

# INTERACTIVE IMAGE SEGMENTATION

**Hazırlayan**

Zafer CÖMERT

2013, Mayıs

# PROGRAMIN TANIMI

Program çalıştırıldığında kullanıcıdan gri ölçekli bir resim seçmesi istenmektedir. Kullanıcıya seçilen resim gösterilecek Foreground (Ön Plan) ve Background (Arka Plan) değerlerini interaktif olarak belirlemesi sağlanacaktır. Elde edilen bu değerler ile Foreground ve Background değerleri Gauss Filtresinden geçirilecektir. Program Foreground ve Background değerlerini grafiksel olarak gösterecektir. Daha sonra resim binary formatına getirilecek ve Geodesic Distance ile her bir noktanın arka plana ve ön plana olan uzaklıkları hesaplanacak ve grafiksel olarak gösterilecektir. Son aşamada ise elde edilen binary mask, geodesic distance hesaplamasında sonra güncellenecek ve Interactive Image Segmentation tamamlanacaktır.

## 1. KULLANICININ RESİM SEÇMESİNİ SAĞLAMAK

Kullanıcıda bir resim seçmesini sağlamak için MATLAB Grafik User Interface (GUI)'den faydalanılmıştır ve sadece uzantısı JPEG olan dosyaların seçilmesi sağlanmıştır.

```
%% Select an image

FileName, PathName] = uigetfile('*.jpg', 'Select an image for
Segmentation Please select an gray scale image');

FullPath = strcat(PathName, FileName);

I = double(imread(FullPath));
```

Seçilen resim gri ölçekli değilse programın sonlandırılması sağlanmıştır.

```
%% Preliminary

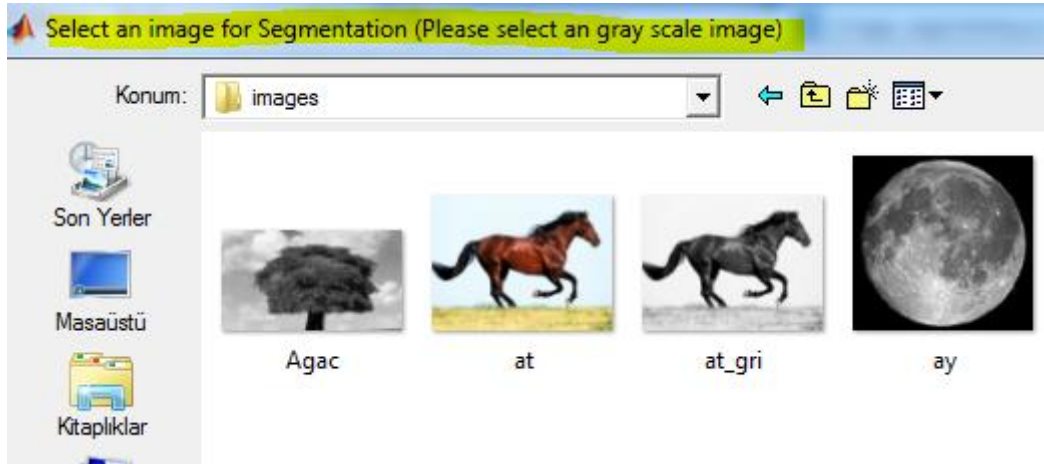
[x y z] = size(I);

if(z>1)

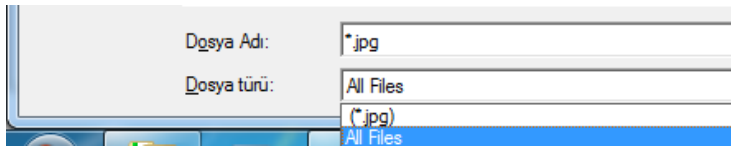
    clear all, close all, clc

    error('Please select a gray scale image...');

end
```



Şekil 1 Resmin Seçilmesini Sağlanmıştır



Şekil 2 İzin Verilen Dosya Formatları

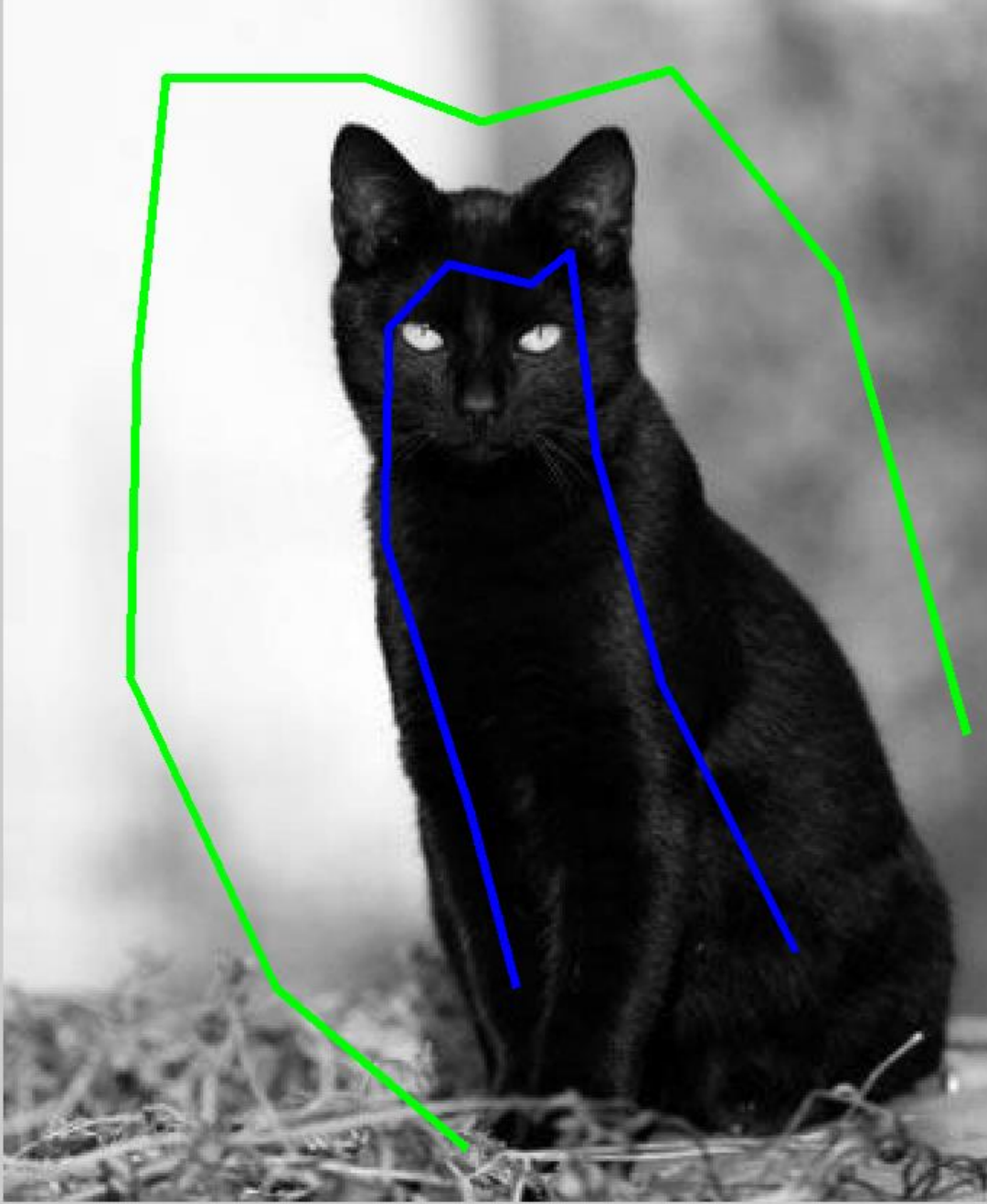
## 1. FOREGROUND VE BACKGROUN DEĞERLERİNİN ALINMASI

Foreground (F) ve Background (B) değerleri kullanıcı tarafından belirlenmektedir. Bunun için kullanıcıdan F ve B değerleri için 10 noktaya tıklanması istenmiştir. Bu noktaların koordinat değerleri bir matriste tutulmuş ve daha sonra koordinat değerleri üzerinden resim üzerindeki renk değerleri sayısal olarak elde edilmiştir.

```
%% USER INTERACTIVE
```

```
message = sprintf('Please click 10 point for ForeGround');  
uiwait(msgbox(message));  
  
disp('Click ForeGround (10 point)');  
  
[xpf ypf] = ginput(10);  
  
ft = abs(int16([xpf ypf]));  
  
fg = FindCoordinateValue (I,ft);  
  
hold on  
  
line(xpf,ypf,'color','b','LineWidth',5);
```

Please select Foreground and Background



## 2. KOORDİNAT DEĞERLERİNİN ELDE EDİLMESİ

Kullanıcı etkileşimi ile elde edilen veriler koordinat değerleridir. Koordinat değerlerine resim üzerinde karşılık gelen sayısal değerleri elde etmek için ***FindCoordinateValue.m*** isimli bir dosya oluşturulmuştur. Resmin sayısal değerlerini ve koordinat değerlerini parametre olarak alan bu fonksiyon geriye bir vektör dönerek tıklanan koordinat değerlerine karışık resimdeki gerçek değerlerin dönemsini sağlamıştır.

```

function y = FindCoordinateValue(I,A)

%% To obtain the coordinate values

% I image

% A the coordinate A(x,y)

    [m n] = size(I);

    [x y] = size(A);

    disp(sprintf('Resolution :  %dx%dpx', n, m));

    counter = 1;

    for i=1:x

        y(counter) = I(A(i,2),A(i,1));

        counter=counter+1;

    end

end

```

### 3. GAUSS FİLTRE

Kullanıcı etkileşimi ile elde edilen veriler Gauss Filtresinden geçirilmiştir. Gauss fonksiyonu aşağıdaki belirtilen şekildedir.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Gauss Fonksiyonu ile istenilen hesaplamaların yapılabilmesi için kullanıcıdan alınan Foreground ve Background değerlerinin ortalama ve standart sapma değerleri hesaplanmalıdır.

```

%% GAUSS FILTER

k = graythresh(I);

% Gauss1 filter for ForeGround

mf = mean(fg);

sf = std(fg);

% Gauss filter for BackGround

mb = mean(bg);

sb = std(bg);

fy = (1/sqrt(2*pi*sf))*exp(0.5 * (I-mf).^2/(sf^2));

by = (1/sqrt(2*pi*sb))*exp(0.5 * (I-mb).^2/(sb^2));

fg_maske = fy<by;

```

#### 4. FOREGROUND VE BACKGROUND GRAFİĞİNİN ÇİZDİRİLMESİ

F ve B değerleri çizdirilirken iki grafik sunulmuştur. Birinci grafik direk seçilen değerleri gösterirken diğer grafik bu değerlerin Gauss Fonksiyonu üzerindeki yerini göstermektedir.

```

%% Plot Fore/BackGround Graphics
message = sprintf('Click OK to see Foreground and Background Graphic');

uiwait(msgbox(message));

t=0:255;
yf = exp(-0.5*(t-mf)/sf);

yb = exp(-0.5*(t-mb)/sb);

```

## Grafiği çizdiren kod bloğu

```
figure(2)

subplot(1,2,1);

set(gcf, 'Position', get(0,'Screensize')); % Maximize
figure.

plot(fg,'b','LineWidth',5);

hold on

plot(bg,'g','LineWidth',5);

xlim([1 10]);

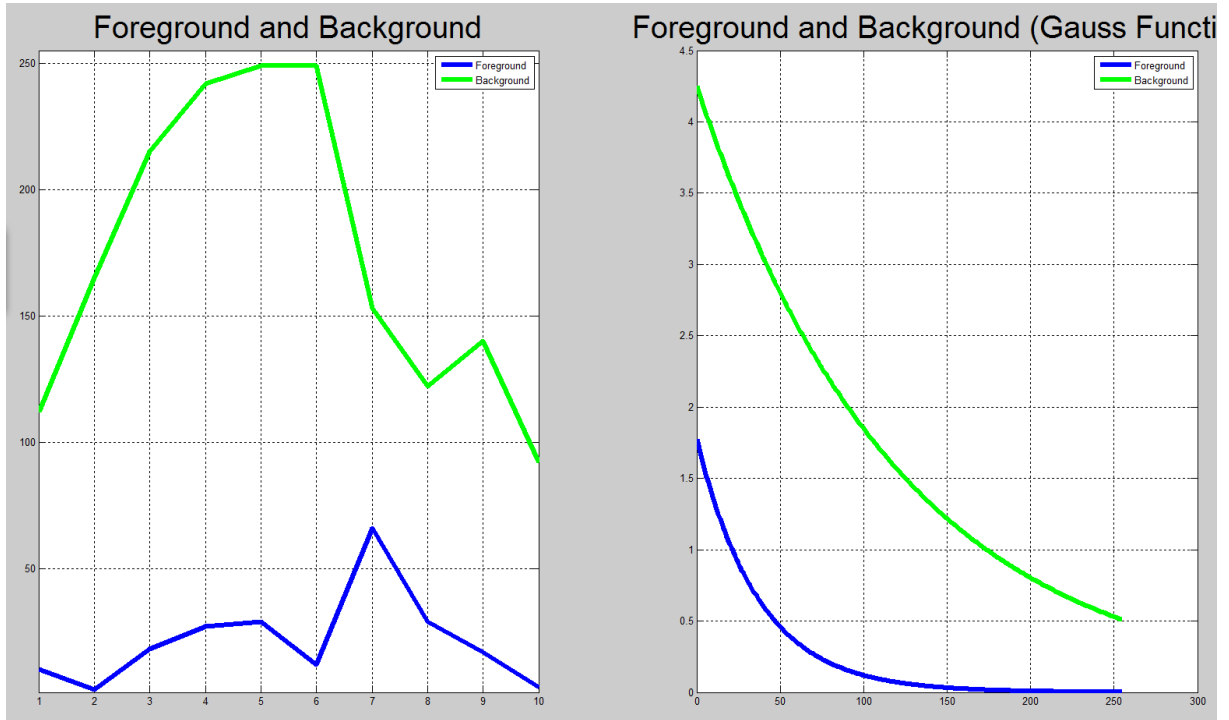
ylim([1 255]);

grid on

title('Foreground and Background','FontSize',30);

% Add a legend

legend('Foreground', 'Background');
```



Şekil 3 Foreground ve Background Grafiklerinin Çizdirilmesi

## 5. MASKENİN ÇİZDİRİLMESİ

F ve B değerleri Gauss Fonksiyonundan geçirildikten sonra maske elde edilir.

```
%% Show the Mask

message = sprintf('Please click OK to see Mask');

uiwait(msgbox(message));

figure(3);

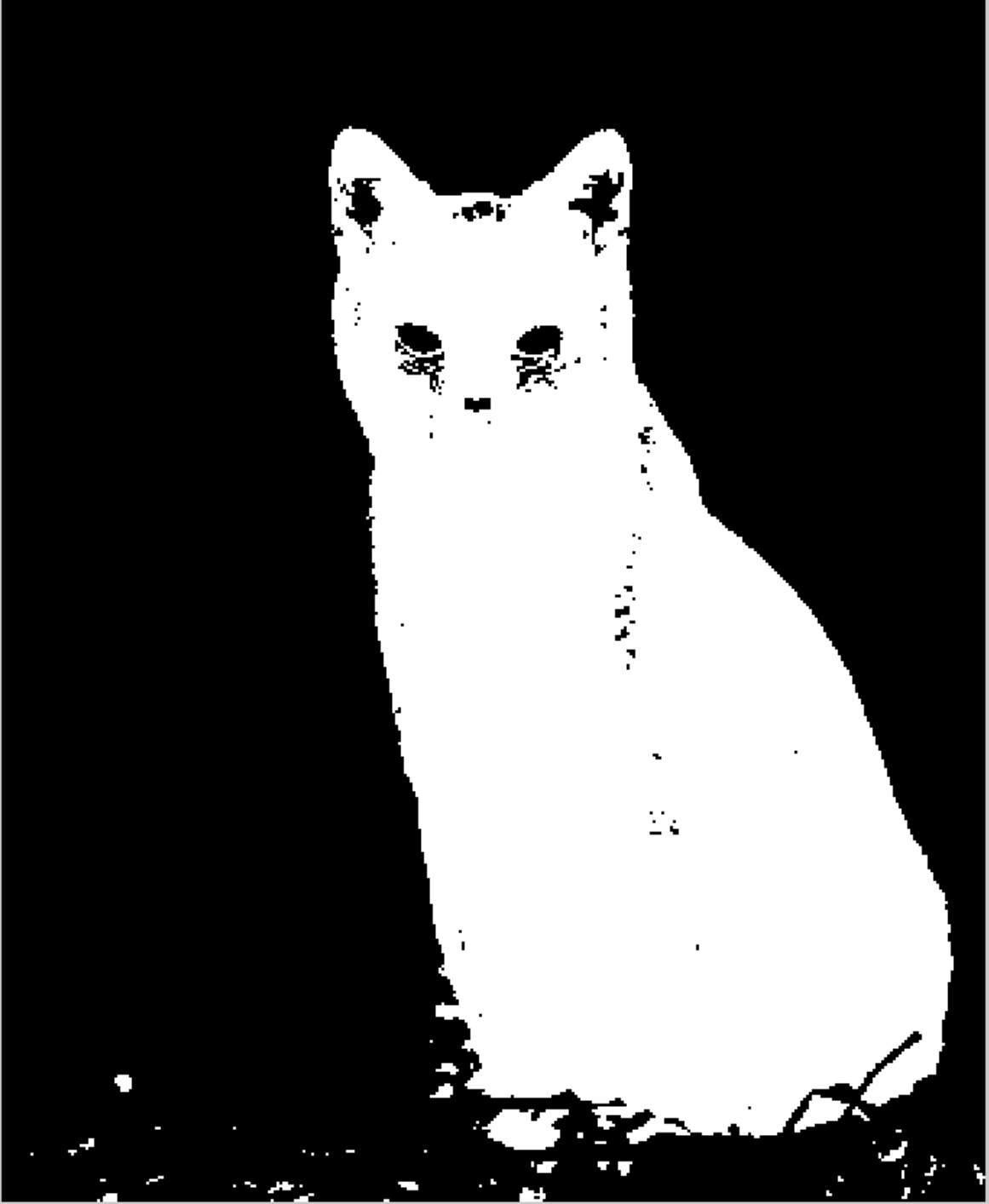
imshow(fg_maske, []);

set(gcf, 'Position', get(0, 'Screensize')); % Maximize
figure.

title('The Mask', 'FontSize', 30);
```



## The Mask



Şekil 4 Maskenin Gösterilmesi

### 6. GEODESIC DISTANCE

Geodesic Distance ile her bir noktanın arka plan ve ön plana olan uzaklığı hesaplanmaktadır. Geodesic Distance hesaplanırken Euclidean, City Block, Chessboard, Quasi-Euclidean metotları kullanılabilir.

## Örnek bir Geodesic Distance Uygulaması

```
% Geodesic

bw = zeros(200,200); bw(50,50) = 1; bw(50,150) = 1;

bw(150,100) = 1;

D1 = bwdist(bw,'euclidean');

D2 = bwdist(bw,'cityblock');

D3 = bwdist(bw,'chessboard');

D4 = bwdist(bw,'quasi-euclidean');

figure

subplot(2,2,1), subimage(mat2gray(D1)), title('Euclidean')

hold on, imcontour(D1)

subplot(2,2,2), subimage(mat2gray(D2)), title('City block')

hold on, imcontour(D2)

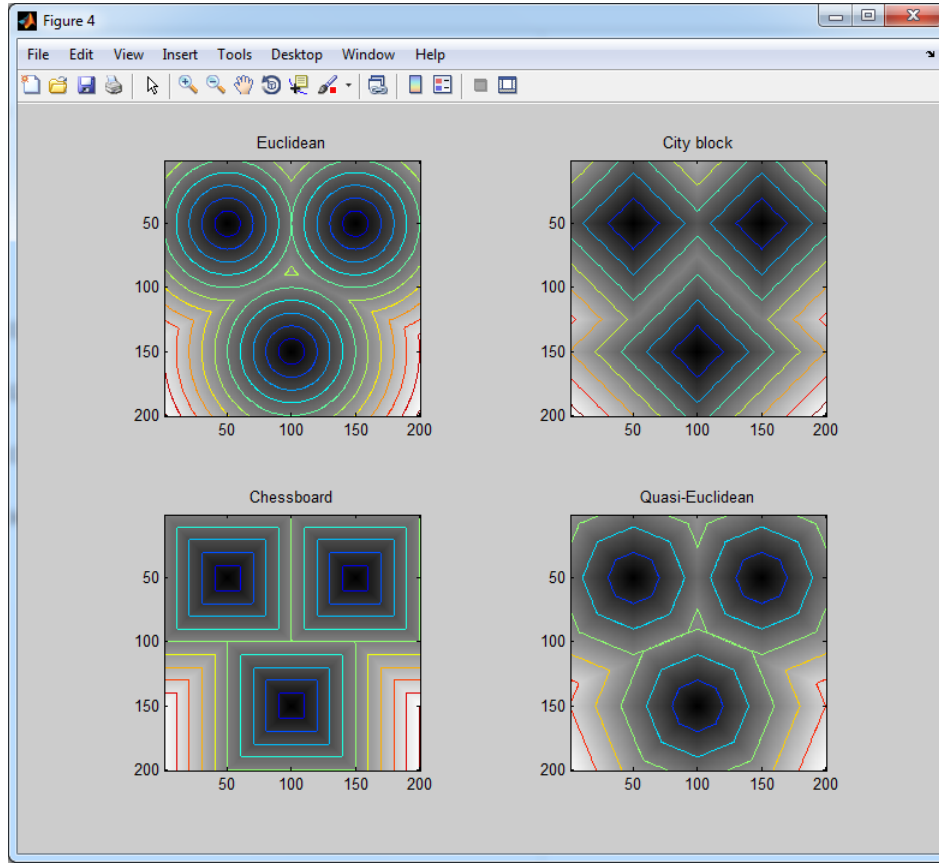
subplot(2,2,3), subimage(mat2gray(D3)), title('Chessboard')

hold on, imcontour(D3)

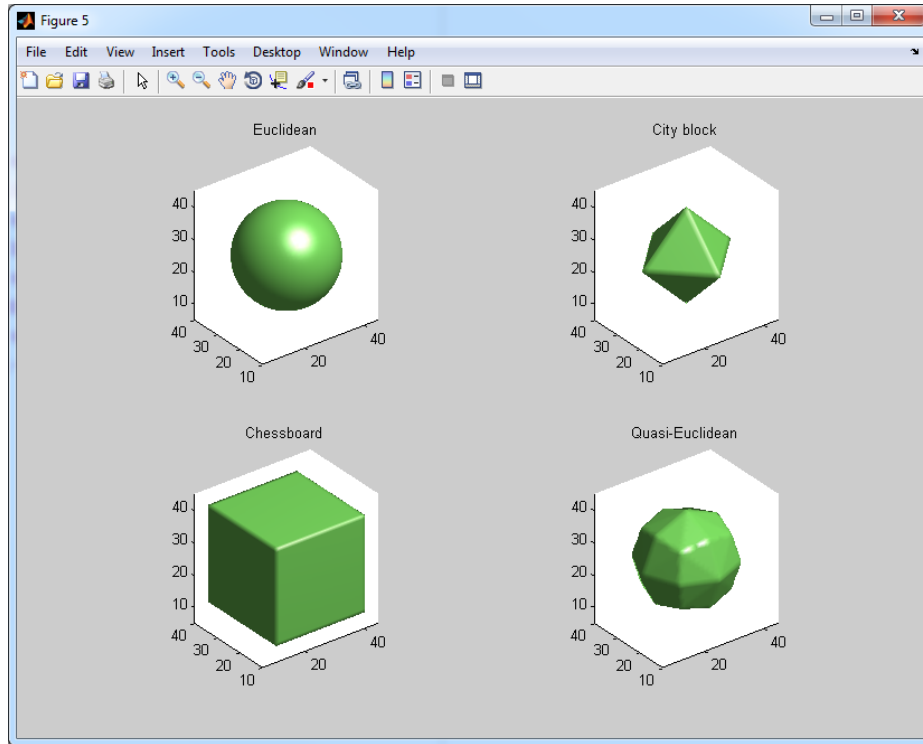
subplot(2,2,4), subimage(mat2gray(D4)), title('Quasi-
Euclidean')

hold on, imcontour(D4)
```

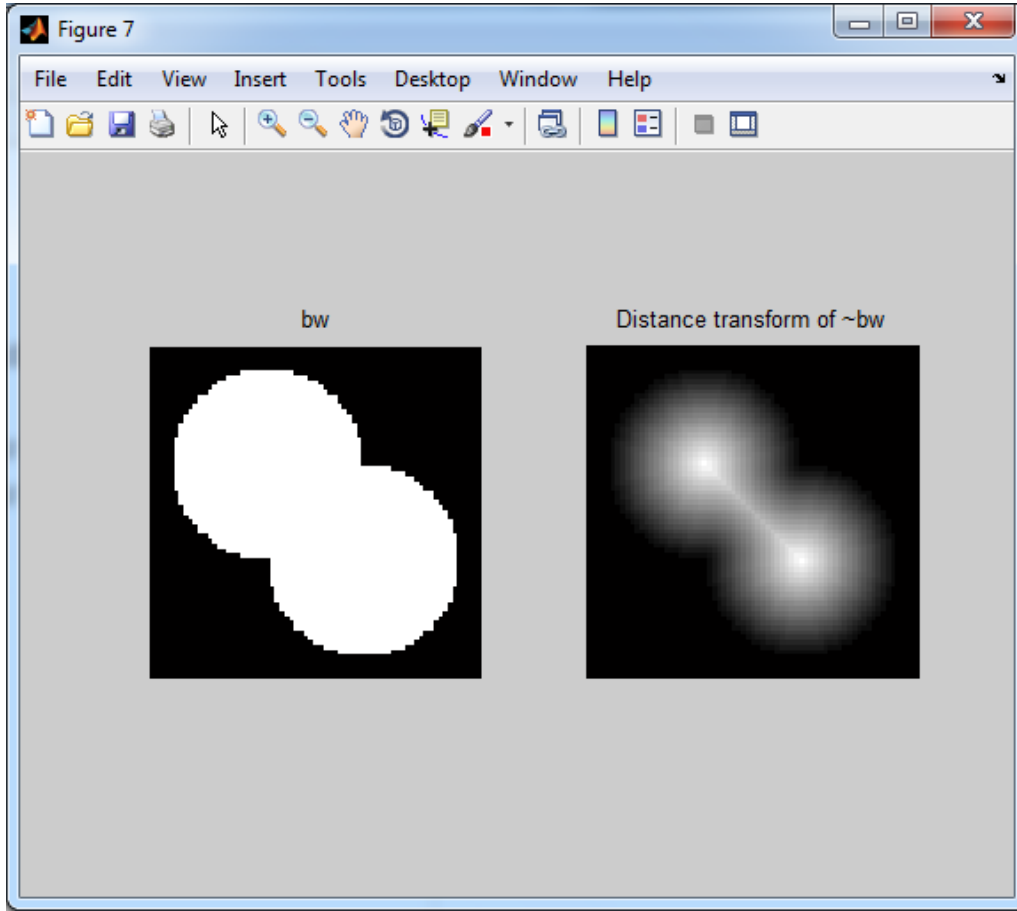
Yapılan araştırmalar neticesinde Geodesic Distance ile ilgili beş farklı uygulama kodlarla beraber eklenmiştir.



Şekil 5 Örnek Geodesic Distance Uygulaması (2)



Şekil 6 Örnek Geodesic Distance Uygulaması (3)



Şekil 7 Örnek Geodesic Distance Uygulaması (4)

Geliştirilen uygulama kapsamında bu dört metot uygulanmıştır.

```

%% Geodesic distance

    message = sprintf('Please click OK to see Geodesic
Distance');

    uiwait(msgbox(message));

    figure(4)

    set(gcf, 'Position', get(0,'Screensize')); % Maximize
figure.

    bw = fg_mask;

    D1 = bwdist(bw, 'euclidean');

    D2 = bwdist(bw, 'cityblock');

    D3 = bwdist(bw, 'chessboard');

    D4 = bwdist(bw, 'quasi-euclidean');


    subplot(2,2,1), subimage(mat2gray(D1)), title('Euclidean')
    hold on, imcontour(D1)

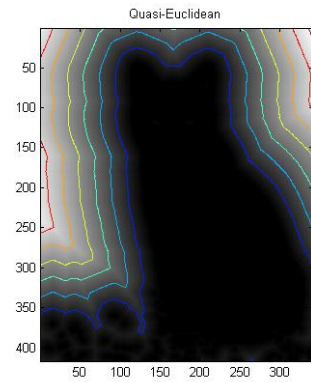
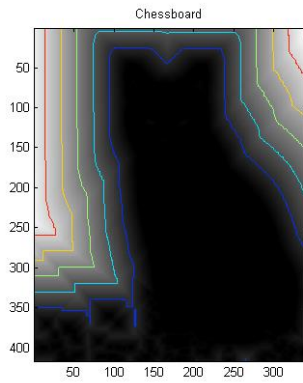
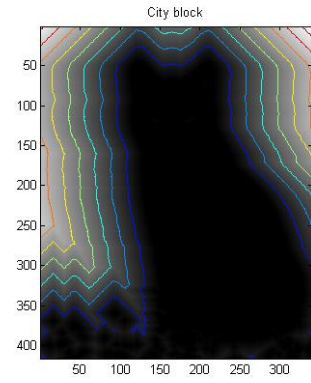
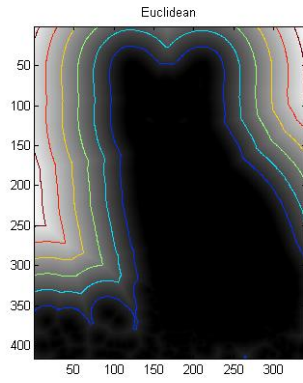
    subplot(2,2,2), subimage(mat2gray(D2)),
        title('City block')
    hold on, imcontour(D2)

    subplot(2,2,3),
        subimage(mat2gray(D3)), title('Chessboard')
    hold on, imcontour(D3)

    subplot(2,2,4), subimage(mat2gray(D4)), title('Quasi-
Euclidean')

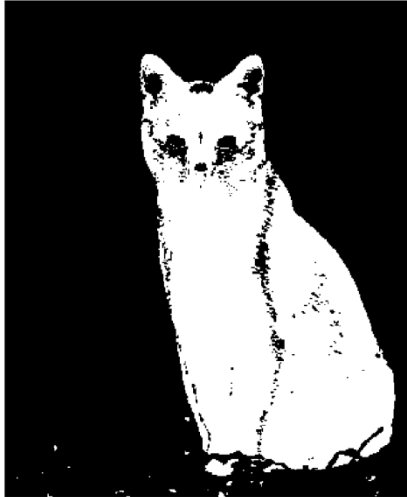
    hold on, imcontour(D4)

```



Şekil 8 Geodesic Distance

The Mask Before Geodesic Distance



The Mask After Geodesic Distance (Euclidean)



Şekil 9 Geodesic Distance'den sonra Maskenin Güncellenmiş Hali