


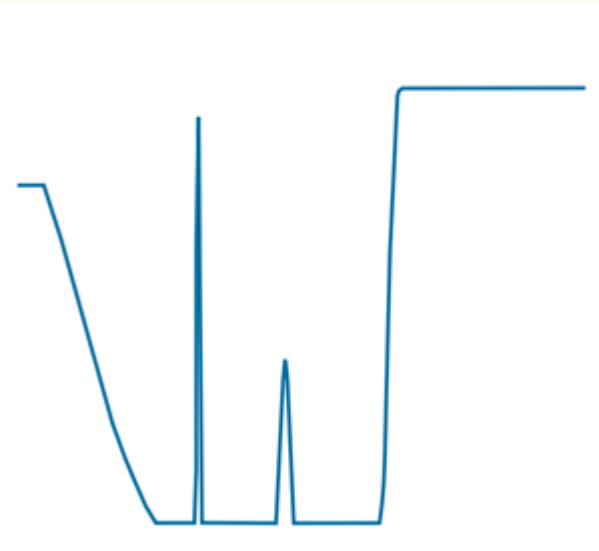
- بر اساس تغییرات مقادیر روشنایی پیکسل‌ها می‌توان لبه‌ها را تشخیص داد.
- نرخ تغییرات را به وسیله‌ی مشتق می‌توان نشان داد.

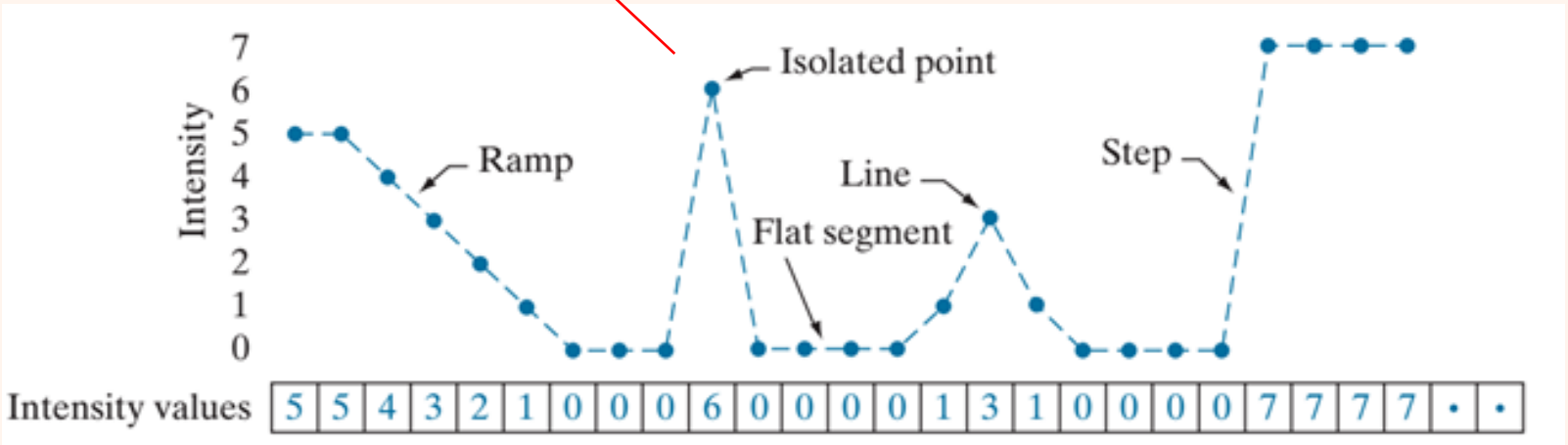


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

تشخیص‌دهنده‌ی لبه





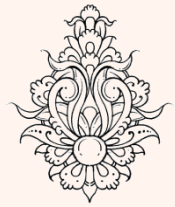


تشخیص‌دهنده‌ی لبه (مشتق اول)

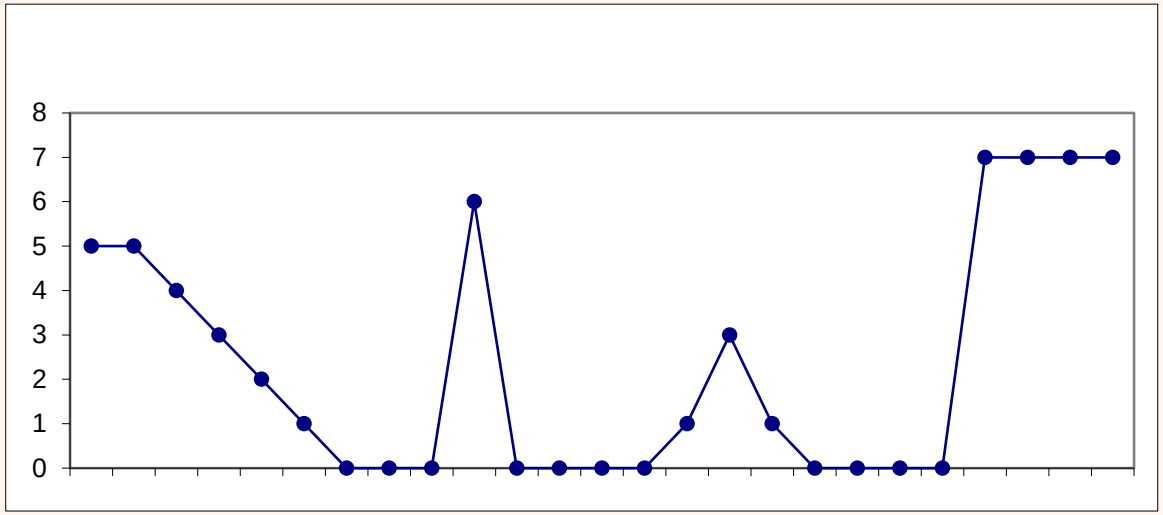
- اگر بخواهیم به صورت یک بعدی مشتق اول را نشان دهیم خواهیم داشت:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- تفاوت میان پیکسل‌های متوالی که نرخ تغییرات را نشان می‌دهد:
  - در نواحی هموار صفر است.
  - در ابتدای لبه مقدار ناصفر دارد.
  - در طول شیب هم ناصفر است.



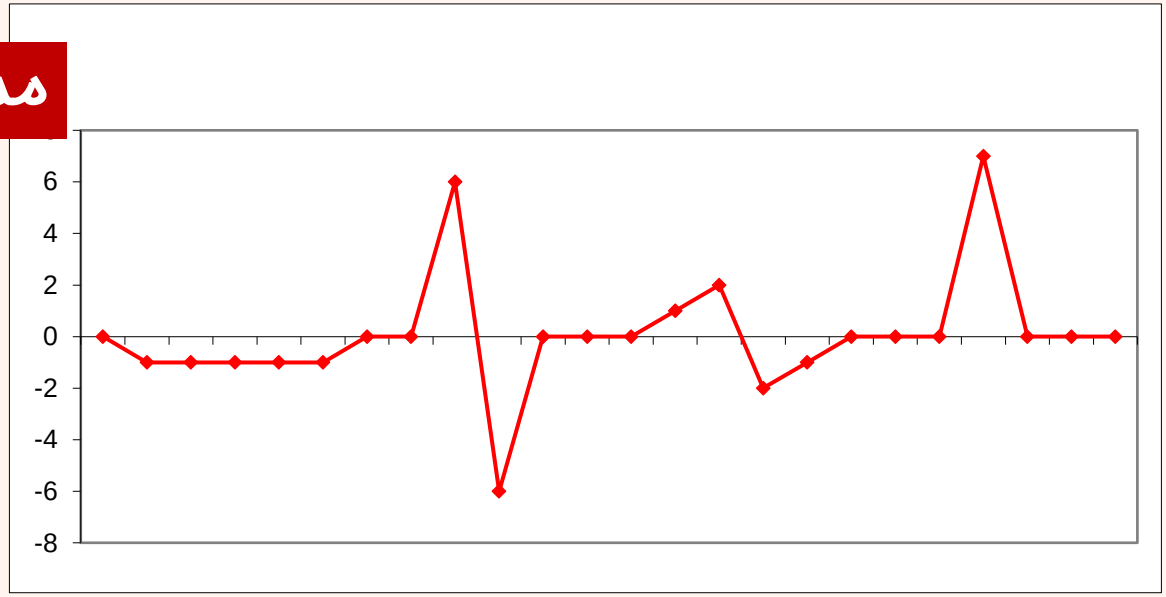
تشخیص‌دهنده‌ی لبه



5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

0	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0

مشتق اول

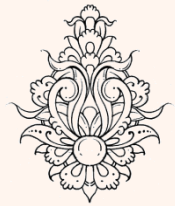


تشخیص‌دهنده‌ی لبه (مشتق دوم)

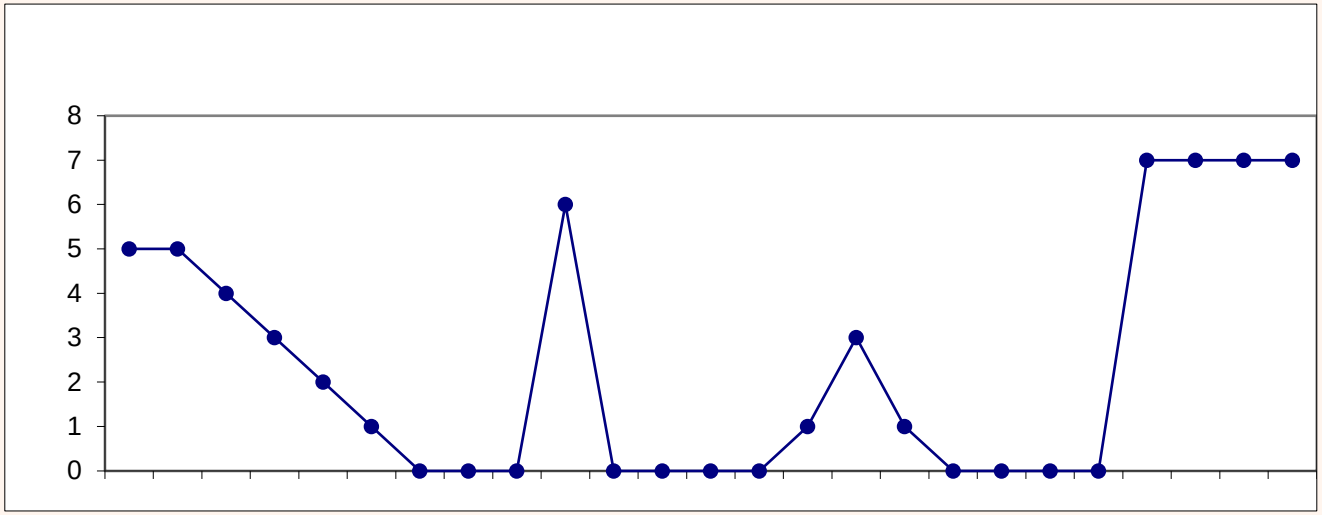
- مشتق دوم از طریق رابطه‌ی زیر محاسبه می‌شود:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

- در مشتق دوم مقادیر پیکسل‌های قبل و بعد از پیکسل جاری در نظر گرفته می‌شود:
  - در نواحی هموار صفر است.
  - در ابتدای و انتهای لبه مقدار ناصفر دارد.
  - در طول شیب هم صفر است.

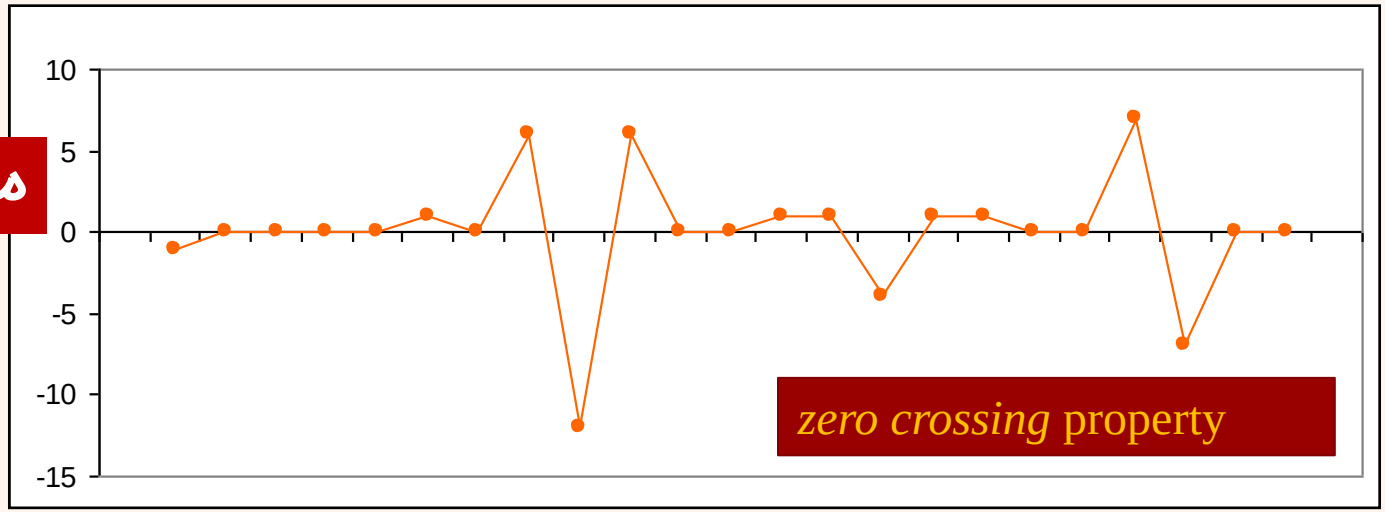


تشخیص‌دهنده‌ی لبه (مشتق دوم)

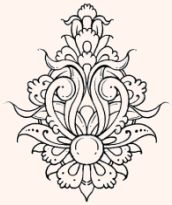


5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

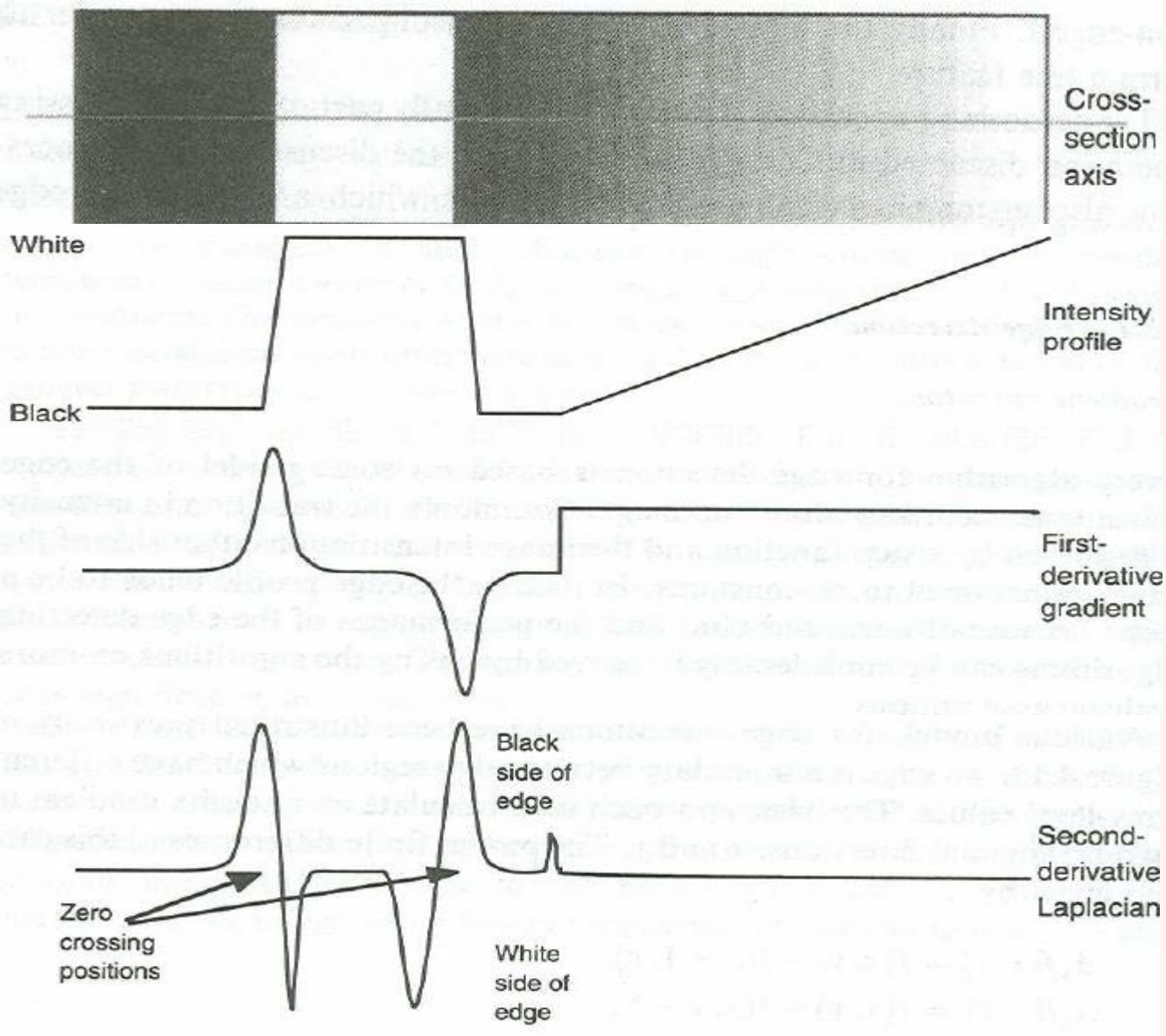
	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



مشتق دوم

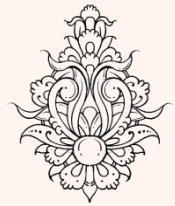


تشخیص‌دهنده‌ی لبه

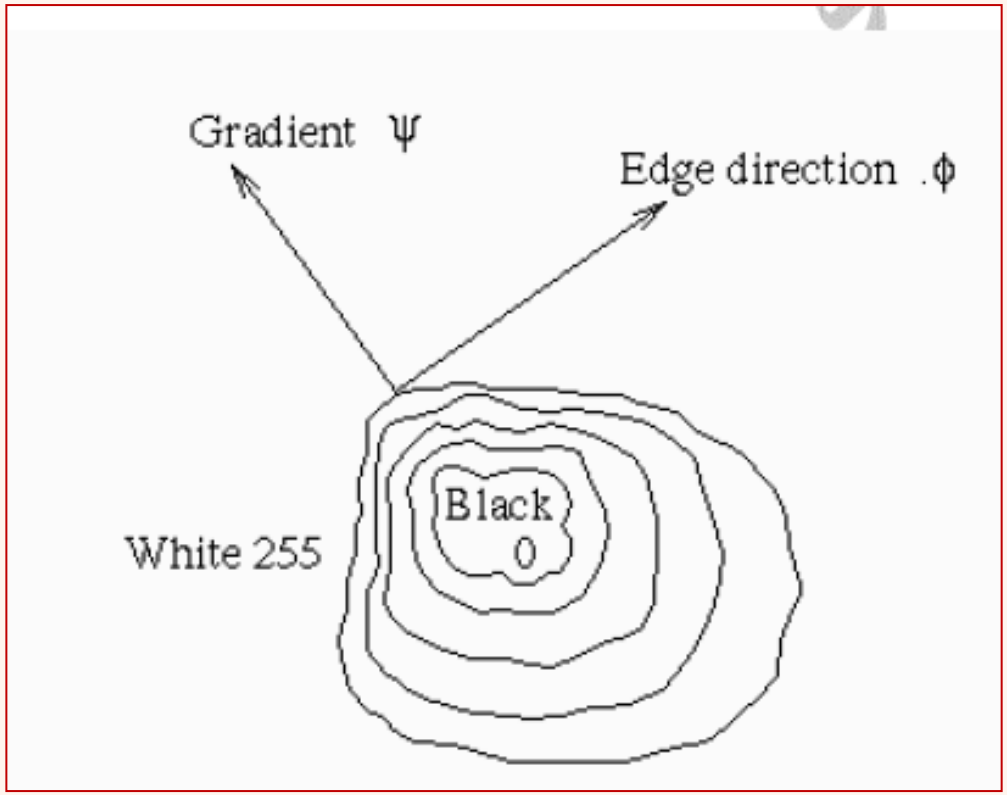


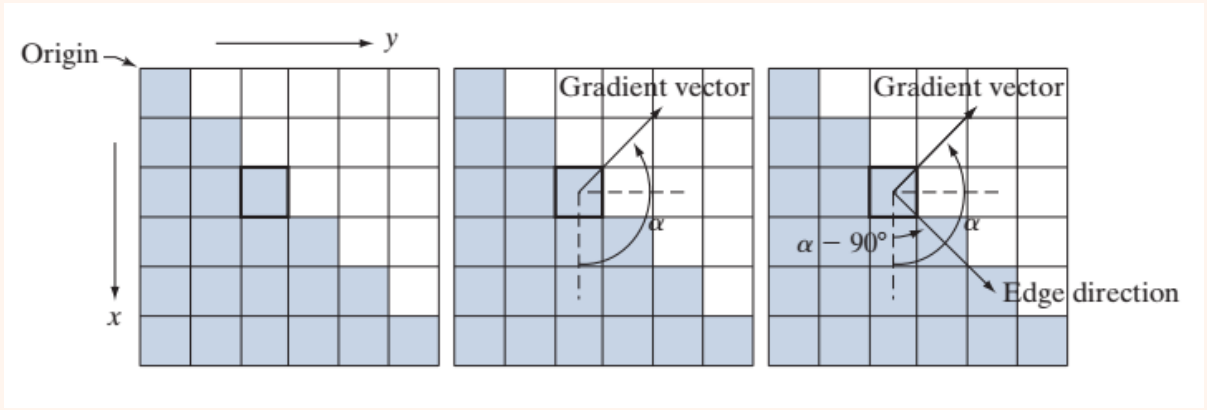
نکات

- مشتق دوم به تغییرات بسیار حساس است و به واسطه‌ی استفاده از آن کوچک‌ترین تغییرات نیز ثبت می‌گردد.
- در استفاده از مشتق اول هنگامی که قدرمطلق مشتق اول یک ماکزیمم محلی باشد لبه خواهیم داشت.
- در مشتق دوم محل‌های گذر از صفر نشان‌گر لبه خواهد بود.



گرادیان


Gradient Ψ
Edge direction .ϕ
Black
0
White 255


Origin
y
x
Gradient vector
α
Gradient vector
α
α − 90°
Edge direction

Gradient

فیلتر با استفاده از مشتق اول

- برای $f(x, y)$ گرادیان f در دو جهت x و y به وسیله‌ی رابطه‌ی زیر تعیین می‌شود:

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- بزرگی بردار زیر به وسیله‌ی رابطه‌ی زیر تعیین می‌شود:

$$\nabla f = mag(\nabla \mathrm{f}) = \left[G_x^2 + G_y^2\right]^{1/2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

فیلتر با استفاده از مشتق اول

- برای سادگی رابطه را به صورت زیر نیز می‌توان در نظر گرفت:

$$\nabla f \approx |G_x| + |G_y|$$

- فیلترهایی که بر پایه‌ی گرادیان(مشتق) طراحی می‌گردند انواع گوناگونی دارند.

فیلترهای مشتق‌گیر Robert

G_x

-1	1

G_y

-1
1

اختلاف روشنایی برای سطرها و ستون‌ها محاسبه می‌شود.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

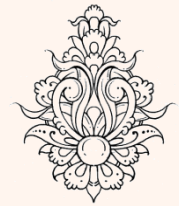
فیلتر با استفاده از مشتق اول

- فیلترهای مشتق‌گیر Robert

-1	0
0	1

0	-1
1	0

- برای تهیه‌ی نقشه‌ی لبه‌ها می‌باید مقدار قدرمطلق مشتق با مقدار آستانه‌ای مقایسه گردد تا وجود یا عدم وجود لبه مشخص گردد. مقدار آستانه بسته به کاربرد می‌تواند متفاوت باشد.



فیلترهای مشتق‌گیر

- فیلترهای مشتق‌گیر Prewitt

h_x

1	0	-1
1	0	-1
1	0	-1

h_y

1	1	1
0	0	0
-1	-1	-1

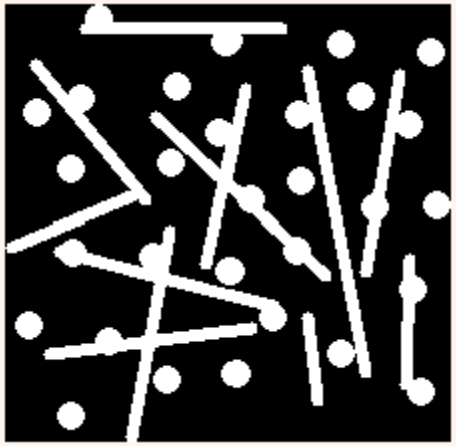
1	1	0
1	0	-1
0	-1	-1

تشخیص خطوط ۴۵ درجه

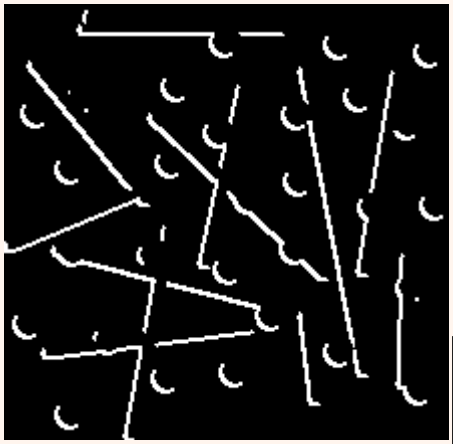
0	1	1
-1	0	1
-1	-1	0

تشخیص خطوط ۴۵ – درجه

فیلترهای مشتق‌گیر قطری



0	1	1
-1	0	1
-1	-1	0



تشخیص خطوط ۴۵- درجه

1	1	0
1	0	-1
0	-1	-1

تشخیص خطوط ۴۵درجه

فیلتر با استفاده از مشتق اول

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

- بر اساس روابط فیلتر زیر به دست می‌آید.

Sobel Operators

h_y

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

h_x

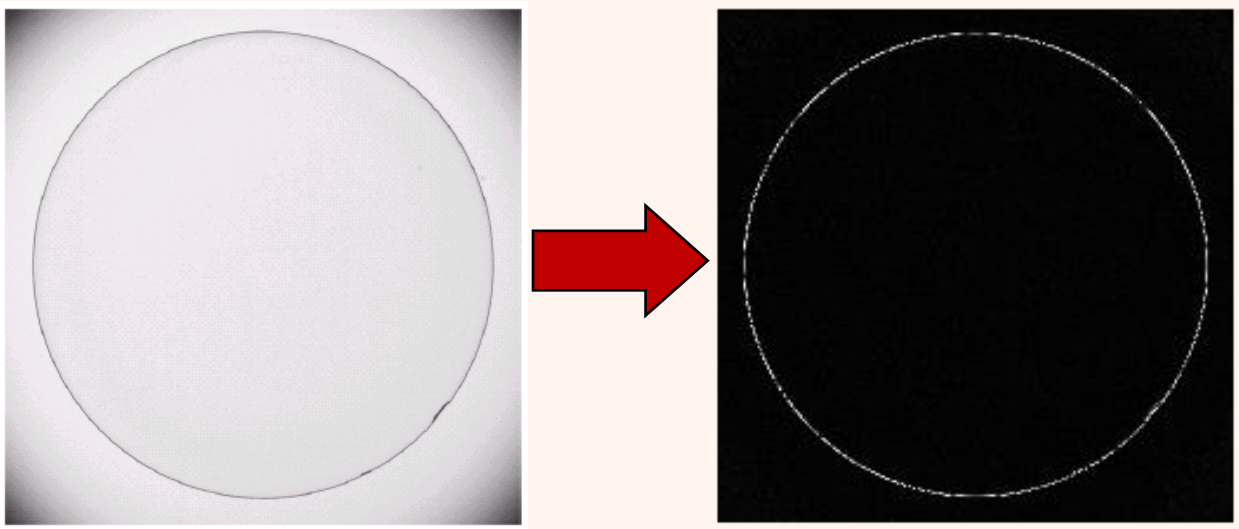
-1	0	1
-2	0	2
-1	0	1

$h_x = h_y^T$

- دو فیلتر به تصویر اصلی اعمال می‌شود و نتایج با هم جمع می‌شود.

فیلتر با استفاده از مشتق اول

- از فیلتر Sobel برای تشخیص لبه به صورت گسترده استفاده می‌شود.



فیلترهای مشتق‌گیر Sobel

h_x

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

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

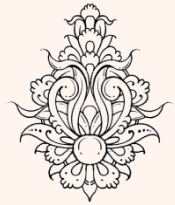
تشخیص خطوط ۴۵ درجه

h_y

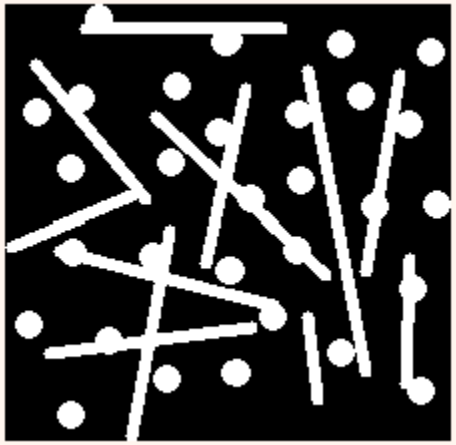
-1	-2	-1
0	0	0
1	2	1

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

تشخیص خطوط ۴۵ – درجه



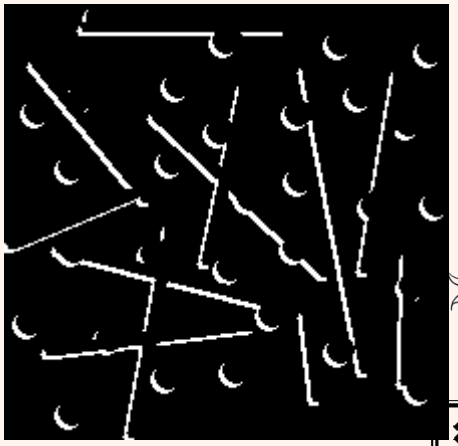
مثال



h_x

تشخیص خطوط ۴۵ درجه

h_y



تشخیص خطوط ۴۵ – درجه

```
BW = edge(I,'Method',thresh,direction)
```

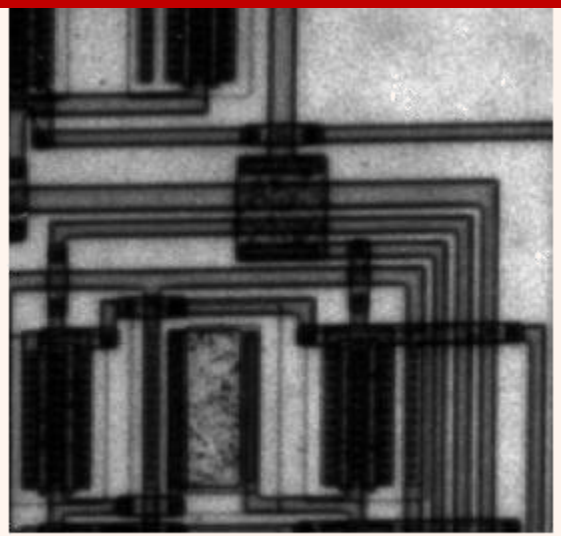
'Sobel'
'Robert'
'Prewitt'
……..

'horizontal' or 'vertical' edges
or 'both' (the default).

```
[BW th] = edge(I,'Method');
```

متوسط گرادیان‌ها را به عنوان آستانه می‌گیرد

```
I = imread('circuit.tif');
BW1 = edge(I,'sobel',0.02);
BW2 = edge(I,'sobel',0.1);
imshow(I,[]);
figure,imshow(BW1);
figure, imshow(BW2)
```

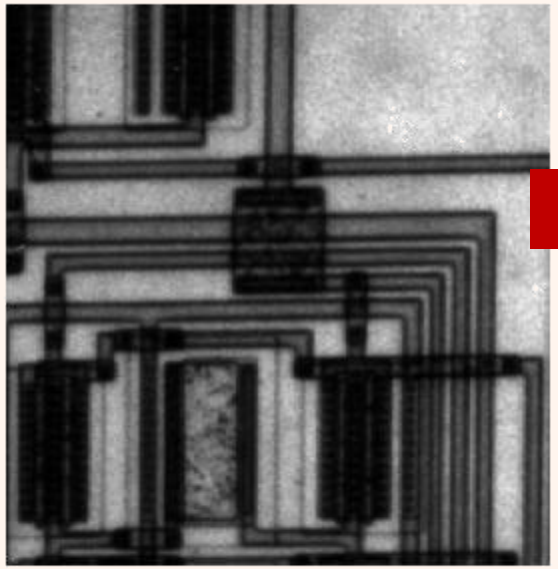


Org

Sobel
Th=0.1

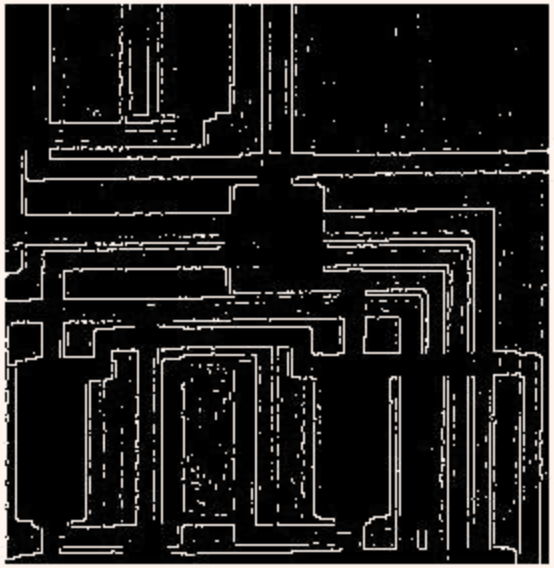
Sobel
Th=0.02

فیلترهای مشتق‌گیر



Org

Robert



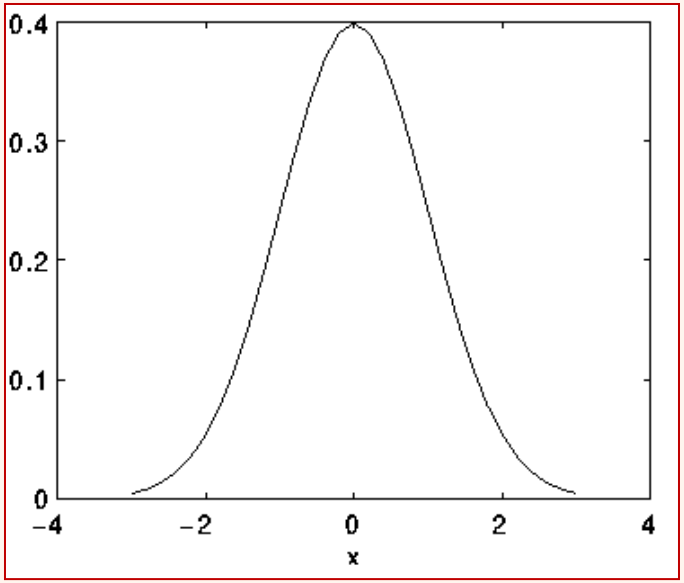
Sobel

Prewitt

روش Canny

f → Gaussian filter (هموارسازی) → گرادیان → g

$$h_{gau}(x) = e^{-\frac{x^2}{2\sigma^2}}$$



$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

$\sigma = 1.4$

Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

روش Canny (ادامه...)

Org

Smoothed

روش Canny (ادامه...)

- پس از به‌دست آوردن تصویر هموار گرادیان تصویر به دست آورده می‌شود (می‌توان از فیلترهای Sobel استفاده نمود).

$$|G| = \sqrt{G_x^2 + G_y^2}$$
$$|G| = |G_x| + |G_y|$$

- سپس جهت لبه‌ها مشخص می‌شود:

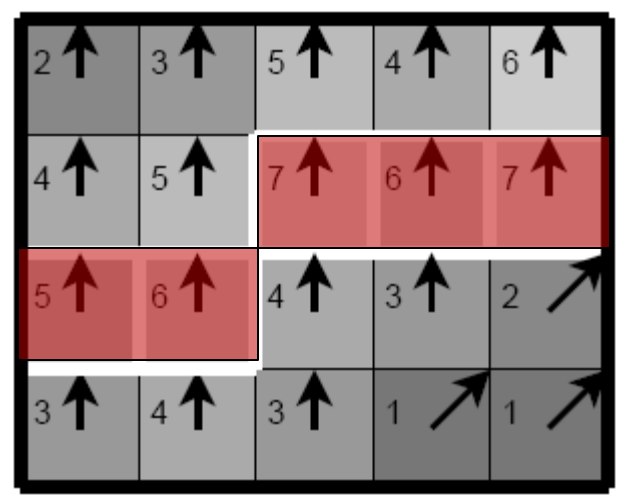
$$\theta = \arctan\left(\frac{|G_y|}{|G_x|}\right)$$

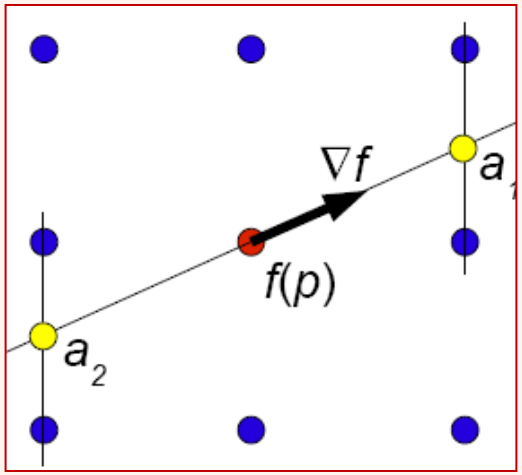
جهت لبه بر جهت مشتق عمود است

- هنگامی‌که جهت مشخص می‌گردد مقدار پیکسل کاندید شده برای لبه بودن تنها در همان جهت بررسی می‌گردد.

Non-Maximum Suppression

- در مثال زیر جهت برای پیکسل‌های شکل عمدتا ۹۰ درجه است پس مقادیر با پیکسل‌های بالا و پایین مقایسه می‌گردد.





$$f(p) > a_1 \wedge f(p) \geq a_2$$
$$f(p) \geq a_1 \wedge f(p) > a_2$$

اگر شرایط برقرار نباشد کاندید حذف می‌گردد

http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf

Double thresholding

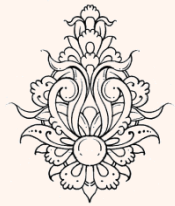
- حتی پس از اعمال فرآیند non-maximum-suppression مقادیری که به عنوان لبه باقی می‌مانند، ممکن است بر اثر نویز به وجود آمده باشند.
- با اعمال دو مقدار آستانه مقادیر آنالیز می‌گردند.

مقدار پیکسل> TH_h	لبه‌ی قوی
TH_l <مقدار پیکسل< TH_h	لبه‌ی ضعیف
مقدار پیکسل< TH_l	حذف می‌گردد

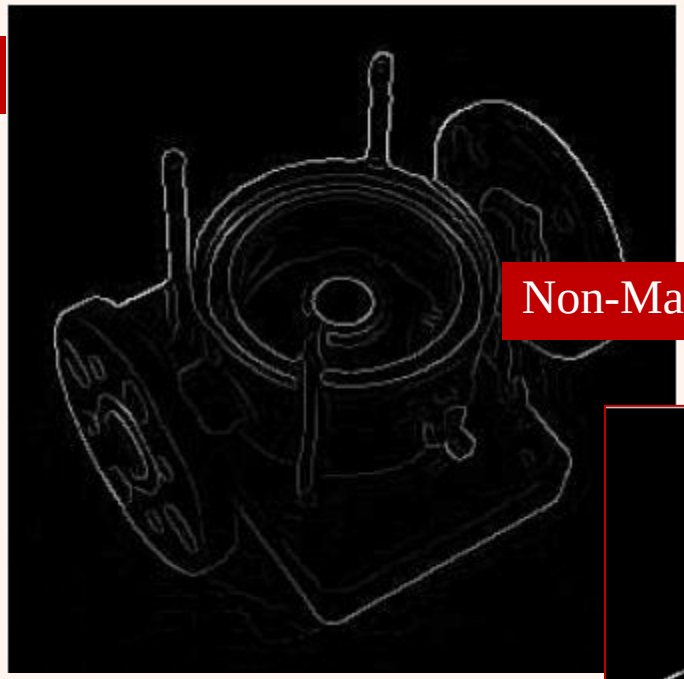
Edge tracking by hysteresis

- اگر مقدار پیکسل از TH_h بزرگ‌تر باشد لبه است.

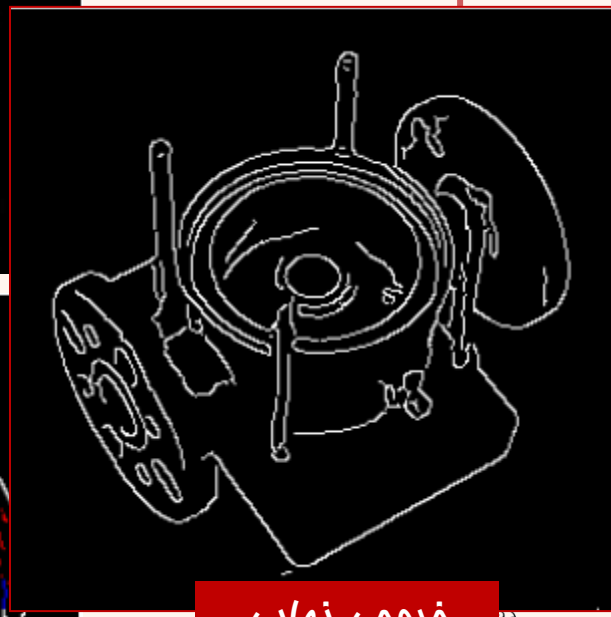
- در همسایگی پیکسل مورد نظر جستجو می‌کنیم، اگر از TH_l کم‌تر نشود همچنان لبه است. در غیر اینصورت پیکسل بعدی زمانی لبه است که از TH_hبزرگ‌تر باشد .



گرادیان تصویر هموار

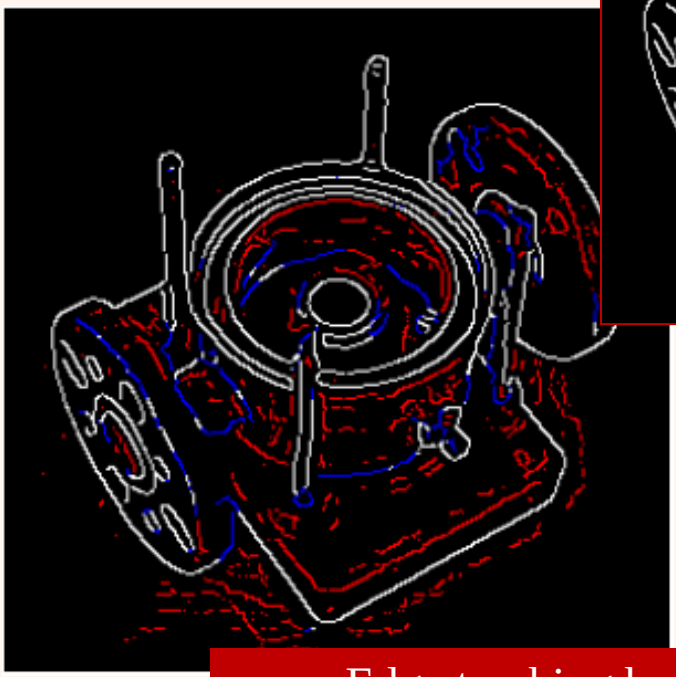


Non-Maximum Suppression



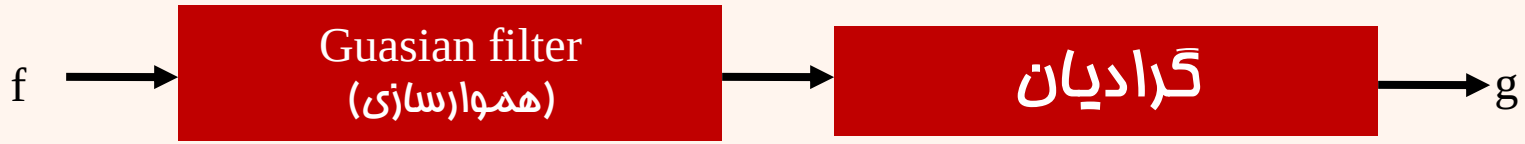
خروجی نهایی

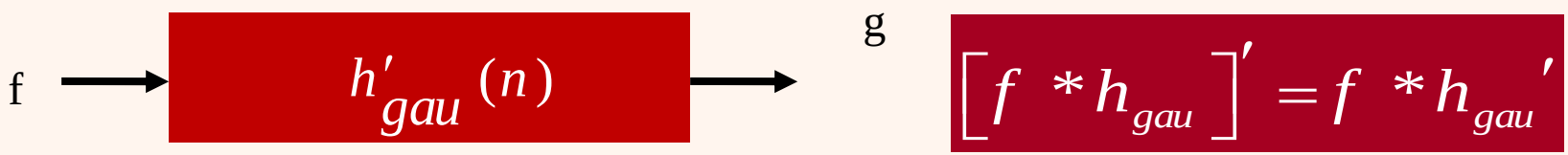
دو مقدار آستانه



Edge tracking by hysteresis

روش Canny (ادامه...)





$$h_{gau}(x) = e^{-\frac{x^2}{2\sigma^2}}$$

$$h'_{gau}(x) = (-\frac{x}{\sigma^2})e^{-\frac{x^2}{2\sigma^2}}$$

$$h_{gau}(n) = e^{-\frac{n^2}{2\sigma^2}} \rightarrow h'_{gau}(n) = (-\frac{n}{\sigma^2})e^{-\frac{n^2}{2\sigma^2}} \qquad -(M-1) \le n \le M-1$$

فیلتر لاپلاسین

Laplacian Filter

- فیلتر لاپلاسین از رابطه‌ی زیر به دست می‌آید:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- برای اعمال به تصویر در جهت محور xها خواهیم داشت:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

- و در جهت محور y :

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

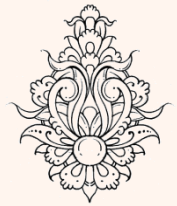
فیلتر لاپلاسین

- بنابراین خواهیم داشت:

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

- بر این اساس فیلتری مانند شکل زیر می‌توان در نظر گرفت:

0	1	0
1	-4	1
0	1	0



فیلتر لاپلاسین

- فیلتر لاپلاسین می‌تواند به صورت‌های زیر در نظر گرفته شود:

0	1	0
1	-4	1
0	1	0

0	-1	0
-1	4	-1
0	-1	0

1	1	1
1	-8	1
1	1	1

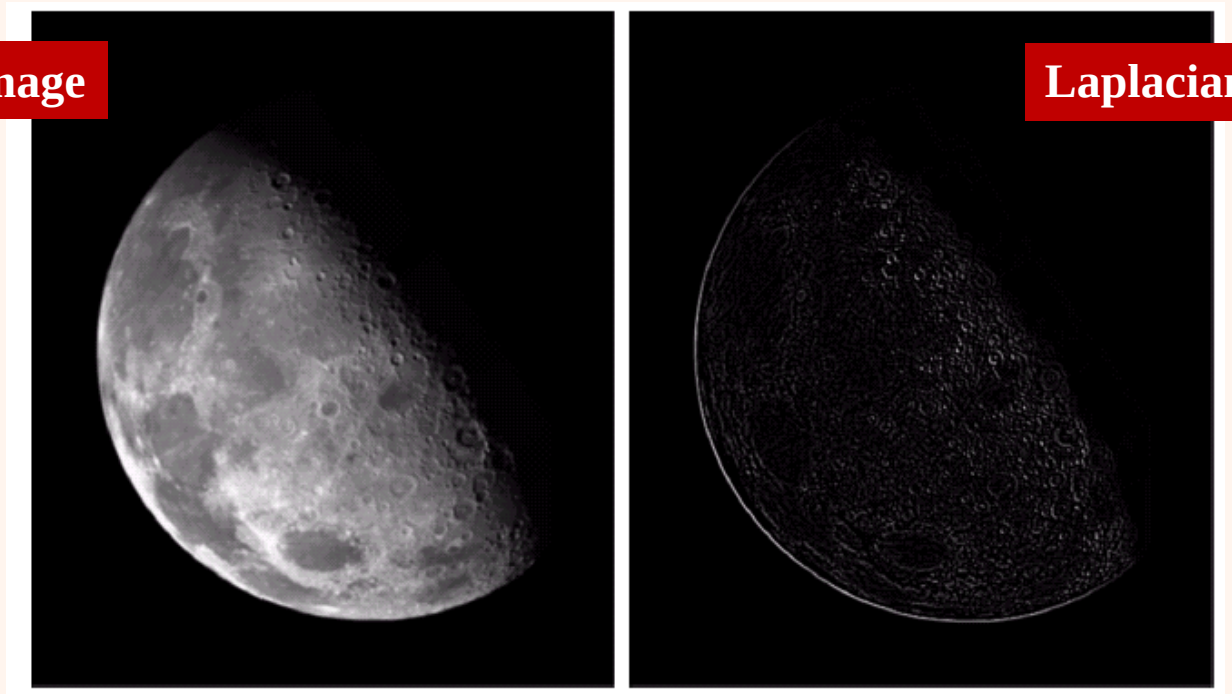
-1	-1	-1
-1	8	-1
-1	-1	-1

- مجموع مقادیر داخل فیلتر باید صفر باشد تا در نواحی هموار پاسخ صفر دهد و تنها لبه را مشخص کند.

فیلتر لاپلاسین

```
h = fspecial('laplacian', alpha);
```

- هنگامی‌که فیلتر لاپلاسین به تصویر اعمال گردد، جزییات و لبه‌های تصویر بهتر آشکار خواهد شد.



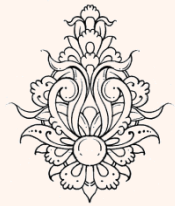
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

دریچه‌ی تیزکننده

- برای هر چه بیشتر آشکار شدن لبه‌ها و جزییات تصویر می‌توان به طریق زیر عمل نمود:

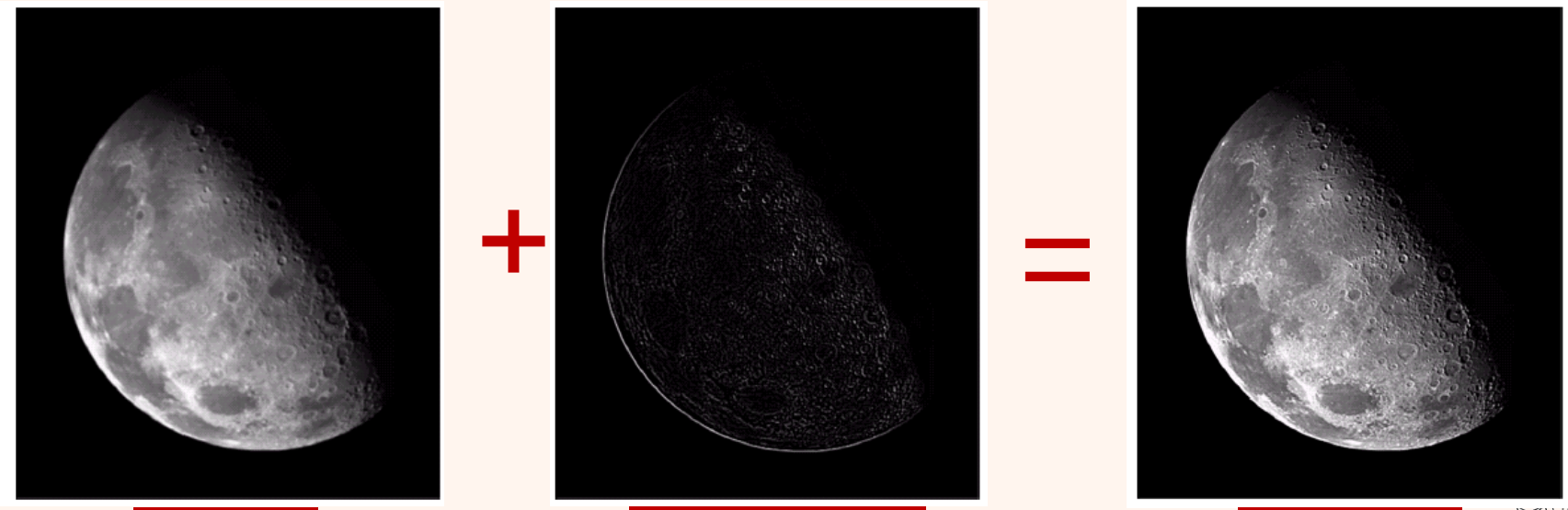
$$g(x,y)=f(x,y)+c[\nabla^2 f(x,y)]$$

- بسته به این که پیکسل مرکزی در فیلتر لاپلاسین مثبت یا منفی باشد c متفاوت است.



دریچه‌ی تیزکننده

- برای این منظور می‌توان به طریق زیر عمل نمود:



دریچه‌ی تیزکننده

- اگر پیکسل مرکزی فیلتر مقدار مثبت داشته باشد به واسطه‌ی جمع نمودن تصویر اصلی و مشتق دوم آن تصویر تیز شده و در غیر این‌صورت از کسر نمودن دو مقدار تصویر مطلوب نتیجه می‌شود.

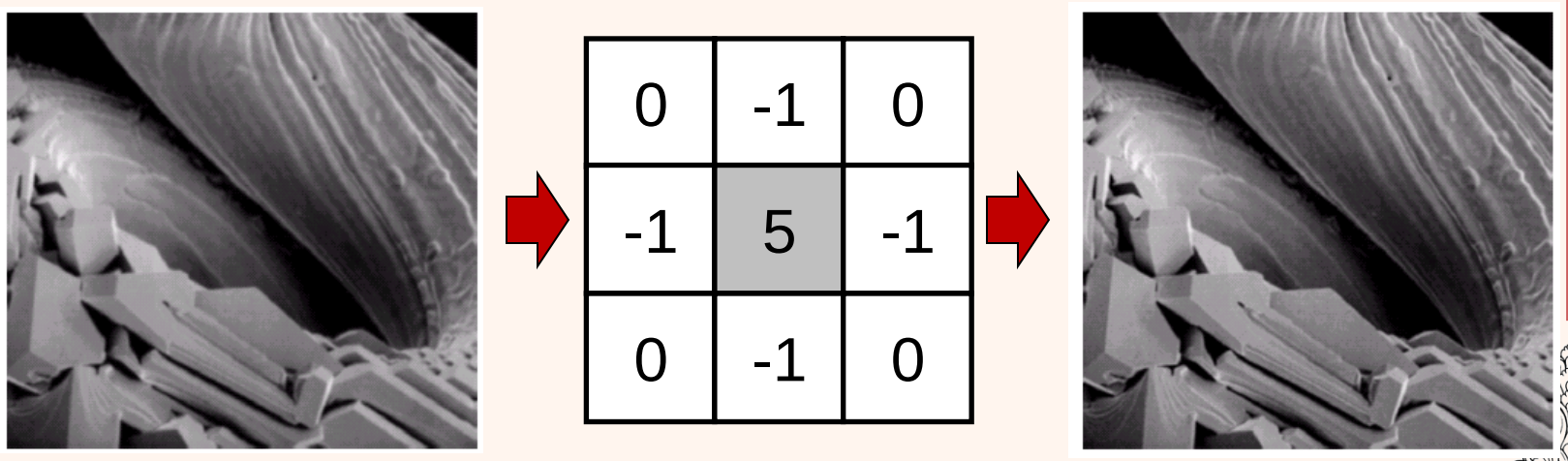
$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \text{if the center of Laplacian mask is negative} \\ f(x,y)+\nabla^2 f(x,y) & \text{if the center of Laplacian mask is positive} \end{cases}$$

دریچه‌ی تیزکننده

بهشتی

دریچه‌ی تیزکننده

- با استفاده از فیلتر به‌دست آمده خواهیم داشت:



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

دریچه‌ی تیزکننده

```
I = imread('barb.gif');
W=[0 -1 0; -1 5 -1;0 -1 0];
gd=imfilter(I,w);imshow(I)
figure, imshow(gd,[]);
```

Org image

Sharped image

دریچه‌ی تیزکننده

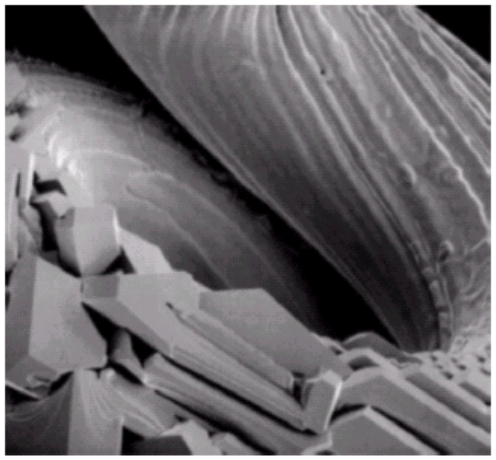
- ساختارهای متفاوتی از فیلترهای لاپلاسی را می‌توان در نظر گرفت:

0	1	0
1	-4	1
0	1	0

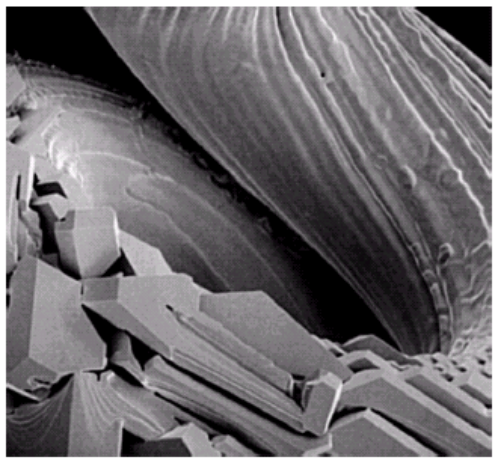
Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian

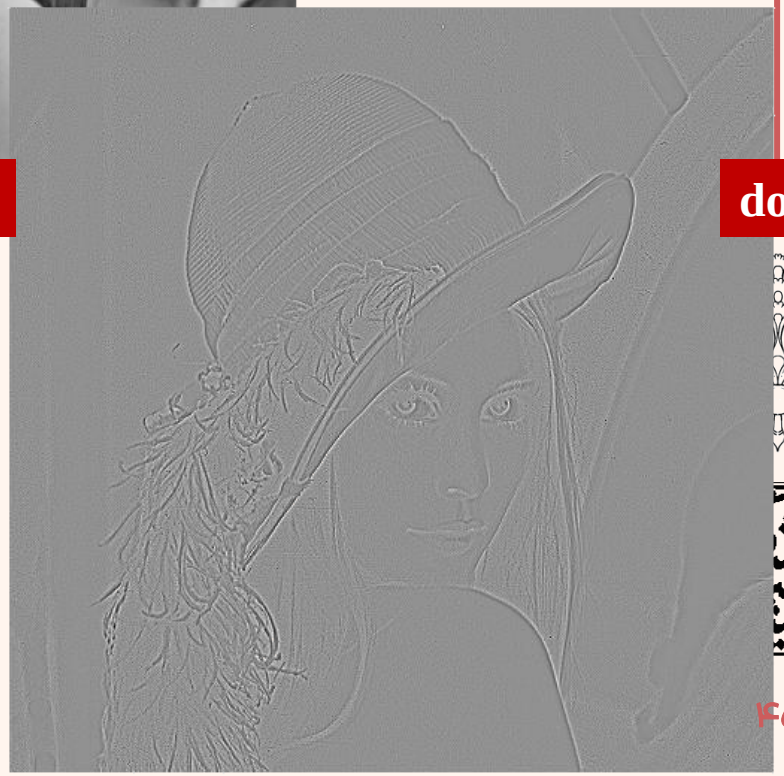


-1	-1	-1
-1	9	-1
-1	-1	-1



Org

Uint8



double

```
I = imread('lena.gif');
w=[0 1 0;1 -4 1;0 1 0];
imshow(I);
f2=im2double(I);
g2=imfilter(f2,w,'replicate');
figure;imshow(abs(g2),[]);
```

دریچه‌ی تیزکننده

```
I = imread('lena.gif');
w=[0 1 0;1 -4 1;0 1 0];
imshow(I);
f2=im2double(I);
g2=imfilter(f2,w,'replicate');
figure;imshow(g2,[]);
g=f2-g2;
figure;imshow(g);
```

```
I = imread('lena.gif');
w4 = fspecial('laplacian',0);
w8=[1 1 1;1 -8 1;1 1 1];
f=im2double(I);
g4=f-imfilter(f,w4,'replicate');
g8=f-imfilter(f,w8,'replicate');
imshow(f);
figure;imshow(g4);
figure;imshow(g8);
```

Org

w4

w8

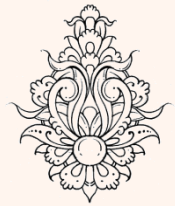
دریچه‌ی تیزکننده

Sharpening Filter

- تأکید به روی جزییات تصویر به ویژه لبه‌ها
- کاربرد در فرآیندهایی چون آشکارسازی رگ‌های خونی و یا آنژیوگرافی

$$\frac{1}{(\alpha+1)}\begin{bmatrix} -\alpha & \alpha-1 & -\alpha \\ \alpha-1 & \alpha+5 & \alpha-1 \\ -\alpha & \alpha-1 & -\alpha \end{bmatrix}$$

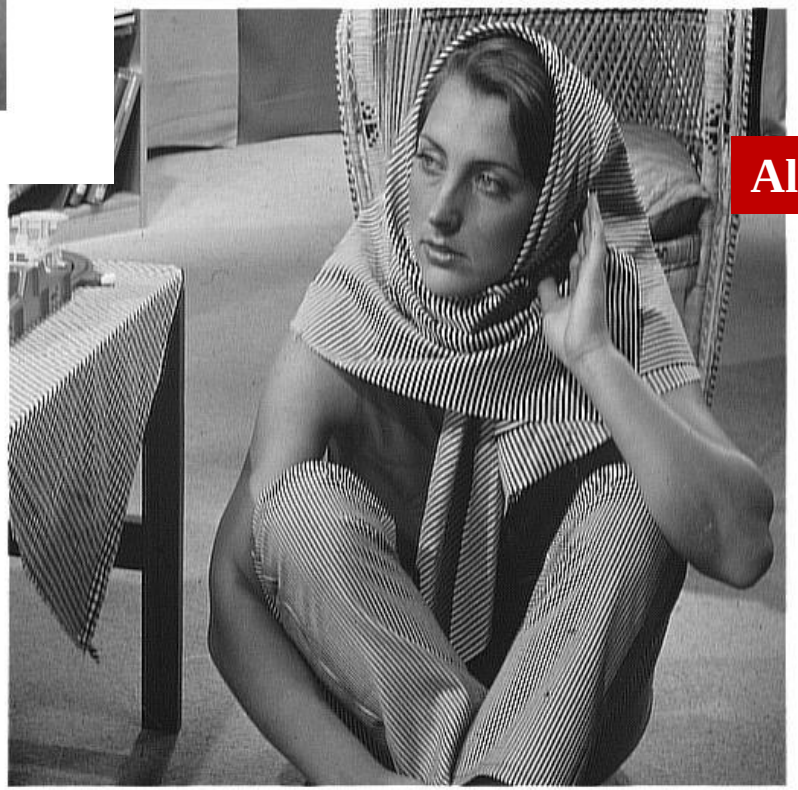
فیلتر استفاده شده در Matlab



مثال

Org

$$\frac{1}{(\alpha+1)}\begin{bmatrix} -\alpha & \alpha-1 & -\alpha \\ \alpha-1 & \alpha+5 & \alpha-1 \\ -\alpha & \alpha-1 & -\alpha \end{bmatrix}$$



Alpha=6

فیلتر High-boost

$$f_s(x,y) = f(x,y) - \bar{f}(x,y)$$

$$f_{hb}(x,y) = Af(x,y) - \bar{f}(x,y)$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f(x,y) - \bar{f}(x,y)$$

$$f_{hb}(x,y) = (A-1)f(x,y) + f_s(x,y)$$

فیلتر High-boost

$$f_{hb}(x,y) = (A-1)f(x,y) + f_s(x,y)$$

تصویر تیز شده به صورت کلی

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center of Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center of Laplacian mask is positive} \end{cases}$$

فیلتر High-boost

- به منظور تیز نمودن تصویر می‌توان از فیلتر زیر استفاده نمود:

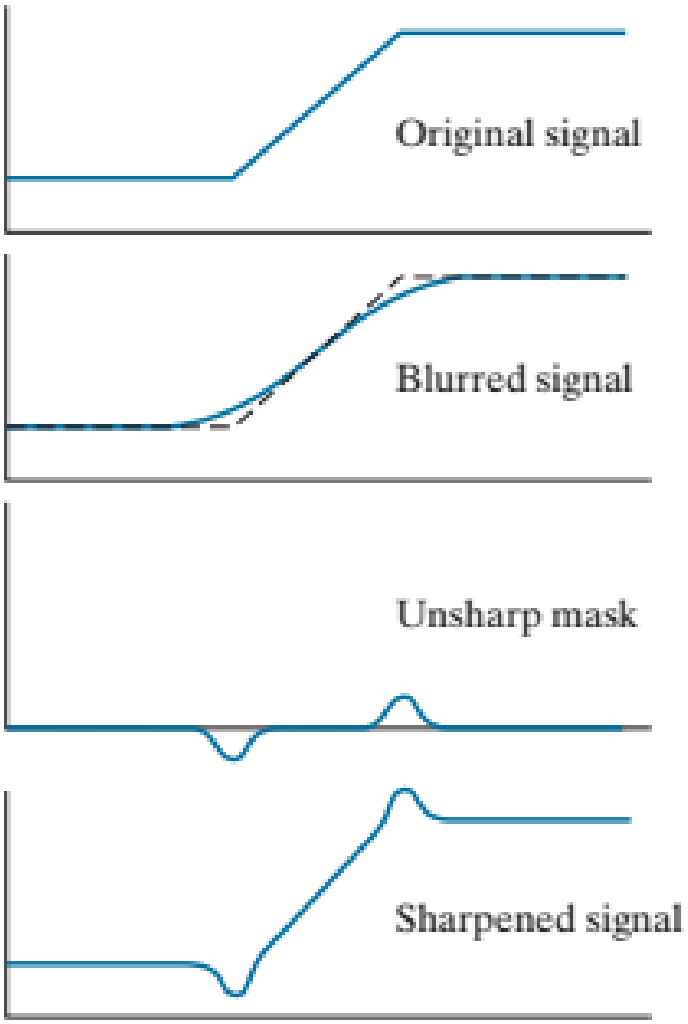
$$f_{hb} = \begin{cases} Af(x,y) - \nabla^2 f(x,y) & \text{if the center of Laplacian mask is negative} \\ Af(x,y) + \nabla^2 f(x,y) & \text{if the center of Laplacian mask is positive} \end{cases}$$

- اگر A=1 باشد تیزکننده با فیلتر لاپلاسین معمولی خواهیم داشت (حالت استاندارد)

0	-1	0
-1	A+4	-1
0	-1	0

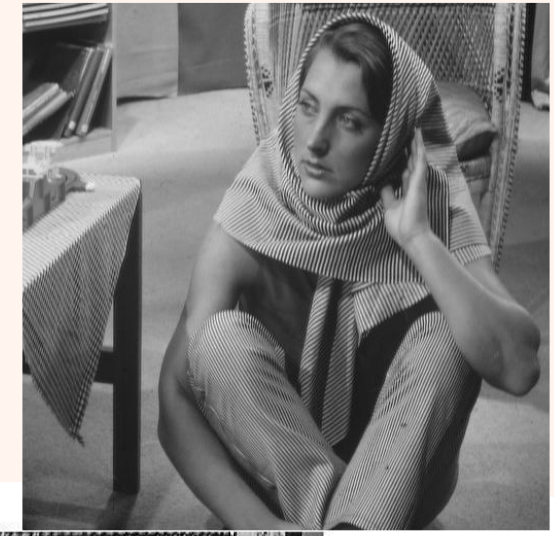
-1	-1	-1
-1	A+8	-1
-1	-1	-1

High-boost فیلتر



مثال

Org



A=1

A=1.7

a b c
d e

FIGURE 3.49 (a) Original image of size 600×259 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$. (c) Mask. (d) Result of unsharp masking using Eq. (3-56) with $k = 1$. (e) Result of highboost filtering with $k = 4.5$.

فیلتر لاپلاسین

- هنگامی‌که فیلتر مذکور به تصویر اعمال می‌شود به وسیله‌ی مشخص نمودن یک مقدار آستانه تعیین می‌شود پیکسل مورد نظر جزیی از لبه خواهد بود یا خیر.
- اگر مقدار آستانه خیلی بزرگ یا خیلی کوچک باشد نتیجه‌ی مطلوب به دست نمی‌آید.

-1	-1	-1
-1	8	-1
-1	-1	-1

1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

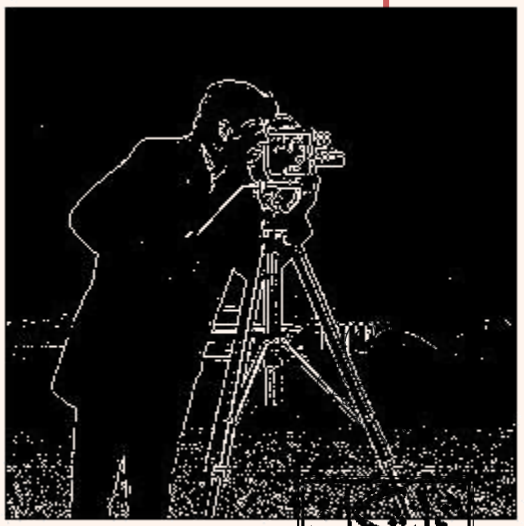
0	1	0
1	-4	1
0	1	0

مثال

Org

Th=0.07

Th=0.14



Th=0.44

Marr-Hildreth

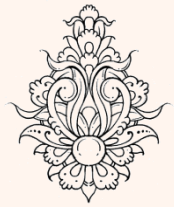
- در روش لاپلاسین همراه با تشخیص لبه، نویز بسیاری توسط سیستم تشخیص داده می‌شود.
- روش مطرح شده در راستای حذف نویزهای مذکور و ارائه‌ی پاسخی بهینه پیشنهاد شده است.

Th=0.14

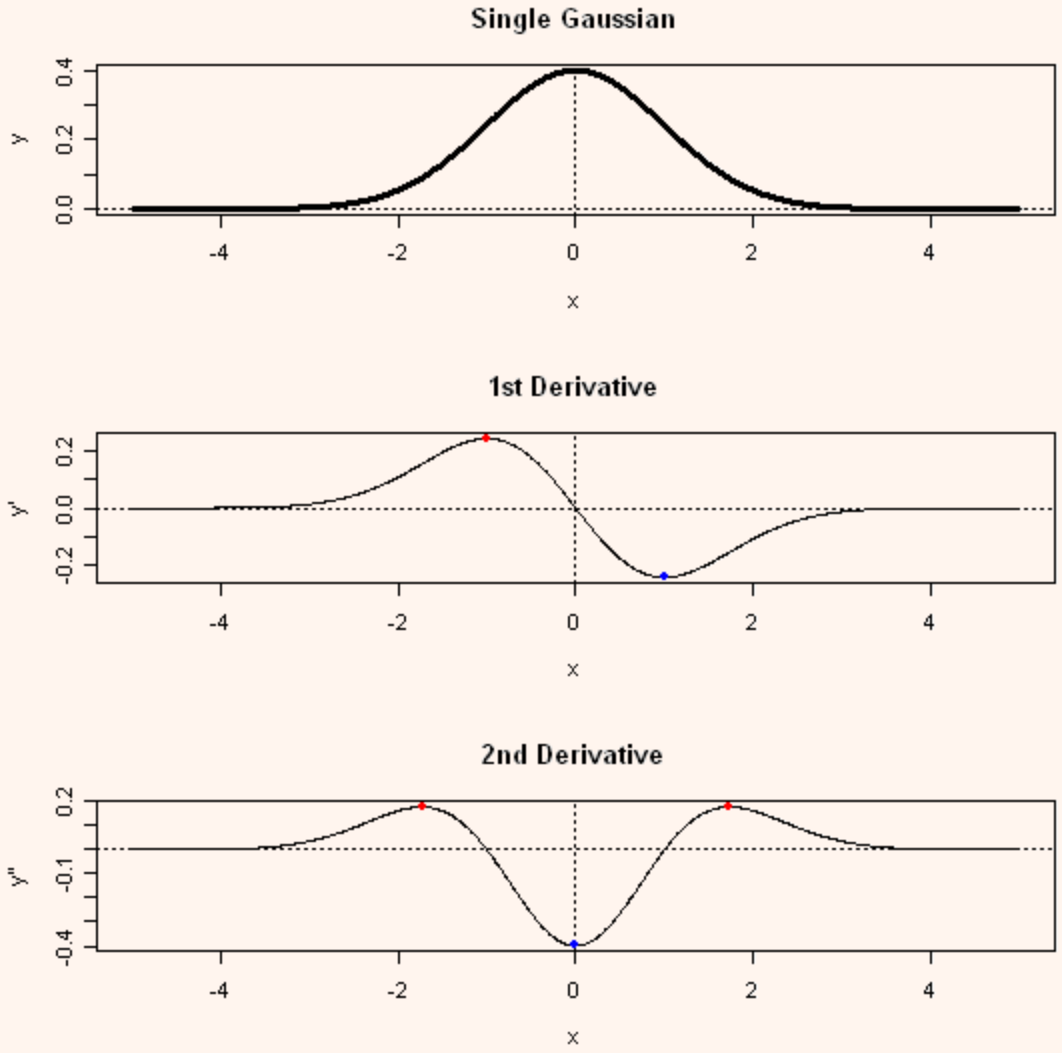
LoG

- برای از میان بردن نویزها ابتدا تصویر را از یک فیلتر گوسی هموارساز عبورمی‌دهیم.
- سپس به روی تصویر به‌دست آمده فیلتر لاپلاسین را اعمال می‌کنیم.

$$\underbrace{\nabla^2(f(x,y)\otimes G(x,y))}_{\text{Laplacian of Gaussian-filtered image}} = \underbrace{\nabla^2 G(x,y)\otimes f(x,y)}_{\text{Laplacian of Gaussian (LoG) -filtered image}}$$



تابع گاوسی و مشتق‌هایش

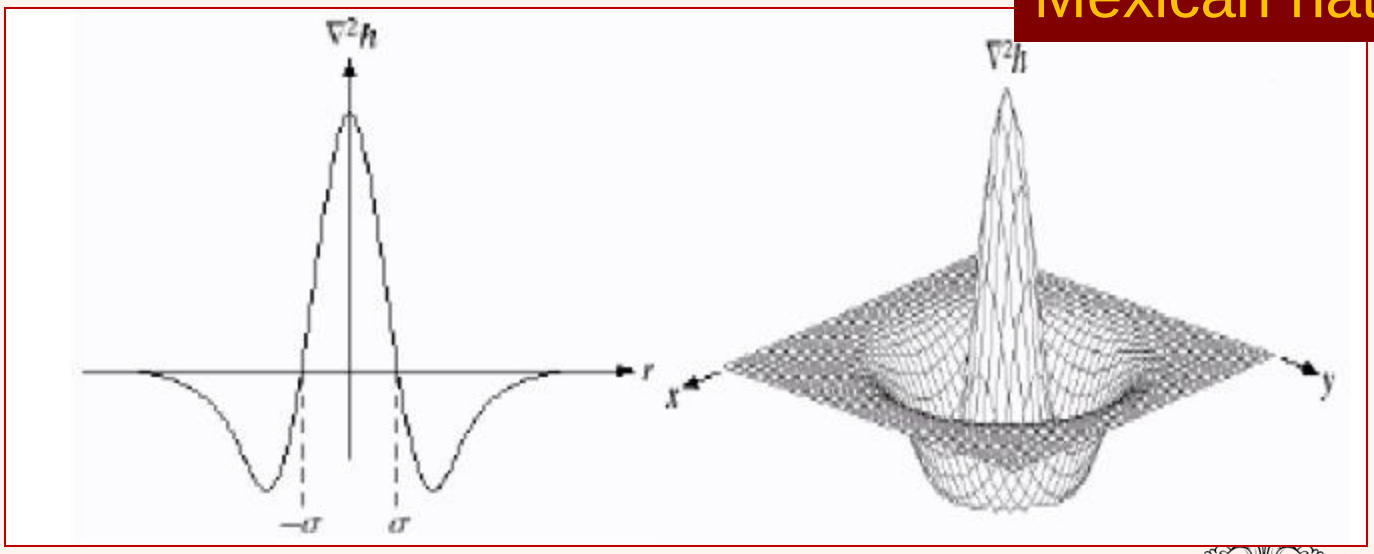


LoG

- نمونه‌ای از دریچه‌ی پیشنهادی

Mexican hat

0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0




```
h = fspecial('log', hsize, sigma)
```

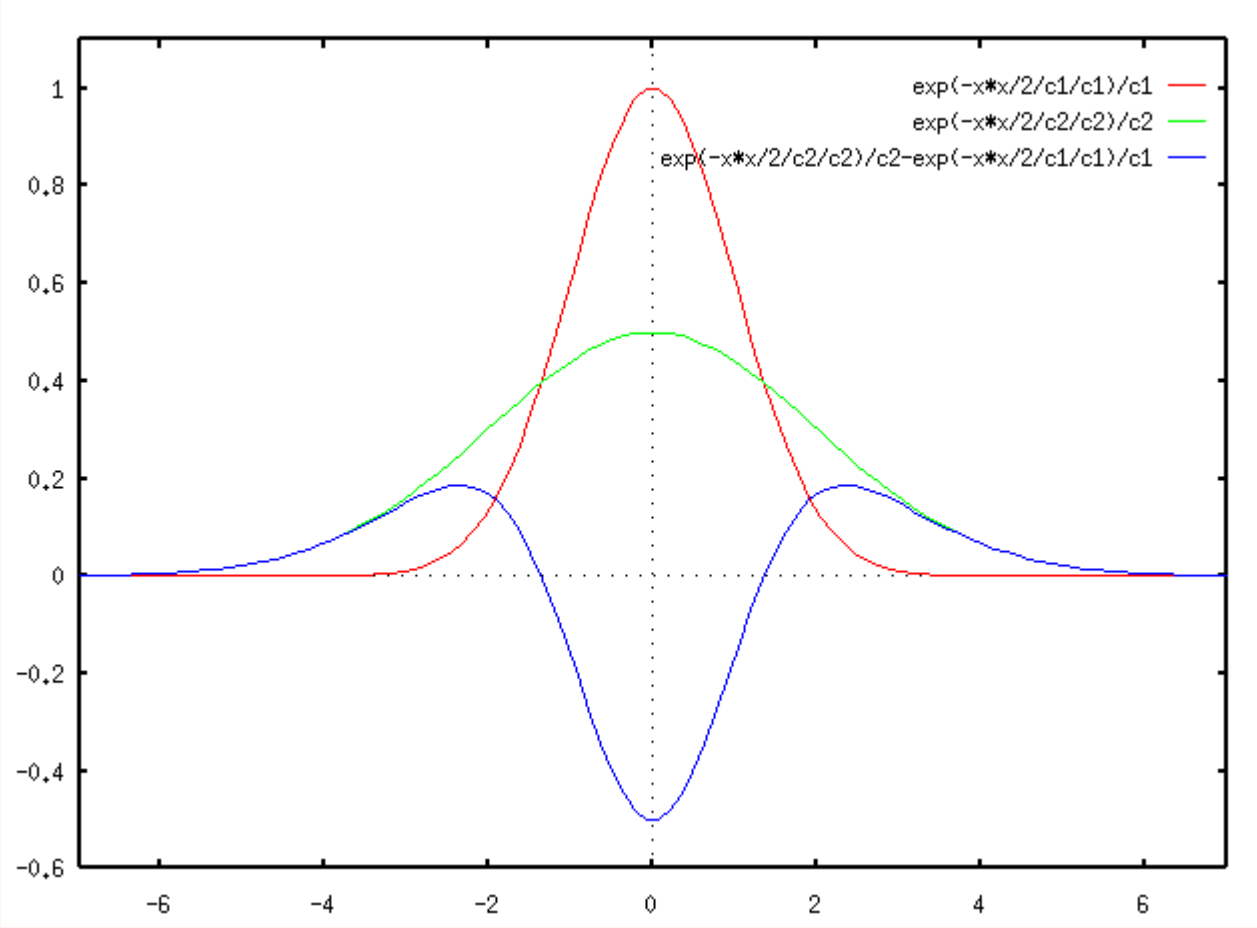
میزان هموارسازی

مثال

```
I = imread('lena.gif');
Id=im2double(I);
h1= fspecial('gaussian', [5 5], .3);
I1=imfilter(Id,h1,'replicate');
h2= fspecial('gaussian', [5 5], .5);
I2=imfilter(Id,h2,'replicate');
imshow(I1-I2,[]);
```

DoG

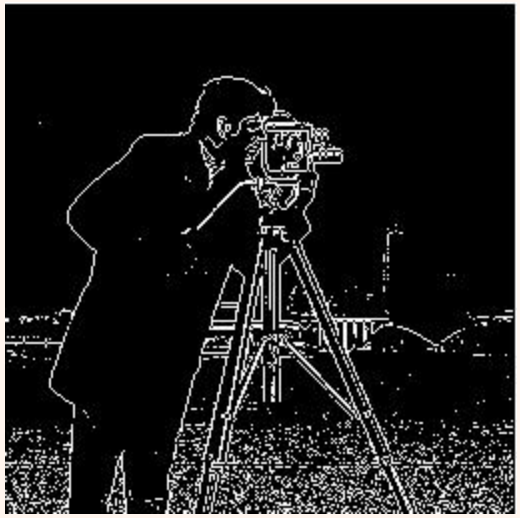
$$DoG \triangleq G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}}\left[\frac{1}{\sigma_1}e^{-(x^2+y^2)/2\sigma_1^2} - \frac{1}{\sigma_2}e^{-(x^2+y^2)/2\sigma_2^2}\right]$$

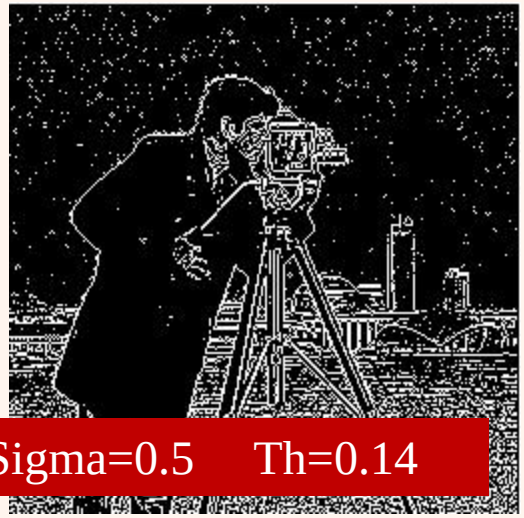


Laplacian of Gaussian

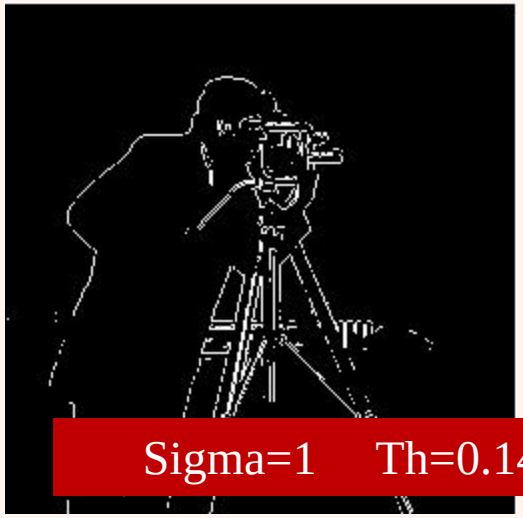
Sigma=0.5 Th=0.05

Sigma=0.5 Th=0.5





Sigma=0.5 Th=0.14



Sigma=1 Th=0.14

مقایسه

```
A = imread('wheel.png');

% Marr/Hildreth edge detection
% with threshold forced to zero
MH1 = edge(A,'log',0,1.0);
MH2 = edge(A,'log',0,2.0);
MH3 = edge(A,'log',0,3.0);
MH4 = edge(A,'log',0,4.0);

EFGH = [ MH1 MH2; MH3 MH4];
imshow(EFGH);
```

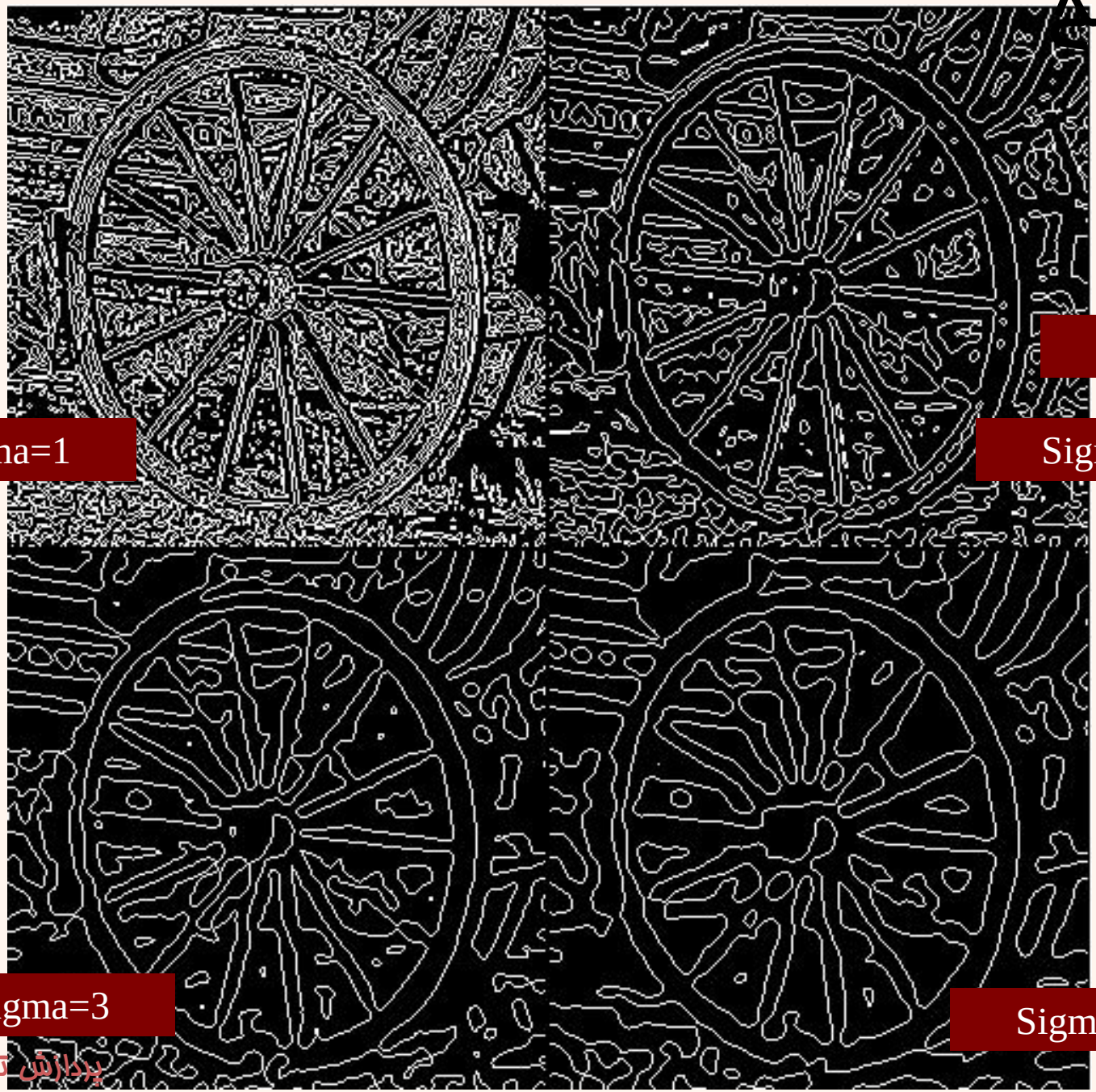
```
[C1, Ct1] = edge(A,'canny',[],1.0);
[C2, Ct2] = edge(A,'canny',[],2.0);
[C3, Ct3] = edge(A,'canny',[],3.0);
[C4, Ct4] = edge(A,'canny',[],4.0);

% Recompute lowering both
automatically computed
% thresholds by fraction k
k = 0.75
C1 = edge(A,'canny',k*Ct1,1.0);
C2 = edge(A,'canny',k*Ct2,2.0);
C3 = edge(A,'canny',k*Ct3,3.0);
C4 = edge(A,'canny',k*Ct4,4.0);

ABCD = [ C1 C2; C3 C4 ];
 imshow(ABCD);
```


https://www.cs.ubc.ca/~woodham/cpsc505/examples/edge.html

پاسخ



Marr-Hildreth

Sigma=1

Sigma=2

Sigma=3

Sigma=4

Canny

Sigma=1

Sigma=2

Sigma=3

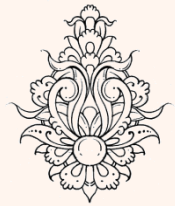
Sigma=4

روش‌های تشخیص لبه

LINKING EDGE POINTS

- معمولا لبه‌های تشخیص داده شده به دلیل نویز، تغییرات روشنایی به صورت کامل و دقیق مرزها را نشان نمی‌دهند.
- از این جهت بعد از اعمال روش‌های تشخیص لبه، اعمال الگوریتم‌هایی برای ایجاد پیوستگی در لبه‌ها ضروری است تا پیکسل‌های خروجی روش‌های تشخیص لبه به مرزهای معنادار تبدیل شوند.

linking algorithm



Using the Hough Transform

- این روش برای یافتن اشکال مختلف در تصویر به کار می‌رود.
- فرض می‌کنیم n پیکسل به عنوان لبه شناسایی شده‌اند.
- تمام خطوط ممکن بین هر دو نقطه را در نظر می‌گیریم.

تعداد خطوط ممکن

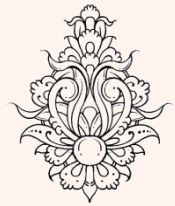
$$\frac{n(n-1)}{2} \sim n^2$$

- حال به ازای هر خط تعداد نقاطی که در آن صدق می‌کنند را به دست می‌آوریم. بدین‌ترتیب خطوطی که بیشترین نقاط(بیشترین رای) را شامل می‌شوند، انتخاب می‌شوند.

تعداد مقایسه‌ها

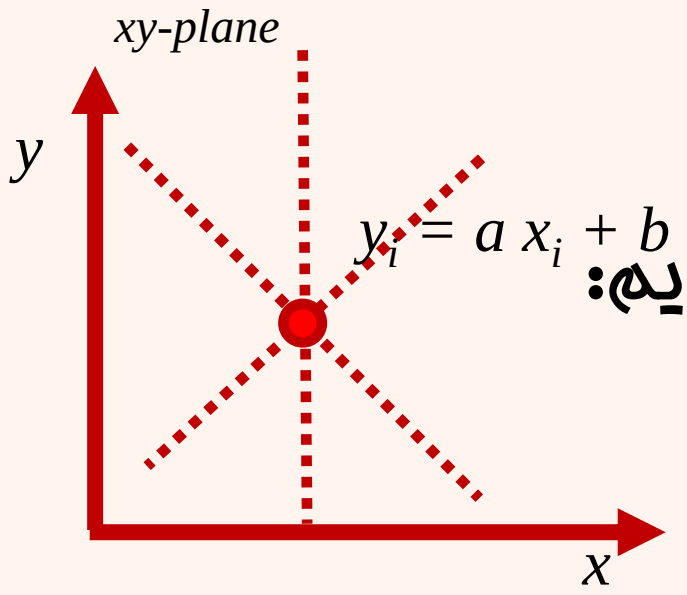
$$n \times \frac{n(n-1)}{2} \sim n^3$$

حجم محاسباتی این شیوه(↑) بازدارنده است.



Developed by Paul Hough in 1962 and patented by IBM

Hough Transform



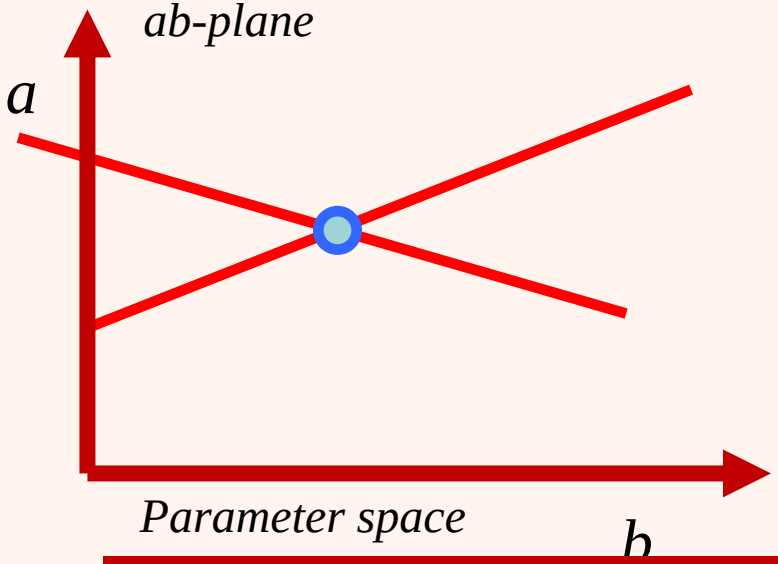
بی‌شمار خط را می‌توان در نظر گرفت که از این نقطه عبور می‌کنند،

- نقطه (j) را در نظر می‌گیریم:

با تغییر مقدار a و b این خطوط به دست می‌آیند.

با بازنویسی رابطه خط به صورت زیر

$$b = -x_i a + y_i$$



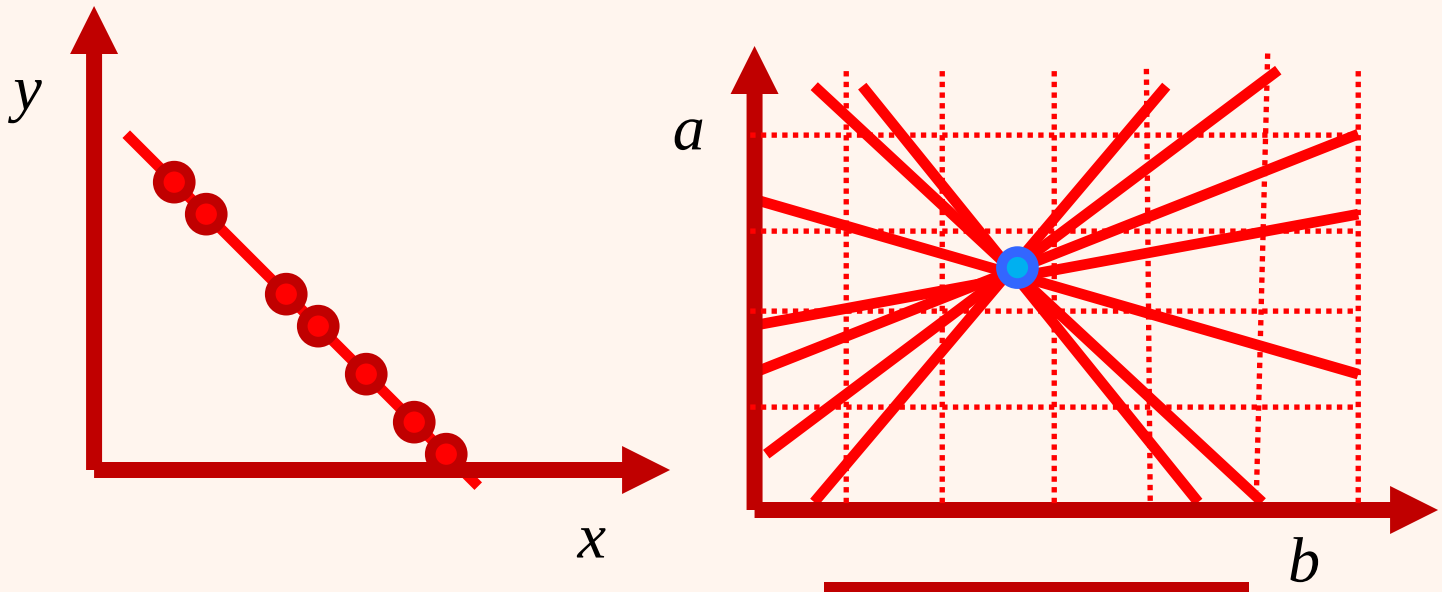
و در نظر گرفتن صفحه ab این خط شامل همه زوج‌هایی (شیب، عرض از مبدا) است که از نقطه‌ی (x_i,y_i) عبور می‌کنند.

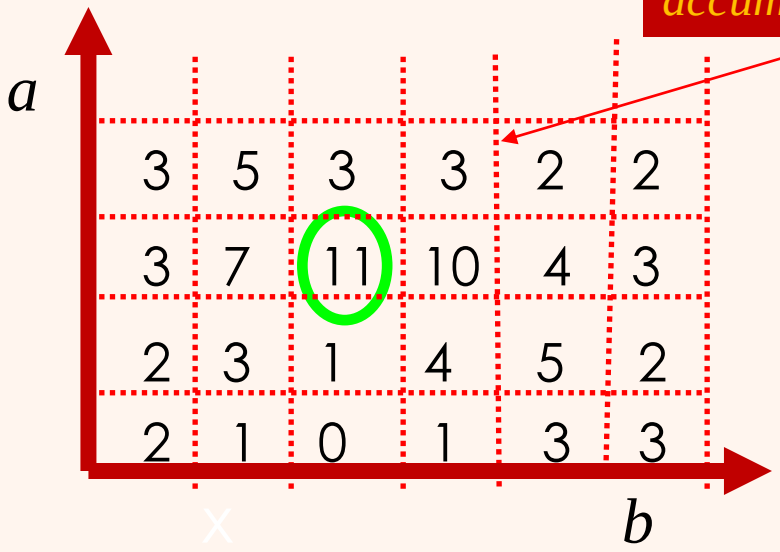
برای نقطه (x_j,y_j) به همین ترتیب

$$b = -x_j a + y_j$$

نقطه تلاقی (a',b') پارامترهای خطی است که از این دونقطه عبور می‌کنند.

Hough Transform





accumulator cells

برای کاهش حجم محاسبات صفحه‌ی ab به یک‌سری سلول تقسیم می‌شود.
این سلول‌ها در ابتدا صفر هستند.
به ازای هر (x_k, y_k) و تمام بازه‌های a مقدار b به دست آمده و به عدد معادل هر سلول یک واحد افزوده می‌شود.

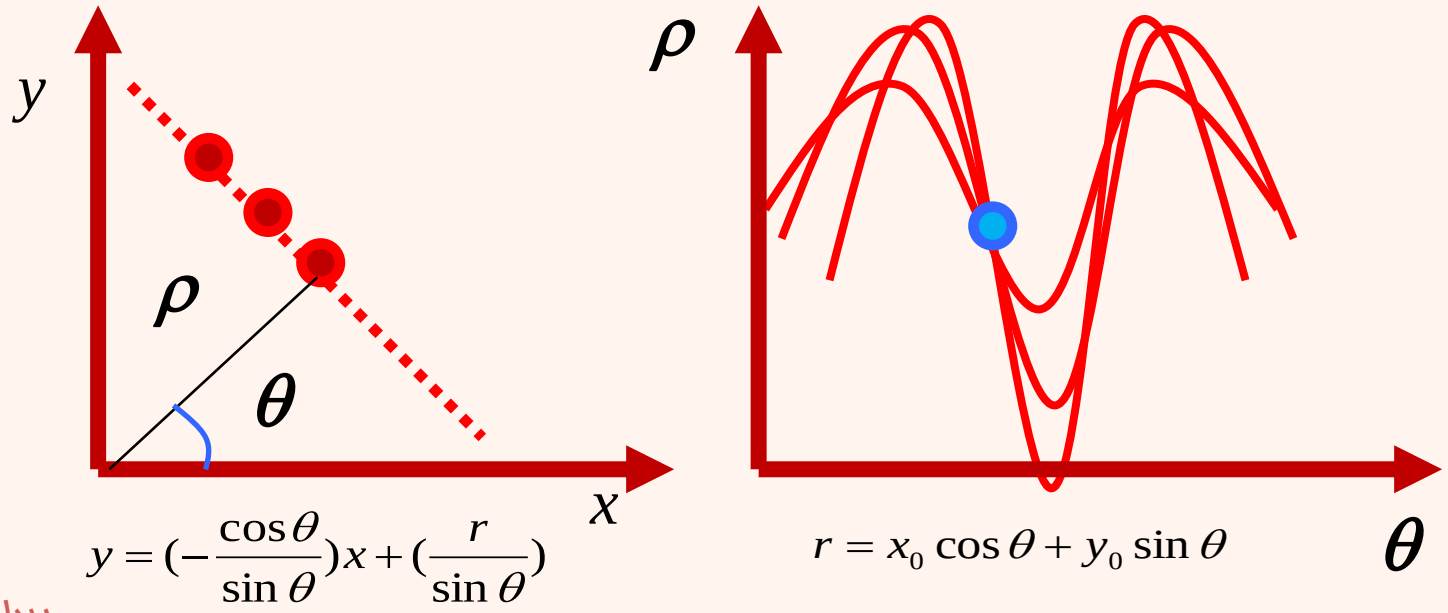
سلول انباشتگری که مقدار ماکزیمم دارد خط مورد نظر خواهد بود

Hough Transform

اگر a به k بخش تقسیم شده باشد، تعداد محاسبات به صورت زیر به دست می‌آید:

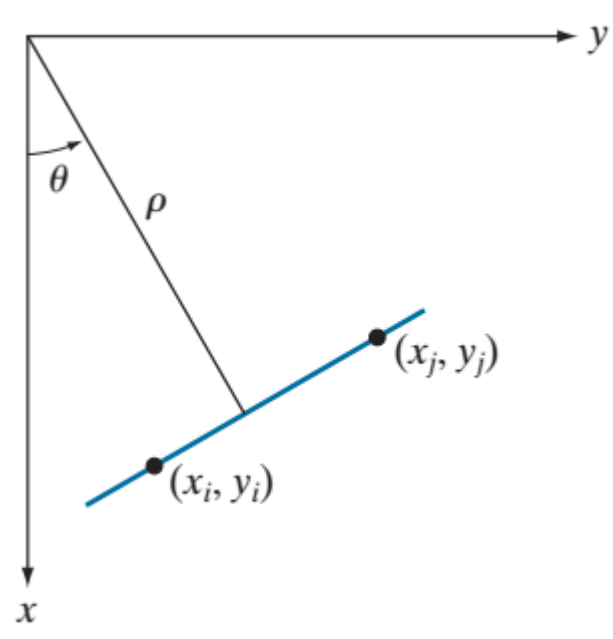
تعداد محاسبه‌ها $n \times k$

- مشکلی که این شیوه دارد این است که پارامترهای مورد استفاده نامحدود هستند، برای خطوط عمودی، شیب خط به سمت بینهایت میل می‌کند!
- برای چیرگی بر این مشکل می‌توان از مختصات قطبی استفاده کرد:

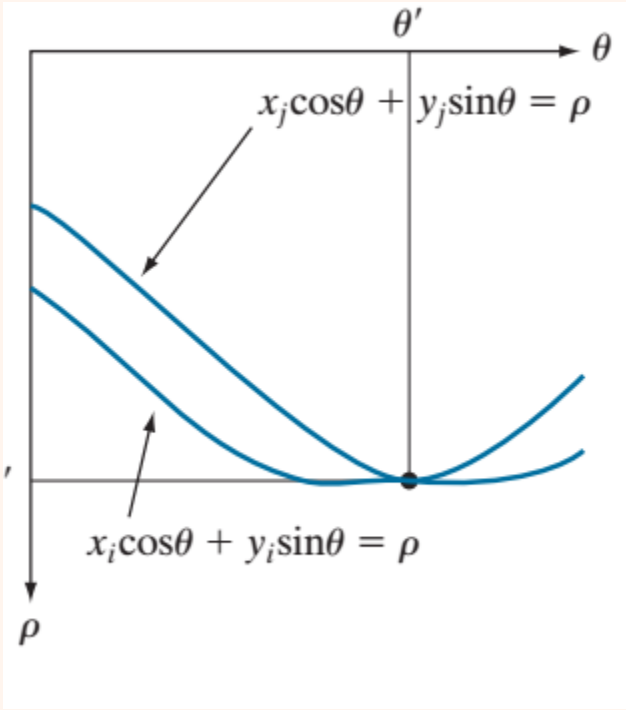


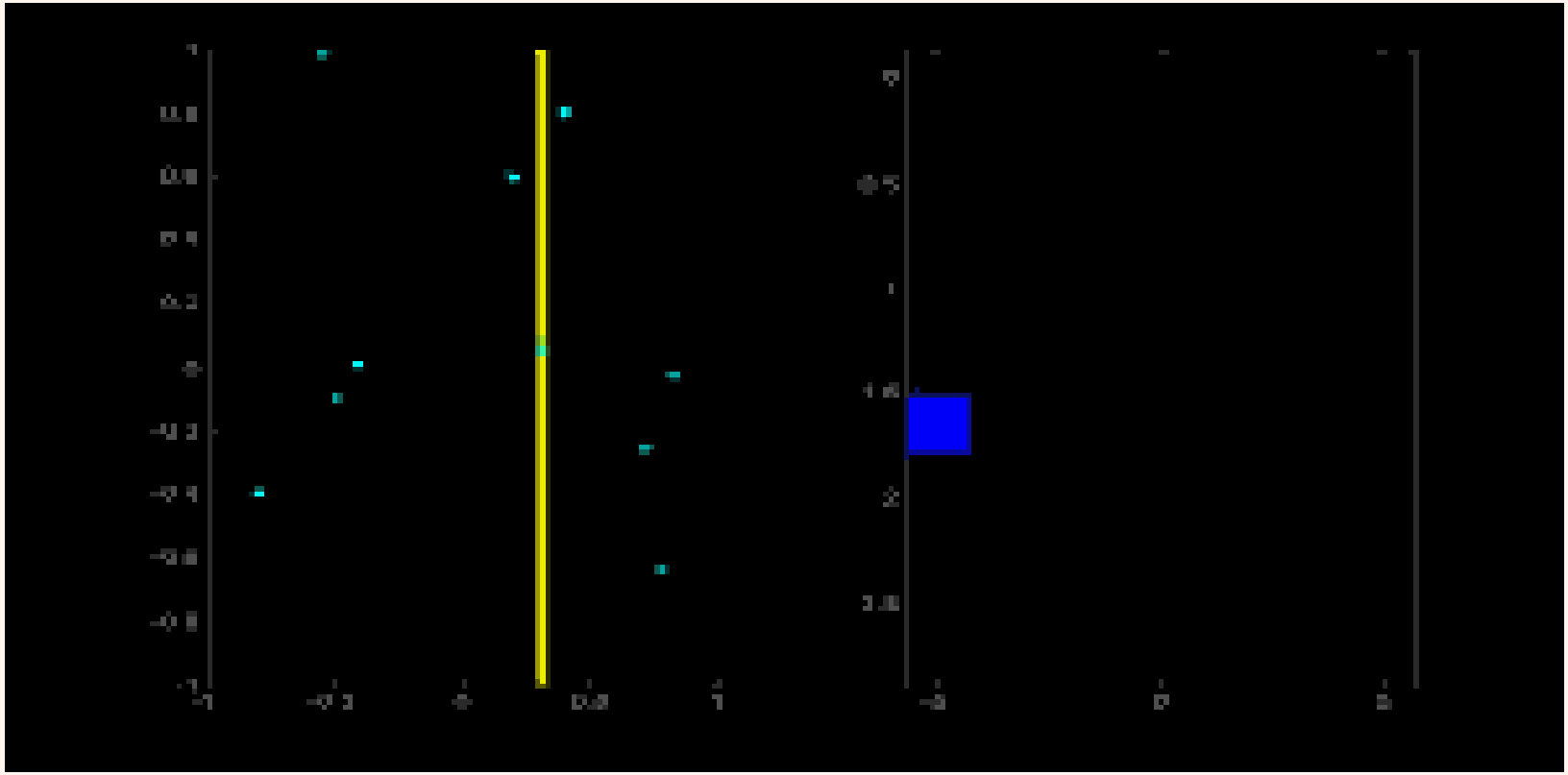
the normal representation of a line

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



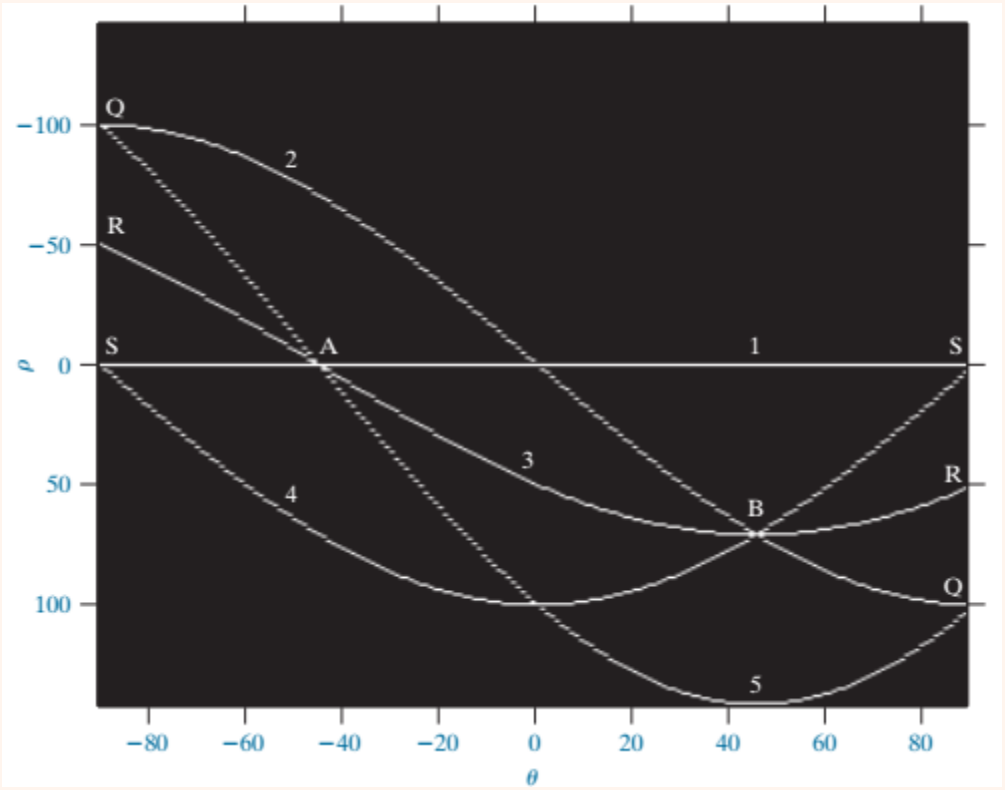
به ازای هر (x_k,y_k) و تمام بازه‌های θ مقدار ρ به دست آمده و به عدد معادل هر سلول یک واحد افزوده می‌شود.





https://docs.opencv.org/3.4/d3/de6/tutorial_js_houghlines.html

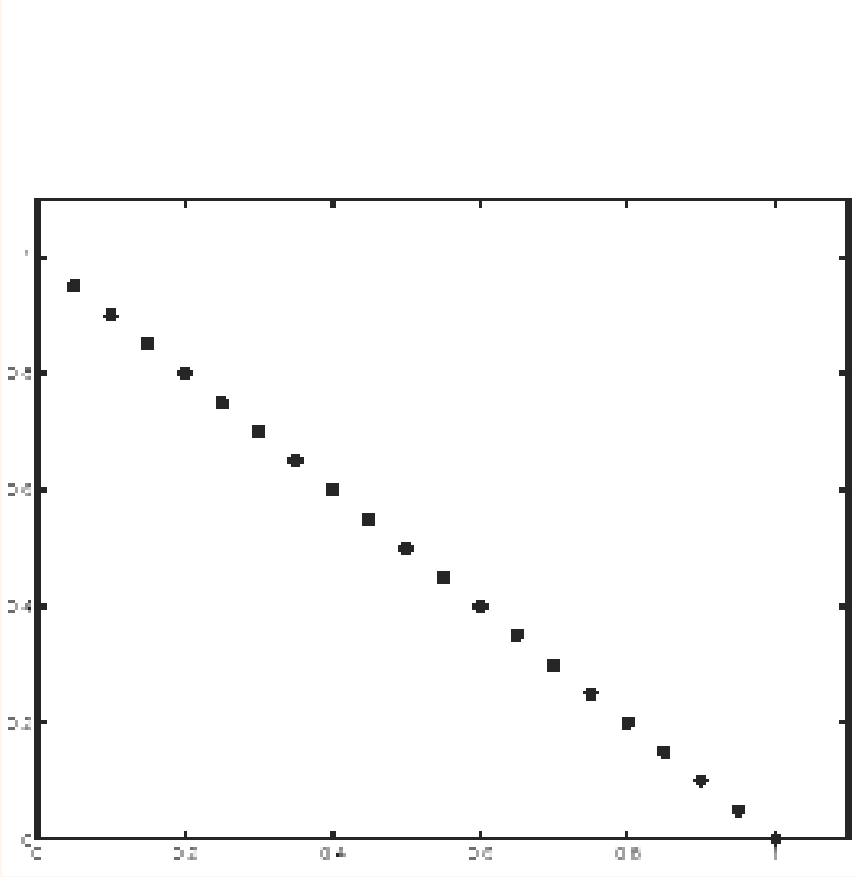
مثال



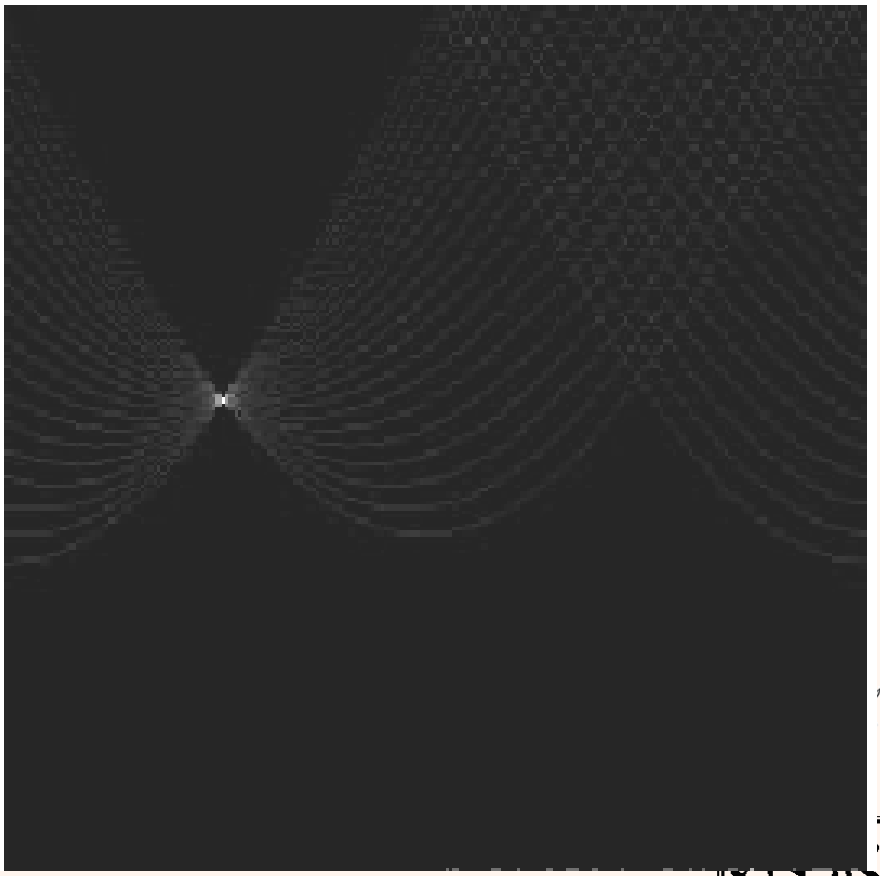
a
b

FIGURE 10.30
(a) Image of size 101×101 pixels, containing five white points (four in the corners and one in the center). (b) Corresponding parameter space.

مثال

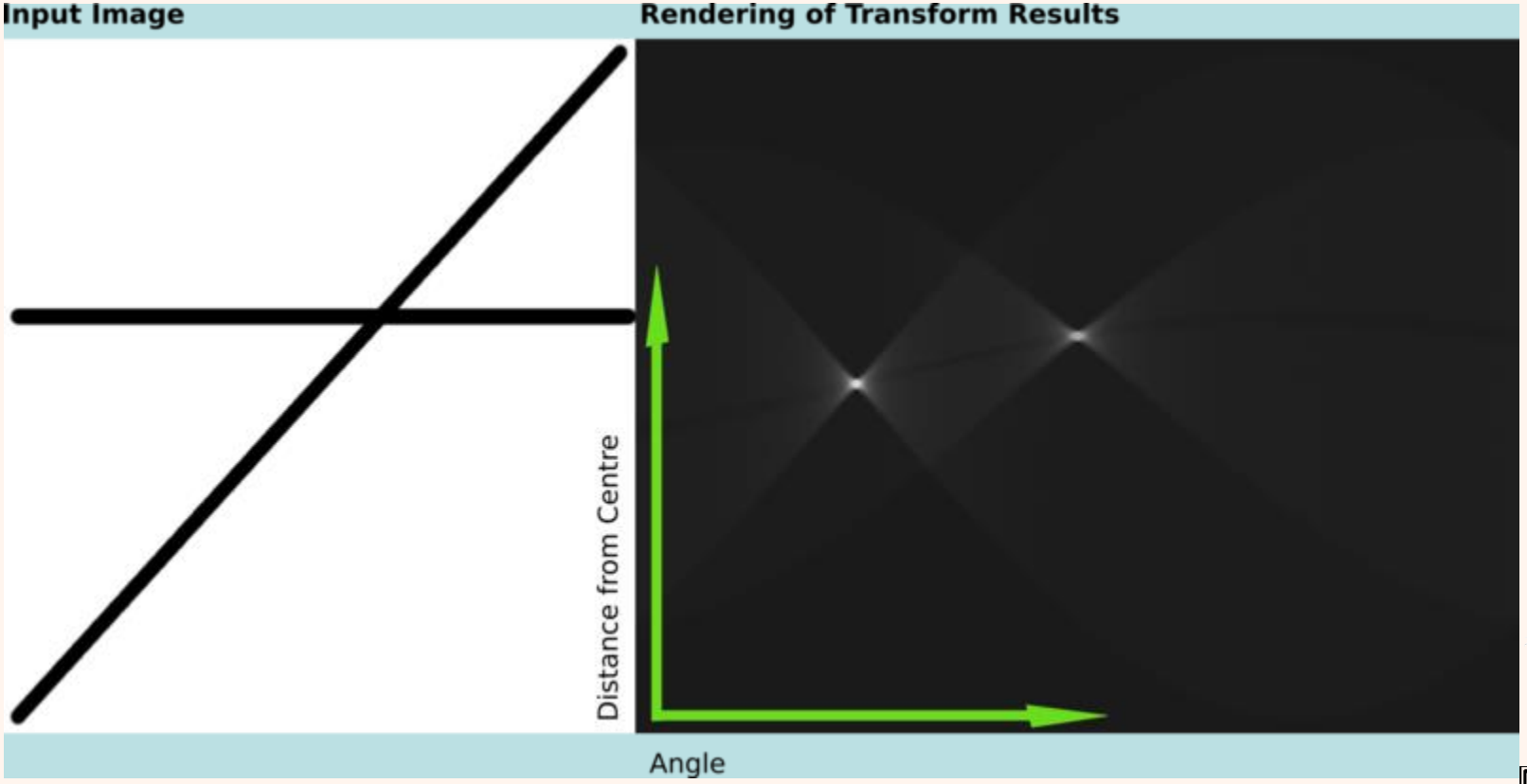


features

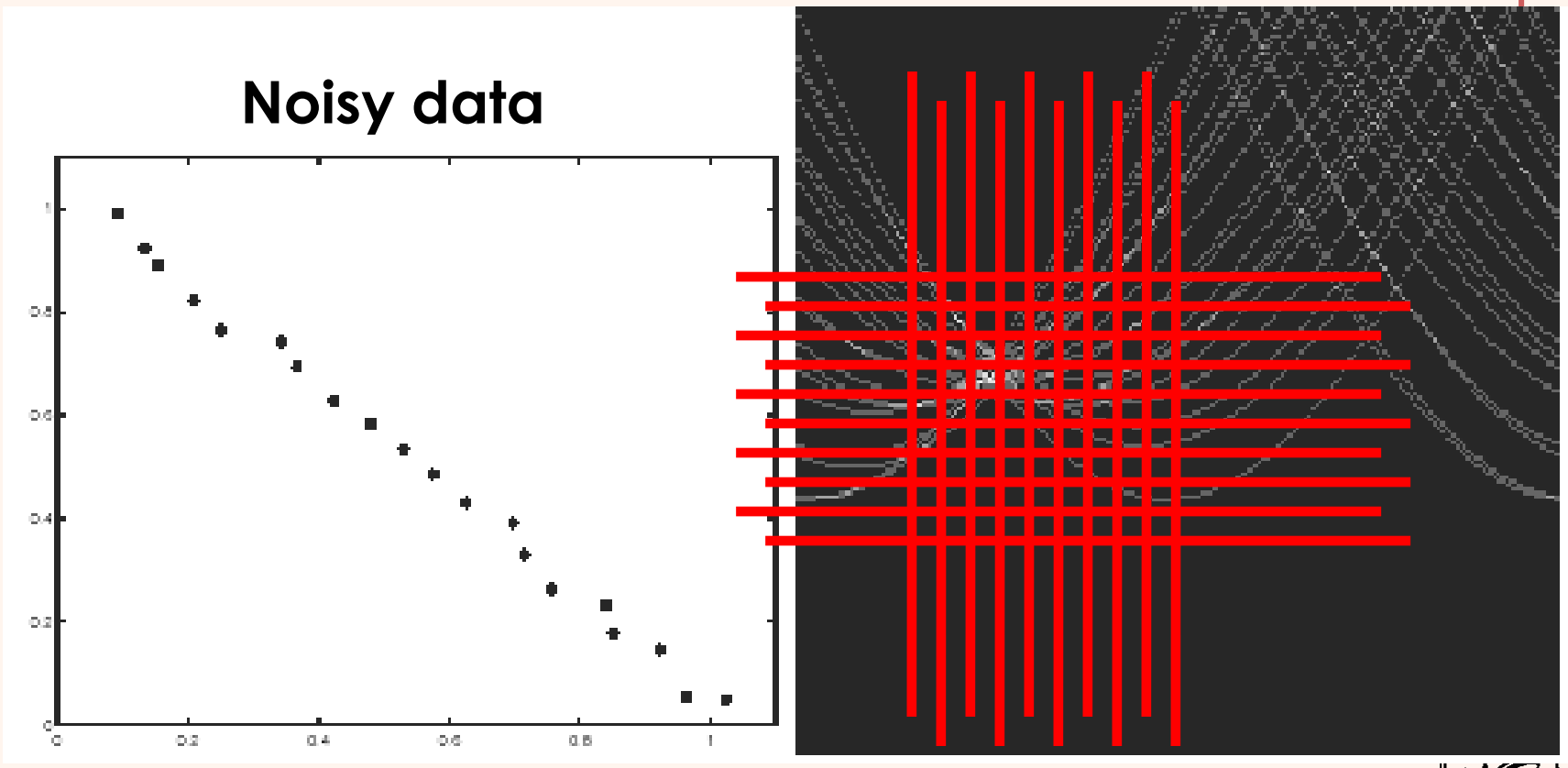


votes

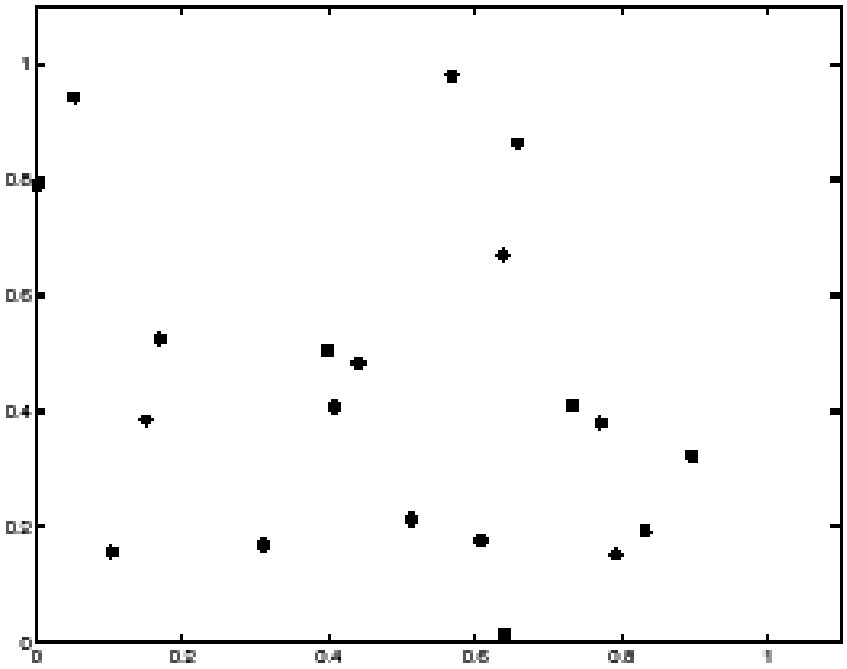
مثال


Input Image
Rendering of Transform Results
Distance from Centre
Angle

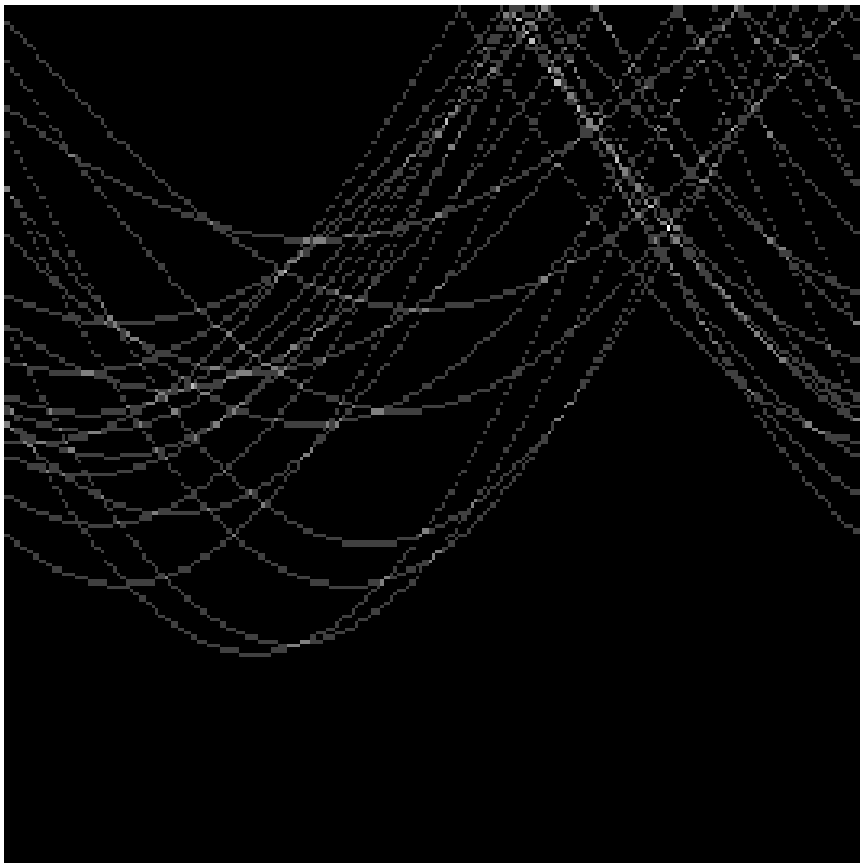
مثال



مثال



features



votes

Hough Transform

1. Image → Canny

مثال

Hough Transform

2. Canny → Hough votes

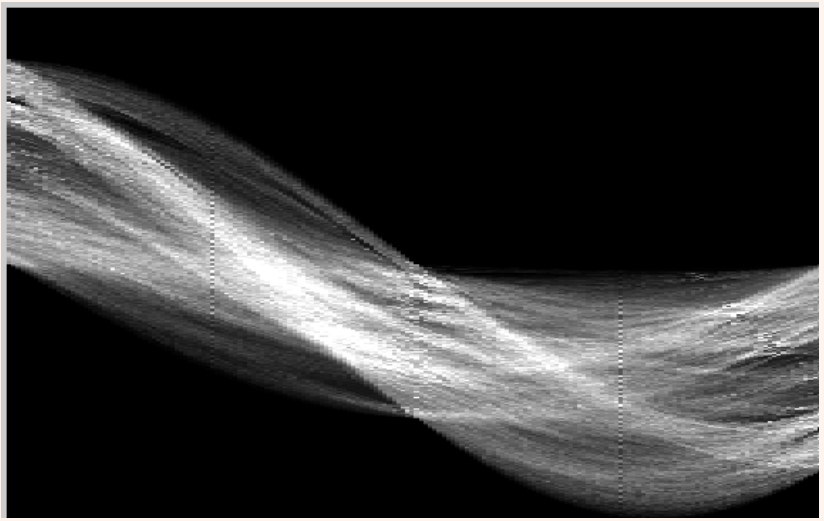
مثال

Hough Transform

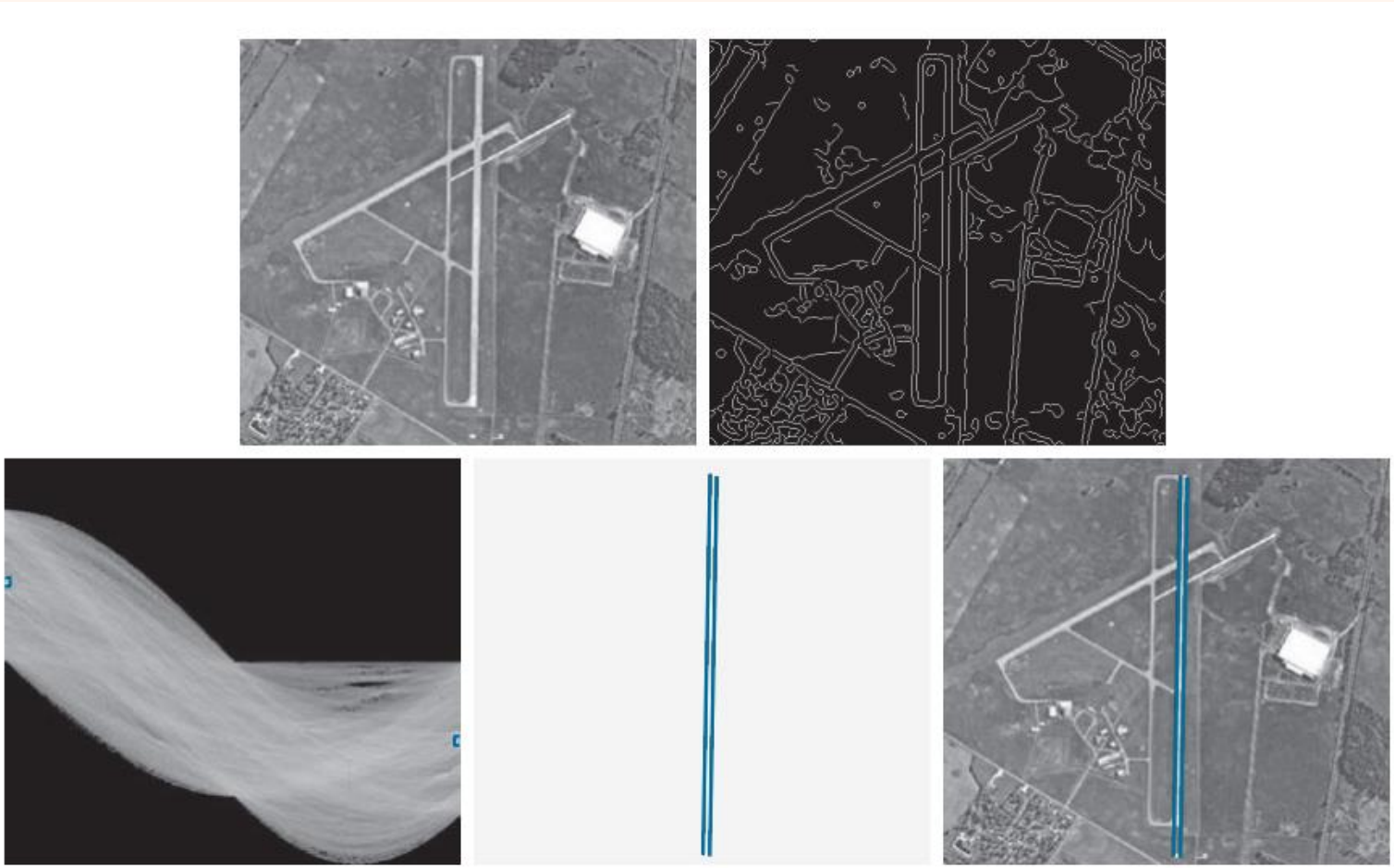
3. Hough votes → Edges

مثال

Find peaks and post-process



مثال



a b
c d e

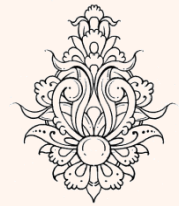
FIGURE 10.31 (a) A 502×564 aerial image of an airport. (b) Edge map obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes. (e) Lines superimposed on the original image.

تشخیص سایر اشکال

- برای تشخیص هر منحنی دیگری که به فرم $g(v, c) = 0$ نوشته شود می‌توان از شیوه استفاده کرد. در این عبارت v مختصات و c ضرایب را نشان می‌دهد.
- به عنوان مثال برای تشخیص دایره

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

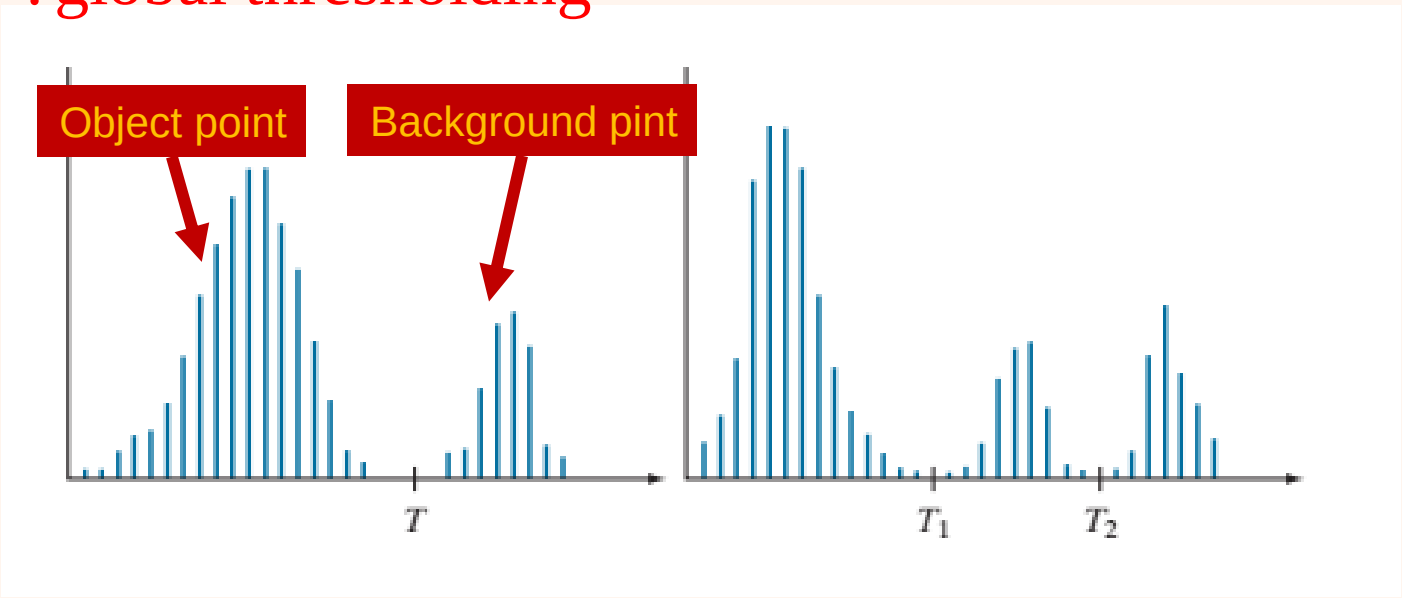
- در اینجا سلول‌های انباشتگر در فضای سه بعدی هستند، به ازای مقادیر c_1 و c_2 مقدار c_3 محاسبه و رای سلول متناظر یک واحد افزایش می‌یابد.



آستانه‌گذاری

$$g(x,y)=\begin{cases}1 & \text{if } f(x,y)>T \quad \text{(object point)}\\ 0 & \text{if } f(x,y)\le T \quad \text{(background point)}\end{cases}$$

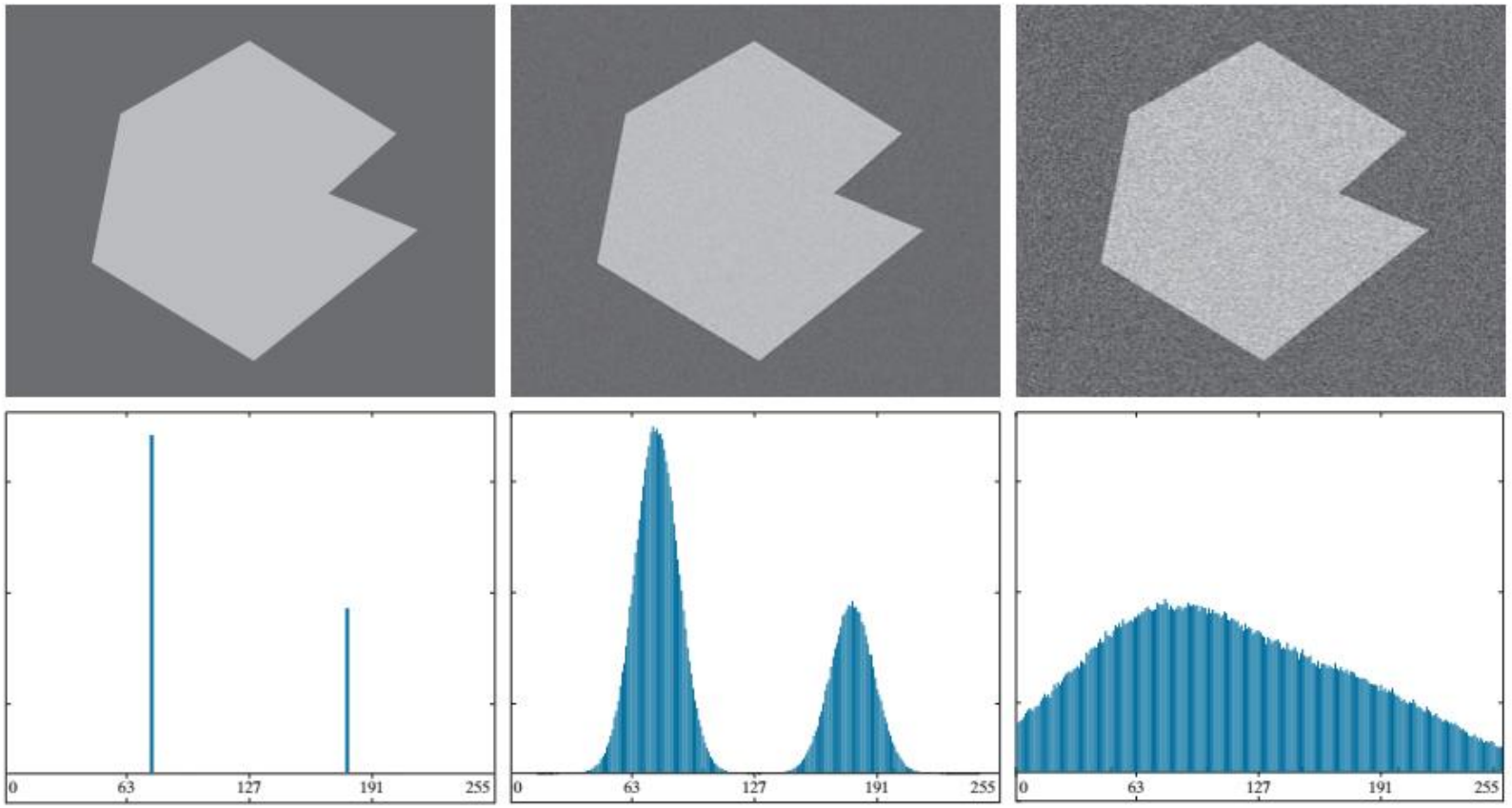
T : global thresholding



Multiple thresholding

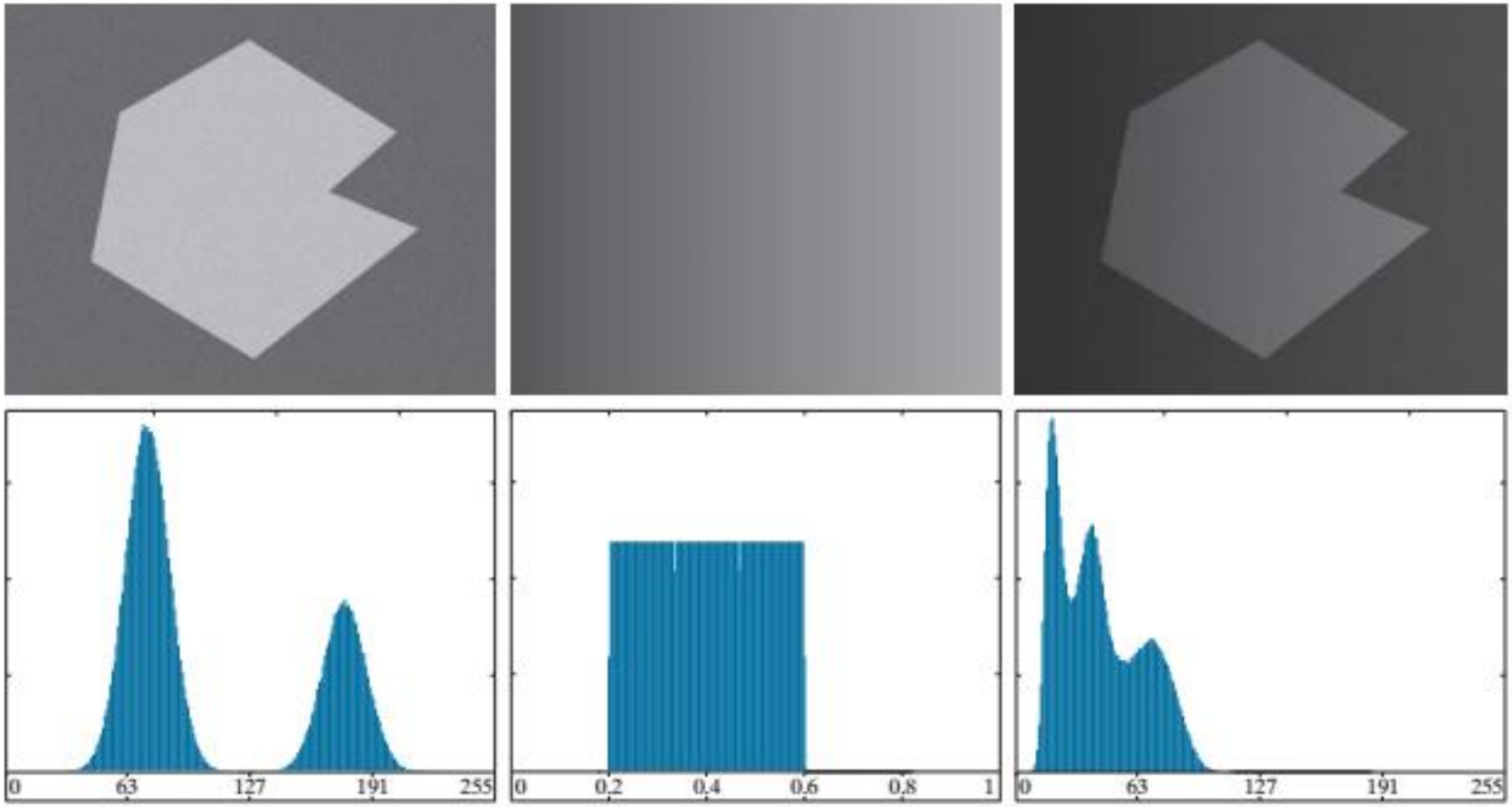
$$g(x,y)=\begin{cases}a & \text{if } f(x,y)>T_2\\ b & \text{if } T_1<f(x,y)\le T_2\\ c & \text{if } f(x,y)\le T_1\end{cases}$$

تاثیر نویز



a b c
d e f

FIGURE 10.33 (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) through (f) Corresponding histograms.



a b c
d e f

FIGURE 10.34 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d) through (f) Corresponding histograms.

الگوریتم پایه حد آستانه عمومی

۱
- یک حد آستانه T به عنوان تخمین اولیه در نظر گرفته شود.

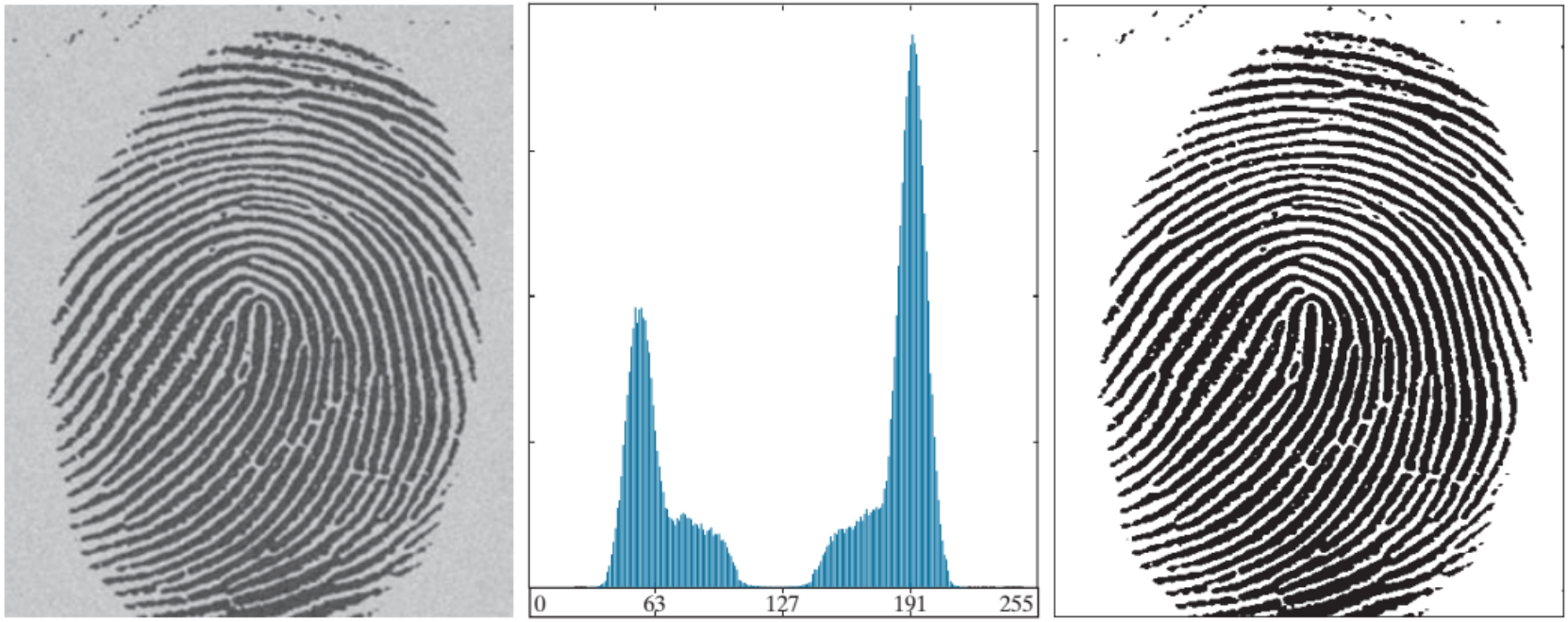
۲
- با استفاده از این حد آستانه پیکسل‌ها به دو گروه تقسیم می‌شوند: G_1 و G_2

۳
- مقدار میانگین روشنایی پیسکل‌های هر گروه محاسبه شود: m_1 و m_2

۴
- حد آستانه جدید به صورت زیر محاسبه شود:

$$T = \frac{m_1 + m_2}{2}$$

۵
- گام‌های ۲ تا ۴ تا رسیدن به همگرایی(زمانی که اختلاف T در دو گام متوالی از مقداری از پیش‌تعیین شده کم‌تر باشد.)



a b c

FIGURE 10.35 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (thin image border added for clarity). (Original image courtesy of the National Institute of Standards and Technology.).

Otsu's Method (1979)

- ایده اصلی این روش بخش‌بندی تصویر به دو دسته است به گونه‌ای که واریانس بین دسته‌ای ماکزیمم شود.

$$m_1(k) = \sum_{i=0}^{k} iP(i/C_1) = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i$$

میانگین پیکسل‌های گروه ۱

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i/C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

میانگین پیکسل‌های گروه ۲

$$P_1 m_1 + P_2 m_2 = m_G$$

میانگین روشنایی تصویر

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

$$= P_1 P_2 (m_1 - m_2)^2$$

$$= \frac{\left[m_G P_1 - m_1 P_1\right]^2}{P_1(1-P_1)}$$

$$= \frac{\left[m_G P_1 - m\right]^2}{P_1(1-P_1)}$$

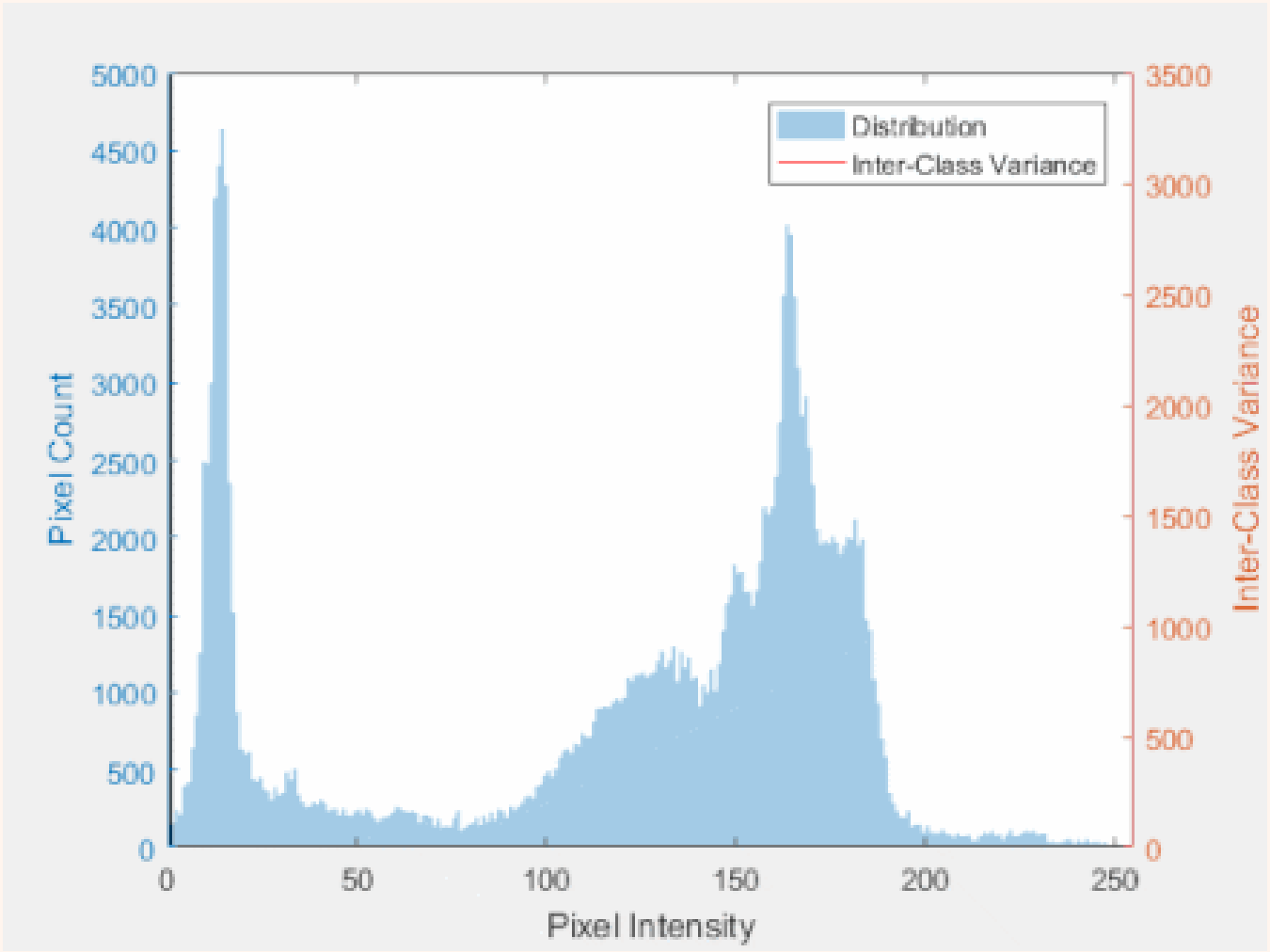
واریانس بین کلاسی

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

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > k^* \\ 0 & \text{if } f(x,y) \le k^* \end{cases}$$

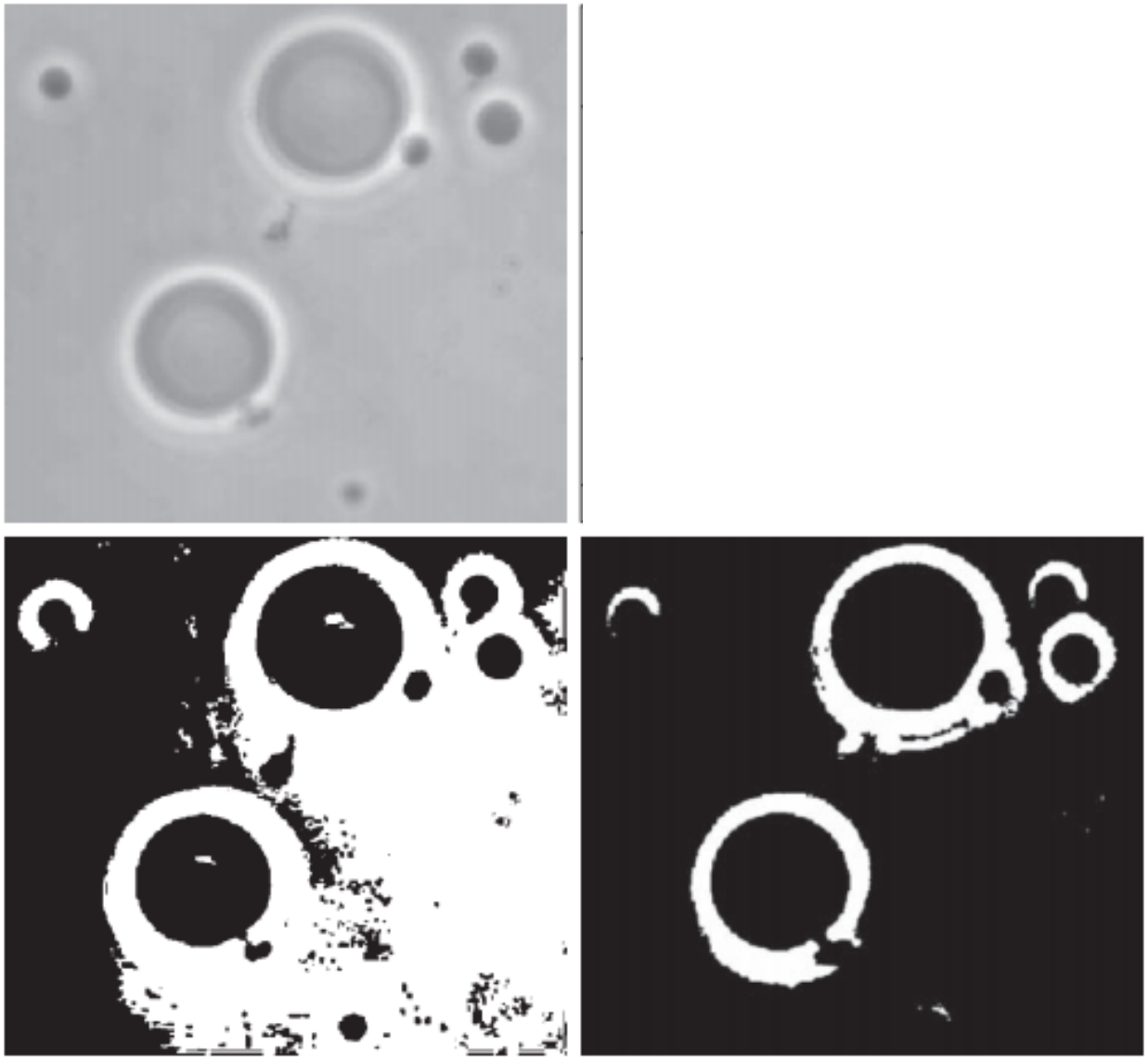
$$\text{Separability measure} \quad \eta = \frac{\sigma_B^2}{\sigma_G^2}$$

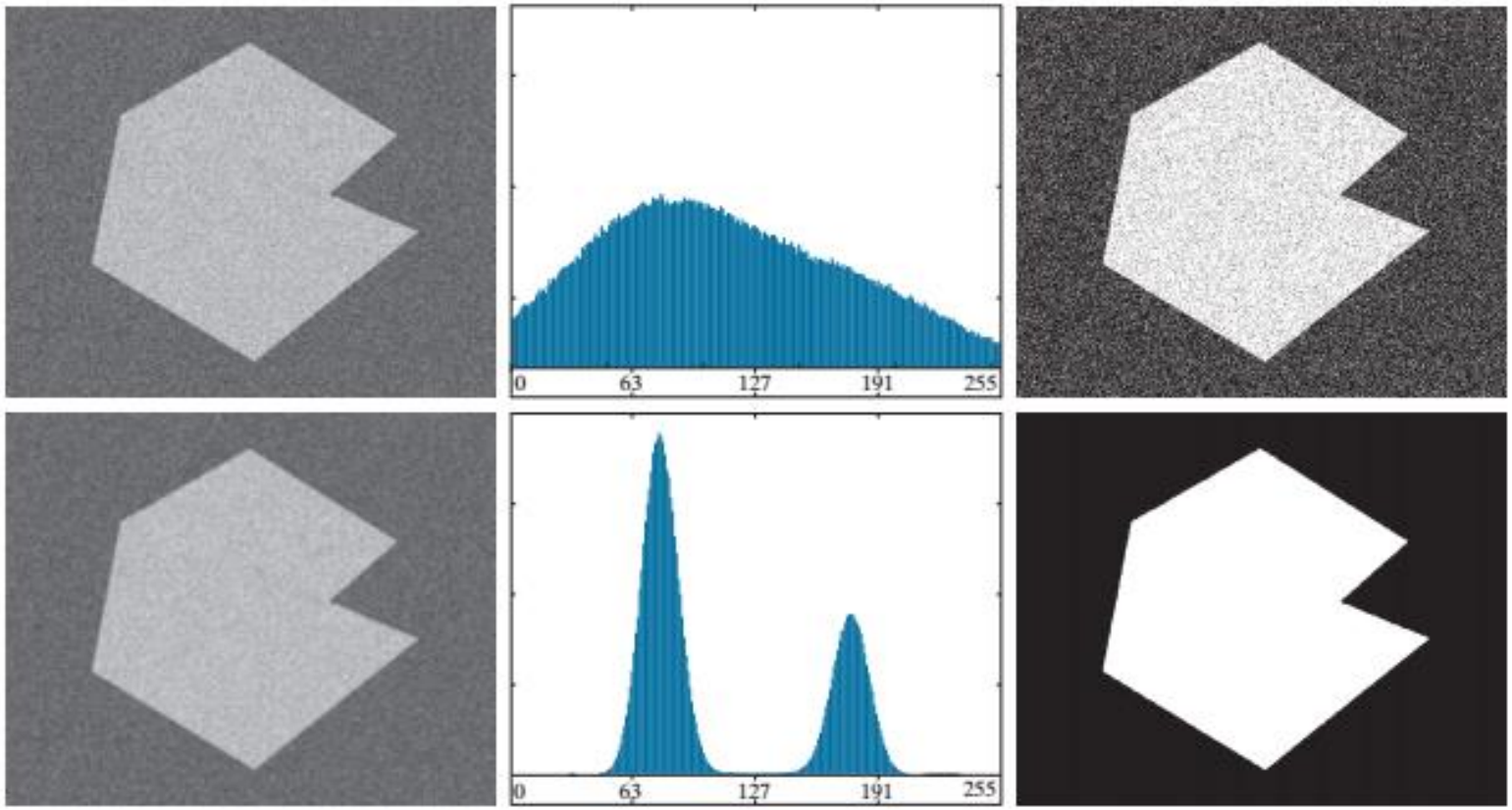
مثال


Distribution
Inter-Class Variance
Pixel Count
Inter-Class Variance
Pixel Intensity
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
0
50
100
150
200
250

a b
c d

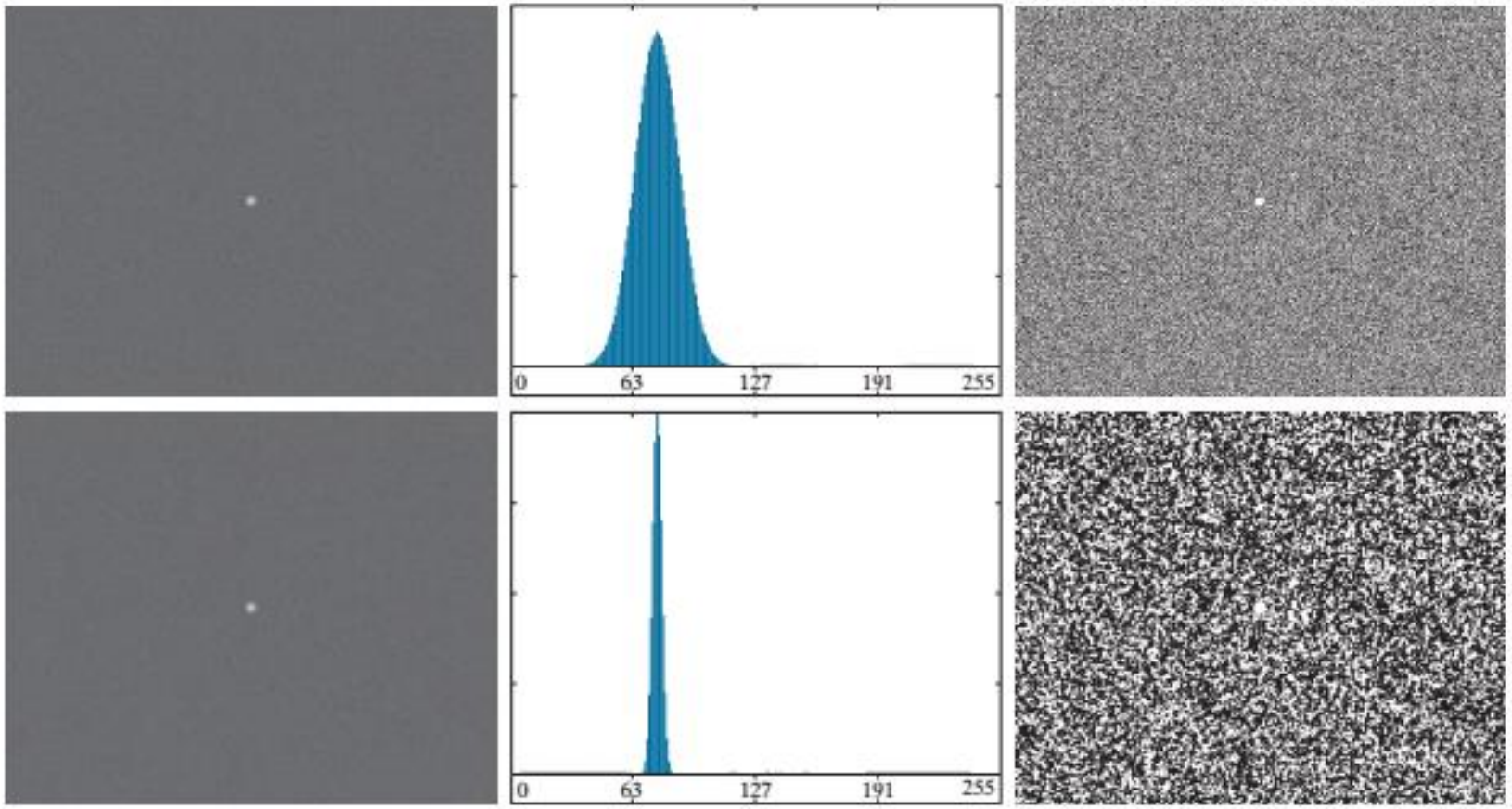
FIGURE 10.36
(a) Original image.
(b) Histogram (high peaks were clipped to highlight details in the lower values).
(c) Segmentation result using the basic global algorithm from Section 10.3.
(d) Result using Otsu's method.
(Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)





a b c
d e f

FIGURE 10.37 (a) Noisy image from Fig. 10.33(c) and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging kernel and (e) its histogram. (f) Result of thresholding using Otsu's method.

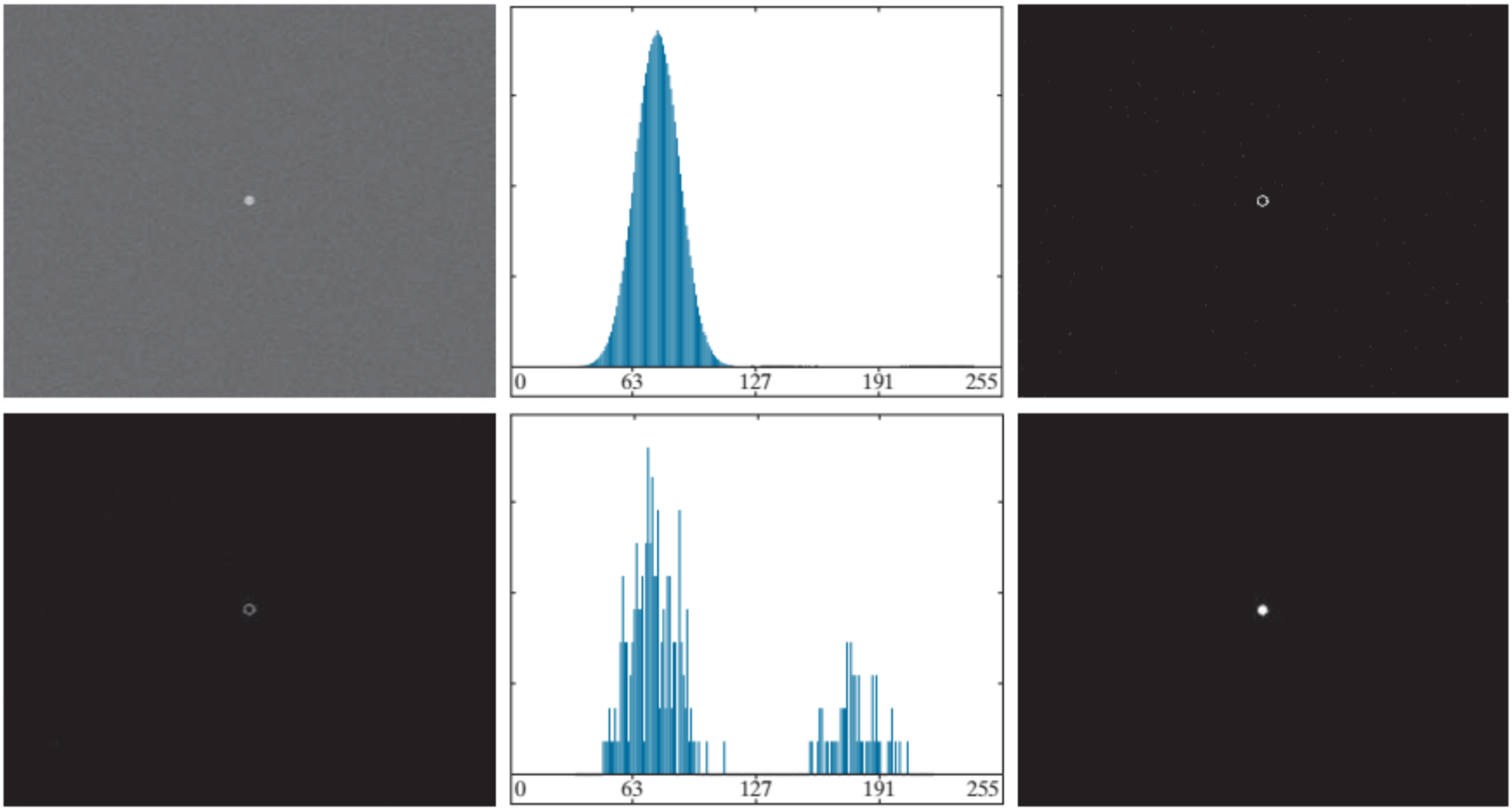


a b c
d e f

FIGURE 10.38 (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5×5 averaging kernel and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases to extract the object of interest. (See Fig. 10.39 for a better solution.)

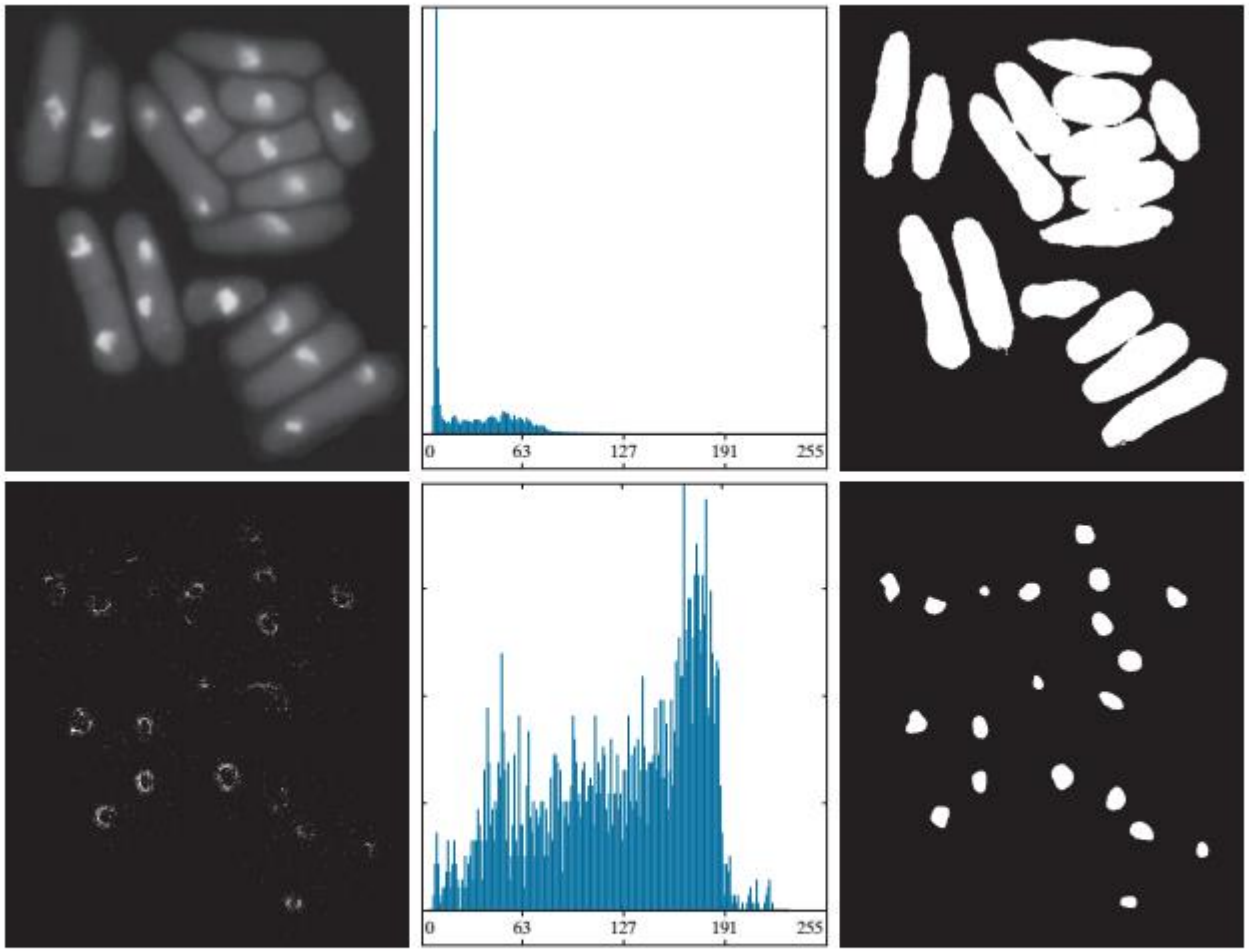
استفاده از اطلاعات لبه

- برای غلبه بر مشکل کوچک بودن ابعاد شی می‌توان تنها از اطلاعات مربوط به لبه‌های و پیکسل‌های اطراف آن استفاده کرد.



a b c
d e f

FIGURE 10.39 (a) Noisy image from Fig. 10.38(a) and (b) its histogram. (c) Mask image formed as the 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.



a b c
d e f

IGURE 10.40 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the istogram in (b). (d) Mask image formed by thresholding the absolute Laplacian image. (e) Histogram of the non-ero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram n (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)