Interactive Image Segmentation

Zafer CÖMERT

www.zafercomert.com

comertzafer@gmail.com

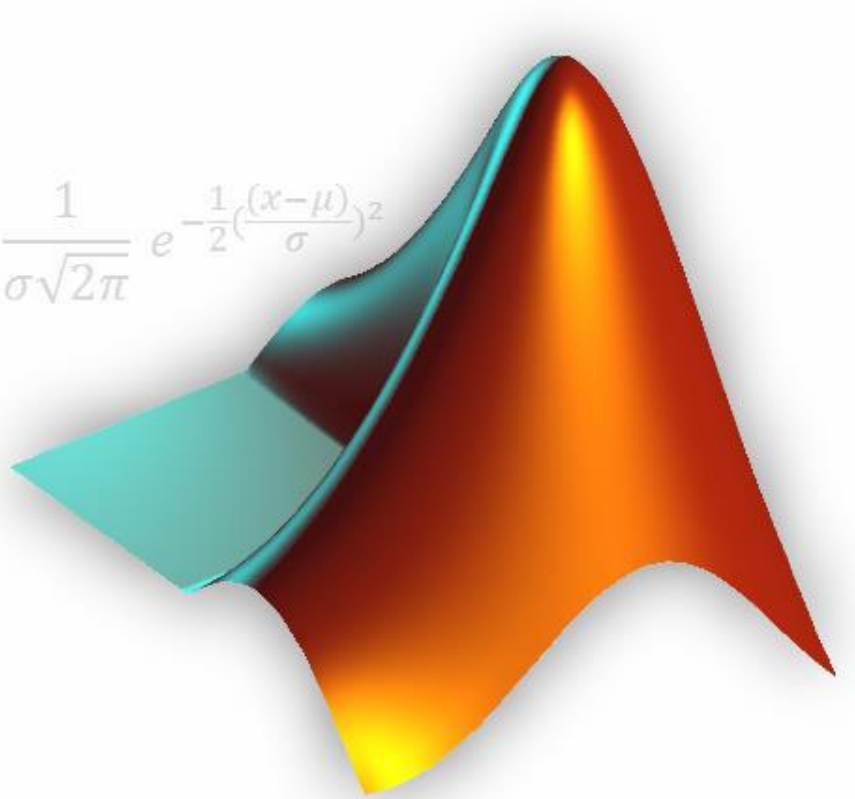


Image Segmentation

- Point, Line and Edge Detection
- Line Detection Using The Hough Transform
- Thresholding
- Region-Based Segmentation
- Segmentation Using the Watershed Transform
- Uygulama

Image Segmentation İmge Bölütleme

- İmge bölütleme, imgeyi birbiriyle çakışmayan fakat imgenin tümünü içeren alt imge gruplarına ayırma işlemidir.
- Bu gruplandırma işlemi imgenin belirli bir veya birden fazla özelliği dikkate alınarak gerçekleştirilebilmektedir.
- Genelde bölütlemenin teorik bir tabanı olmadığından bölütlemeye ilişkin standart bir yöntem bulunmamakta, sezgisel (ad-hoc) ya da probleme özgü yöntemlerle bölütleme işlemi gerçekleştirilmektedir.

Image Segmentation İmge Bölütleme

- İmge bölütleme bir imgeyi anlamlı parçalara yada nesnelere ayırma işlemidir ve otonom hedef takiplerinde oldukça önemli bir işleve sahiptir .

- Şimdiye kadar literatüre yüzlerce segmentasyon algoritması sunulmuştur . Tüm imgeler için iyi sonuç veren genel bir algoritma olmadığı gibi aynı algoritma farklı imgeler üzerinde farklı sonuçlar verebilmektedir.

Point Detection

- Resim içerisinde belirli noktaların izole edilmesini sağlamak amacıyla kullanılabilir.

R, bir maskeyi ifade eder.

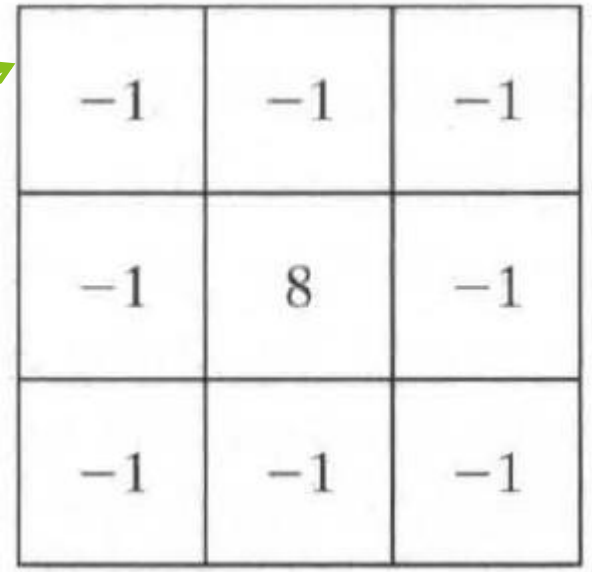
$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

$$|R| \geq T$$

- Burada T negatif olmayan bir sınır değeridir. imfilter fonksiyonu ile birlikte maskeler kullanılarak çözüm elde edilebilir.

Point Detection

```
g = abs(imfilter(tofloat(f), w)) >= T;
```



Line Detection

- Maskenin resim etrafında gezdirilmesi mantığı ile çalışır. Maskenin yöne tanımlanan değerler ile ayarlanabilir.

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

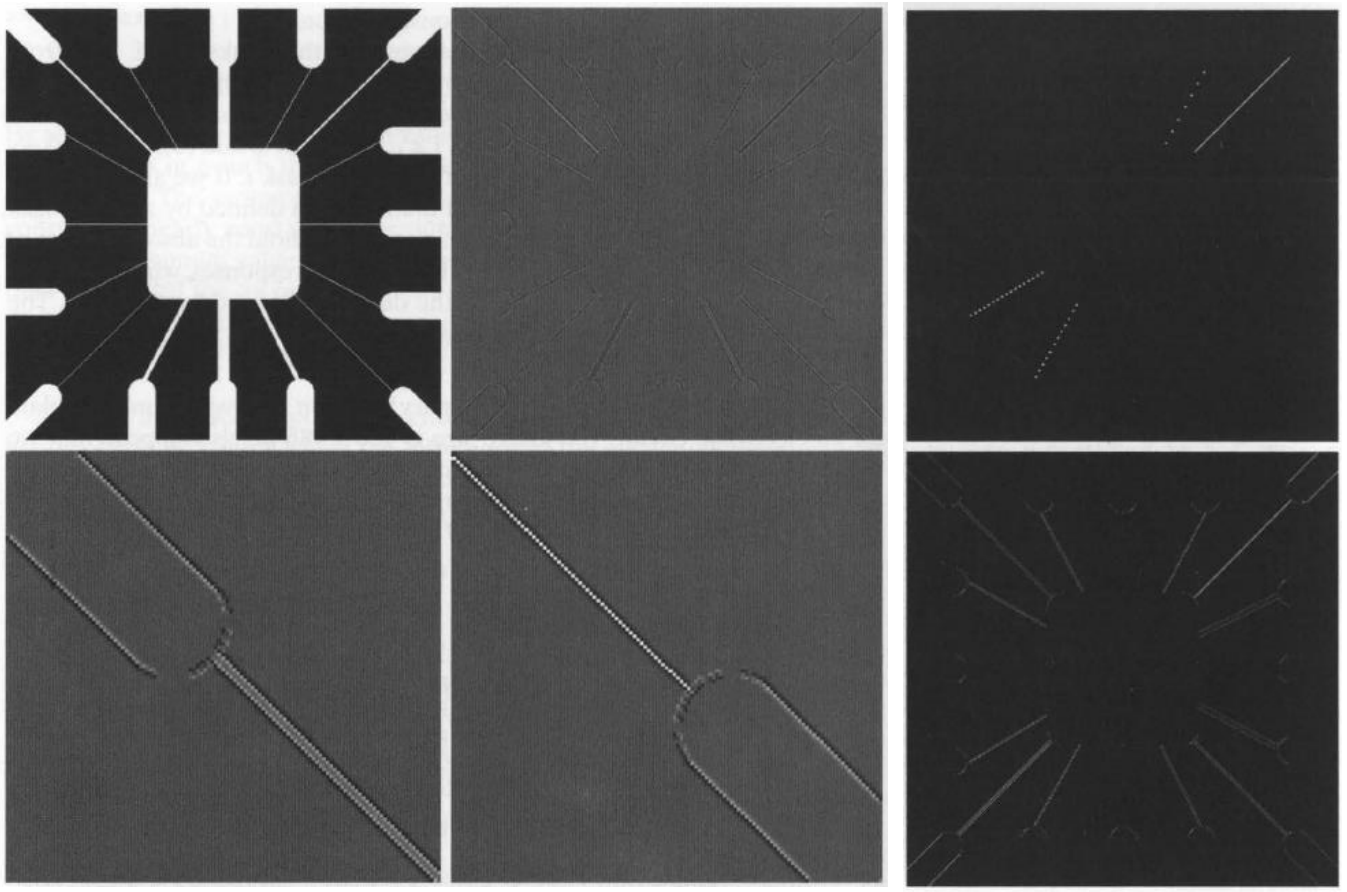
−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

Line Detection



(a) Image of a wire-bond template. (b) Result of processing with the +45o detector in Fig. (c) Zoomed view of the top, left region of (b). (d) Zoomed view of the bottom, right section of (b). (e) Absolute value of (b). (f) All points (in white) whose values satisfied the condition 9 >= T, where 9 is the image in (e). (The points in (f) were enlarged to make them easier to see.)

Edge Detection Using Function **edge**

- Point ve Line Detection görüntü işlemede son derece önemli olmasına rağmen Edge Detection yöntemi, görüntü işlemede yaygın olarak kullanılmaktadır.

Edge Detection Using Function **edge**

```
[g, t] = edge(f, 'method', parameters)
```

Edge Detector	Description
Sobel	Finds edges using the Sobel approximation to the derivatives in Fig. 11.5(b)
Prewitt	Finds edges using the Prewitt approximation to the derivatives in Fig. 11.5(c).
Roberts	Finds edges using the Roberts approximation to the derivatives in Fig. 11.5(d).
Laplacian of a Gaussian (LoG)	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a Laplacian of a Gaussian filter.
Zero crossings	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a specified filter.
Canny	Finds edges by looking for local maxima of the gradient of $f(x, y)$. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Therefore, this method is more likely to detect true weak edges.

Edge Detection Using Function **edge**

```
[g, t] = edge(f, 'sobel', T, dir)
```

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image neighborhood

−1	−2	−1
0	0	0
1	2	1

$g_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$

−1	0	1
−2	0	2
−1	0	1

$g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$

Sobel

−1	−1	−1
0	0	0
1	1	1

$g_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$

−1	0	1
−1	0	1
−1	0	1

$g_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

Prewitt

−1	0
0	1

$g_x = z_9 - z_5$

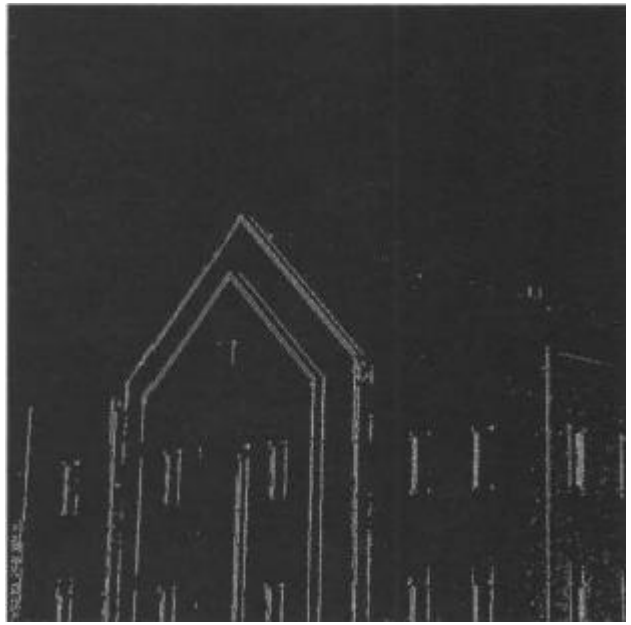
0	−1
1	0

$g_y = z_8 - z_6$

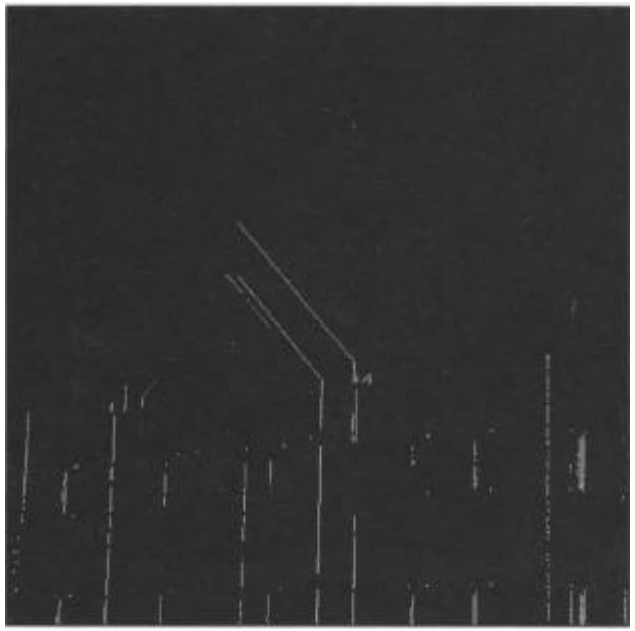
Roberts

Edge Detection Using Function **edge**

Original Image



Vertical Sobel Mask



Specified Threshold

Line Detection Using The Hough Transform

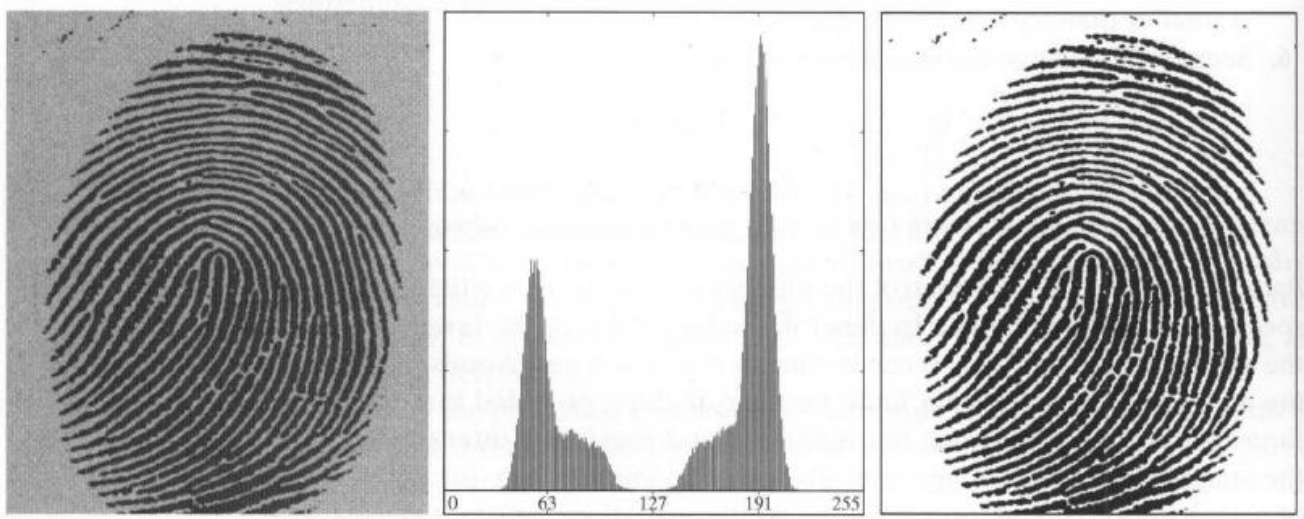
```
[H, theta, rho] = hough(f, 'ThetaRes', val1, 'RhoRes', val2)
```

Thresholding

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

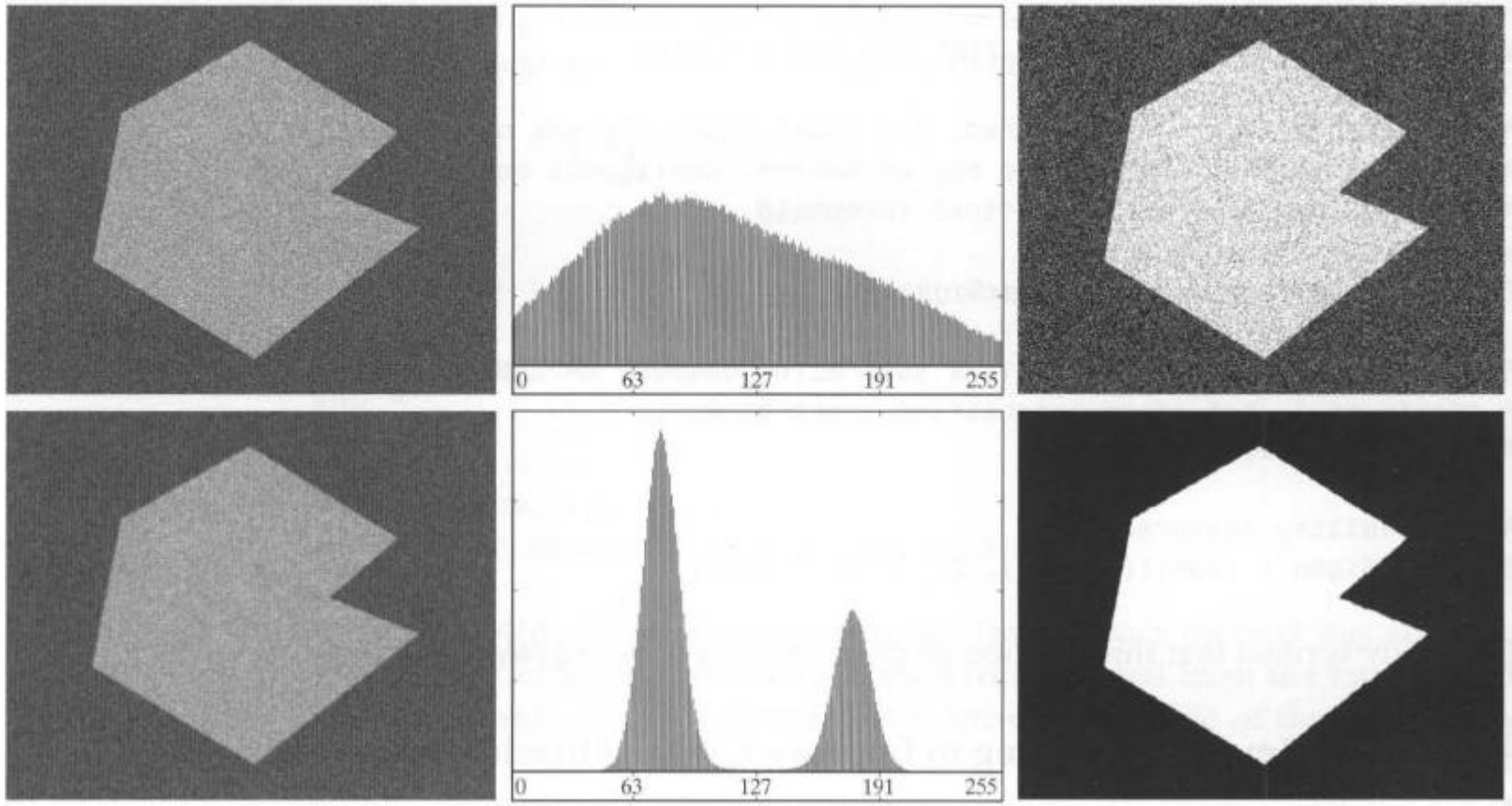
```
g = im2bw(f, T/den)
```

```
>> [T, SM] = graythresh(f)
T =
    0.4902
SM =
    0.9437
```



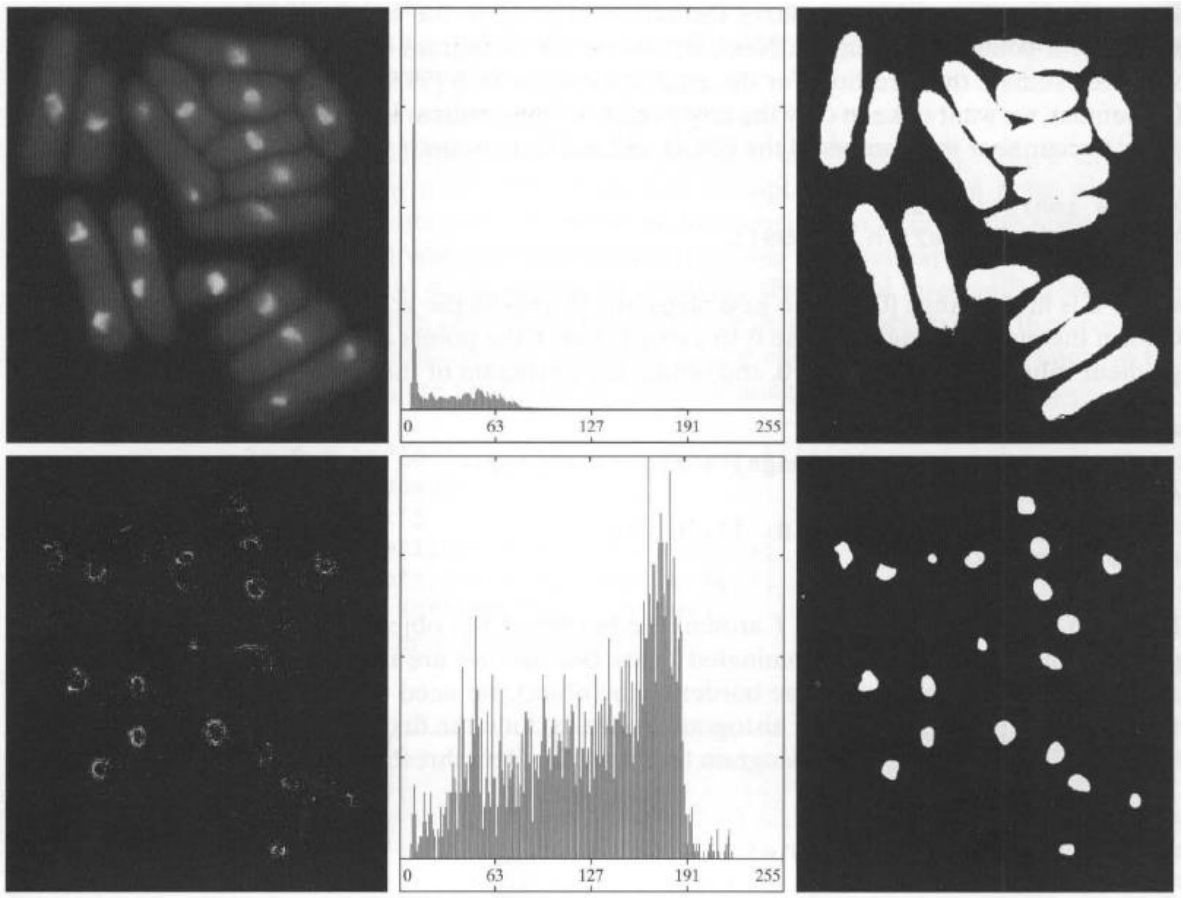
where f is the input image, T is the resulting threshold, normalized to the range [0, 1], and SM is the separability measure

Using Image Smoothing to Improve Global Thresholding



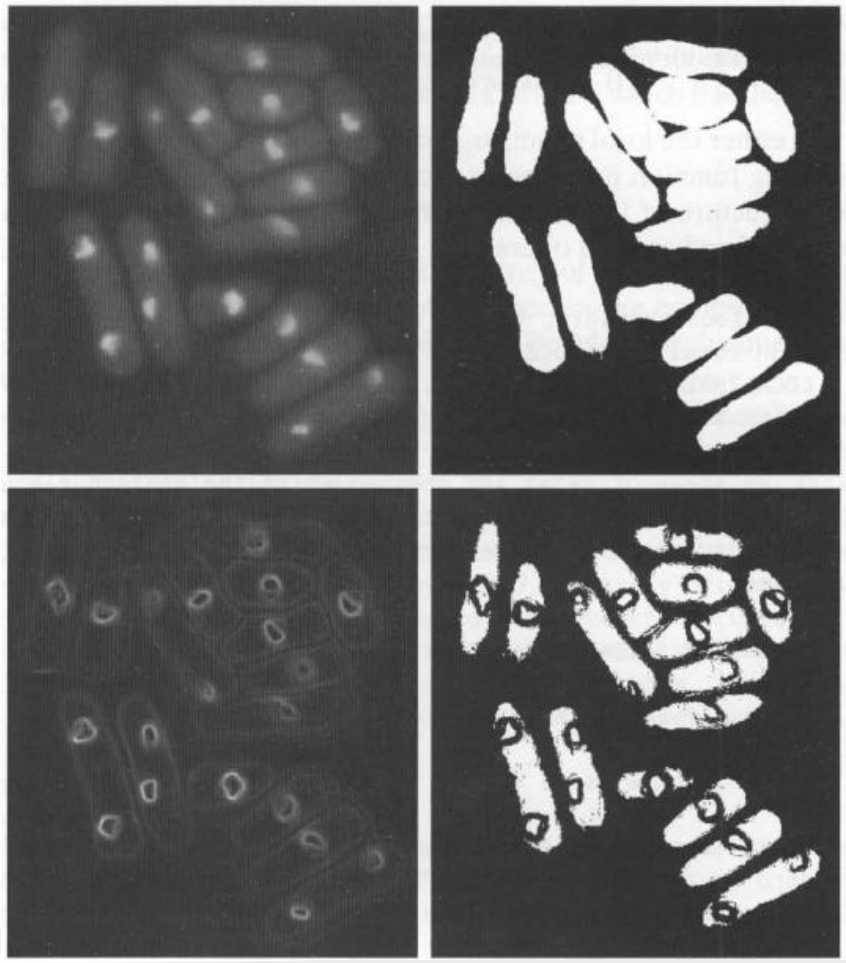
a) Noisy image, and
b) its histogram.
c) Result obtained using Otsu's method.
d) Noisy image smoothed using a 5 X 5 averaging mask,
(e) its histogram.
(f) Result of thresholding using Otsu's method

Using Edges to Improve Global Thresholding



(a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) using function gray thresh. (d) Product of the marker and original images. (e) Histogram of the nonzero pixels in (d). (f) Image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

Variable Thresholding Based on Local Statistics



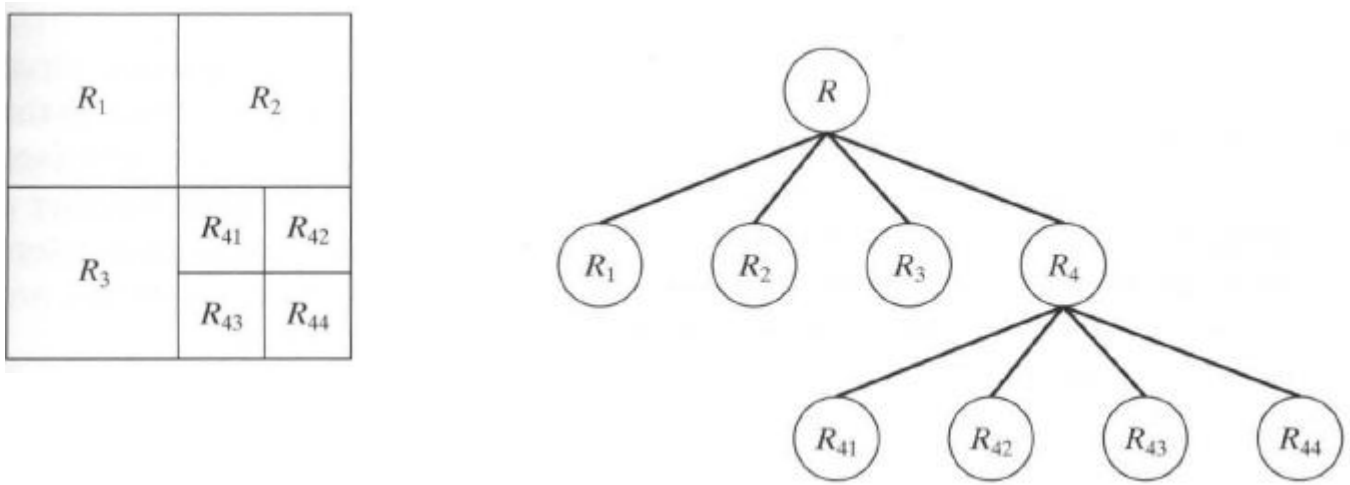
(a) Yeast cell image. (b) Image segmented using Otsu's method. (c) Image of local standard deviations. (d) Image segmented using local thresholding.

```
g = stdfilt(f, nhood)
```

```
>> [TGlobal] = graythresh(f);
>> gGlobal = im2bw(f, TGlobal);
>> imshow(gGlobal)
```

```
g = localthresh(f, ones(3), 30, 1.5, 'global');
SIG = stdfilt(f, ones(3));
```

Region-Based Segmentation



Region-Based Segmentation

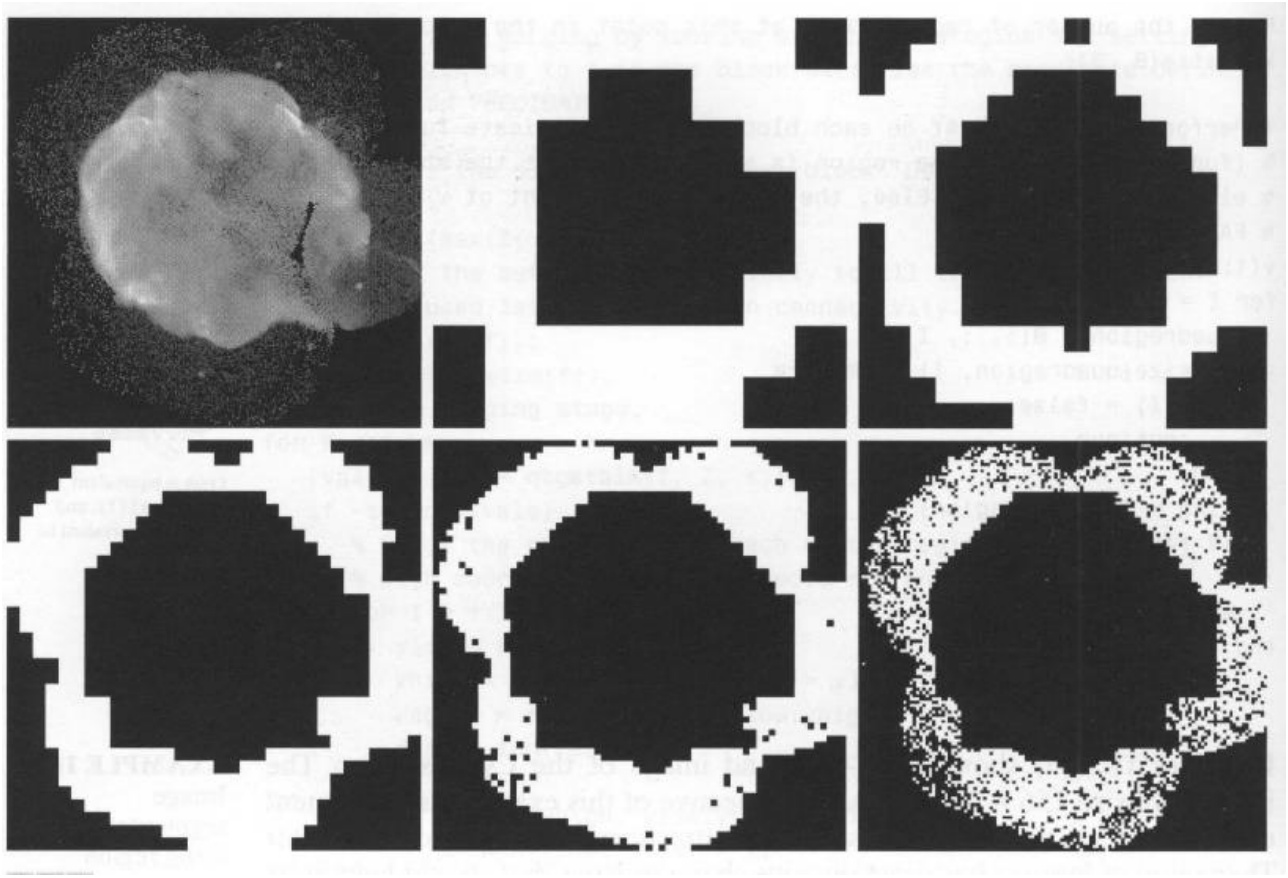
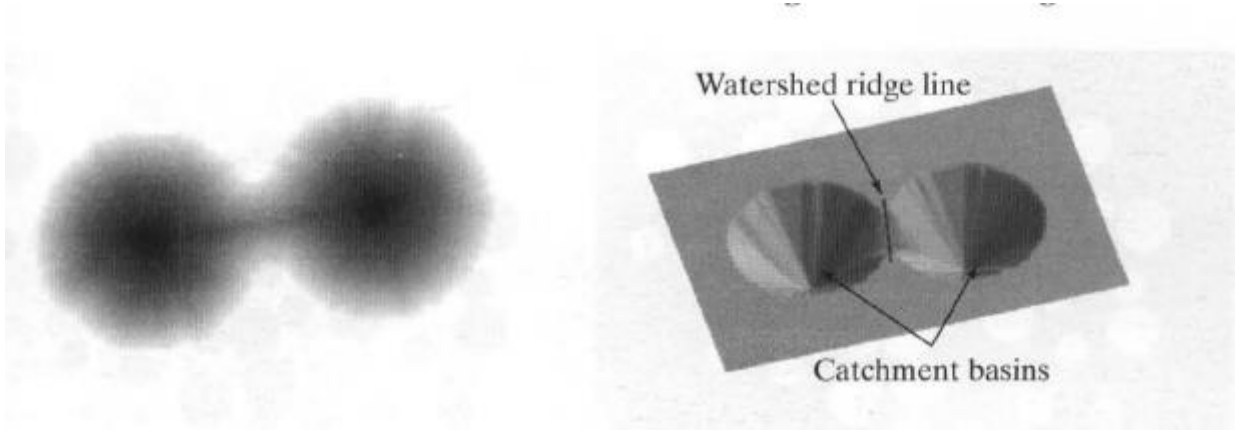
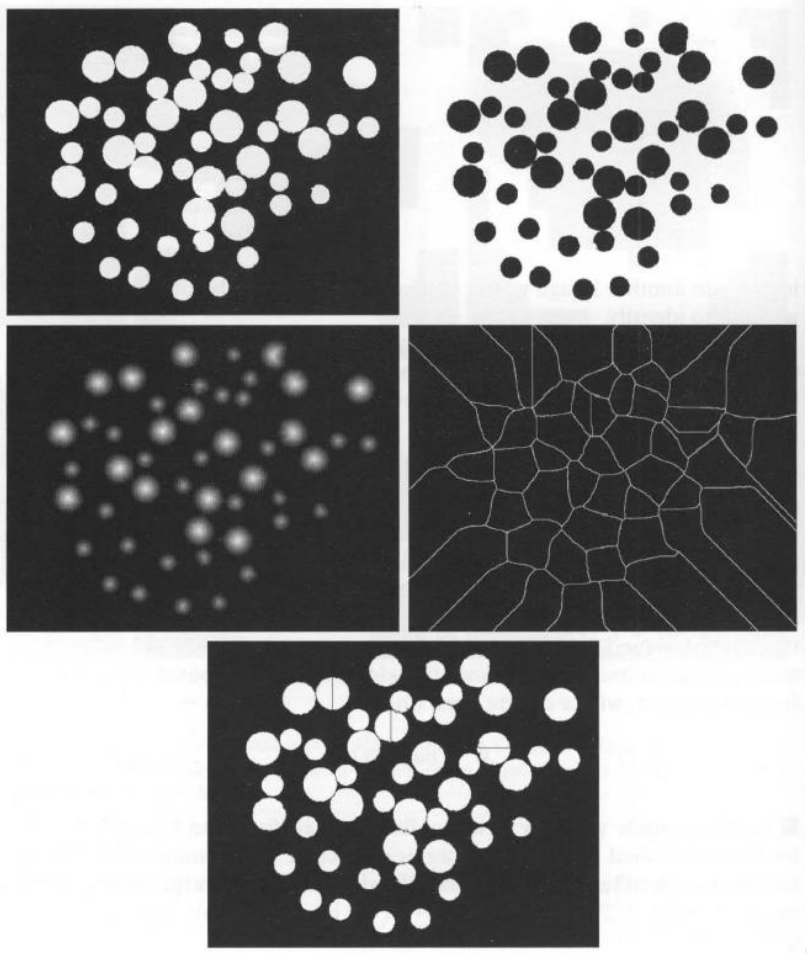


Image segmentation using a split-and-merge algorithm. (a) Original image. (b) through (f) Results of segmentation using function splitmerge with values of mindim equal to 32, 16, 8, 4, and 2, respectively. (Original image courtesy of NASA.)

Segmentation Using the Watershed Transform



Segmentation Using the Watershed Transform



1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
0	0	0	0	0	1.00	1.00	1.41	2.00	2.24
0	0	0	0	0	1.41	1.00	1.00	1.00	1.41
0	1	1	1	0	1.00	0.00	0.00	0.00	1.00

Binary image

Distance Transform

(a) Binary image. (b) Complement of image in (a). (c) Distance transform. (d) Watershed ridge lines of the negative of the distance transform. (e) Watershed ridge lines superimposed in black over original binary image. Some oversegmentation is evident.

Segmentation Using the Watershed Transform

```
>> h = fspecial('sobel');
>> fd = tofloat(f);
>> g = sqrt(imfilter(fd, h, 'replicate') .^ 2 + ...
          imfilter(fd, h', 'replicate') .^ 2);
>> L = watershed(g);
>> wr = L == 0;
```

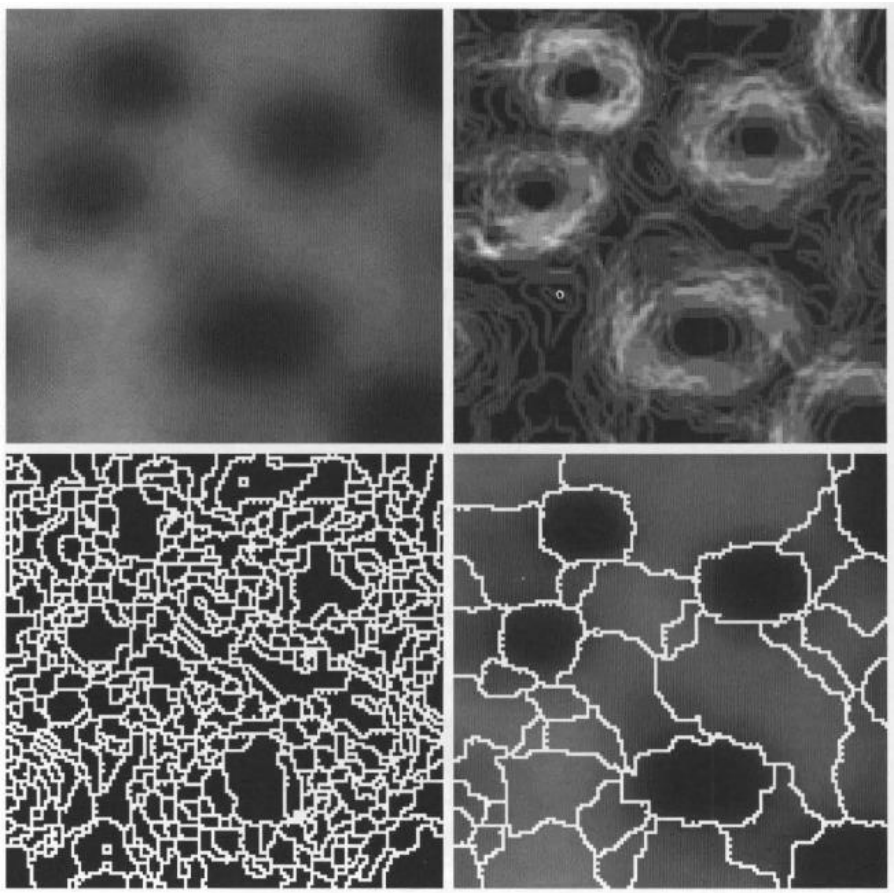
1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
0	0	0	0	0	1.00	1.00	1.41	2.00	2.24
0	0	0	0	0	1.41	1.00	1.00	1.00	1.41
0	1	1	1	0	1.00	0.00	0.00	0.00	1.00

Binary image

Distance Transform

(a) Binary image. (b) Complement of image in (a). (c) Distance transform. (d) Watershed ridge lines of the negative of the distance transform. (e) Watershed ridge lines superimposed in black over original binary image. Some oversegmentation is evident.

Segmentation Using the Watershed Transform



1	1	0	0	0
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	1	1	1	0

Binary image

0.00	0.00	1.00	2.00	3.00
0.00	0.00	1.00	2.00	3.00
1.00	1.00	1.41	2.00	2.24
1.41	1.00	1.00	1.00	1.41
1.00	0.00	0.00	0.00	1.00

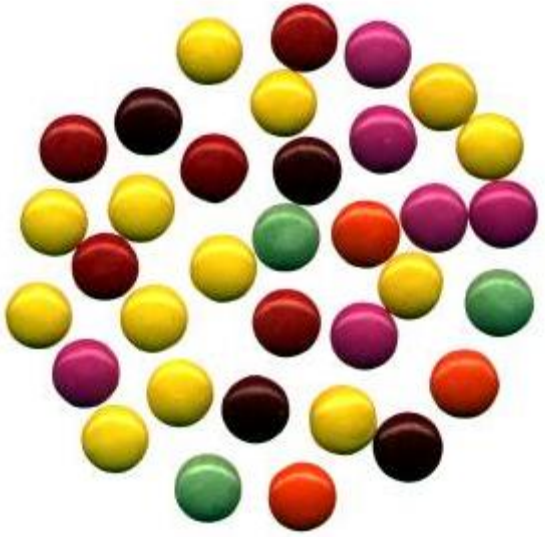
Distance Transform

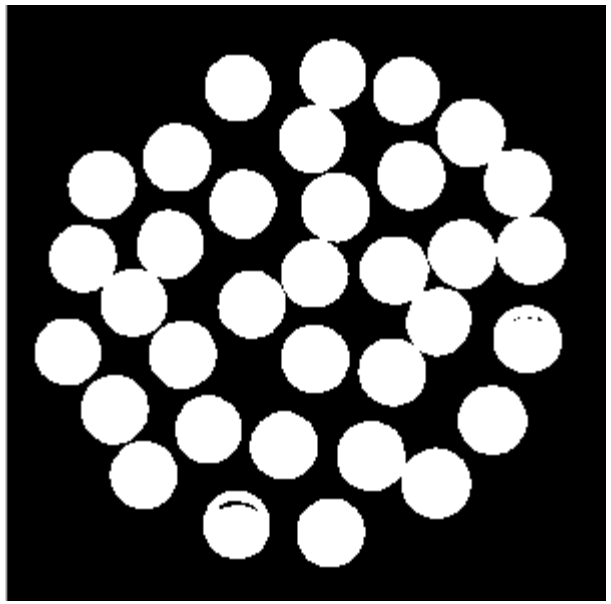
(a) Binary image. (b) Complement of image in (a). (c) Distance transform. (d) Watershed ridge lines of the negative of the distance transform. (e) Watershed ridge lines superimposed in black over original binary image. Some oversegmentation is evident.

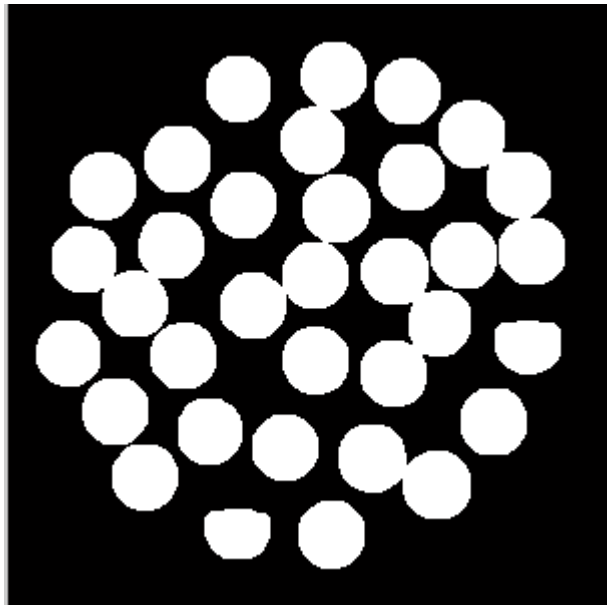
Örnek Uygulamalar

Image Segmentation

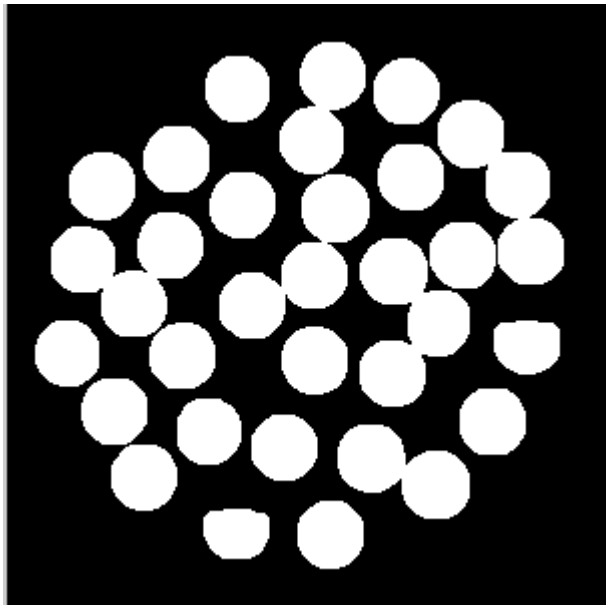
Color Segmentation

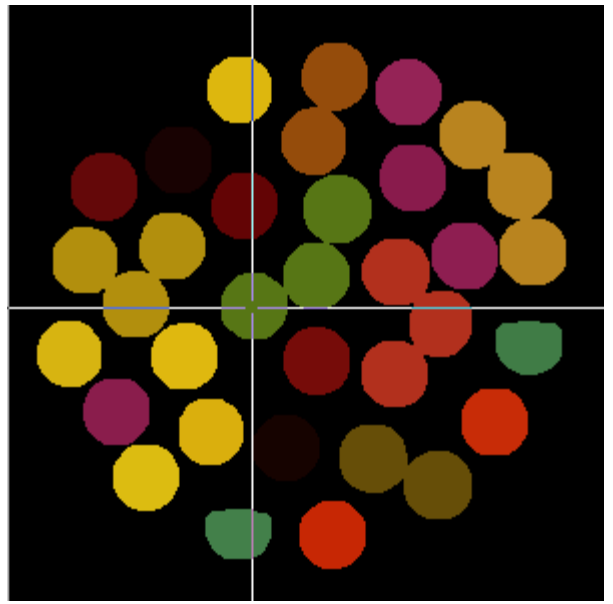




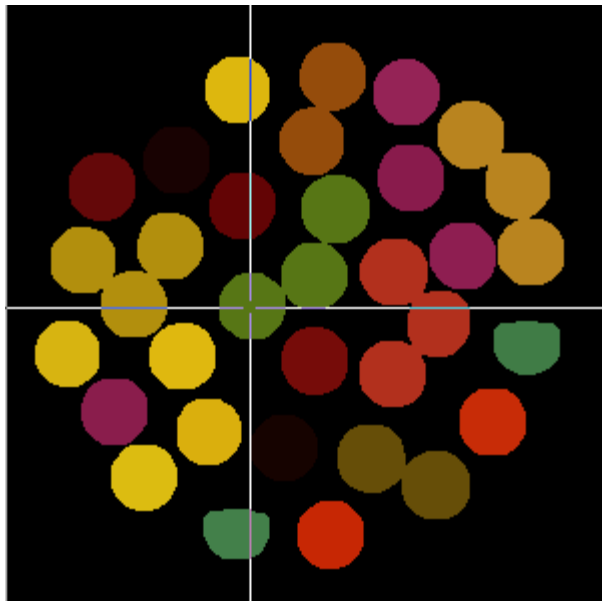


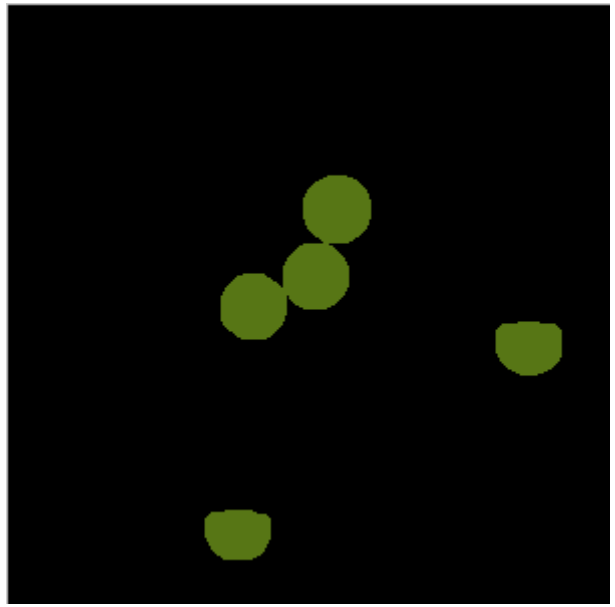
Color Segmentation

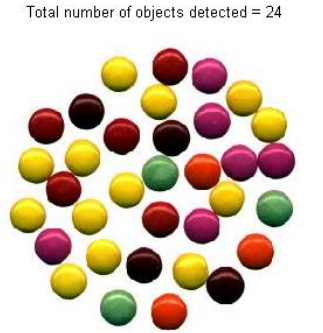


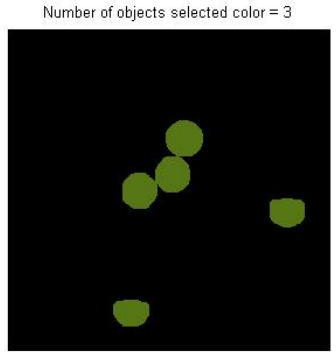


Color Segmentation







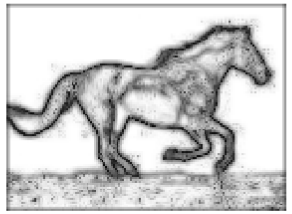


ColorSegmentation Klasöründe örnek uygulama yer almaktadır.

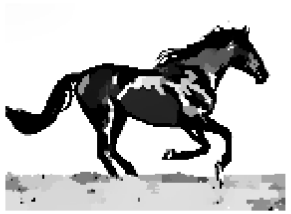
Central Segmentation

input

segment size

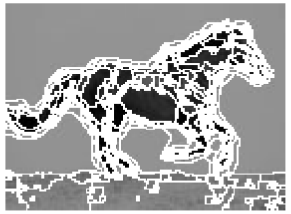


segment label



boundary of continuous segment labels

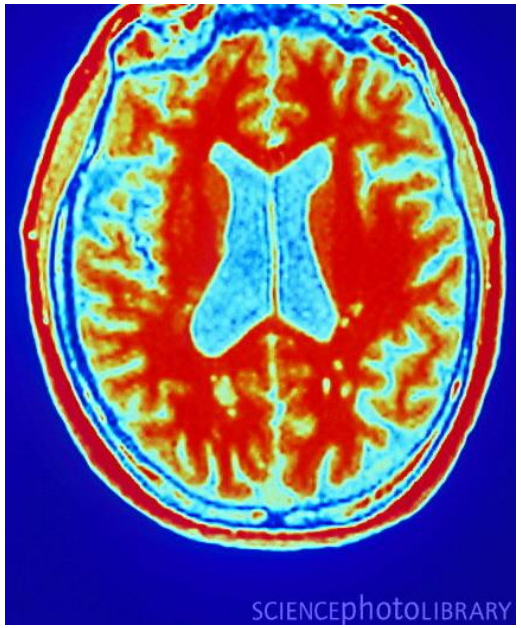
boundary of discretized segment labels (final segmentation)

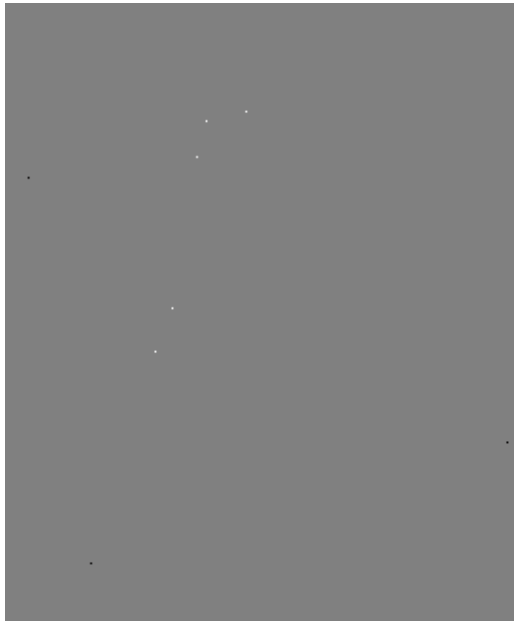
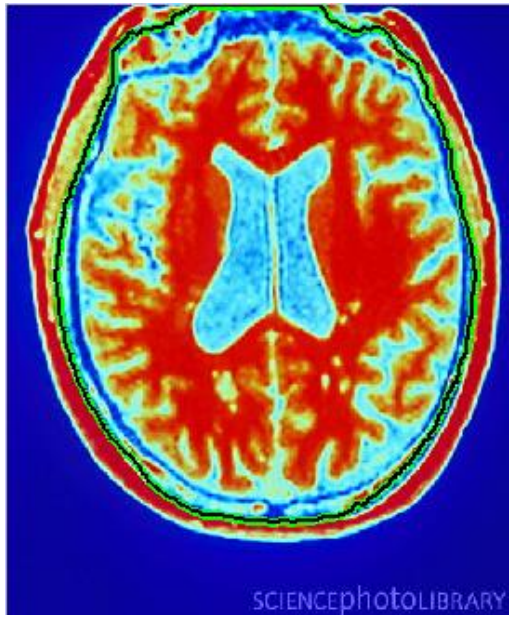


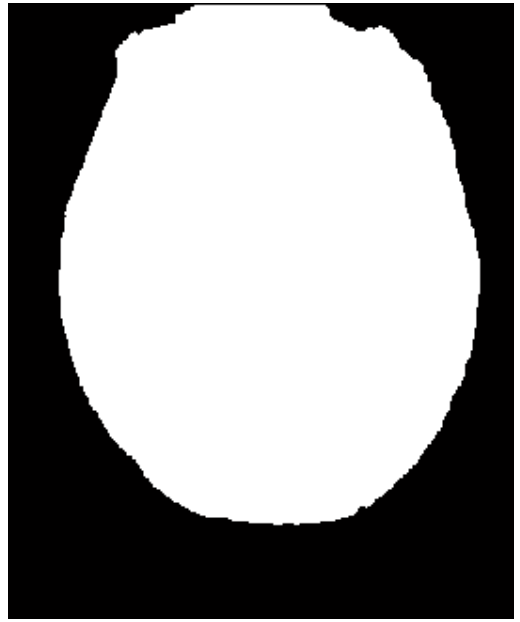
derivative

CentralSeg Klasöründe örnek uygulama yer almaktadır.

GrowCut



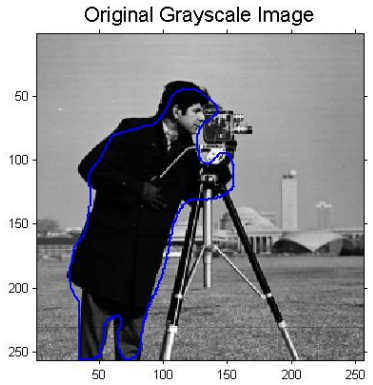



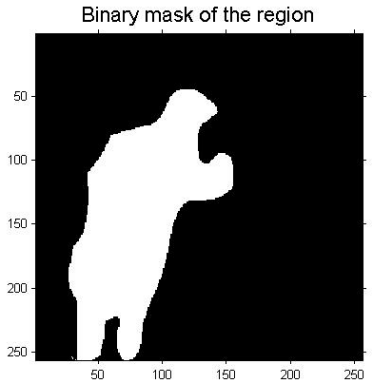


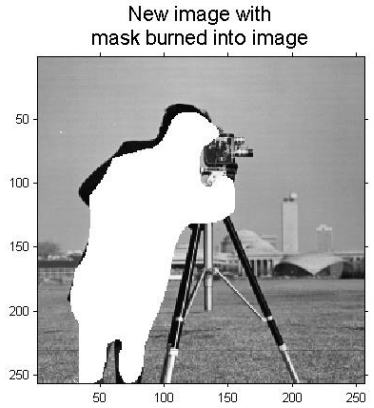
Grow_cut Klasöründe örnek uygulama yer almaktadır.

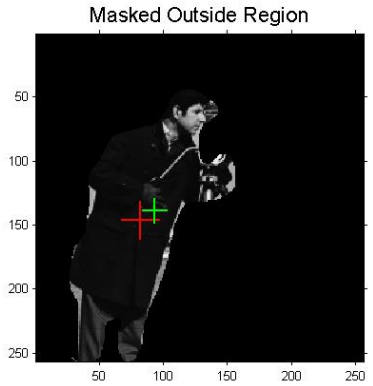
Freehand Segmentation

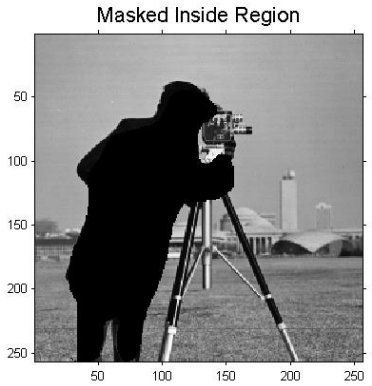
Freehand Segmentation

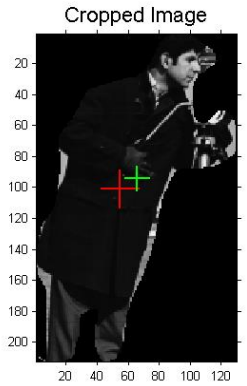












Freehand Klasöründe örnek uygulama yer almaktadır.

Image Segmentation and Matting

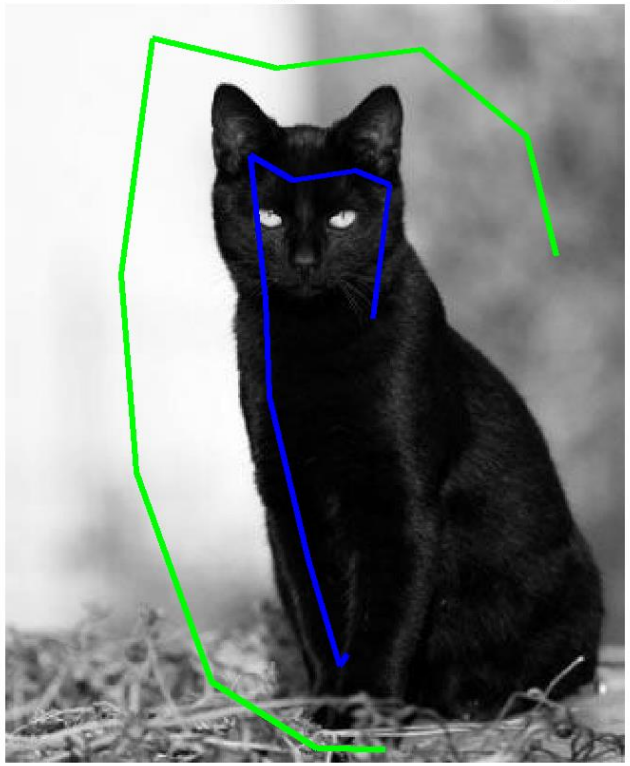
Image Segmentation and Matting

Image Segmentation and Matting

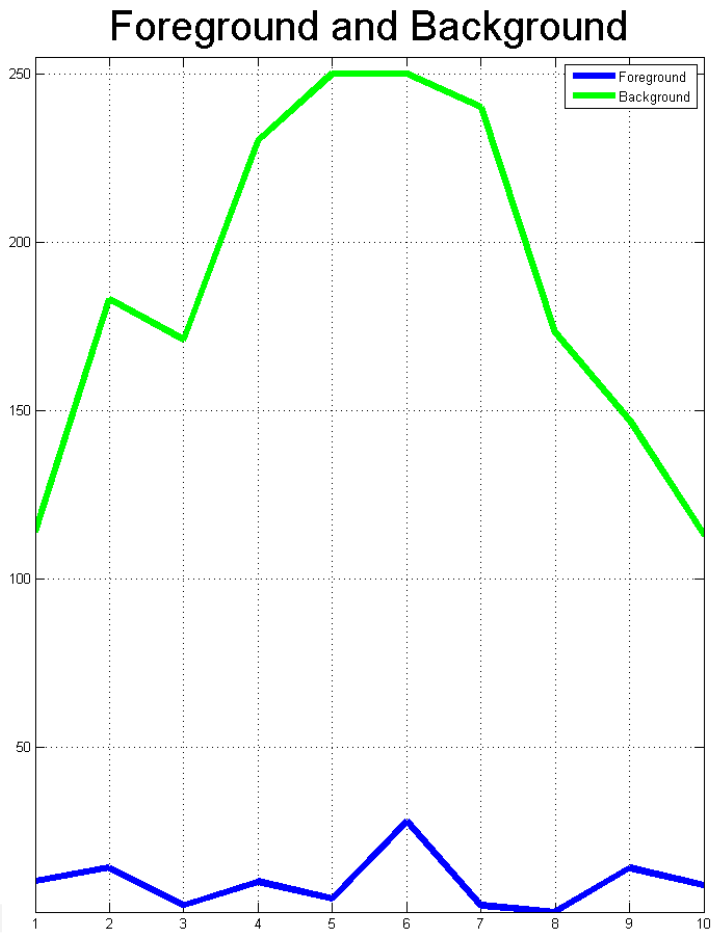
Interactive Image Segmentation

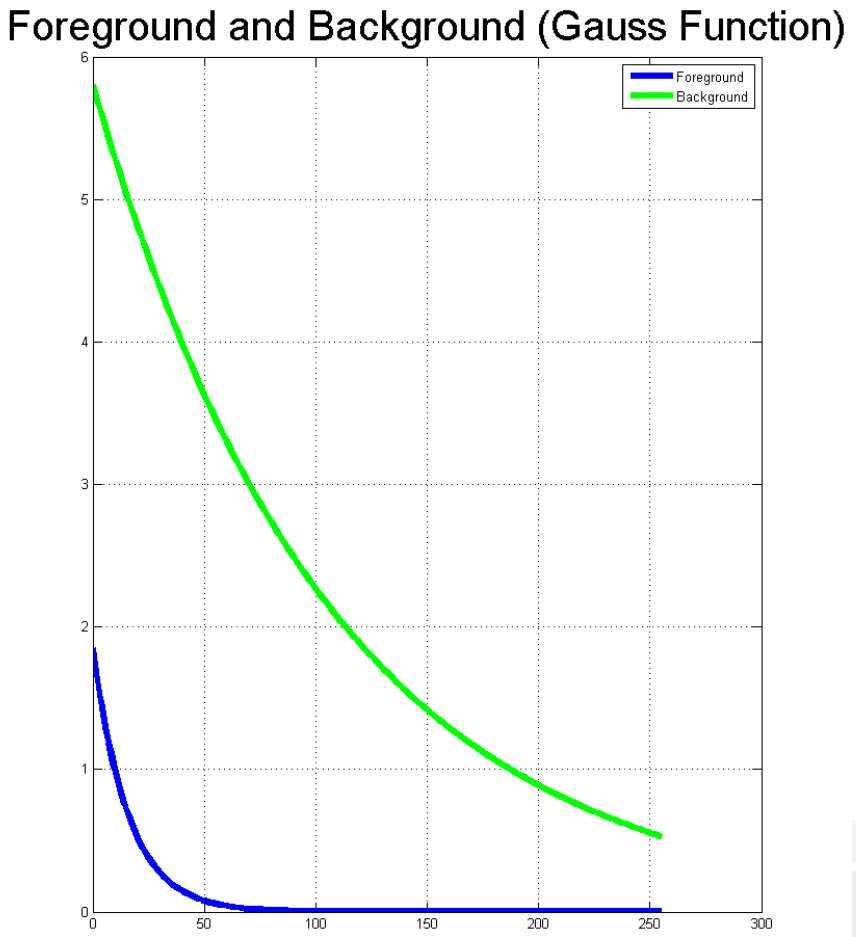
Please Select Foreground and Background

Please Select Foreground and Background



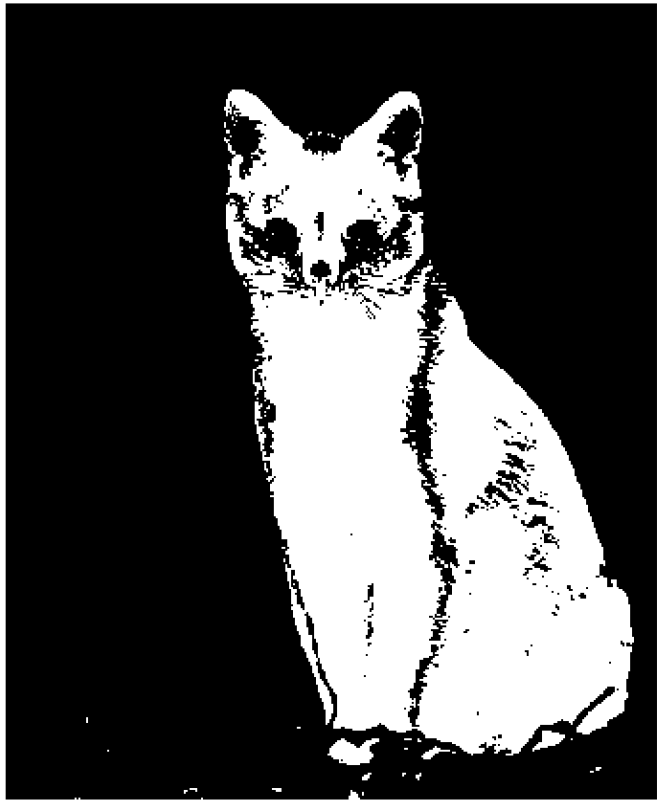
Interactive Image Segmentation



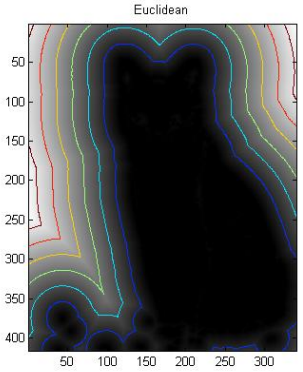


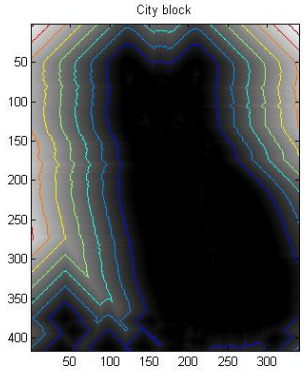
Interactive Image Segmentation

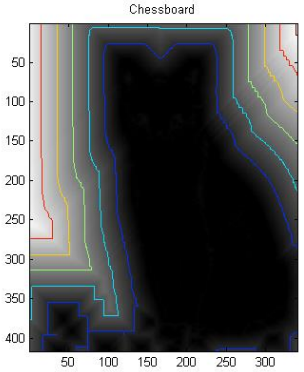
The Mask

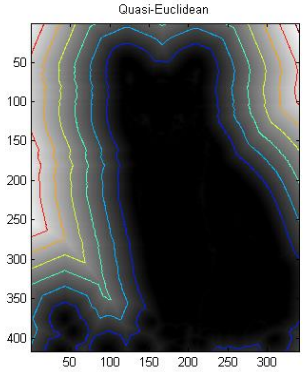


Interactive Image Segmentation



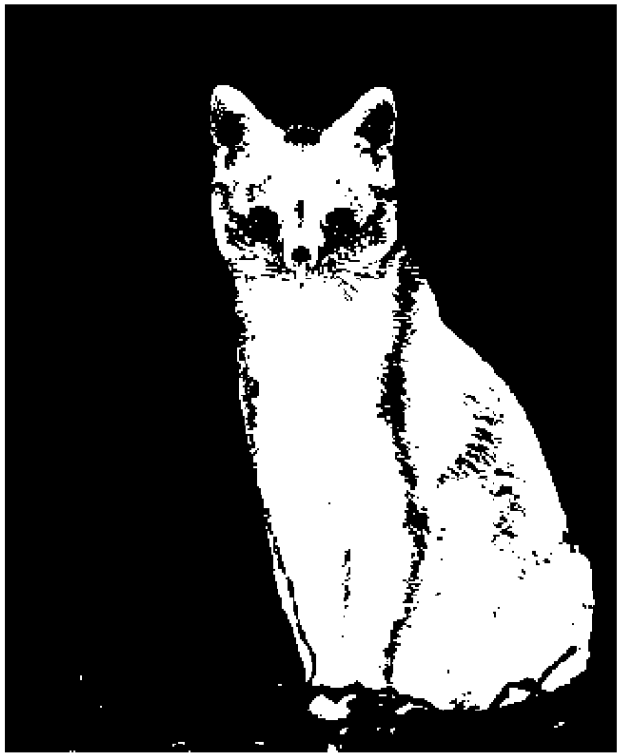




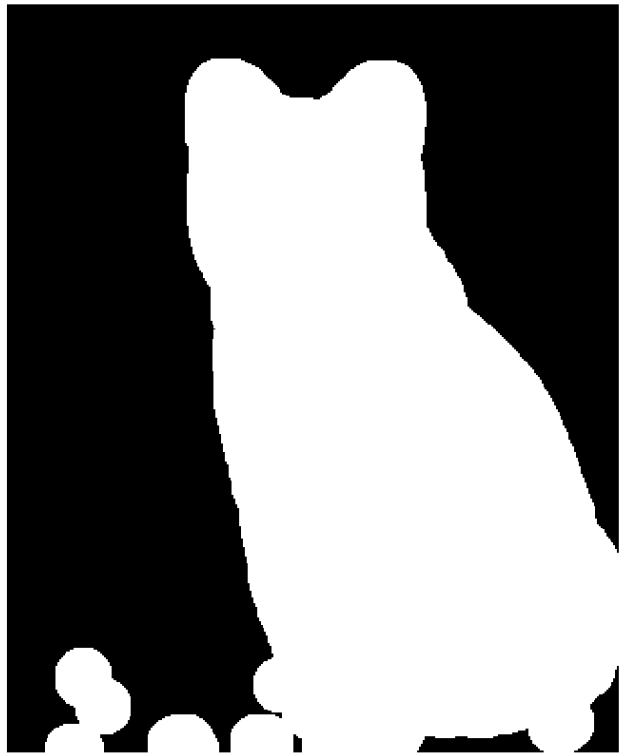


Interactive Image Segmentation

The Mask Before Geodesic Distance



The Mask After Geodesic Distance (Euclidean)



Teşekkürler

www.zafercomert.com

comertzafer@gmail.com