DIGITALNA OBRADA SLIKE

ČAS 7 – DETEKCIJA IVICA, LINIJA, ĆOŠKOVA; SEGMENTACIJA SLIKE

Kako detektovati ivice?

Ideja:

Nagle promene intenziteta u slici se mogu detektovati diferencijalnim operatorima prvog i drugog reda

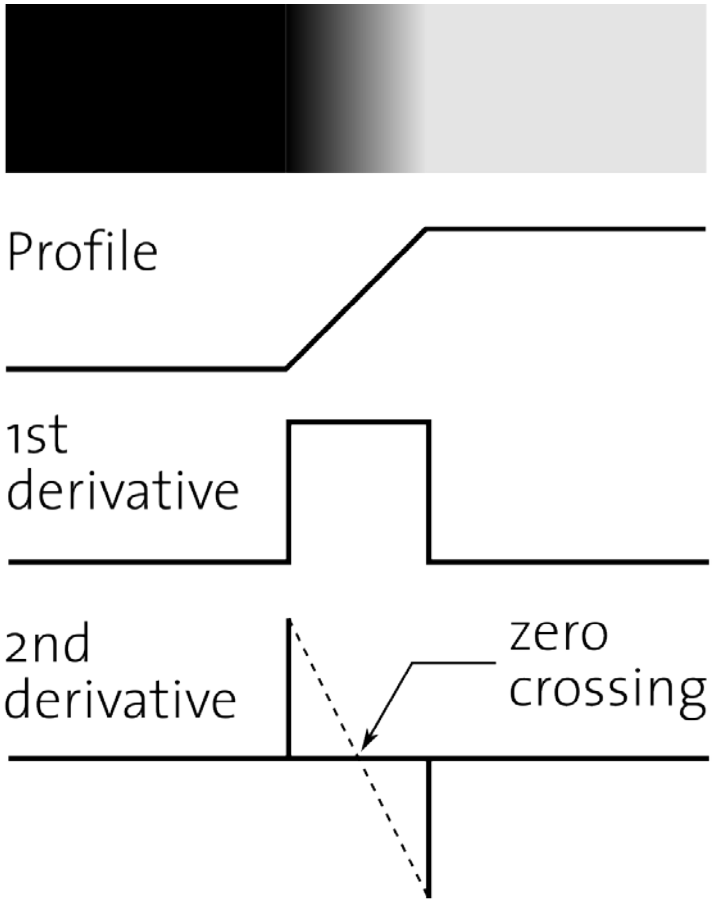
$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

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

$$= f(x+2) - f(x+1) - f(x+1) + f(x) =$$

$$= f(x+2) - 2f(x+1) + f(x)$$

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



Osnovni koraci pri detekciji ivica

1) **Niskofrekventno filtriranje**

Potrebno je kako bi se potisnuo šum i nerelevantni sitni detalji

2) **Detekcija ivičnih tačaka**

Bazira se na pronalaženju tačaka u čijoj okolini postoji značajna promena intenziteta. U ovom koraku se izdvajaju tačke koje potencijalno pripadaju ivici

3) **Lokalizacija ivice**

Od svih kandidata za ivične piksele biraju se samo oni koji najverovatnije pripadaju ivici

Detekcija ivica bazirana na gradijentu

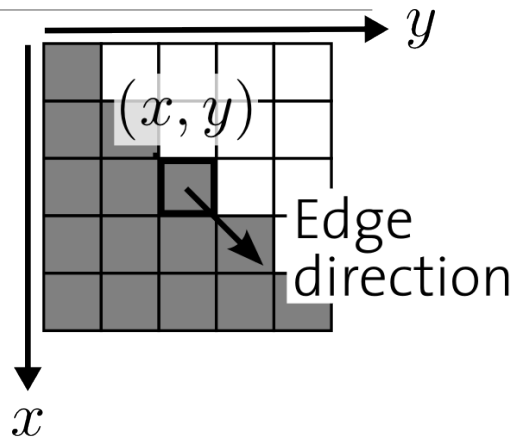
Gradijent slike sadrži informaciju o pravcu i amplitudi maksimalne promene intenziteta za svaki piksel u slici pa se stoga može iskoristiti za detekciju ivica

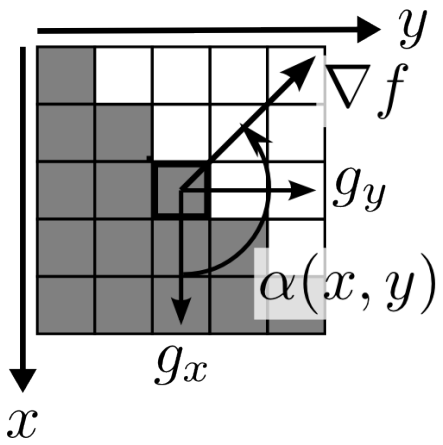
$$G[f(x,y)] = \nabla f(x,y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (f_x, f_y)$$

$$\theta = \arctan\left(\frac{f_y}{f_x}\right)$$

$$|\nabla f| = \sqrt{f_x^2 + f_y^2}$$

$$|\nabla f| = |f_x| + |f_y|$$





Najčešći operatori bazirani na gradijentu

Robertsov operator

$$h_x[m,n] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$h_y[m,n] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Pruit operator

$$h_x[m,n] = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_y[m,n] = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Sobelov operator

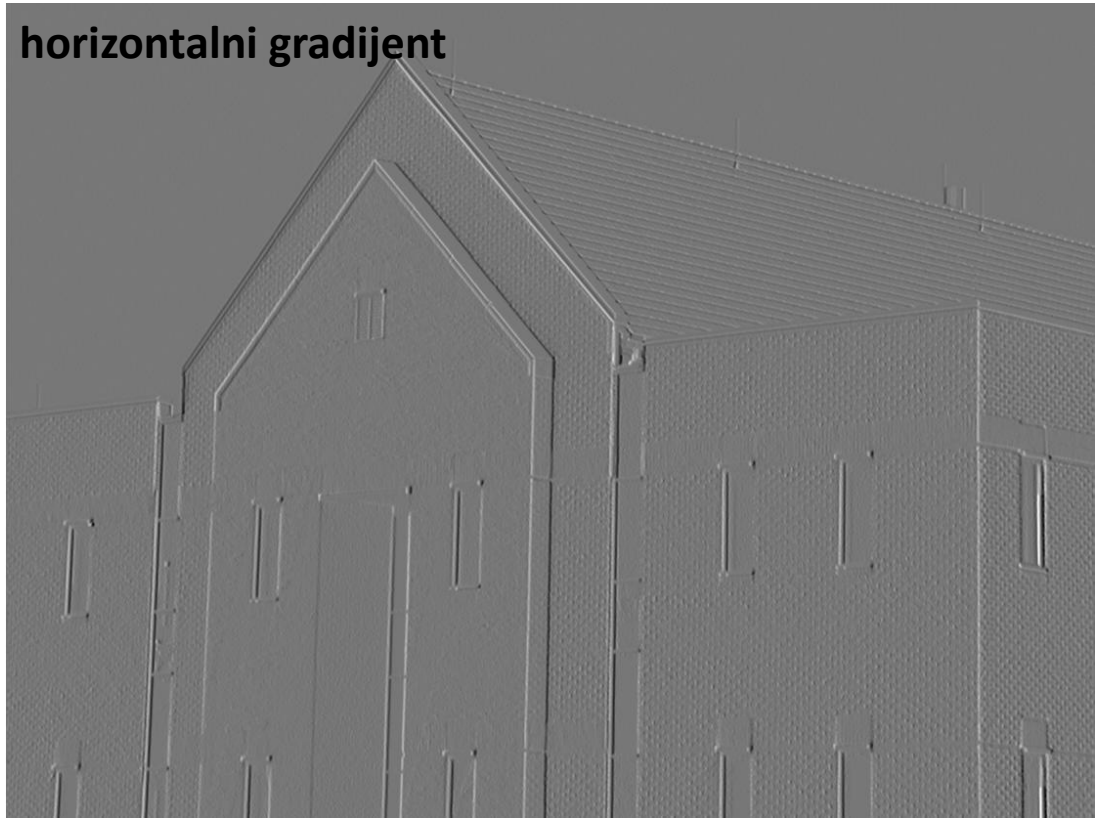
$$h_x[m,n] = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

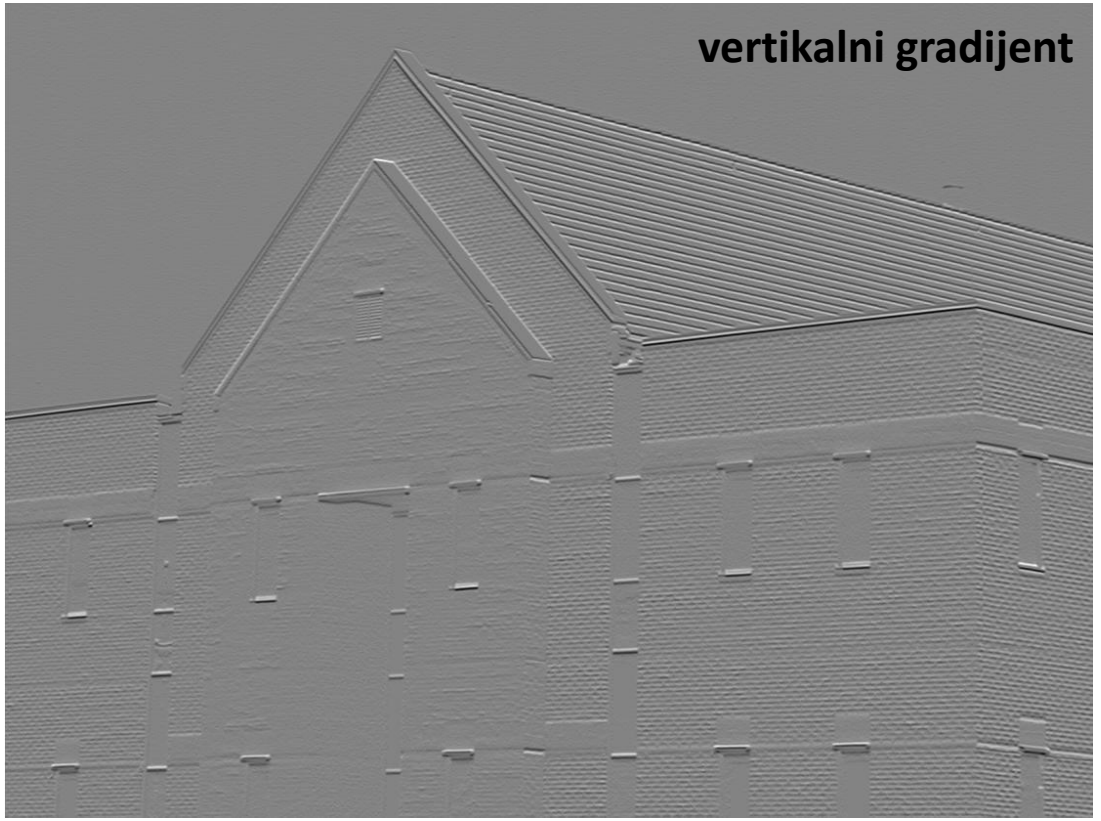
$$h_y[m,n] = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Sobelov operator - primer

```
I = imread('building.tif');
I = double(I)/255;

Hx = [-1 -2 -1; 0 0 0; 1 2 1]
Hy = Hx'

Gx = imfilter(I, Hx, 'replicate', 'same');
Gy = imfilter(I, Hy, 'replicate', 'same');

figure('Name', 'Vertikalni gradijent'); imshow(Gx, []);
figure('Name', 'Horizontalni gradijent'); imshow(Gy, []);

Gm = (Gx.^2 + Gy.^2).^0.5;
Ga = atan(Gy./Gx);
Ga(Gx == 0) = 0;

figure('Name', 'Magnituda gradijenta'); imshow(Gm, []);
figure('Name', 'Ugao gradijenta'); imagesc(Ga);
```

Sobelov operator - primer

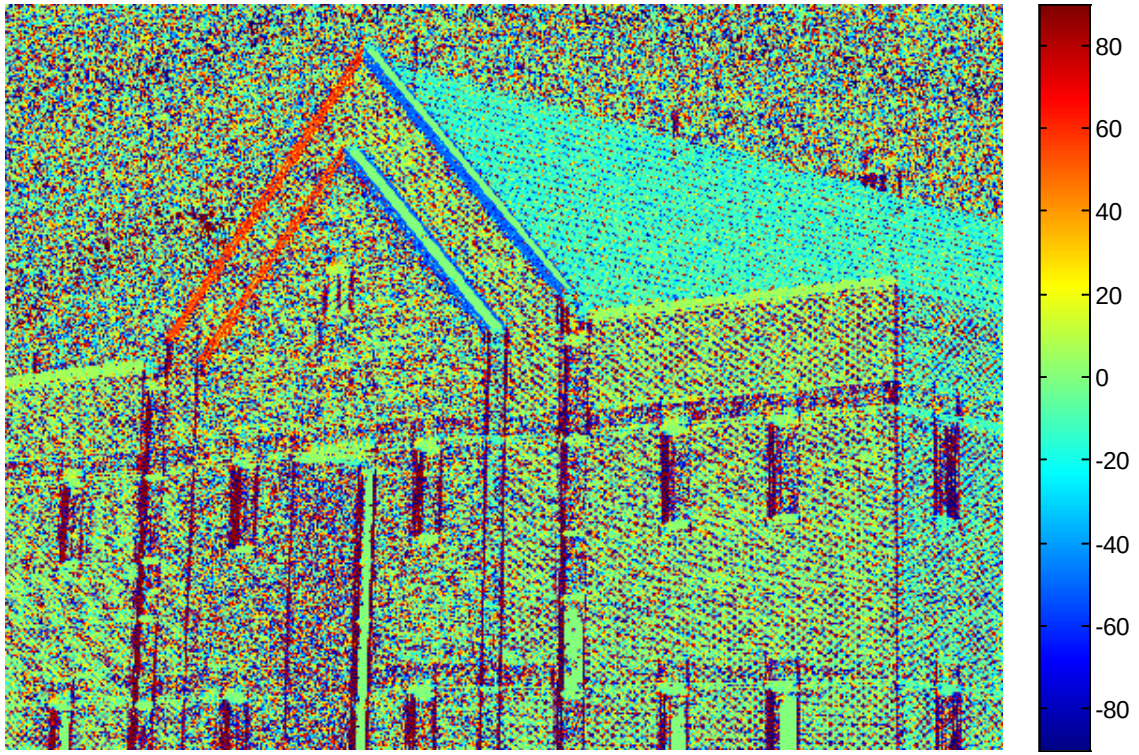




Sobelov operator - primer

magnituda gradijenta

ugao gradijenta



Kombinacija sa NF filtrom

magnituda gradijenta

magnituda gradijenta slike isfiltrirane Gausovim filtrom 9x9, σ=5

Detekcija ivica na bazi gradijenta

Ivični piksel predstavlja onaj piksel kod koga magnituda gradijenta ima značajnu vrednost

Prema tome detekcija ivica se može obaviti poređenjem magnitude gradijenta sa pragom. Pikseli čija magnituda gradijenta prelazi prag bivaju označeni kao ivični pikseli.

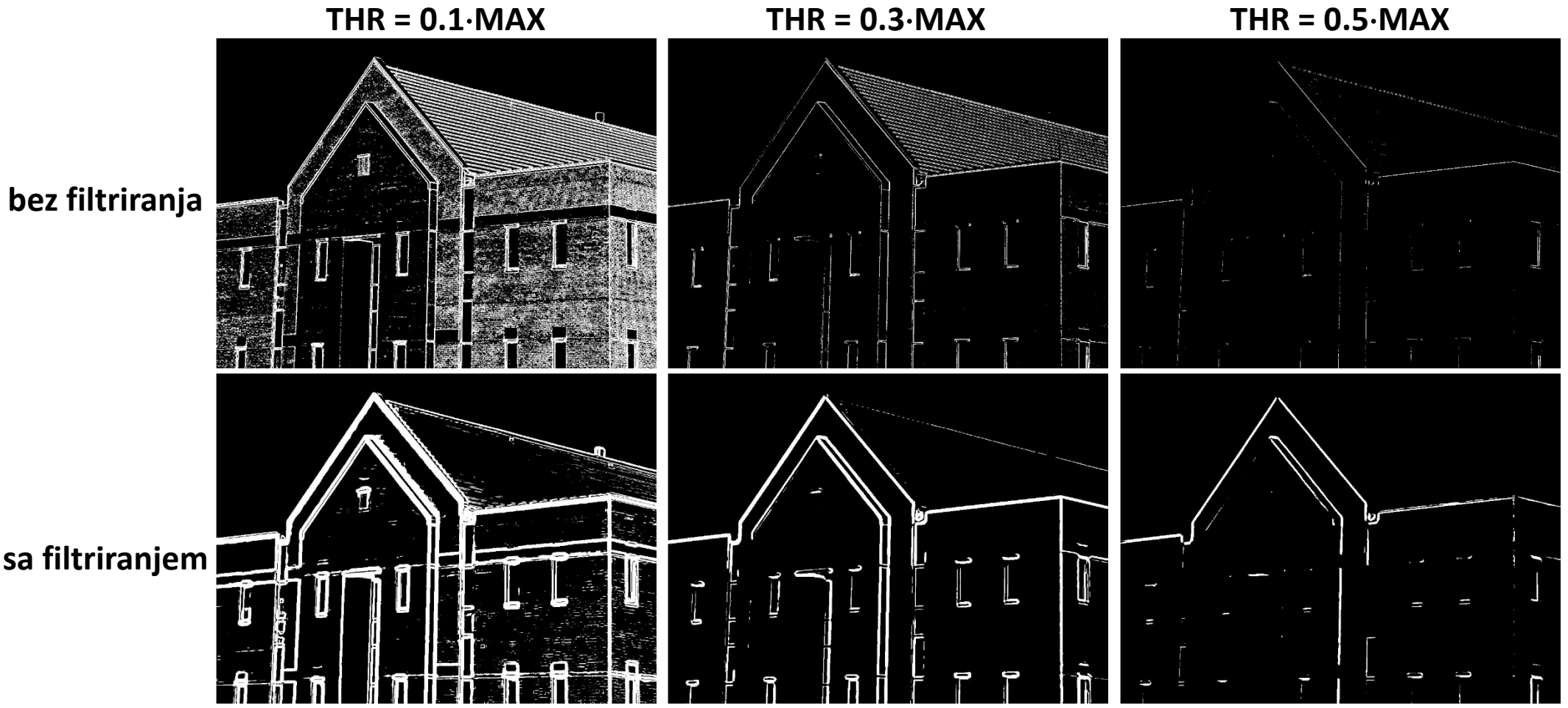
$$E[m,n] = \begin{cases} 1, \left|\nabla f[m,n]\right| \geq T \\ 0, \left|\nabla f[m,n]\right| < T \end{cases}$$

Poređenje gradijenta s pragom

```
Gx = imfilter(I, Hx, 'replicate', 'same');
Gy = imfilter(I, Hy, 'replicate', 'same');
Gm_filt = (Gx.^2 + Gy.^2).^0.5;

thr1 = 0.1*max(Gm(:));
thr2 = 0.3*max(Gm(:));
thr3 = 0.5*max(Gm(:));
E = zeros(size(Gm));
E(Gm>thr1) = 1;
figure('Name', 'THR=0.1*MAX'); imshow(E, []);
E = zeros(size(Gm));
E(Gm>thr2) = 1;
figure('Name', 'THR=0.3*MAX'); imshow(E, []);
E = zeros(size(Gm));
E(Gm>thr3) = 1;
figure('Name', 'THR=0.5*MAX'); imshow(E, []);
```

```
Ifilt = imfilter(I, fspecial('gaussian', [9 9], 5),
'replicate', 'same');
Gx = imfilter(Ifilt, Hx, 'replicate', 'same');
Gy = imfilter(Ifilt, Hy, 'replicate', 'same');
Gm_filt = (Gx.^2 + Gy.^2).^0.5;

thr1 = 0.1*max(Gm_filt(:));
thr2 = 0.3*max(Gm_filt(:));
thr3 = 0.5*max(Gm_filt(:));
E = zeros(size(Gm_filt));
E(Gm_filt>thr1) = 1;
figure('Name', 'FILT THR=0.1*MAX'); imshow(E, []);
E = zeros(size(Gm_filt));
E(Gm_filt>thr2) = 1;
figure('Name', 'FILT THR=0.3*MAX'); imshow(E, []);
E = zeros(size(Gm_filt));
E(Gm_filt>thr3) = 1;
figure('Name', 'FILT THR=0.5*MAX'); imshow(E, []);
```

Poređenje gradijenta s pragom



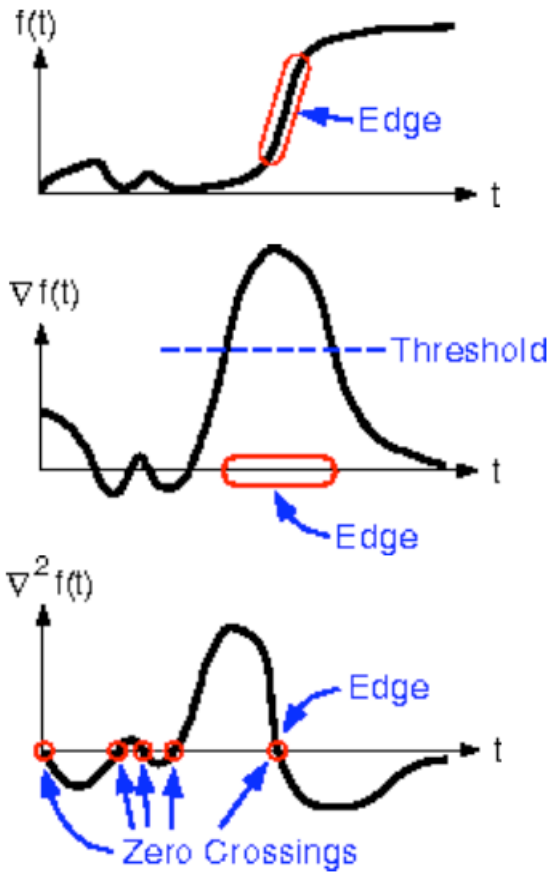
Laplasov operator

Gradijentni operatori imaju tendenciju da daju široke ivice – lošija lokalizacija

Postepeni prelazi će rezultovati malim gradijentom i verovatno neće biti detektovani

Drugi izvod daje bolju lokalizaciju ivica!

Položaj ivice se određuje prolascima kroz nulu drugog izvoda slike



Laplasov operator

Detekcija ivica Laplasovim operatorom se izvodi tako što se najpre odredi laplasijan slike:

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

a potom se detektuju značajne vrednosti laplasijana i za njih odrede prolasci kroz nulu.

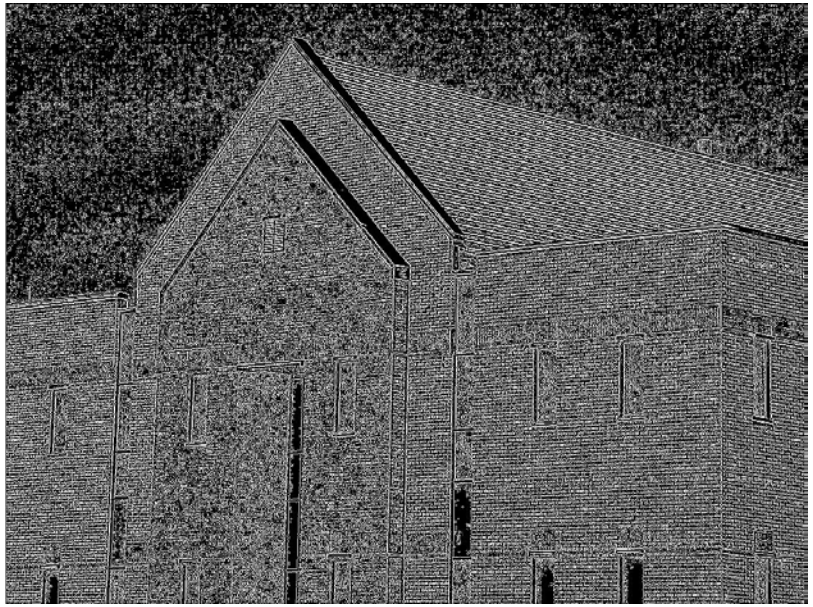
Diskretne aprosksimacije laplasijana:

$$h_L[m,n] = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad h_L[m,n] = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$h_L[m,n] = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad h_L[m,n] = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

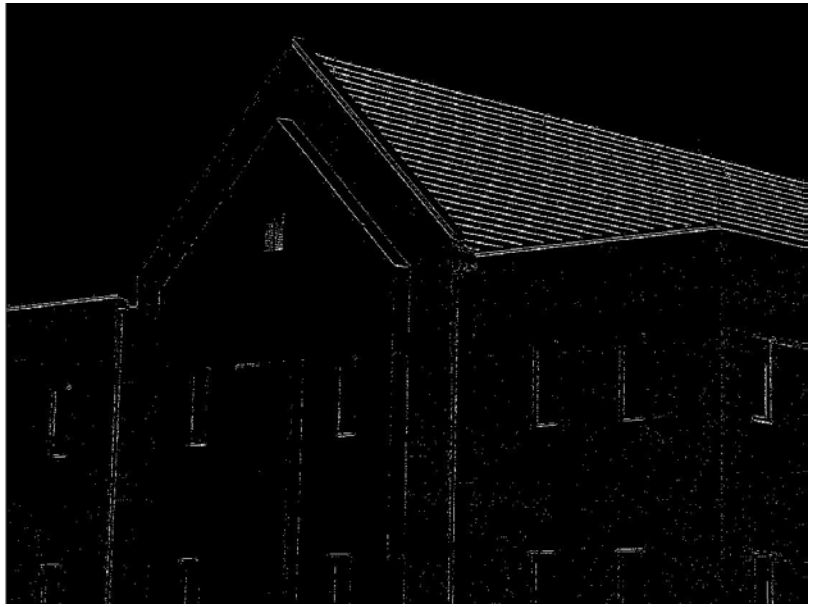
Laplasov operator - primer

THR = 0.1



THR = 0.4

THR = 0.8



Laplasov operator

Dosta osetljiviji na šum od operatora baziranih na gradijentu

Detekcija nultih preseka rezultuje tankim ivicama širine 1 piksel

Laplasijan je izotropan tako da ne daje informaciju o smeru ivice

LoG (Laplacian of Gaussian) operator

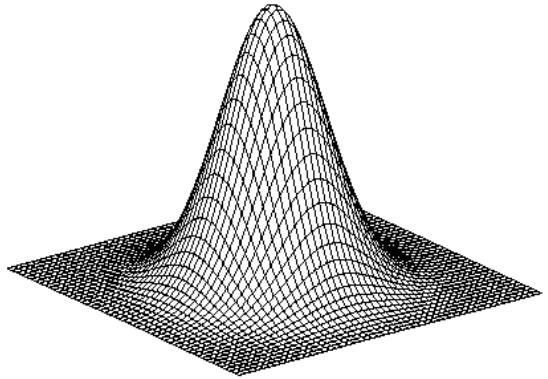
Osetljivost na šum se smanjuje prethodnim niskofrekventnim filtriranjem

Može se spojiti u jedan korak primenom laplasijana na imuplsni odziv NF filtra. Ako je NF filtar Gausov onda se takav operator naziva LoG (Laplacian of Gaussian)

$$h_{NF}(x,y) = \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}\right]\left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}}\right]$$

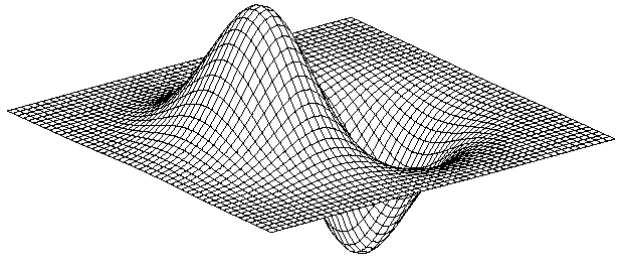
$$h_{LoG}(x,y) = \frac{1}{\pi\sigma^4}\left(1-\frac{x^2+y^2}{2\sigma^2}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Operatori izvedeni iz Gausove funkcije



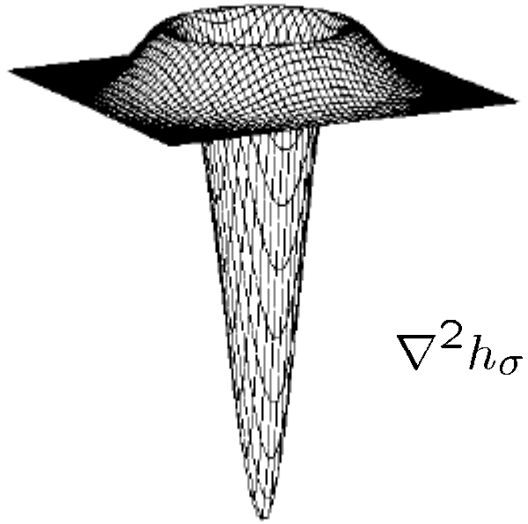
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

Gaussian



$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

DroG – Derivative of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

LoG – Laplacian of Gaussian
Mexican Hat (Sombrero)

Kanijev (Canny) algoritam za detekciju ivica

Nastao 1986 na MIT-u

I dalje ima superiorne karakteristike u pogledu detekcije ivica

Kani je formulisao tri uslova koja bi idealni detektor trebalo da ispuni:

1) Mali procenat lažnih detekcija, visok procenat pravih detekcija
2) Dobra lokalizacija
3) Jedinični odziv na ivicu (sve ivice su širine 1 piksel)

Kani je matematički opisao postavljene kriterijume i pokušao da nađe optimalno rešenje

U opštem slučaju analitičko rešenje ovog optimizacionog problema ne postoji ali se on može rešiti numeričkim metodama

Kanijev algoritam za detekciju ivica

Numeričko rešenje Kanijevog problema dovodi do toga da jedobra aproksimacija optimalnog detektora idealnih ivica za 1-D slučaj ivica zašumljenih Gausovim šumom prvi izvod Gausijana:

$$\frac{d}{dx}e^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}}$$

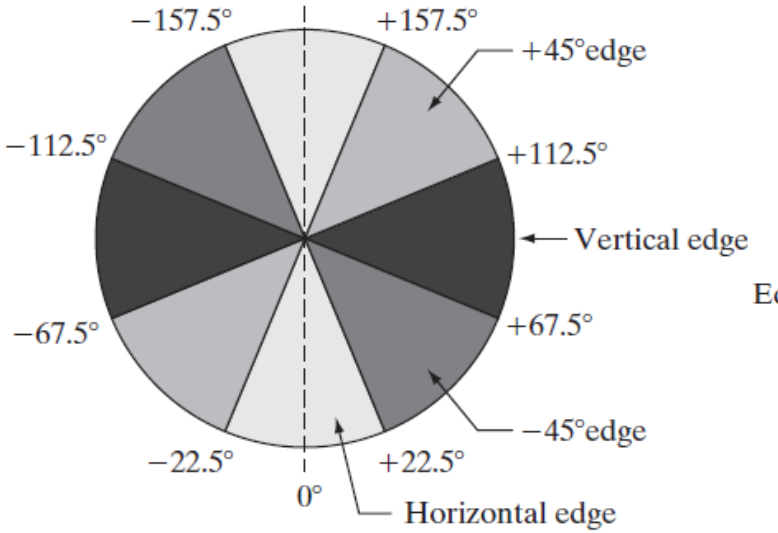
U 2-D slučaju detekcija se radi u smeru dominantnog gradijenta (normala na ivicu)

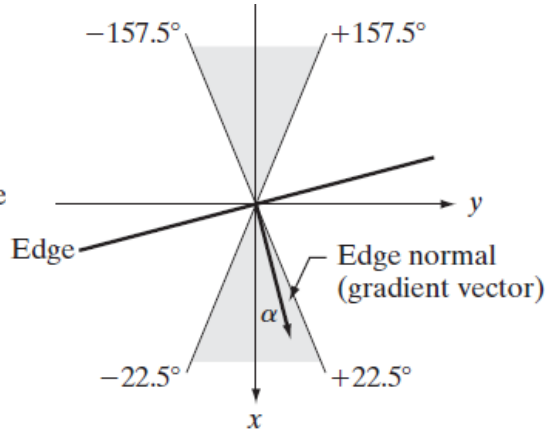
Kanijev algoritam za detekciju ivica

Najpre se primeni odgovarajući operator za detekciju gradijenta. U Kanijevom algoritmu se koristi DroG operator.

Podešavanjem σ Gausove funkcije postiže se kompromis između lokalizacije i osetljivosti na šum.

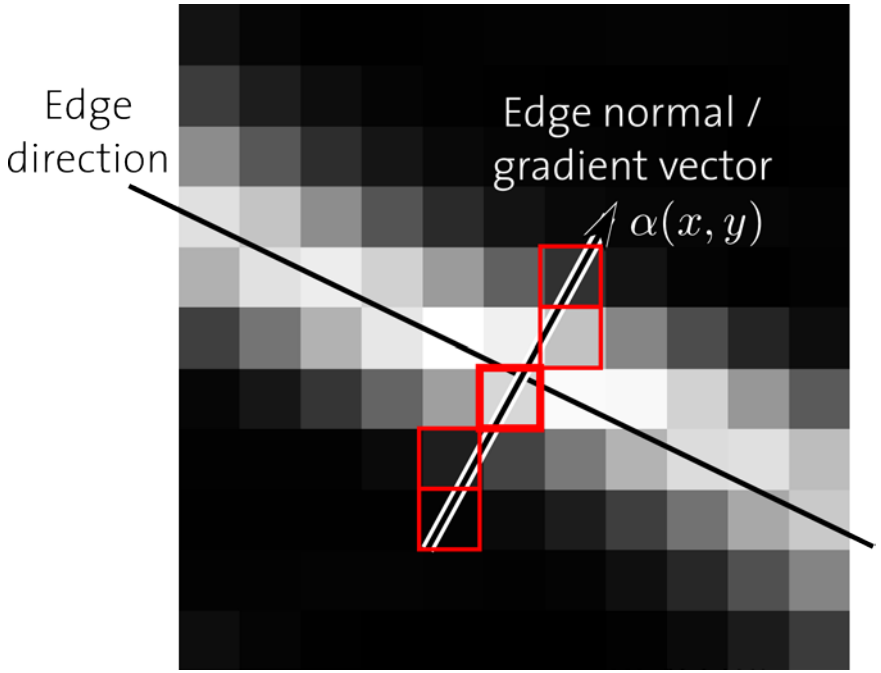
Sledeći korak predstavlja potiskivanje lokalnih ne-maksimuma.

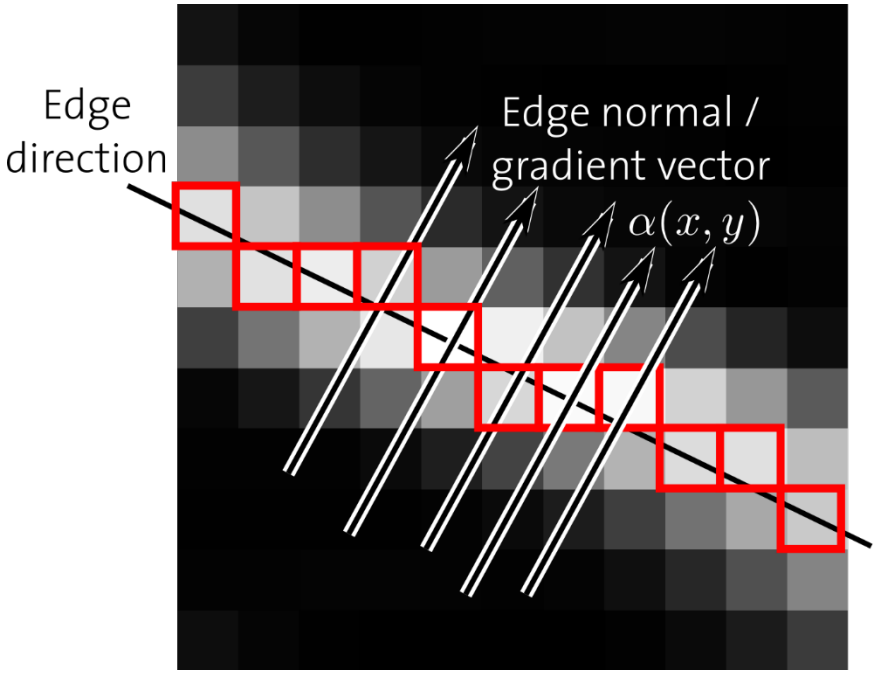




horizontalna	-45° dijagonalna	vertikalna	+45° dijagonalna
$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

Potiskivanje lokalnih ne-maksimuma





Histerezisno poređenje s pragom

Poređenje s pragom se obavlja u dva koraka kako bi se obezbedila što bolja robusnost na šum

Zadaju se dve vrednosti praga. Svi pikseli čija magnituda gradijenta prelazi viši prag predstavljaju sigurne ivice, dok pikseli čija magnituda gradijenta prelazi samo niži prag predstavljaju potencijalne ivice

U narednom koraku, zadržavaju se samo one potencijalne ivice koje su povezane sa nekom sigurnom ivicom

Podrška u MATLAB-u

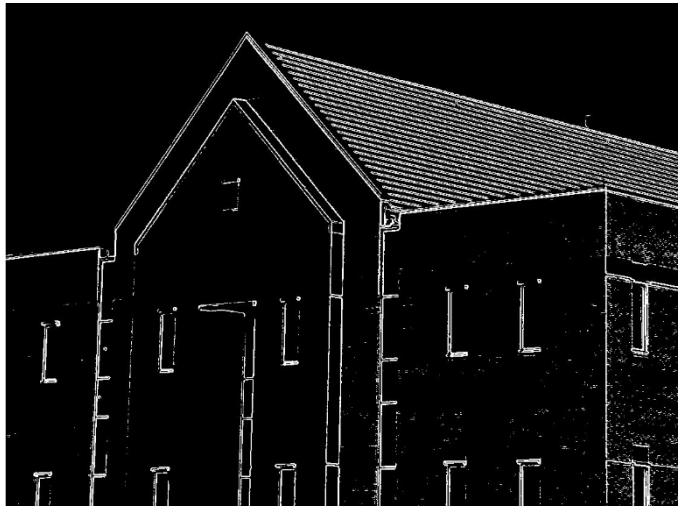
binarna_matrica_ivica = edge(ulazna_slika, metod, opcije)

```
help edge

Es = edge(I, 'sobel', 'nothinning');
figure('Name',  'Sobel operator'); imshow(Es, []);

El = edge(I, 'log', 0.0006, 4);
figure('Name',  'LoG operator'); imshow(El, []);

Ec = edge(I, 'canny', 0.2, 2);
figure('Name',  'Canny operator'); imshow(Ec, []);
```

Isprobati funkciju edge za različite vrednosti parametara.

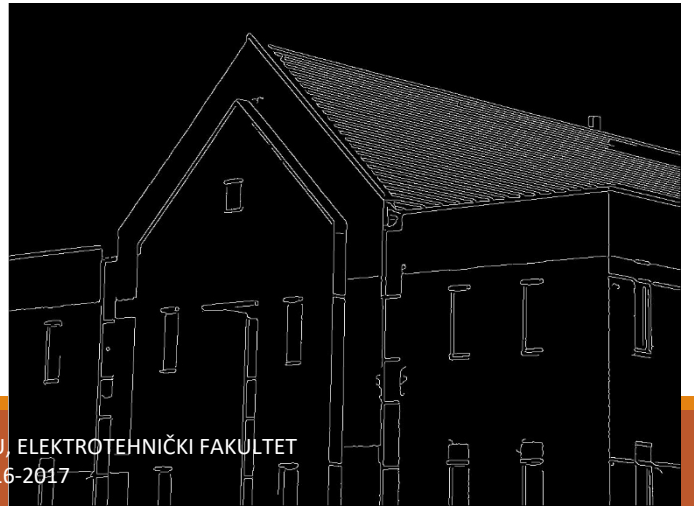
Primer

ulaz



Sobel

LoG



Canny

Harisov detektor – detekcija ćoškova

$$A(x,y) = \left[\nabla_x I(x,y)\right]^2$$

$$B(x,y) = \left[\nabla_y I(x,y)\right]^2$$

$$C(x,y) = \nabla_x I(x,y) \cdot \nabla_y I(x,y)$$

$$\bar{A}(x,y) = H_\sigma * A(x,y)$$

$$\bar{B}(x,y) = H_\sigma * B(x,y)$$

$$\bar{C}(x,y) = H_\sigma * C(x,y)$$

$$M(x,y) = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

$$\bar{M}(x,y) = \begin{bmatrix} \bar{A} & \bar{C} \\ \bar{C} & \bar{B} \end{bmatrix}$$

$$R = \det \bar{M} - k\left(\mathrm{tr}\bar{M}\right)^2$$

$$R = \bar{A}\bar{B} - \bar{C}^2 - k\left(\bar{A} + \bar{B}\right)^2$$

Harisov detektor – detekcija ćoškova

```
I = rgb2gray(imread('shapes_gray.png'));
I = im2double(I);

Sx = [-1 -2 -1; 0 0 0; 1 2 1]./8;
Sy = [-1 0 1; -2 0 2; -1 0 1]./8;

Ix = imfilter(I, Sx, 'replicate');
Iy = imfilter(I, Sy, 'replicate');

H = fspecial('gaussian', [5 5], 1);
A = imfilter(Ix.^2, H, 'replicate');
B = imfilter(Iy.^2, H, 'replicate');
C = imfilter(Ix.*Iy, H, 'replicate');

figure; imshow(I);
figure; imshow(A, []);
figure; imshow(B, []);
figure; imshow(C, []);
```

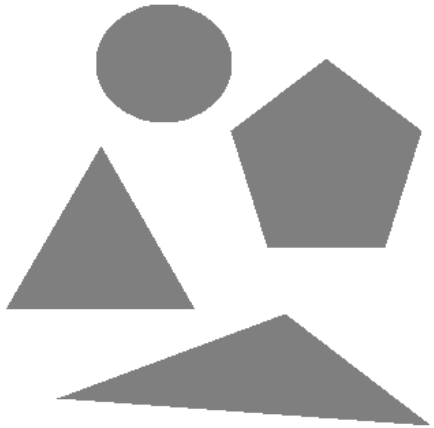
```
k = 0.05;
R = A.*B - C.^2 - k*((A+B).^2);
Q = R./max(R(:));

thr = 0.15;

corner = Q;
corner(Q<thr) = 0;
corner(Q>=thr) = 1;

figure; imshow(corner);
```

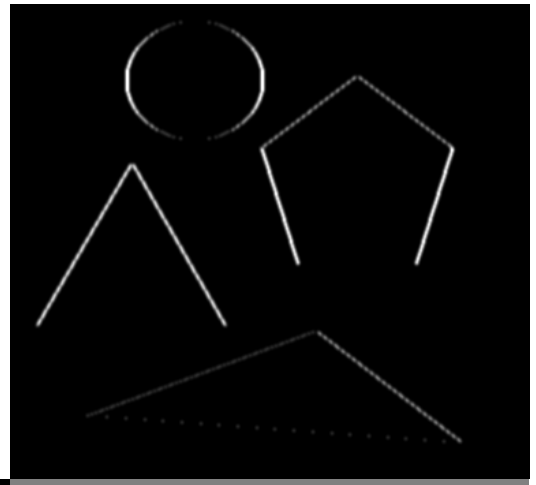
Probati funkciju za vrednosti praga 0.1, 0.2

Probati sa slikama shapes_gray1.png i shapes.png

Harisov detektor - primer

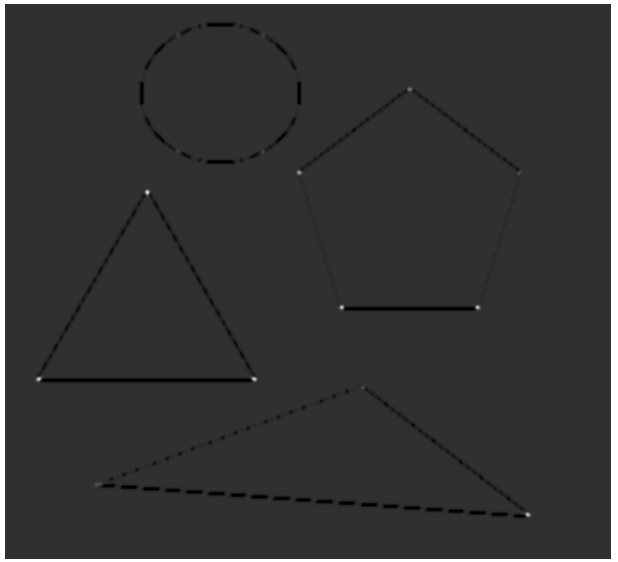
I



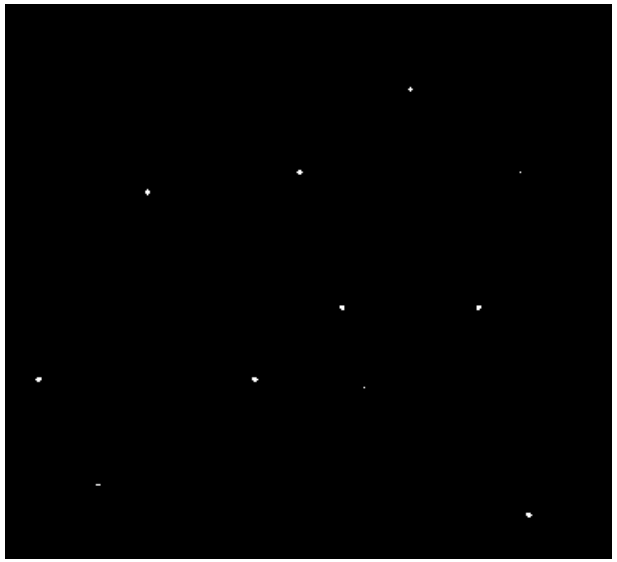


$\left[\nabla_y I(x,y)\right]^2$

Q

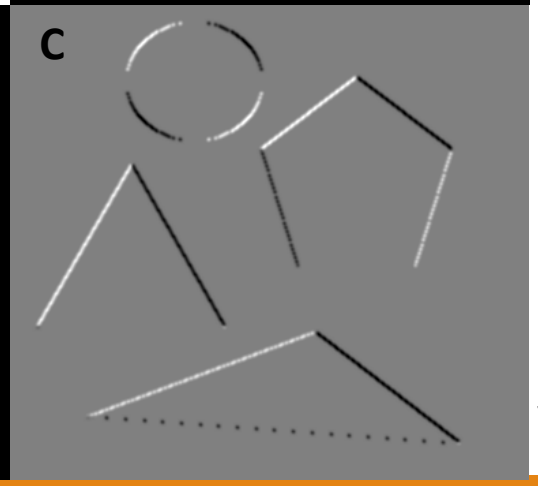


detektovani ćoškovi



$\left[\nabla_x I\right]^2$

C

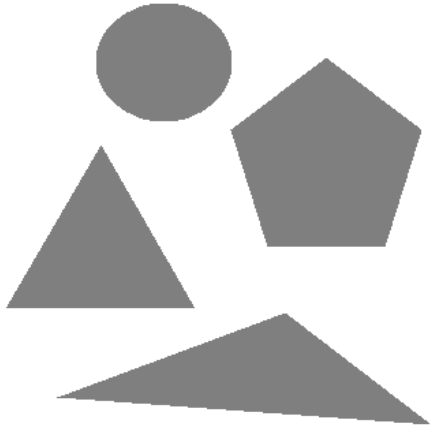


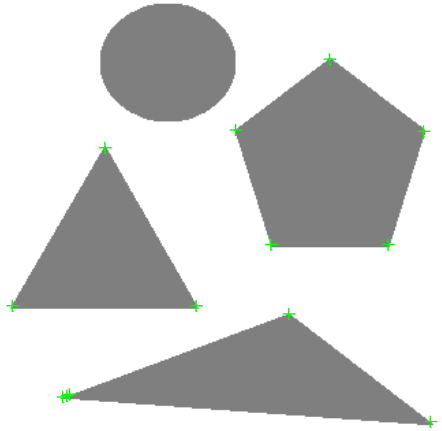
$\nabla_x I(x,y)\cdot\nabla_y I(x,y)$

Harisov detektor – detekcija ćoškova

```
I = rgb2gray(imread('shapes_gray.png'));
I = im2double(I);

corners = detectHarrisFeatures(I, 'MinQuality', 0.25, 'FilterSize', 5);
figure; imshow(I); hold on; plot(corners); hold off;
```

Zbog čega se prag toliko razlikuje ovde?





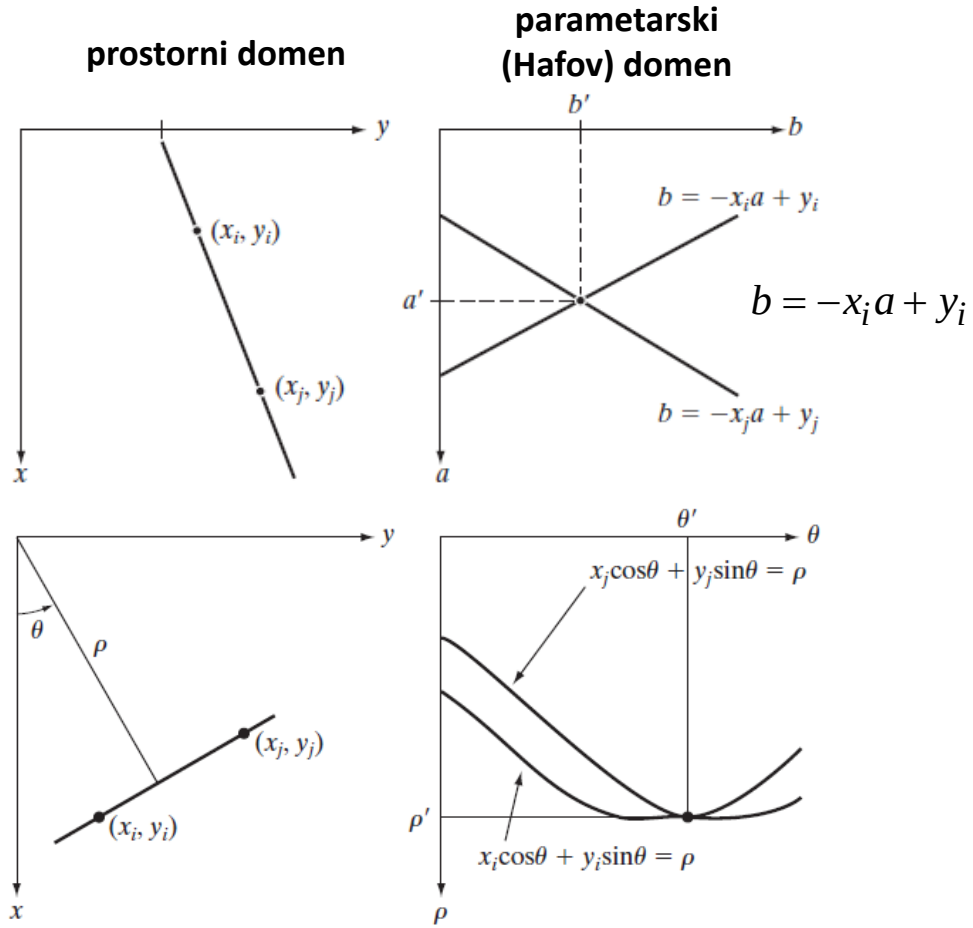
Harisov detektor - primer

```
corners = detectHarrisFeatures(I, 'MinQuality', 0.05, 'FilterSize', 11);
figure; imshow(I); hold on; plot(corners); hold off;
```

```
J = imfilter(I, fspecial('gaussian', [7 7], 2), 'replicate');
corners = detectHarrisFeatures(J, 'MinQuality', 0.05, 'FilterSize', 11);
figure; imshow(I); hold on; plot(corners); hold off;
```

Detekcija linija – Hafova transformacija

$y_i = ax_i + b$



$b = -x_i a + y_i$

prostorni domen

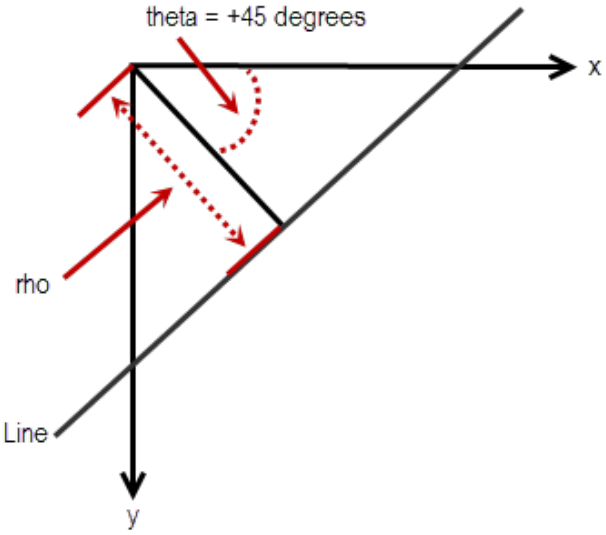
$$x\cos\theta + y\sin\theta = \rho$$

parametarski domen (θ,ρ):

Svaka tačka iz prostornog domena predstavlja sinusoidalnu funkciju u parametarskom domenu. Svaka prava u prostornom domenu predstavlja tačku u parametarskom domenu. Ako se tri ili više sinusoidalnih kriva seku u istoj tački parametarskom domenu to znači da su odgovarajuće taške kolinearne u prosotrnom domenu.

Ideja: Naći tačke nagomilavanja (sa najviše preseka) u parametarskom domenu.

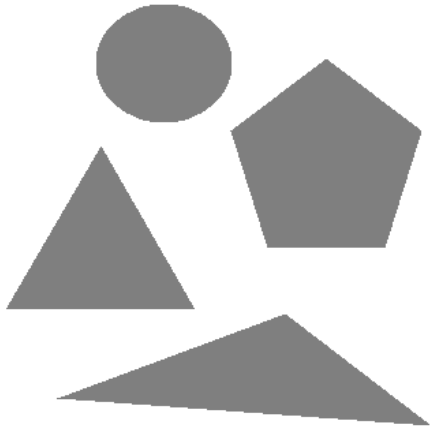
Funkcija hough u Matlabu za ugao normale uzima vrednosti iz opsega +-90° u smeru prikazanom na slici

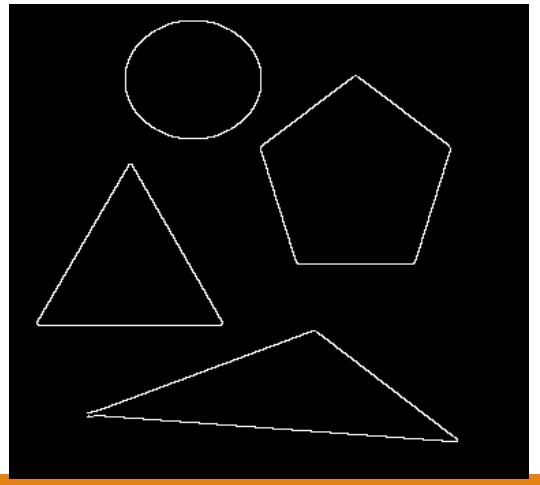


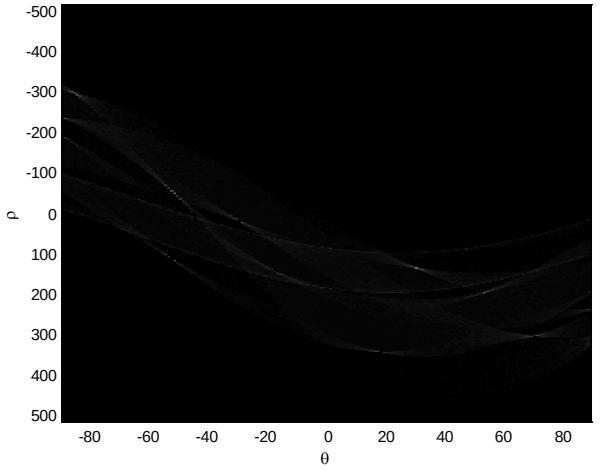
Hafova transformacija

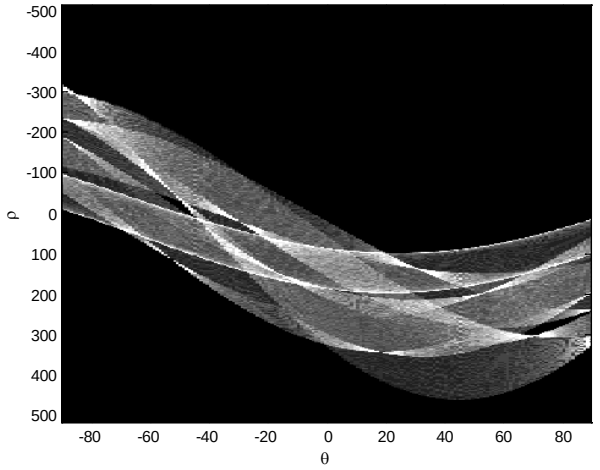
```
I = rgb2gray(imread('shapes_gray.png'));
I = im2double(I);

E = edge(I, 'canny');
figure; imshow(E);

[H,T,R] = hough(E);

figure; imagesc(H,'XData',T,'YData',R); colormap('gray');
xlabel('\theta'), ylabel('\rho'); axis on, axis normal;
figure; imagesc(imadjust(mat2gray(H)),'XData',T,'YData',R); colormap('gray');
xlabel('\theta'), ylabel('\rho'); axis on, axis normal;

P  = houghpeaks(H, 11, 'Threshold', 0.2*max(H(:)));
figure; imagesc(H, 'XData',T,'YData',R);
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
plot(T(P(:,2)),R(P(:,1)),'s','color','white');
```

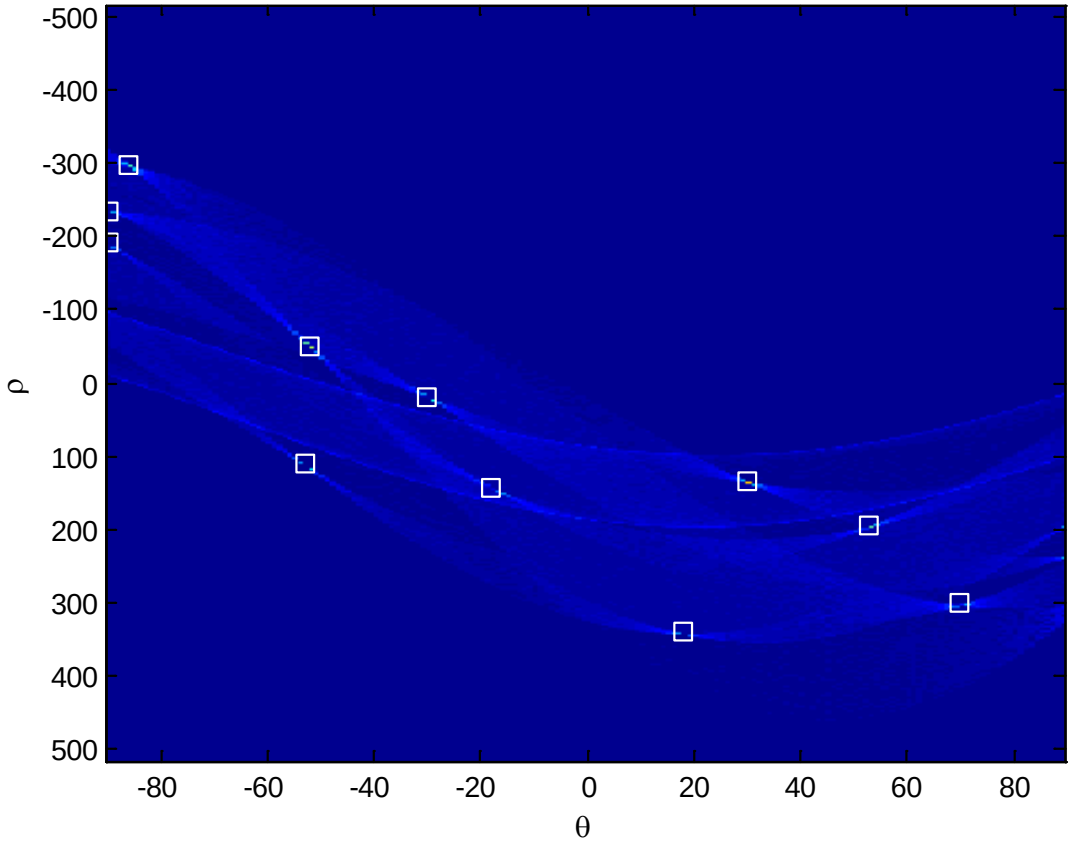
Hafova transformacija - primer





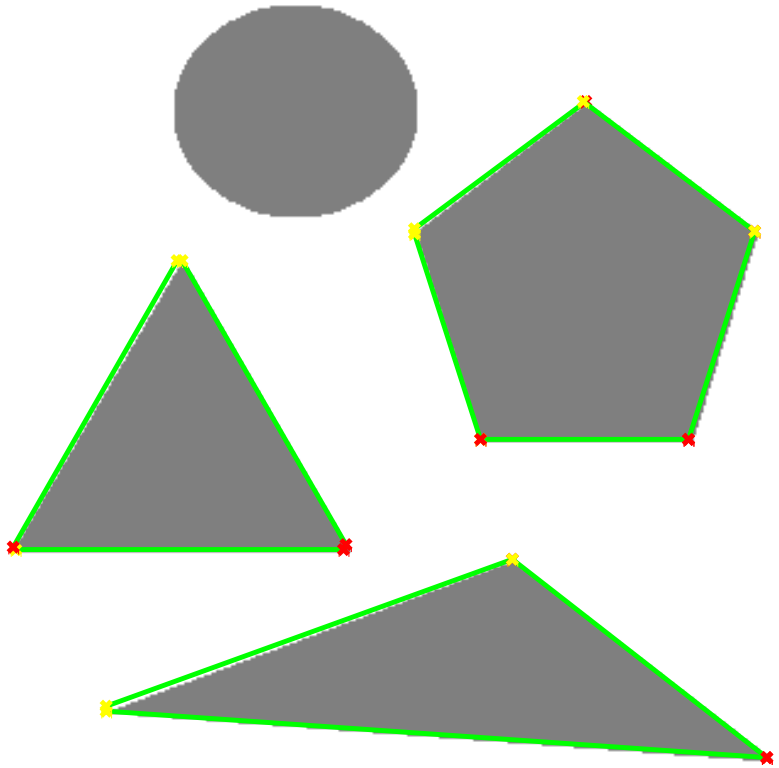






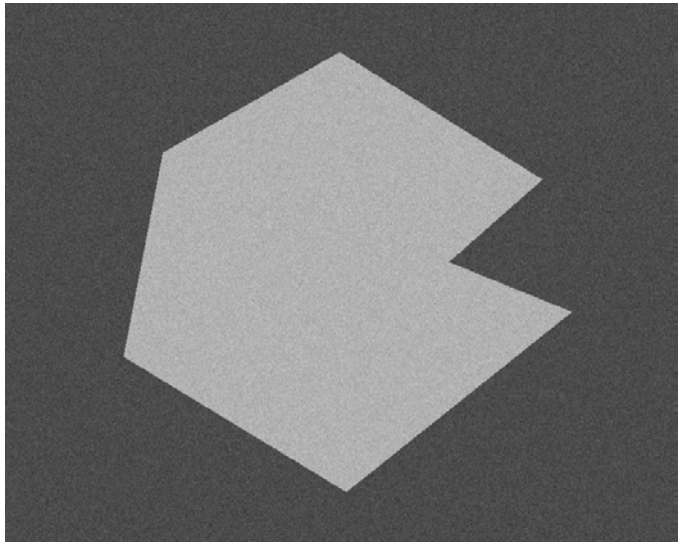
Hafova transformacija – iscrtavanje linija

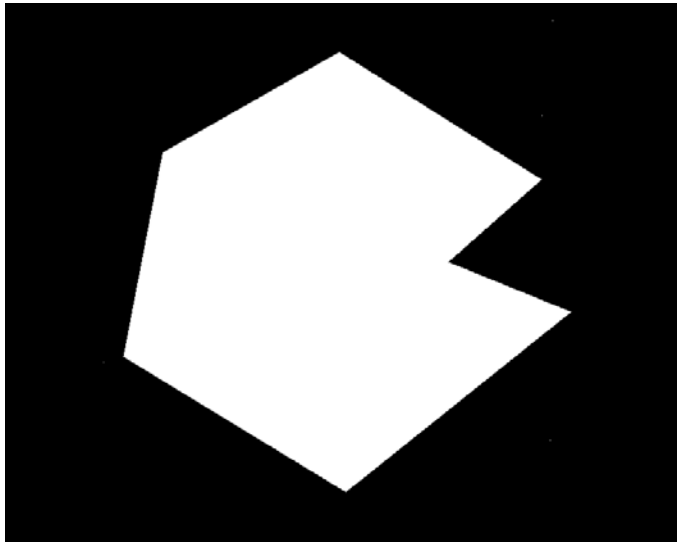
```
lines = houghlines(E,T,R,P);

figure, imshow(I), hold on

for k = 1:length(lines)
  xy = [lines(k).point1; lines(k).point2];
  plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

  % plot beginnings and ends of lines
  plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
  plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
end

hold off;
```

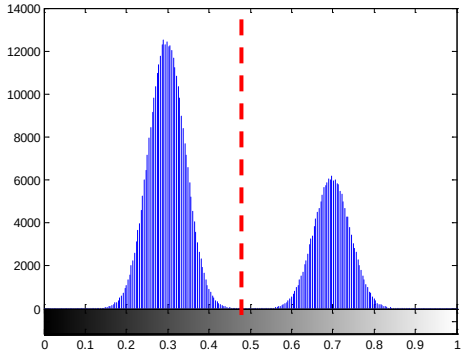


Segmentacija slike – poređenje s pragom

```
I = imread('septagon.tif');
I = im2double(I);
figure; imshow(I);
figure; imhist(I); ylim('auto');

[T, EM] = graythresh(I)

S = I > T;
figure; imshow(S);
```



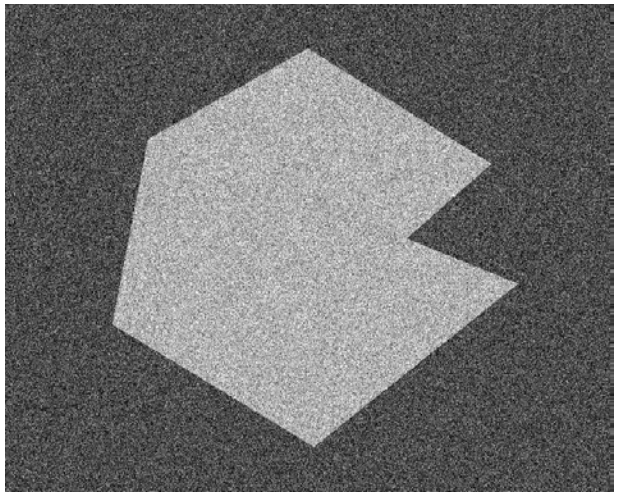


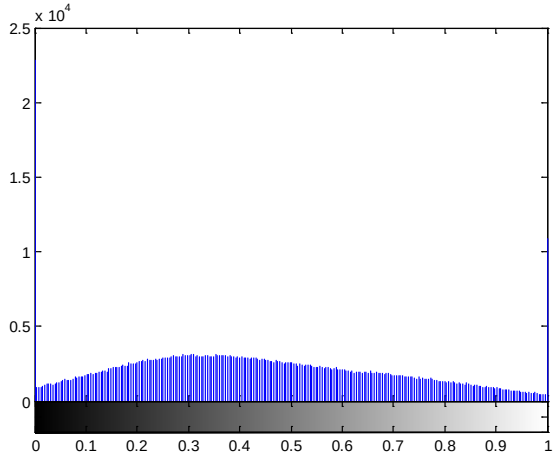


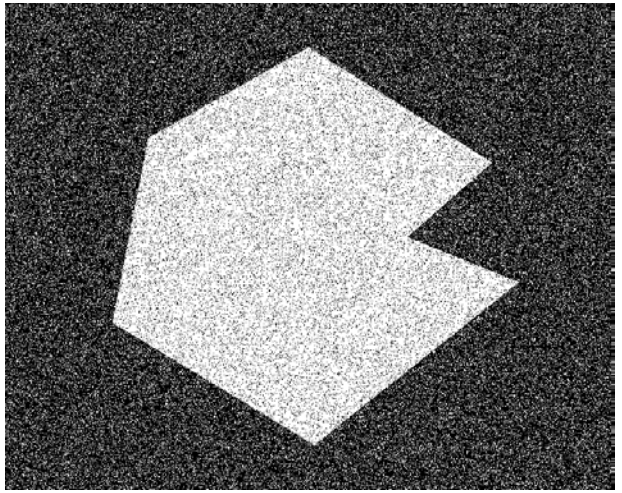
Segmentacija zašumljenih slika

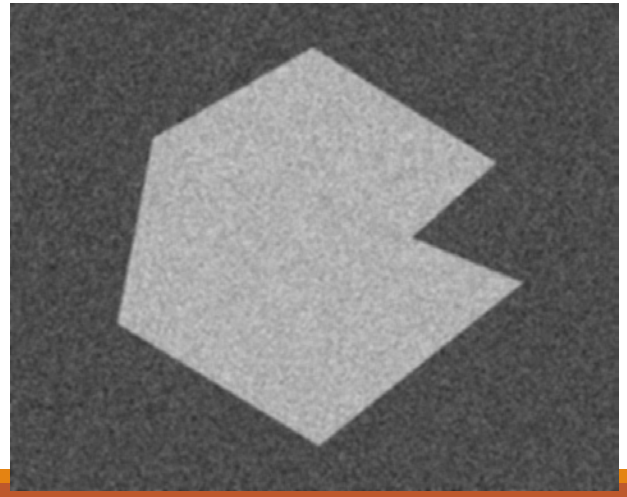
```
I = imread('septagon_noisy.tif');
I = im2double(I);
figure; imshow(I);
figure; imhist(I); ylim('auto');

[T, EM] = graythresh(I)

S = I > T;
figure; imshow(S);

If = imfilter(I, fspecial('gaussian', [7 7], 3), 'replicate');
figure; imshow(If);
figure; imhist(If); ylim('auto');

[T, EM] = graythresh(If)

S = If > T;
figure; imshow(S)
```

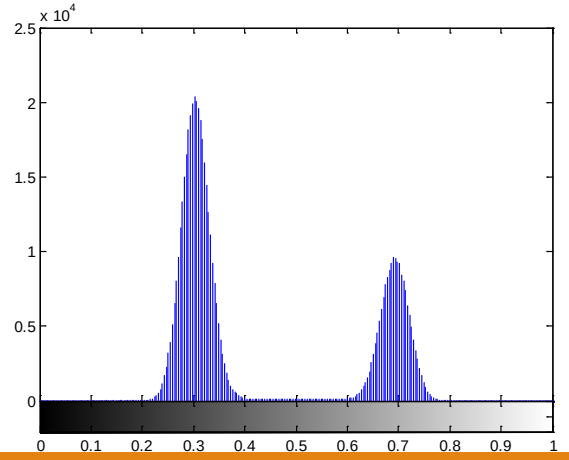
Segmentacija zašumljenih slika

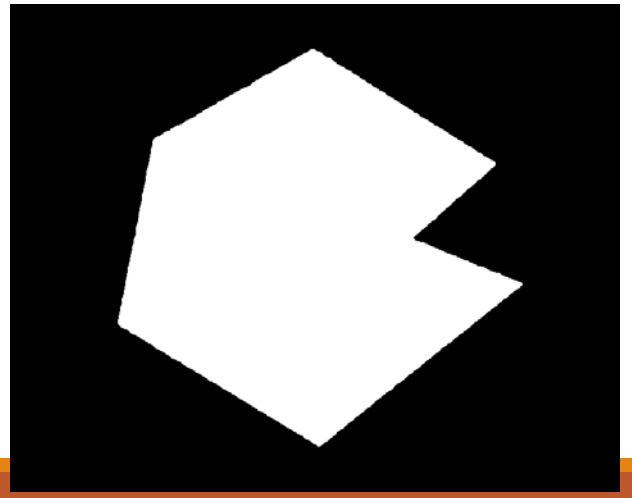








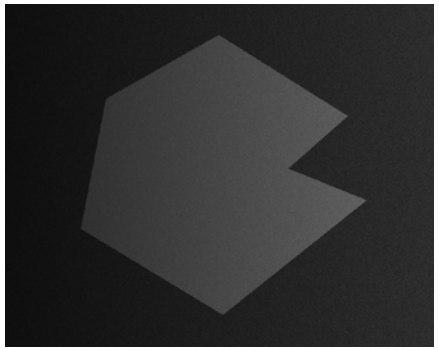


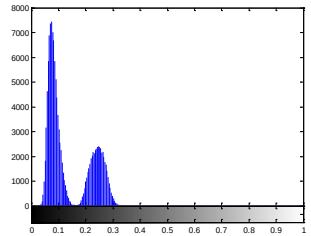


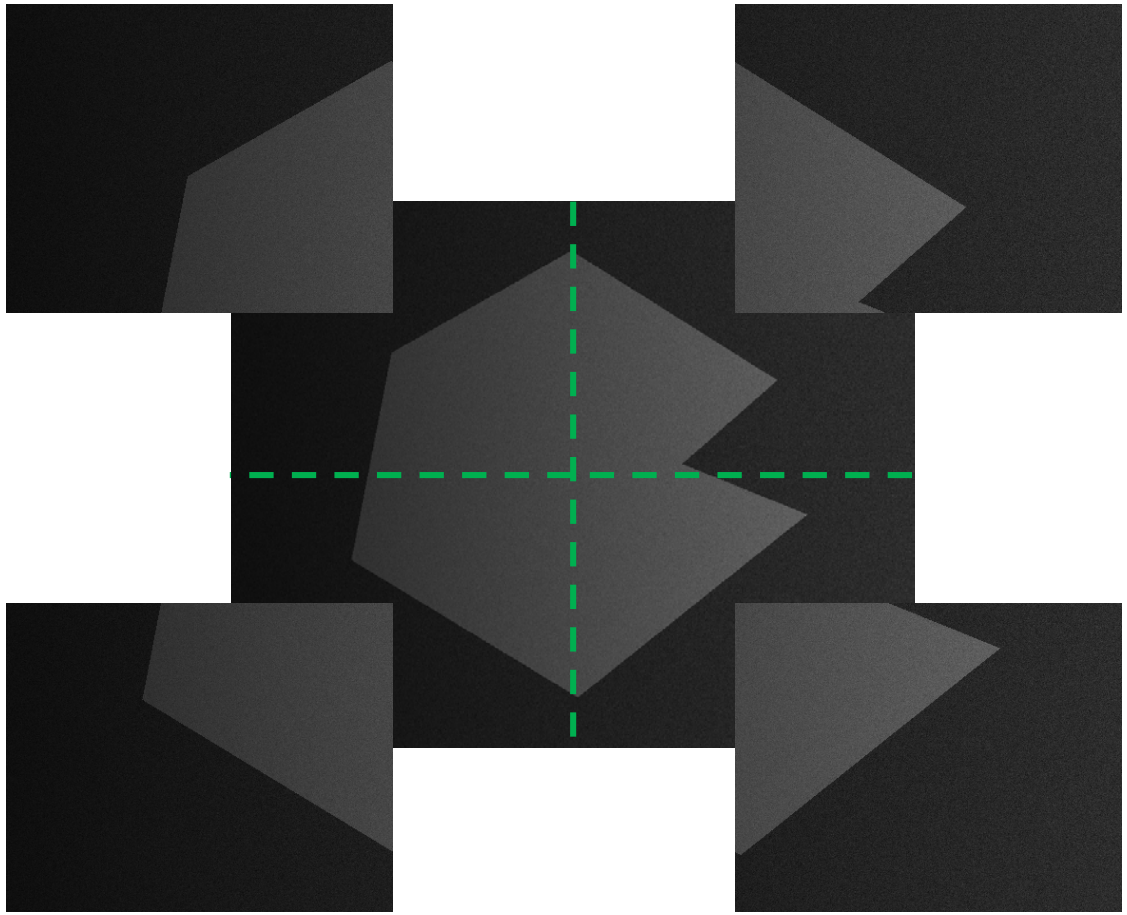
Prostorno promenljivi prag

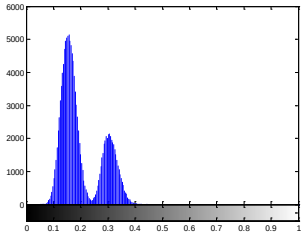
```
I = imread('septagon_noisy_shaded.tif');
I = im2double(I);
figure; imshow(I);
figure; imhist(I); ylim('auto');

[T, EM] = graythresh(I)

S = I > T;
figure; imshow(S);

I1 = I(1:round(size(I,1)/2), 1:round(size(I,2)/2));
I2 = I(1:round(size(I,1)/2), round(size(I,2)/2)+1:end);
I3 = I(round(size(I,1)/2)+1:end, 1:round(size(I,2)/2));
I4 = I(round(size(I,1)/2)+1:end, round(size(I,2)/2)+1:end);
figure; imshow(I1); figure; imhist(I1); ylim('auto');
figure; imshow(I2); figure; imhist(I2); ylim('auto');
figure; imshow(I3); figure; imhist(I3); ylim('auto');
figure; imshow(I4); figure; imhist(I4); ylim('auto');
```

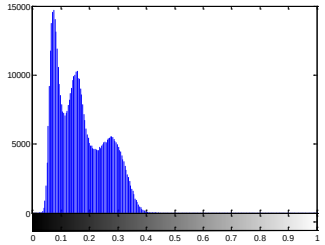
Prostorno promenljivi prag

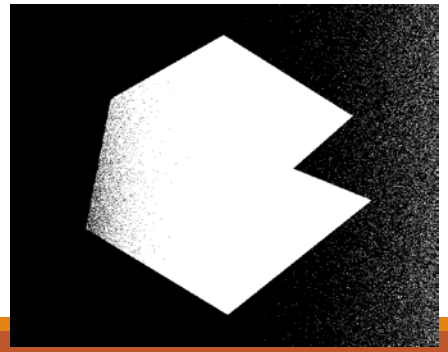


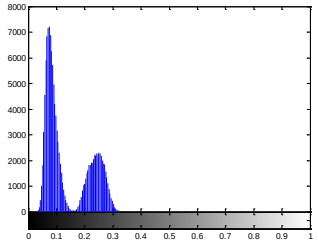


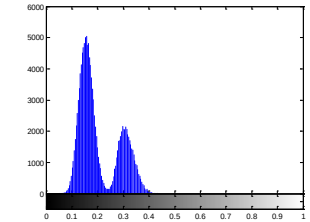








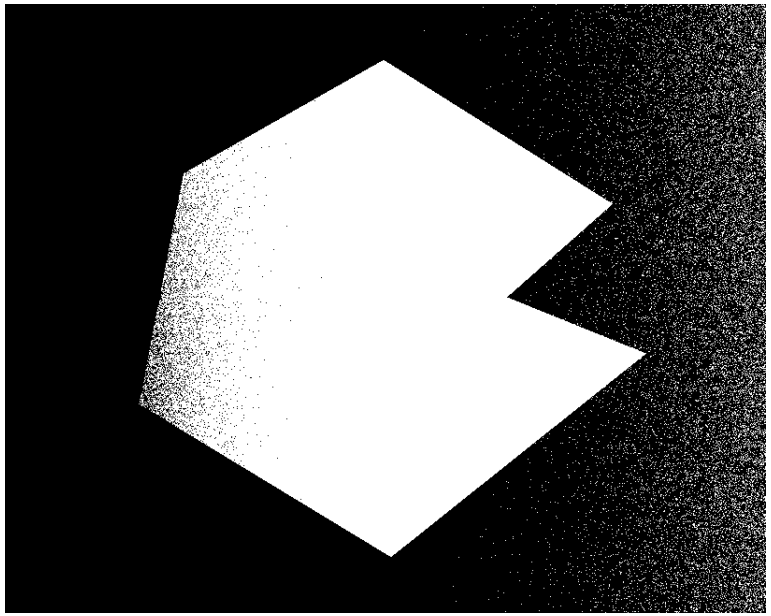




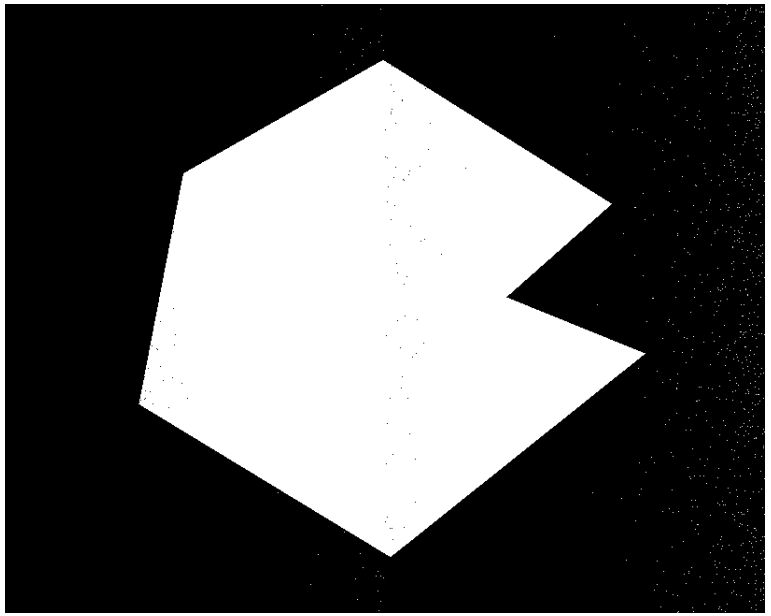
Prostorno promenljivi prag

```
S = zeros(size(I));

S(1:round(size(I,1)/2), 1:round(size(I,2)/2)) = I1 > graythresh(I1);
S(1:round(size(I,1)/2), round(size(I,2)/2)+1:end) = I2 > graythresh(I2);
S(round(size(I,1)/2)+1:end, 1:round(size(I,2)/2)) = I3 > graythresh(I3);
S(round(size(I,1)/2)+1:end, round(size(I,2)/2)+1:end) = I4 > graythresh(I4);

figure; imshow(S);

S1 = medfilt2(S, [3,3], 'symmetric');
figure; imshow(S1);
```

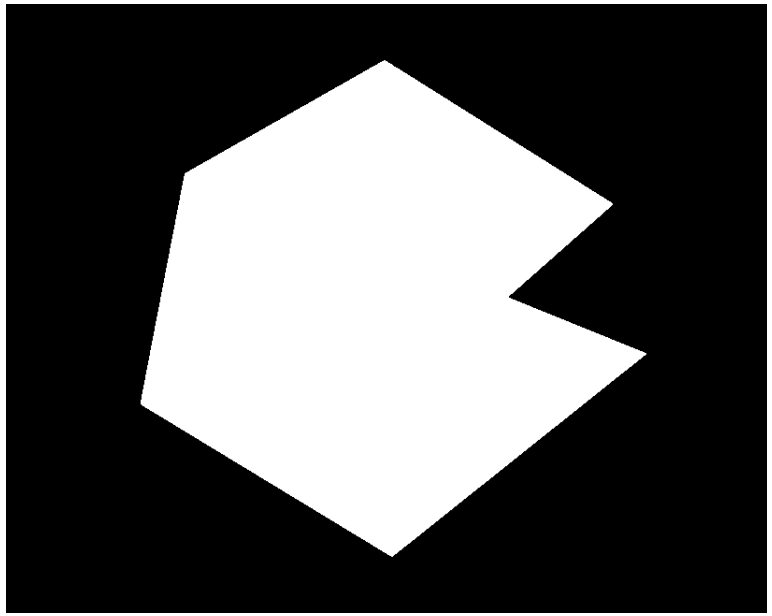
Prostorno promenljivi prag



globalni prag



lokalni prag

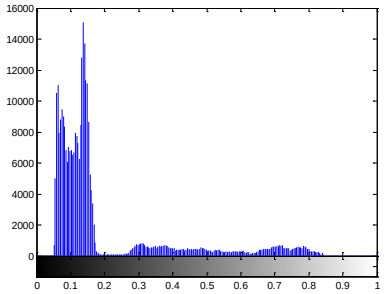


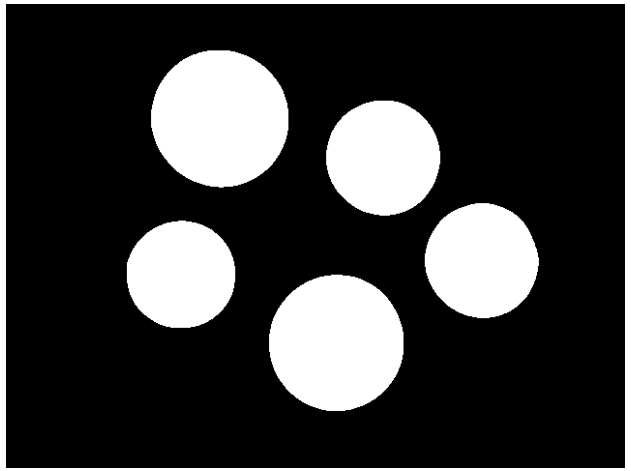
**lokalni prag +
medijan 3x3**

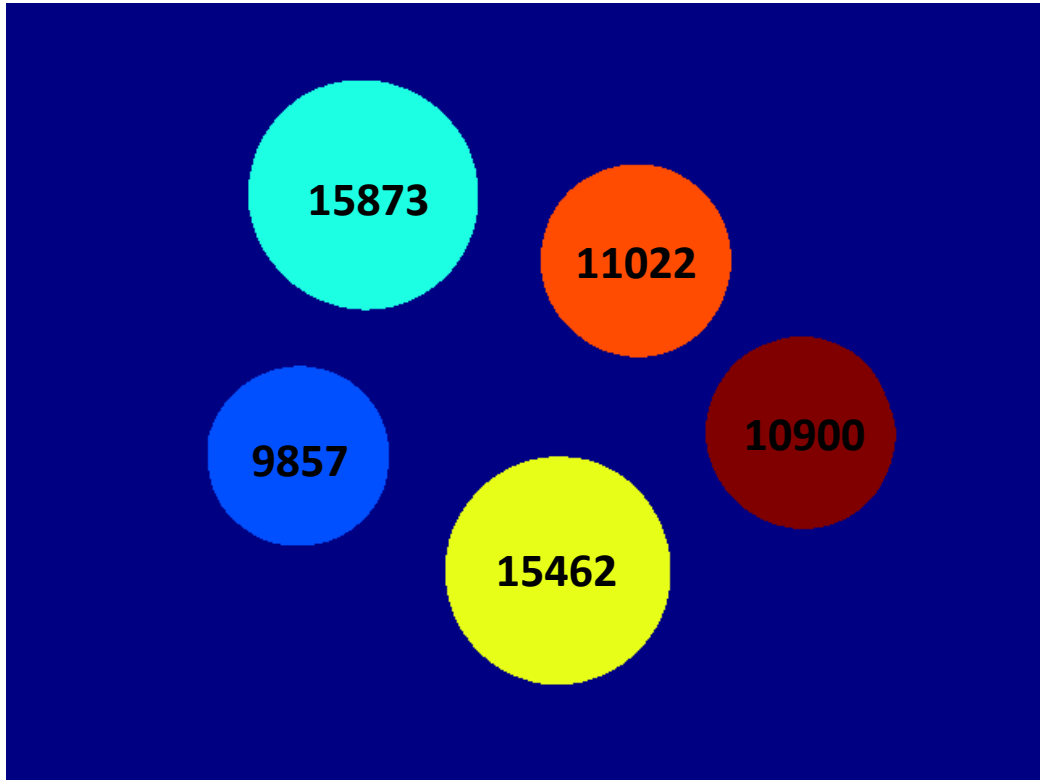
Analiza slike

```
I = rgb2gray(imread('coins.jpg'));
I = im2double(I);
figure; imshow(I);

If = medfilt2(I, [13 13], 'symmetric');
figure; imshow(If);
figure; imhist(If); ylim('auto');

S = If > 0.2;
figure; imshow(S);
[coins, num] = bwlabel(S);
figure; imshow(coins, []); colormap('jet');

area = regionprops(S, 'Area');
```

Analiza slike







Analiza slike

```
I = rgb2gray(imread('coins.jpg'));
I = im2double(I); figure; imshow(I);
Ie = edge(I,'canny',0.4,4); figure; imshow(Ie);

[center, radius] = imfindcircles(Ie, [50, 150]);

coin2_pos = find(radius<60);
coin5_pos = find((radius>60)&(radius<70));
coin20_pos = find(radius>70);

figure; imshow(I);
viscircles(center(coin2_pos,:), radius(coin2_pos), 'Color', 'b', 'LineWidth', 2);
viscircles(center(coin5_pos,:), radius(coin5_pos), 'Color', 'g', 'LineWidth', 2);
viscircles(center(coin20_pos,:), radius(coin20_pos), 'Color', 'r', 'LineWidth', 2);
```

Analiza slike