Yalova Universitesi
Fen Bilimleri Enstitusu
Bilgisayar Muhendisligi Ana Bilim Dali

Bilgisayarli Gorme Dersi

Sevdanur GENC - 135105008

Bilgisayarli Gorme

BSM 562 - Odev II Soru ve Cevaplari

1) Varolan armutlu.jpg image'i icin;

a) Asagidaki ilgili Canny Algoritmasi ile birlikte elimizdeki image goruntusunun kenarlari bulunmustur.

1. Compute the gradient of image $f(x,y)$ by convolving it with the first derivative of Gaussian in x and y directions.

$$f_x(x,y) = f(x,y) * (\frac{-x}{\sigma^2})\, e^{\frac{-(x^2+y^2)}{2\sigma^2}},$$
$$f_y(x,y) = f(x,y) * (\frac{-y}{\sigma^2})\, e^{\frac{-(x^2+y^2)}{2\sigma^2}}.$$

2. Perform non-maxima suppression on the gradient magnitude.

3. Apply hysteresis thresholding to the non-maxima suppressed magnitude. Scan the image from left-right, top-bottom. If the gradient magnitude at the pixel is above the high threshold, declare that as an edge point. Then recursively look at its neighbors (4 connected, or 8 connected). If the gradient magnitude is above the low threshold, declare that as an edge point.

Yapmis oldugumuz islemler su sirasiyla su sekildedir; elimizdeki goruntunun oncelikle image'imizi imread ile okutup bunu rgb2gray ile boyutunu azaltmis oluyorum sonrasinda double islemine donusturuyorum. Gaussian ve Hysteresis degerleri icin degiskenler atamalarini gerceklestiriyorum. Image'in konvulasyon degerleri icin gassuian sigma degerini belirleyerek ekrana sigma degerine gore gaussian filtrelemesini bastiriyoruz. Image'in pixel degerleri sayesinde tahmini koseleri belirleyebilmek icin normal yon hesaplaniyor. Buldugumuz pixellere gore yonleri ayiriyoruz. Gradyant yontemi ile bulmus oldugumuz kenarlarin zayif kenar noktalarini non-maxima suppression yontemi ile basitiyoruz. En son olarak Hysteresis thersholding yani ikili esik yontemini kullanarak alt ve ust esiklere gore sonuclari ekrana bastirmis oluyoruz. Matlab kodu ve ekran goruntuleri asagida bulunmaktadir.

```
clear all;
close all;
clc; clf;

filtreOlcusu = 5;
sigma = 14;
dusukEsikDegeri = 0.1;
yuksekEsikDegeri = 0.3;

image = rgb2gray(imread('armutlu.jpg'));
image = double(image);
figure(1), imshow(image, []), title('Orjinal Image');

gaussian_filtreleme = fspecial('gaussian', filtreOlcusu, sigma);
```

```
convulationImage = conv2(image, gaussian_filtreleme, 'same');
figure (2), imshow(convulationImage, []), title(['Gaussian Filterleme sonucu
\sigma = ', num2str(sigma)]);

[gaussian_filtreleme_x gaussian_filtreleme_y] =
gradient(gaussian_filtreleme);
image_gradient_x = conv2(convulationImage, gaussian_filtreleme_x, 'same');
image_gradient_y = conv2(convulationImage, gaussian_filtreleme_y, 'same');

normalYon = atan2(image_gradient_y, image_gradient_x);
normalYon = normalYon*180/pi;
normalYon_dis = zeros(512, 512);

for i = 1  : 512
    for j = 1 : 512
        if ((normalYon(i, j) > 0 ) && (normalYon(i, j) < 22.5) ||
(normalYon(i, j) > 157.5) && (normalYon(i, j) < -157.5))
            normalYon_dis(i, j) = 0;
        end
        if ((normalYon(i, j) > 22.5) && (normalYon(i, j) < 67.5) ||
(normalYon(i, j) < -112.5) && (normalYon(i, j) > -157.5))
            normalYon_dis(i, j) = 45;
        end
        if ((normalYon(i, j) > 67.5 && normalYon(i, j) < 112.5) ||
(normalYon(i, j) < -67.5 && normalYon(i, j) > 112.5))
            normalYon_dis(i, j) = 90;
        end
        if ((normalYon(i, j) > 112.5 && normalYon(i, j) <= 157.5) ||
(normalYon(i, j) < -22.5 && normalYon(i, j) > -67.5))
            normalYon_dis(i, j) = 135;
        end
    end
end

figure(3), imshow(image_gradient_x), title('X Gradyan');
figure(4), imshow(image_gradient_y), title('Y Gradyan');
figure(5), imagesc(normalYon_dis); colorbar, title('Normal Yon');
imageGradientMagnitude = sqrt(image_gradient_x.^2 + image_gradient_y.^2);
figure(6), imshow(imageGradientMagnitude, []), title('Gradyan Buyuklugu');

imageSupressed = zeros(512, 512);
for i = 2  : 511
    for j = 2 : 511
        if (normalYon_dis(i, j) == 0)
            if (imageGradientMagnitude(i, j) > imageGradientMagnitude(i, j -
1) && imageGradientMagnitude(i, j) > imageGradientMagnitude(i, j + 1))
                imageSupressed(i, j) = imageGradientMagnitude(i, j);
            else
                imageSupressed(i, j) = 0;
            end
        end

        if (normalYon_dis(i, j) == 45)
            if (imageGradientMagnitude(i, j) > imageGradientMagnitude(i + 1,
j - 1) && imageGradientMagnitude(i, j) > imageGradientMagnitude(i - 1, j +
1))
```

```
                imageSupressed(i, j) = imageGradientMagnitude(i, j);
            else
                imageSupressed(i, j) = 0;
            end
        end

      if (normalYon_dis(i, j) == 90)
            if (imageGradientMagnitude(i, j) > imageGradientMagnitude(i - 1,
j) && imageGradientMagnitude(i, j) > imageGradientMagnitude(i + 1, j))
                imageSupressed(i, j) = imageGradientMagnitude(i, j);
            else
                imageSupressed(i, j) = 0;
            end
        end

      if (normalYon_dis(i, j) == 135)
            if (imageGradientMagnitude(i, j) > imageGradientMagnitude(i - 1,
j - 1) && imageGradientMagnitude(i, j) > imageGradientMagnitude(i + 1, j +
1))
                imageSupressed(i, j) = imageGradientMagnitude(i, j);
            else
                imageSupressed(i, j) = 0;
            end
        end
    end
end
figure(7), imshow(imageSupressed), title('Image Non-Maximal Suppression');

thersholdDusukDegeri = dusukEsikDegeri * max(max(imageSupressed));
thersholdYuksekDegeri = yuksekEsikDegeri * max(max(imageSupressed));

imageThreshold = zeros(512, 512);

for i = 1  : 512
    for j = 1 : 512
        if (imageSupressed(i, j) < thersholdDusukDegeri)
            imageThreshold(i, j) = 0;
        elseif (imageSupressed(i, j) > thersholdYuksekDegeri)
            imageThreshold(i, j) = 1;
        else
            if ((imageSupressed(i + 1, j) > thersholdYuksekDegeri) ||
(imageSupressed(i - 1, j) > thersholdYuksekDegeri) || (imageSupressed(i, j +
1) > thersholdYuksekDegeri) || (imageSupressed(i, j - 1) >
thersholdYuksekDegeri))
                imageThreshold(i, j) = 1;
            end
        end
    end
end
figure(8), imshow(imageThreshold), title('Hysteresis Thresholding');
```

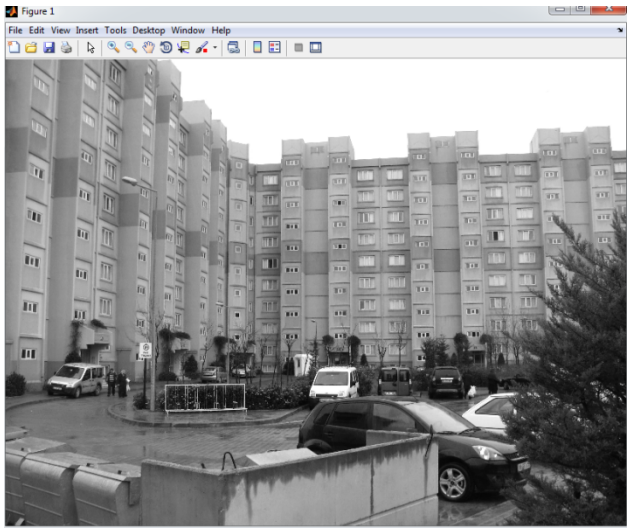


Figure 1
File Edit View Insert Tools Desktop Window Help

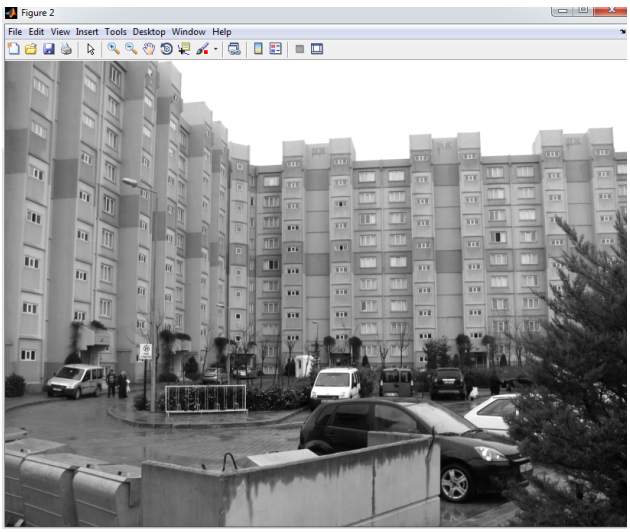


Figure 2
File Edit View Insert Tools Desktop Window Help

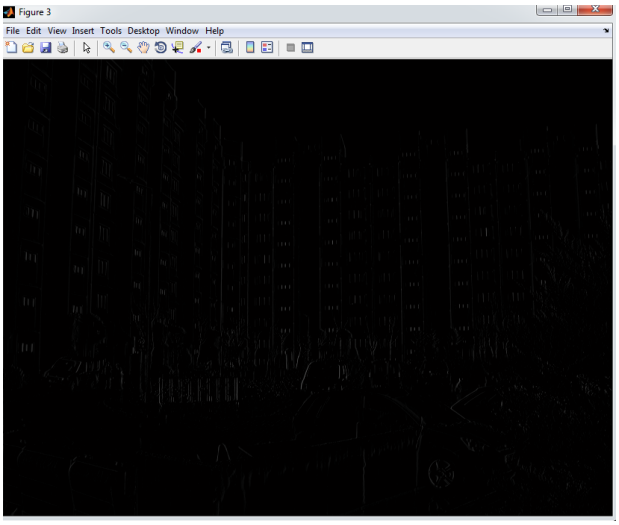


Figure 3
File Edit View Insert Tools Desktop Window Help

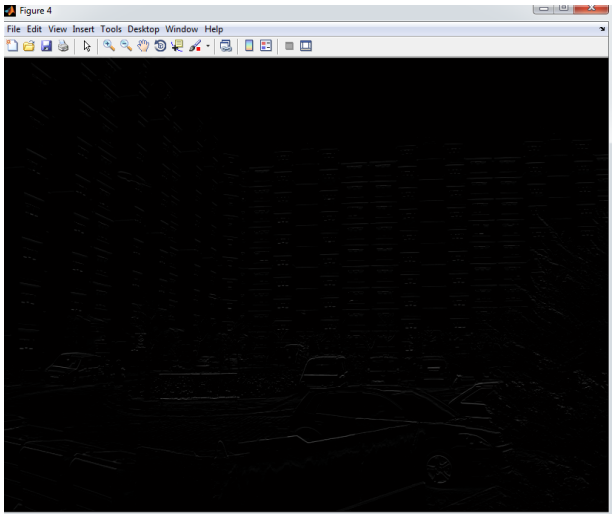


Figure 4
File Edit View Insert Tools Desktop Window Help

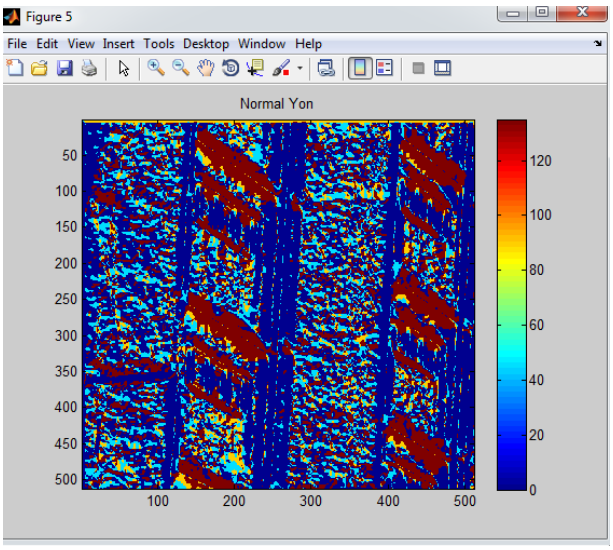


Figure 5
File Edit View Insert Tools Desktop Window Help
Normal Yon
50
100
150
200
250
300
350
400
450
500
100
200
300
400
500
120
100
80
60
40
20
0

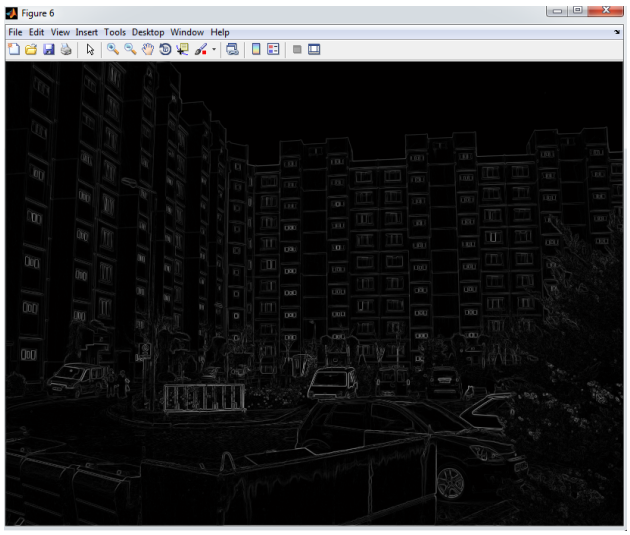
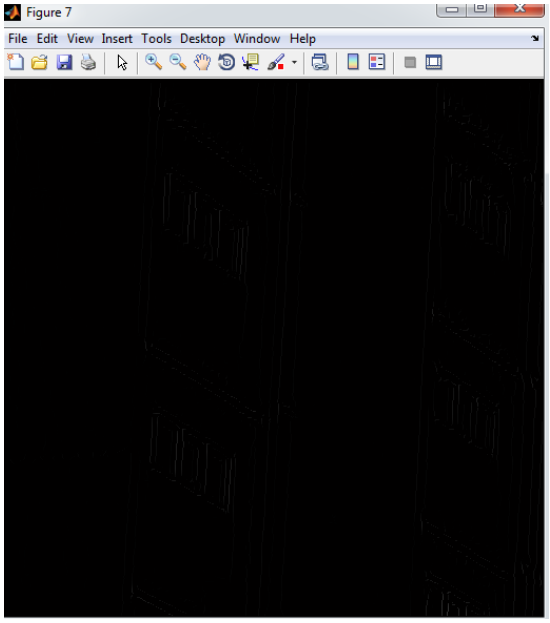
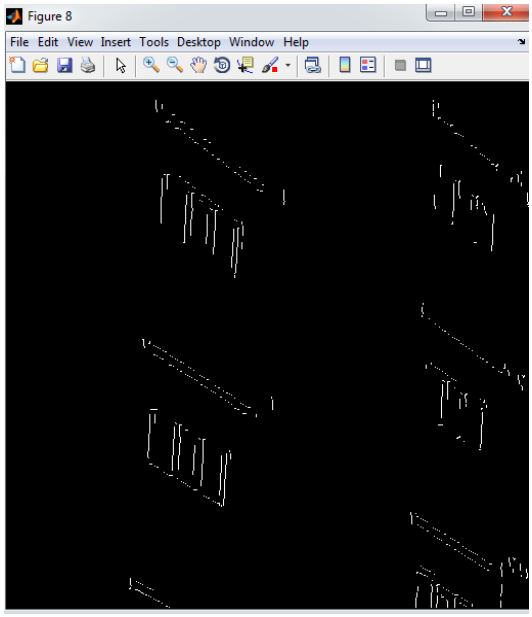


Figure 6
File Edit View Insert Tools Desktop Window Help





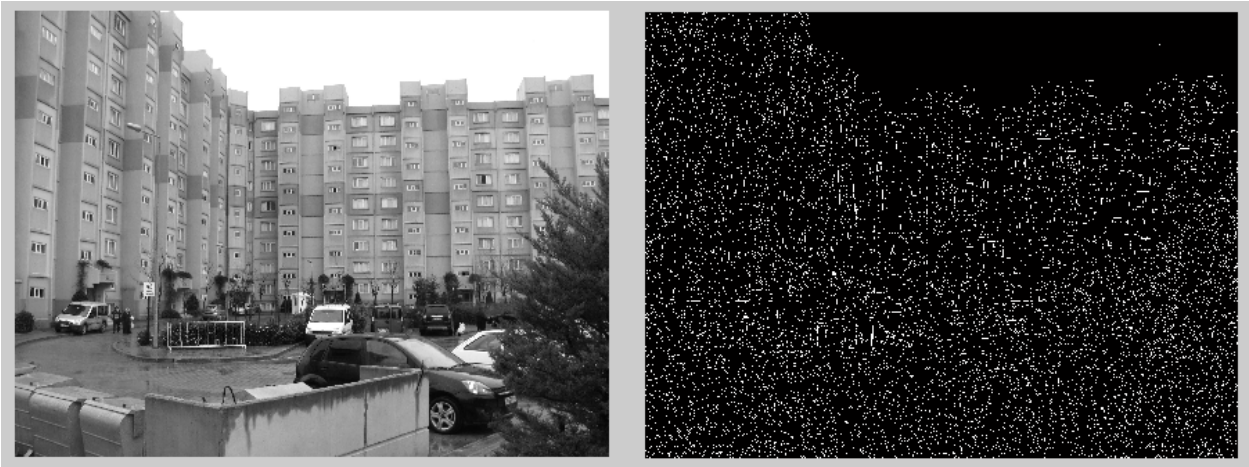
b) Asagidaki ilgili Haralick Algoritmasi ile birlikte elimizdeki image goruntusunun kenarlari bulunmustur.

1. Find $k_1, k_2, k_3, \ldots, k_{10}$ using least square fit, or masks given in Figure 2.8.
2. Compute θ, $\sin\theta$, $\cos\theta$.
3. Compute C_2, C_3.
4. If $C_3 < 0$ and $|\frac{C_2}{3C_3}| < \rho_0$ then that point is an edge point.

Elimizdeki image'in haralick edge detection methoduna ait bi-kubik algoritmasina gore cozuyoruz. Oncelikle image'imizi okutuyoruz, sonrasinda her piksel icin bi-kubik polinom katsayisina gore maskelemeyi hazirliyoruz. Gradyan icin ikinci ve ucuncu turevlerini belirliyoruz. Gradyan icin hesapladigimiz ikinci turev sifira esitken, ucuncu turevde negatiftir. Gaussian filtreleme kullanarak orjinal goruntuyu ayirt ederiz. Ilgili image goruntusune ait matlab kodu ve ekran goruntusu asagidadir.

```
function [Edges]=haralickKenarTespitYontemi(image, esikDegeri)

close all;
clear all;
clc;

    image=imread('armutlu.jpg');
    esikDegeri=0.5;
    image = double(rgb2gray(image));
    image = double(image);

imageGaussian = fspecial('gaussian', 15, 2.5);
imageGaussianFiltreleme=imfilter(double(image), imageGaussian,
'replicate','conv');
Maske=maskeOlustur();
```

```
imageMaske=cell(10);

for i=2:10
    imageMaske{i}=zeros(size(image));
    imageMaske{i}=imfilter(double(imageGaussianFiltreleme), Maske{i},
'replicate','conv');
end

Maske2=imageMaske{2};
Maske3=imageMaske{3};
Maske4=imageMaske{4};
Maske5=imageMaske{5};
Maske6=imageMaske{6};
Maske7=imageMaske{7};
Maske8=imageMaske{8};
Maske9=imageMaske{9};
Maske10=imageMaske{10};

KNormInv=1.0./(sqrt(Maske2.*Maske2+Maske3.*Maske3)+eps);
sinTheta=Maske2.*KNormInv;
cosTheta=Maske3.*KNormInv;
C2=Maske4.*sinTheta.*sinTheta+Maske5.*sinTheta.*cosTheta+Maske6.*cosTheta.*co
sTheta;
C3=Maske7.*sinTheta.*sinTheta.*sinTheta+Maske8.*sinTheta.*sinTheta.*cosTheta+
Maske9.*sinTheta.*cosTheta.*cosTheta+Maske10.*cosTheta.*cosTheta.*cosTheta;

ind=find(C3<0 & abs(C2./(3*C3))<esikDegeri);
Edges=zeros(size(image));
Edges(ind)=255;
figure,subplot(1,2,1), imshow(uint8(image),[]);
subplot(1,2,2), imshow(uint8(Edges),[]);

end

function [Maske]=maskeOlustur()

Maske=cell(10);
Maske{1}=(1/175)*[-13 2 7 2 -13; 2 17 22 17 2 ; 7 22 27 22 7;2 17 22 17 2 ; -
13 2 7 2 -13];
Maske{2}=(1/420)*[31 -5 17 -5 31; -44 -62 -68 -62 -44; 0 0 0 0 0; 44 62 68 62
44 ; -31 5 17 5 -31];
Maske{3}=Maske{2}';
Maske{4}=(1/70)*[2 2 2 2 2; -1 -1 -1 -1 -1; -2 -2 -2 -2 -2; -1 -1 -1 -1 -1; 2
2 2 2 2];
Maske{5}=(1/100)*[4 2 0 -2 -4; 2 1 0 -1 -2; 0 0 0 0 0; -2 -1 0 1 2; -4 -2 0 2
4];
Maske{6}=Maske{4}';
Maske{7}=(1/60)*[-1 -1 -1 -1 -1; 2 2 2 2 2; 0 0 0 0 0; -2 -2 -2 -2 -2; 1 1 1
1 1];
Maske{8}=(1/140)*[-4 -2 0 2 4; 2 1 0 -1 -2; 4 2 0 -2 -4; 2 1 0 -1 -2; -4 -2 0
2 4];
Maske{9}=Maske{8}';
Maske{10}=Maske{7}';

end
```



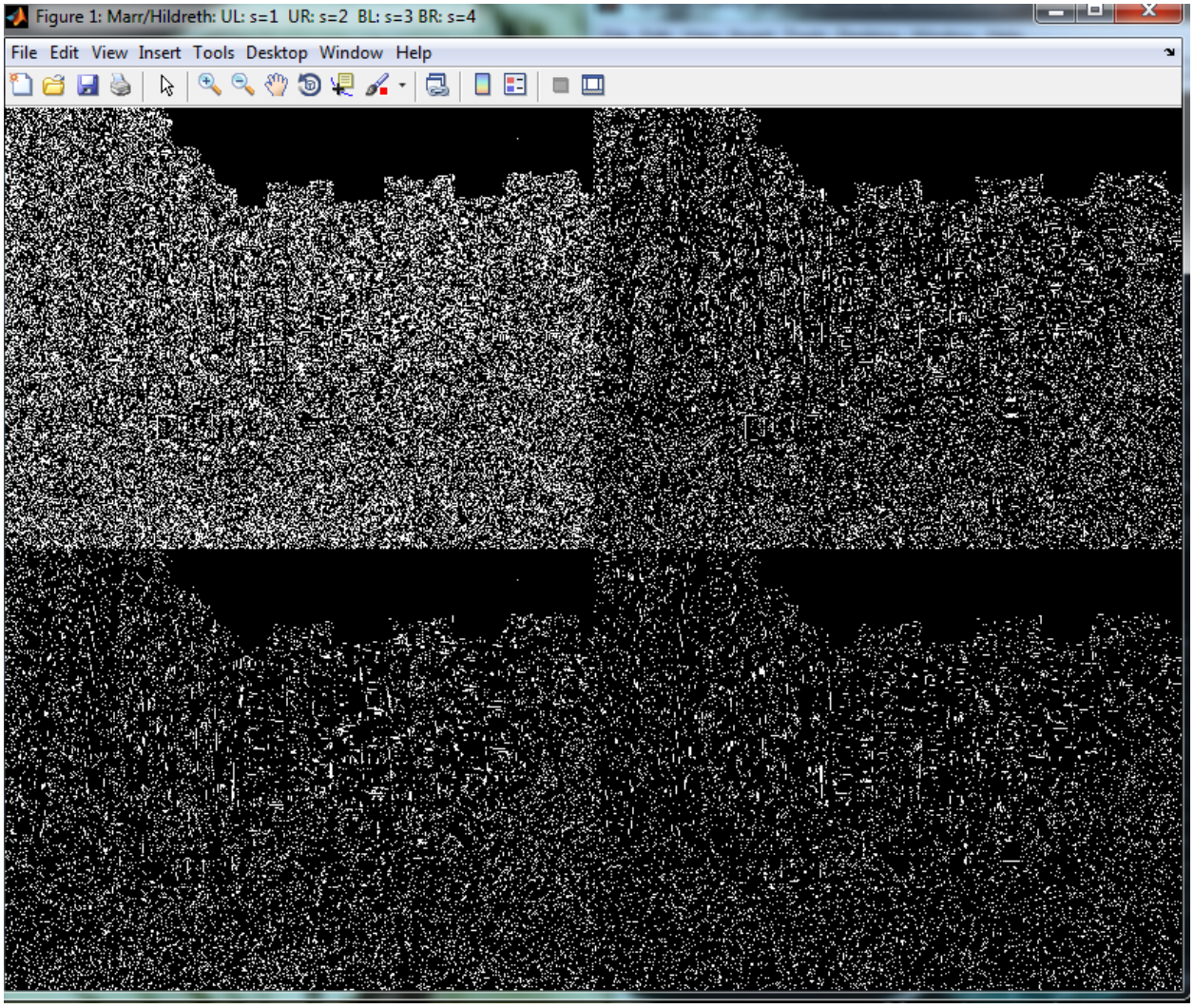
d) Asagidaki ilgili Laplacian Of Gaussian (LOG) Algoritmasi ile birlikte elimizdeki image goruntusunun kenarlari bulunmustur.

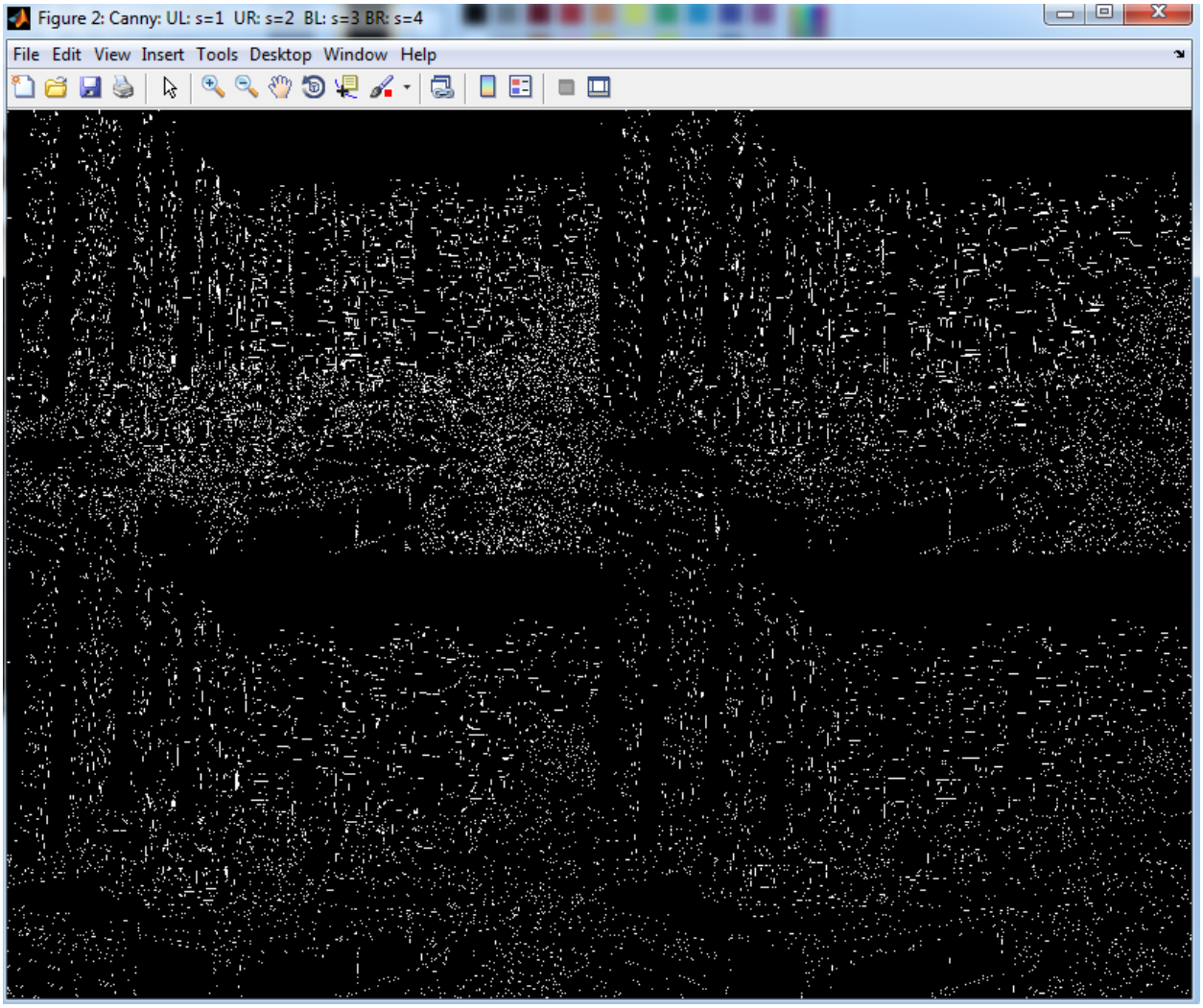
1. Generate a mask for LG for a given σ using equation 2.34.
2. Apply mask to the image.
3. Detect Zerocrossings.
   (a) Scan along each row, record an edge point at the location of zerocrossing.
   (b) repeat step (a) along each column.

Elimizdeki Image'in Marr ve Hildreth'in Log kenar bulma yontemi ile cozuyoruz. Bu cozume ait matlab komutlari ve ekran goruntuleri asagidadir.

```
close all;
clear all;
clc;

image = rgb2gray(imread('armutlu.jpg'));

marrHildreth1 = edge(image,'log',0,1.0);
marrHildreth2 = edge(image,'log',0,2.0);
marrHildreth3 = edge(image,'log',0,3.0);
marrHildreth4 = edge(image,'log',0,4.0);

goruntu1 = [ marrHildreth1 marrHildreth2; marrHildreth3 marrHildreth4];
log = figure('Name','Marr/Hildreth: UL: s=1  UR: s=2  BL: s=3 BR: s=4');
iptsetpref('ImshowBorder','tight');
imshow(goruntu1,'InitialMagnification',100);

[Canny1, CannytDetection1] = edge(image,'canny',[],1.0);
[Canny2, CannytDetection2] = edge(image,'canny',[],2.0);
[Canny3, CannytDetection3] = edge(image,'canny',[],3.0);
```

```
[Canny4, CannytDetection4] = edge(image,'canny',[],4.0);

k = 0.75
Canny1 = edge(image,'canny',k*CannytDetection1,1.0);
Canny2 = edge(image,'canny',k*CannytDetection2,2.0);
Canny3 = edge(image,'canny',k*CannytDetection3,3.0);
Canny4 = edge(image,'canny',k*CannytDetection4,4.0);

goruntu2 = [ Canny1 Canny2; Canny3 Canny4 ];
canny = figure('Name','Canny: UL: s=1  UR: s=2  BL: s=3 BR: s=4');
iptsetpref('ImshowBorder','tight');
imshow(goruntu2,'InitialMagnification',100);
```





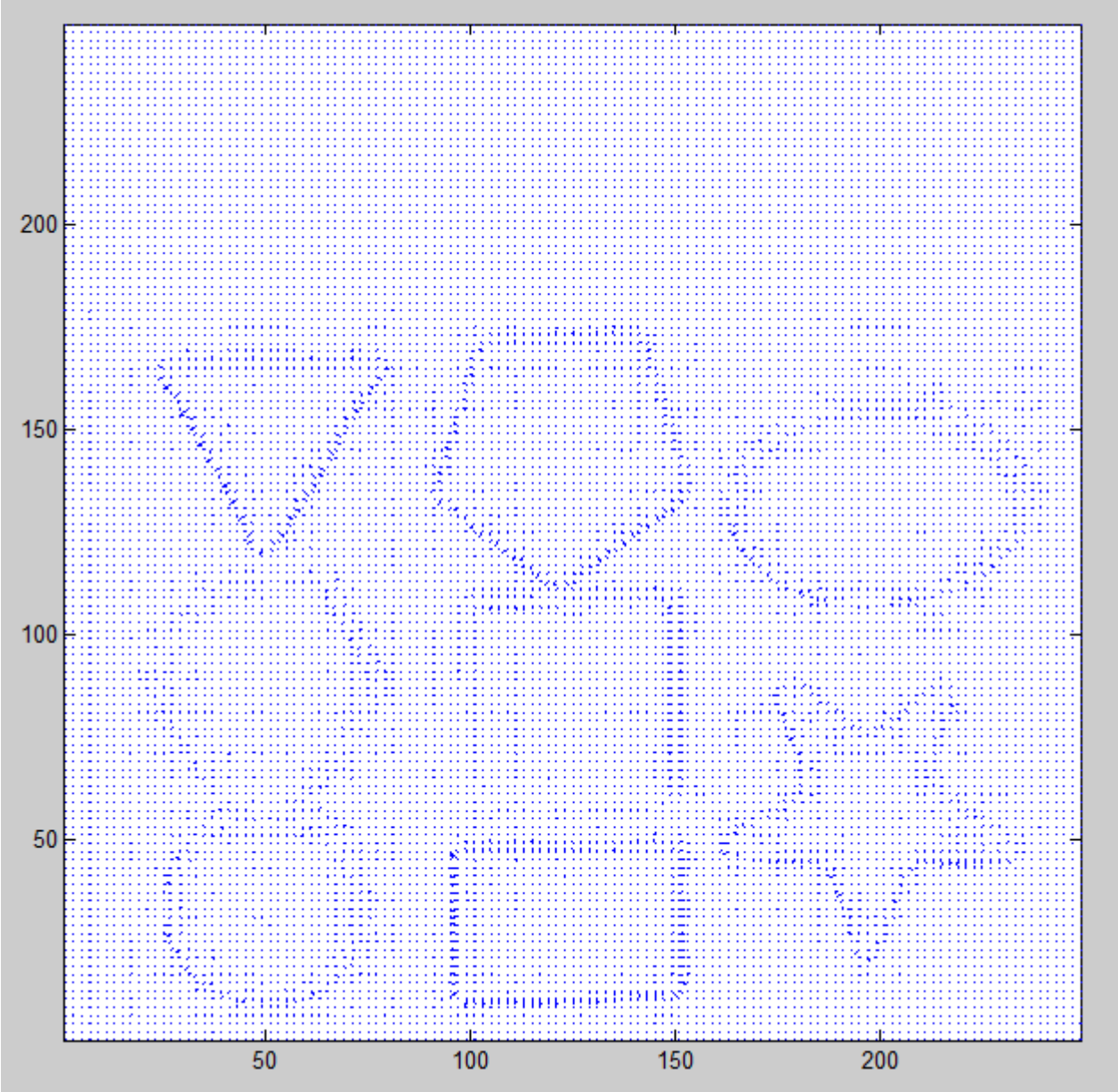
2) Elimizdeki armutlu.jpg image'inin Edge komutunun Roberts, Previtt, Sobel, Log ve Canny yontemlerini kullanarak cozumlenmis matlab kodlari ve ekran goruntuleri asagidadir.

```
clc;
clear all;
close all;

image = imread('armutlu.jpg');
image_gray = rgb2gray(image);

imageRobert = edge(image_gray,'roberts');
imagePrewitt = edge(image_gray,'prewitt');
imageSobel = edge(image_gray, 'sobel');
imageLog = edge(image_gray, 'Log');
imageCanny = edge(image_gray, 'Canny');

subplot(2,4,1), imshow(image), title('Image - Orjinal');
subplot(2,4,3), imshow(image_gray),title('Image - Gray');
subplot(2,4,4), imshow(imageRobert),title('Image - Robert');
```

```
subplot(2,4,5), imshow(imagePrewitt),title('Image - Prewitt');
subplot(2,4,6), imshow(imageSobel),title('Image - Sobel');
subplot(2,4,7), imshow(imageLog),title('Image - Log');
subplot(2,4,8), imshow(imageCanny),title('Image - Canny');
```

3) Elimizdeki sekiller.jpg image'inin Gradient ve Quiver methodlari kullanilarak Mag - Gradyan buyuklugu resminin olusturulmustur. Buna ait matlab kodu ve ekran goruntuleri asagidadir.

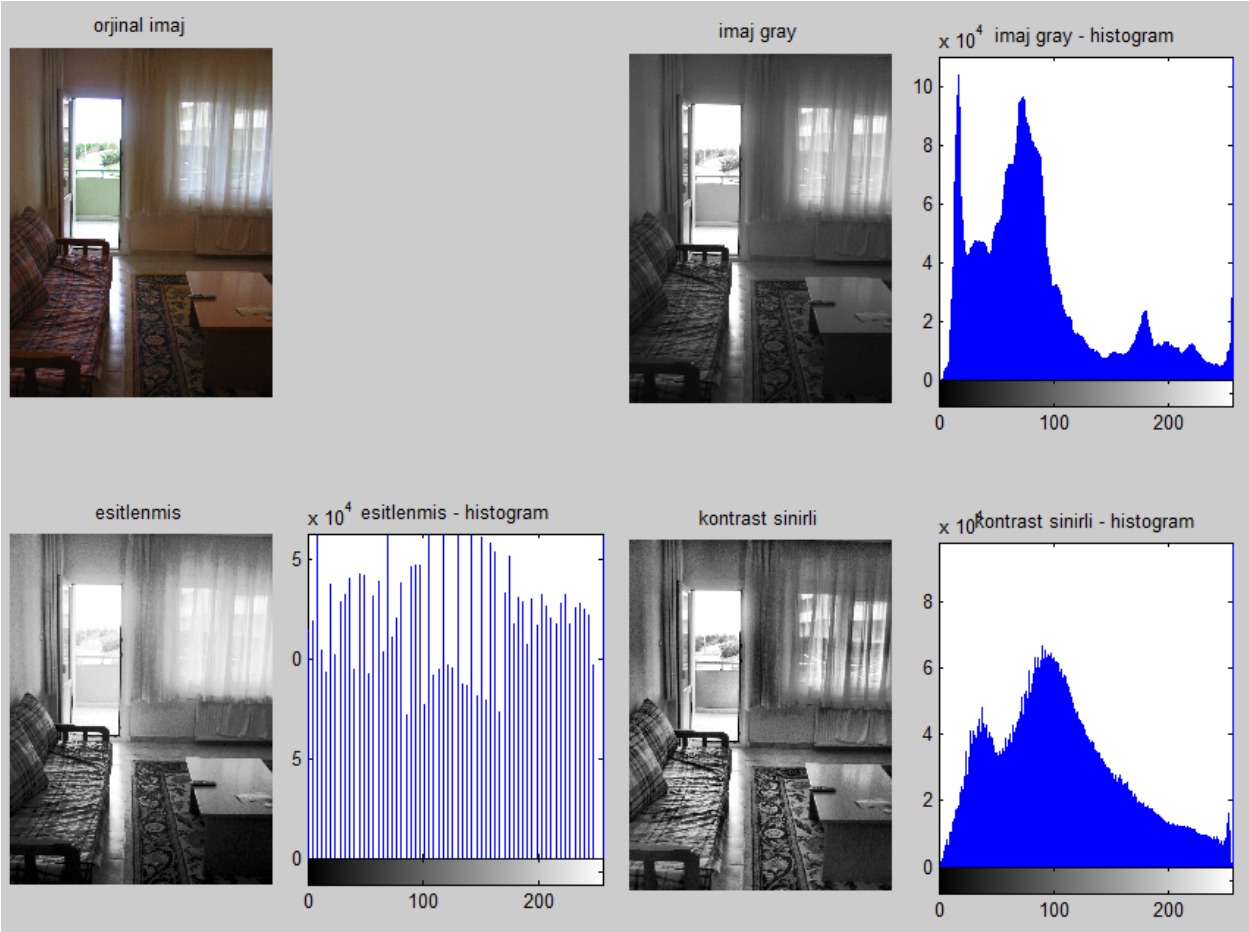
```
clc;
clear all;
close all;

image = imread('sekiller.jpg');
figure, imshow(image);
image = double(image(:,:,1));

[Fx,Fy] = gradient(image);
xspace = (1:2:size(image,2));
yspace = (1:2:size(image,2));

qx = interp2(Fx,xspace,yspace');
qy = interp2(Fy,xspace,yspace');

figure;
quiver(xspace,yspace,qx,qy);
axis image;
```



4) Elimizdeki oda.jpg Image'nin imhist ve adapthisteq komutlari ile urettigi sonuclar matlab komutlari ve ekran goruntuleri ile birlikte asagidadir.

```
clc;
clear all;
close all;

image = imread('oda.jpg');
image_gray = rgb2gray(image);

image_histogram = histeq(image_gray);

image_adapthisteq = adapthisteq(image_gray);
```

```
subplot(2,4,1), imshow(image), title('orjinal imaj');
subplot(2,4,3), imshow(image_gray), title('imaj gray');
subplot(2,4,4), imhist(image_gray);,title('imaj gray - histogram');
subplot(2,4,5), imshow(image_histogram), title('esitlenmis');
subplot(2,4,6), imhist(image_histogram);,title('esitlenmis - histogram');
subplot(2,4,7), imshow(image_adapthisteq), title('kontrast sinirli');
subplot(2,4,8), imhist(image_adapthisteq);,title('kontrast sinirli -
histogram');
```



5) Elimizdeki kompozit.tif isimli image'in Strel komutu ile image'in morfolojik islemlerden open, close, erode ve dilate birlikte kullanilarak matlab komutlari ile cozumlenmesi ve ekran goruntuleri asagidadir.

```
clc;
clear all;
close all;

image = imread('kompozit.tif');
figure, imshow(image), title('orjinal imaj');

imageBackground = 255 - image;
figure, imshow(imageBackground), title('imajin arka plani');

se = strel('disk', 20);
```

```
erode = imerode(imageBackground, se);
figure, imshow(erode), title('imajin asindirilmis hali');
dilate = imdilate(imageBackground, se);
figure, imshow(dilate), title('imajin genisletilmis hali');
opening = imopen(imageBackground, se);
figure, imshow(opening), title('imajin acilmis hali');
closing = imclose(imageBackground, se);
figure, imshow(closing), title('imajin kapanmis hali');
subplot(331), imshow(image), title('orjinal imaj');
subplot(332), imshow(imageBackground), title('imajin arka plani');
subplot(333), imshow(erode), title('imajin asindirilmis hali');
subplot(334), imshow(dilate), title('imajin genisletilmis hali');
subplot(335), imshow(opening), title('imajin acilmis hali');
subplot(336), imshow(closing), title('imajin kapanmis hali');
```