İmage segmentasyon (Görüntü Bölütleme)

Segmantasyon (Bölütleme)

- Segmentasyon genellikle görüntü analizinin ilk aşamasıdır. Görüntü bölütleme, bir görüntüyü her biri içerisinde farklı özelliklerin tutulduğu anlamlı bölgelere ayırmak olarak tarif edilebilir. Örneğin, görüntü içerisindeki benzer parlaklıklar olabilir ve bu parlaklıklar ilgili görüntünün farklı bölgelerindeki nesneleri temsil edebilir.
- Uygulamaya bağlı olarak değişebilen bu segmentlere (bölütler-elemanlar) başka bir örnek olarak; hava-yer fotoğrafında, yolda hareket eden araçları ve çevreyi yoldan ayırt edebilmek için bir segmentleme yapılabilir. (Yolu çevreden ayırabilen bir segmentleme)
- Unutulmamalıdır ki, tüm görüntülere uygulanabilecek genel (universal) bir bölütleme yöntemi yoktur ve hiçbir bölütleme yöntemi mükemmel değildir. Başka bir deyişle, görüntü iyileştirme ve onarma problemlerinde olduğu gibi görüntü bölütleme için tasarlanan yöntemler ve bu yöntemlerin başarımları, görüntüden görüntüye ve uygulamaya dayalı olarak değişiklik arz eder.
- Otomatik görüntü segmentasyonu görüntü işlemenin en zor işlemlerinden biridir.

Segmantasyonun (Bölütleme) özellikleri

- Görüntü bölütleme; R görüntüsünü R1, R2…. Rn bölgelerine bölme sürecidir.

"Segmantasyon tamamlandığında: $\bigcup_{i=1}^{n} R_i = R$

- Herbir bölge uniform'dur (Tek tiptir).
- Bölgeler biribirini üzerine çakışmaz. $R_i \bigcap R_j = \varnothing, \ \forall i \neq j$
- Bir bölgenin pikselleri ortak özelliğe sahiptir. $P(R_i) = TRUE$
- Komşu bölgelerin farklı özellikleri vardır. $P(R_i) \neq P(R_j), \forall R_i, R_j$ komsudur.
- Gri seviyeli görüntülerde segmantasyon algoritmaları ;gri seviye (parlaklık) değerlerinin süreksizlik ve benzerlik özelliğine dayandırılır.
- Süreksizlik tabanlı bölütleme algoritmaları; Izole nokta, ince çizgi veya resim kenarları gibi (gri seviye değerleri birden değişen) süreksizlikleri, düşük ve yüksek filtrelemedeki gibi benzer maskeler kullanarak tespit edebilmeye dayanır.
- Benzerlik tabanlı bölütleme algoritmaları ; *ya eşikleme, bölgede büyüyen*, ya da *bölge bölme ve birleştirmeye* dayanmaktadır.

Benzerlik Tabanlı Görüntü bölütleme algoritmaları

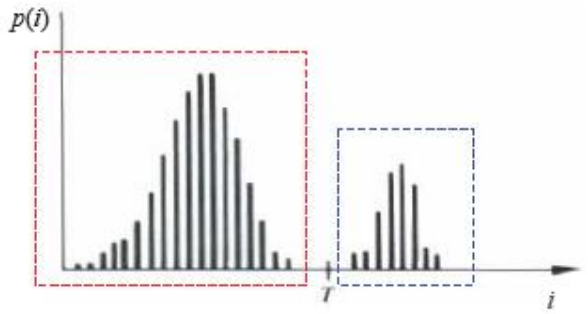
- Gri seviye değerlerindeki benzerliklere göre görüntü bölütleme, bölge bölütlemesi (region segmentation) olarak bilinir.
- Eşikleme (thresholding), büyütme (growing), ve yarma - kaynaştırma (split- and -merge) işlemlerine dayalı olarak gerçekleştirilir.
- Piksellerin gri seviye değerlerindeki benzerlik veya farklılıklara dayalı olarak bir görüntünün bölütlenmesi kavramı hem durağan (static) hem de dinamik (zamanla değişen) görüntülere uygulanabilir[R].

Eşikleme (Thresholding)

Eşikleme, görüntü bölütleme amacı için kullanılan en önemli yaklaşımlardan birisidir. Eşikleme işleminden amaç, görüntü içerisindeki nesneleri görüntü arka planından ayırmaktır.

Eşikleme için, görüntüdeki gri seviye dağılımlarını gösteren görüntü histogramından faydalanılır. Örneğin, koyu bir arka plan üzerinde açık renkli nesnelerden oluşan *f(i, j) görüntüsüne ilişkin histogramı Şekil* (a)'daki biçime sahip olacaktır. Bu histograma göre, nesnelere ve arka plana ait pikseller olmak üzere, görüntüyü iki ana grupta değerlendirmek mümkündür.

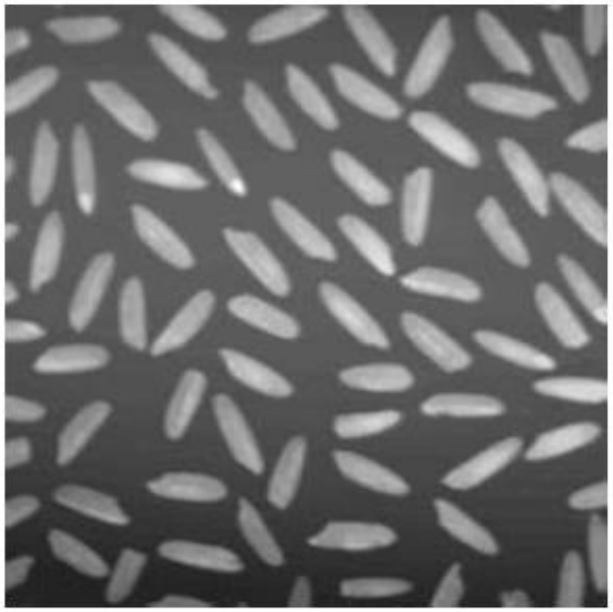
Bu durumda nesneleri arka plandan ayırmak için en kolay yol, histogramdan göreceli olarak belirlenen bir *T eşik değeri ile görüntüdeki piksel değerlerini karşılaştırmak* olacaktır. Buna göre, görüntüdeki herhangi bir *(i, j) pikseli için; f(i, j) > T ise (i, j) pikseli nesneye ait bir nokta, f(i,j) ≤ T ise (i, j) pikseli arka plana ait bir nokta olacaktır.*



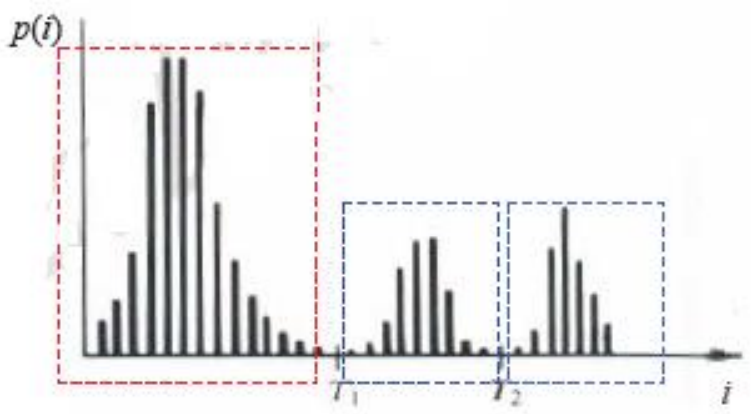
(a)

Eşiklemeye Örnek

```
>> r=imread('rice.tif');
>> imshow(r),figure,imshow(r>110)
```



- *Görüntüye ilişkin Histogram Şekil.* b'deki gibi; ikisi nesneye biri de arka plana ait olmak üzere üç gri seviye grubundan da oluşabilir.
- Buna göre; görüntüdeki herhangi bir (*i, j) pikselleri için;*
- *T1 < f(i, j) ≤ T2 aralığındaki ise bir nesneye,*
- *f(i, j) > T2* aralığındaki pikseller diğer bir nesneye ve
- *f(i, j) ≤ T1 aralığındaki pikseller de görüntü arka planına karşı* düşecektir.



(b)

- *Aşağıda genel bir eşikleme fonksiyonu ve* Eşiklenmiş bir g(*x,y) görüntüsünün bağıntısı verilmiştir. 1 ile belirlenen pikseller nesneye, 0 ile etiketlenen pikseller ise arka plana denk gelir.*
- T Eşikleme fonksiyonun da; *f(x,y), (x,y) noktasındaki gri seviyenin değerini, p(x,y) ise bu noktanın bazı bölgesel özelliklerini belirtir.* Örneğin bu özellik (*x,y) noktasının komşuluğundaki piksel gri seviye değerlerinin ortalaması olabilir*
- Eğer ki *T sadece f(x,y)'ye bağlı ise bu durumda aşağıdaki T eşitliği ile belirlenen eşik değeri bütünsel (global) eşik* olarak adlandırılır.
- Şekil (a), böyle bir eşik değerine örnektir.
- Eğer *T hem f(x,y) hem de p(x,y)'ye bağlı ise bu* durumda aşağıdaki T denklemindeki eşik değeri *bölgesel (local) eşik olarak adlandırılır.*
- *Buna ek olarak, Eğer T, x ve y* uzaysal koordinatlarına bağlı ise bu durumda T denkleminde belirlenen eşik değeri *dinamik eşik olarak adlandırılır.*

$$T = T\,[x, y, f(x,y), p(x,y)]$$

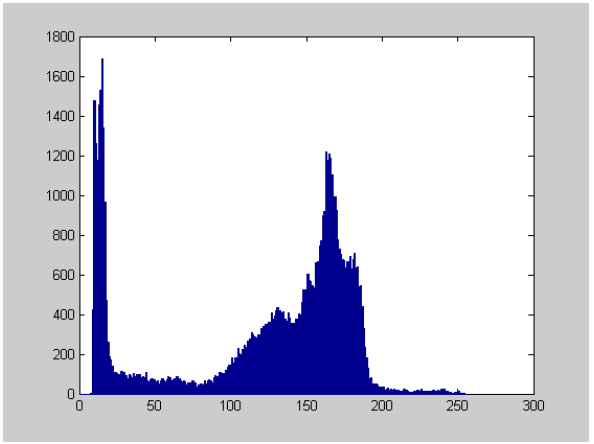
$$g(x,y) = \begin{cases} 1 & ; \quad eger\ f(x,y) > T \\ 0 & ; \quad eger\ f(x,y) \le T \end{cases}$$

Bütünsel (Global) Eşikleme

Tüm eşikleme tekniklerinin en basiti, Şekil (a)'da gösterildiği gibi tek bir eşik değeri kullanarak görüntü histogramını ve dolayısıyla görüntüyü bölütler. Bu bölütleme yönteminin başarısı, görüntü histogramının iyi bir biçimde bölmelenmesine bağlıdır. Şekilde bütünsel eşikleme işlemi sonucu elde edilen görüntü sonuçları verilmiştir (Anlaşılacağı gibi sonuç resim bir binary-siyah-beyaz) resimdir.

Graythresh fonksiyonu resmin historamını alıp, histogramı en uygun şekilde 2 kısma bölütler. (OTSU modeline göre)

```
> a=imread('cameraman.tif');
>> imshow(a);
>> T= graythresh(a)
T =
    0.3451
>> T= graythresh(a);
>> d=im2bw(a,T);
>> imshow(d);
```



Global eşiklemenin uygulanacağı görüntülerde, aydınlatmanın önemli bir payı vardır. Şöyle ki, farklı aydınlatmalara maruz kalmış nesnenin bütünsel eşikleme sonucunda üreteceği ikili görüntüler farklı olacaktır. Bu çerçeveden bakıldığında, büyük oranda kontrol edilebilen ortamlara bütünsel eşiklemenin başarıyla uygulanabileceği açıktır.

Global eşiklemenin kullanılabileceği alanlar, endüstriyel muayene uygulamaları olarak verilebilir çünkü bu tür uygulamalarda aydınlatmanın kontrol edilebilmesi genellikle mümkündür.

Global eşiklemede uygun eşik değeri aşağıdaki algoritmaya göre belirlenebilir.

1- Görüntü içindeki en küçük ve enbüyük gri seviye değerini belirleyip orta değeri T olarak belirle.

2- T'yi kullanarak görüntüyü segmentle.Bu iki G1 ve G2 gibi iki farklı pixel gurubu oluşturacaktır. Pixel değerleri >= T ve pixel değerleri < T olarak).

3- Herbir segmentin ortalama gri seviye değerini $\mu 1$ ve $\mu 2$ olarak hesapla.

4- Bu ortalamalardan yeni T değerini aşağıdaki bağıntıyla hesap et.

$$T = \frac{1}{2}(\mu_1 + \mu_2)$$

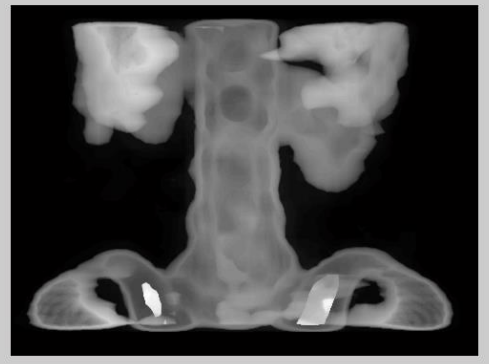
5- 2. ve 4. adımları, T0 toleransından küçük oluncaya kadar iteratif olarak yap.

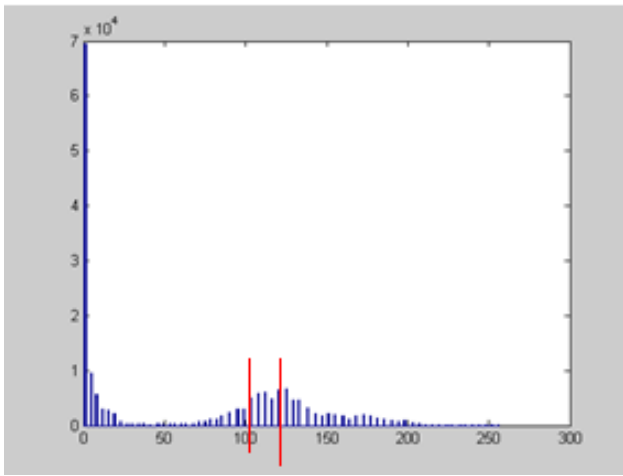
Ödev

- OTSU yöntemini MATLAB'ta gerçekleştiren m-file fonksiyonunu yazınız?
- Ödev Teslim Tarihi: 25.12.2013

Çift Eşikleme

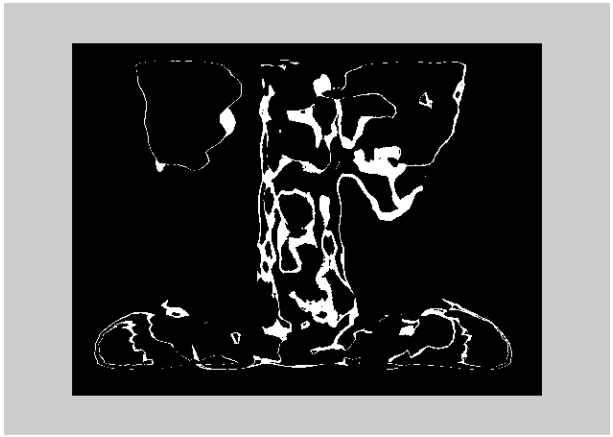
```
>> [x,map]=imread('spine.tif');
>> s=uint8(256*ind2gray(x,map));
>> imshow(s),
>> b=imhist(s);
>> imshow(s),figure, bar(b)
>> imshow(s>115 & s<125)
>> imshow(s>115)
```

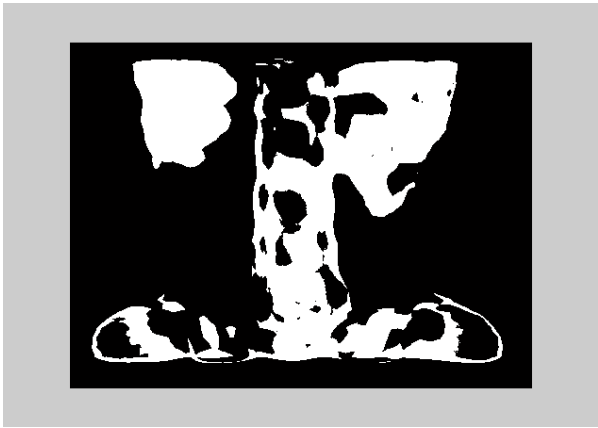




Buradaki spine.tif görüntüsünün indekslenmiş bir görüntü olduğuna dikkat ediniz.

Burada çift eşikleme T1=115, T2=125 arasındaki gri seviyelerin beyaz diğer değerlerin siyah olarak algılanması işlemidir. Buda resim üzerinde tek eşiklemeye göre daha çok sayıda segmantasyon oluşması anlamındadır.

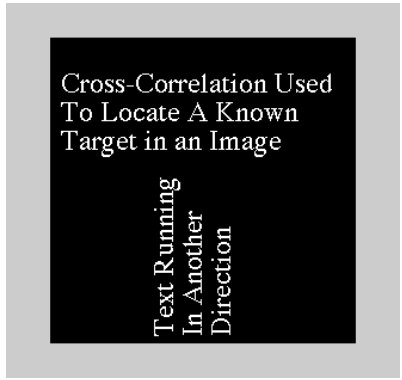


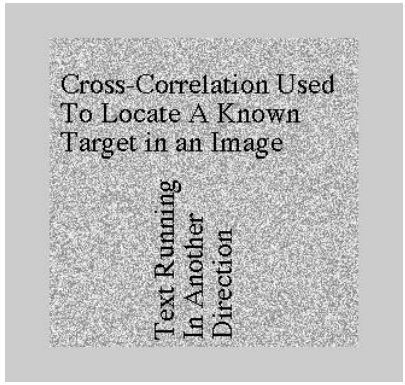


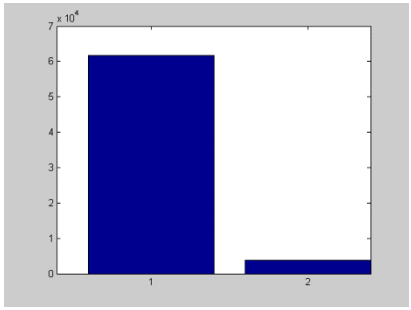
Eşikleme Uygulaması

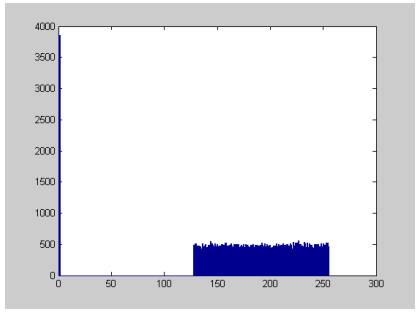
Arka planı gürültülü olan text'leri (örneğin kotü çekilmiş fotokopileri) düzeltemek için kullanılan basit bir uygulamadır.

```
>> t=imread('text.tif');
>> K=imhist(t)
>> r=rand(256)*128+127;
>> tr=uint8(r.*double(not(t));
>> imshow(tr)
>> imshow(tr>100)
```



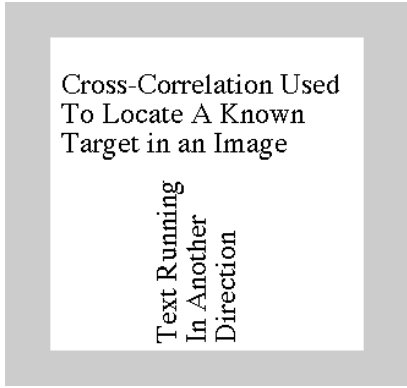






3.Satırdaki rand fonksiyon ile 256x256'lık bir matris oluşturulur. Matrisin elemanları 0-1 arasında değişir. Üretilen bu sayılar 127-255 arasına dönüştürülür. Orijinal text görüntüsü, siyah arka plan üzerine beyaz yazıdır. Bunu not(t) ile tersine çevirip r gürültü fonksiyonu ile elemanter çarpma yapalım.

Yeni görüntünün histogramını incelediğimizde, gray seviye değerini T=100 ile eşiklediğimizde aşağıdaki görüntü elde edilir.



Süreksizlik tabanlı bölütleme algoritmaları

Izole nokta, *ince çizgi* veya *resim kenarları* gibi (gri seviye değerleri ani değişen) süreksizlikleri, düşük ve yüksek filtrelemedekine benzer maskeler kullanarak tespit edebilmeye dayanır.

Bu maskelerin temelini türev işlemleri (1. ve 2. türev) oluşturur.

Temel Bilgi

- 1.derece Türev İşlemi (First-order derivative-tek değişkenli)

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

- 2.derece türev İşlemi (Second-order derivative - tek değişkenli)

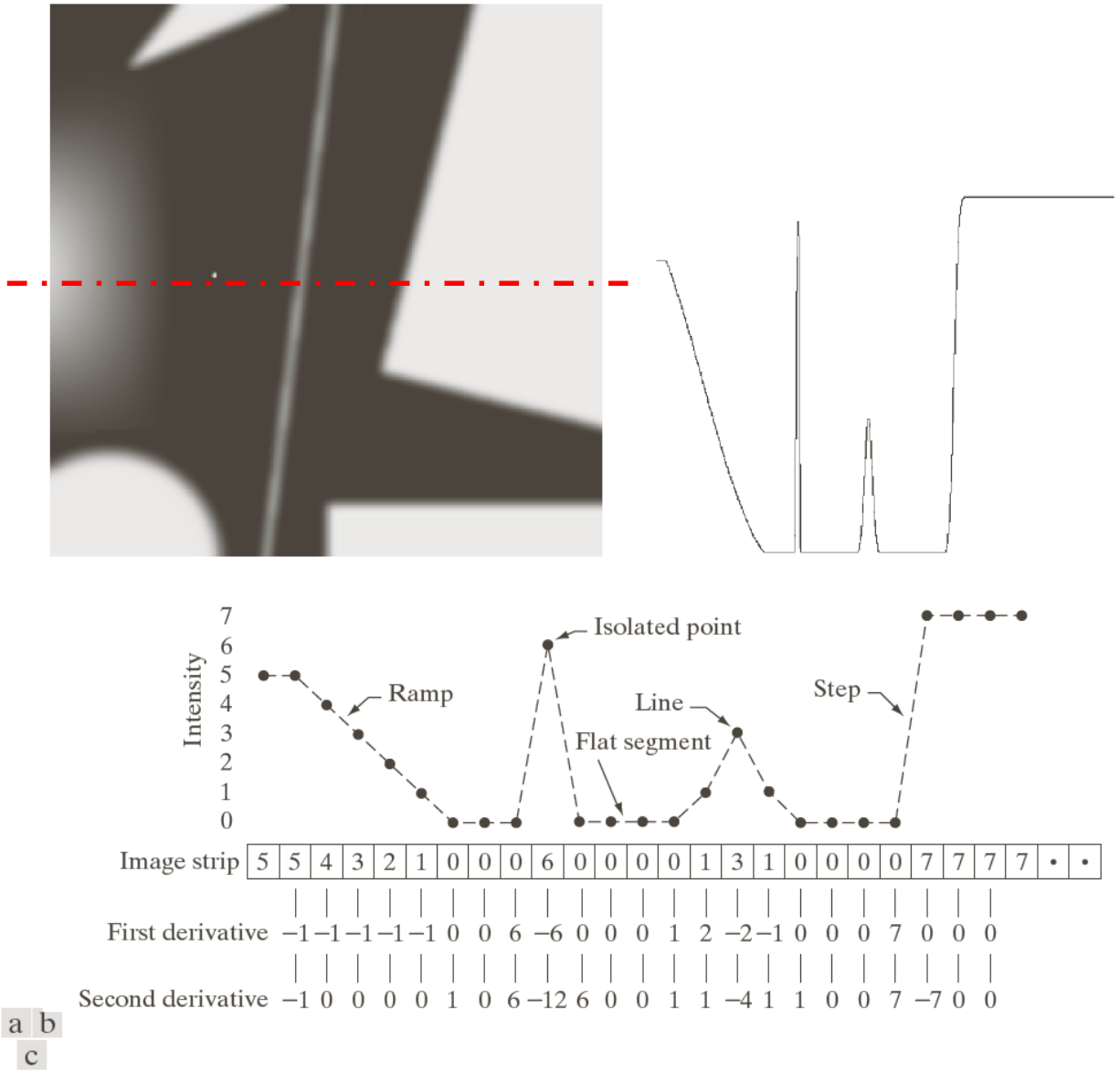
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

•Birinci dereceden türevler, bir görüntüde genellikle kalın kenarlar üretir.

•İkinci dereceden türevler ince çizgiler, izole noktaları ve gürültü gibi ince ayrıntıları bulmak için uygundur.

•Ikinci dereceden türev rampa ve basamak(keskin)geçiş de bir çift kenar tepki oluşturur.

•ikinci türevin işareti aydınlık- karanlık veya karanlık-aydınlık bir kenar geçiş olup olmadığını belirlemek için kullanılabilir



a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

İzole nokta belirlenmesi

- Bir görüntüdeki izole nokta tespiti aşağıdaki maske kullanarak yapılabilir. Belirlenen bu izole nokta maskenin merkezine karşılık gelen eleman değeridir.

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

Image Pixels

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

The Mask

R= - (*x1+x2+ x3+ x4+ x5+ x6+ x7+ x8+ x9)+8*x5*

hesaplanır; IRI > T ise bu nokta (x5) bir izole noktadır. T burada bir eşik değeridir. (wk=xk, zk: maske elemanları)

$$g(x,y)=\begin{cases}1 & \text{if } |R(x,y)|\geq T\\ 0 & \text{otherwise}\end{cases} \qquad R=\sum_{k=1}^{9} w_k z_k$$

- Yani x5'in gri seviye değeri komşu piksellerin gri seviye değerlerinden çok farklı ise bu nokta izole bir noktadır.

İzole nokta maskesi nasıl oluşturulmuştur?

2 boyutlu bir dizinin x ve y yönündeki 2. türevlerinin toplamına Laplacian denir. Sayısal türev işlemine göre aşağıdaki eşitlik bize maskenin nasıl elde edildiğine dair fikir vermektedir. Aynı prensibe göre değişik maskeler verilmiştir.

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

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$
$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

0	0	0
1	-2	1
0	0	0

+

0	1	0
0	-2	0
0	1	0

=

0	1	0
1	-4	1
0	1	0

0.5	0.0	0.5
1.0	-4.0	1.0
0.5	0.0	0.5

+

0.5	1.0	0.5
0.0	-4.0	0.0
0.5	1.0	0.5

=

1	1	1
1	-8	1
1	1	1

1	-2	1
1	-2	1
1	-2	1

+

1	1	1
-2	-2	-2
1	1	1

=

2	-1	2
-1	-4	-1
2	-1	2

Kenar Çıkarma (Belirleme)-Edge Detection

- Bir görüntüdeki kenar, aydınlatma veya yüzey yansımaları gibi bir görüntünün fiziksel görünüşünde oluşan önemli bir değişime karşı düşer ki bu değişim kendisini parlaklık, renk ve doku olarak gösterir.
- Burada kenar anlamında, sadece görüntü parlaklıklarındaki değişikliklerle ilgilenilecektir. Bu anlamda, bir görüntünün gri seviyelerinde ani değişikliklerin olduğu bölgelere *kenar adı verilecektir.*
- Görüntüye ilişkin kenarların belirlenmesi birçok durumda kullanışlıdır. Nesne tanıma problemi buna bir örnek olarak verilebilir. Nesne tanımada temel adım, bir görüntüyü farklı nesnelere karşı düşen farklı bölgelere bölmelemektir.
- Kenar belirleme işleminin kullanışlı olduğu diğer bir örnek, sadece görüntüye ilişkin kenarların kodlandığı düşük bit oranlarında görüntü kodlama uygulamasıdır.
- Kenar belirlemenin önemli uygulamalarından biriside, görüntü içerisindeki belirli nesnenelerin boyutunun doğru bir şekilde ölçülmesi işlemidir.
- Cisimlerin fiziksel özellikleri ile kenarları arasında doğrudan bir ilişki söz konusudur. Dolayısıyla görüntünün birçok fiziksel özelliği kenar bilgisinden ortaya çıkarılabilir. Dolayısıyla kenar belirleme konusu, görüntü analizindeki önemli konulardan birisidir.

Kenar belirleme yöntemleri

- Kenar belirleme yöntemlerine geçmeden önce bazı tanılamalar;
- Görüntüye ilişkin ilgilenilen bölgelerin kendi içerisinde yeterince homojen oldukları varsayılacak ve dolayısıyla iki bölge arasındaki geçiş sadece gri seviye süreksizliklerine dayalı olarak tanımlanabilecektir.

İyi bir kenar belirleyici;

- Kenarları iyi bir biçimde sezebilmelidir,
- Kenarları doğru konumlarda belirleyebilmelidir,
- Bir kenar için tek bir kenar görüntüsü oluşturabilmeli yani yapay kenarlar üretmemelidir.

- Görüntü içerisinde gürültünün varlığı (gürültü yüksek frekanslı bileşenlerden oluştuğu için gürültü ile kenarları birbirinden ayırt etmek güçleşecektir), kenar belirleme ve konumlama ölçütleri arasındaki karşılıklı ilişki (tradeoff) ve kenarların çok ölçekli yapısı, kenar belirleme aşamasında karşılaşılabilecek önemli sorunlardır.

Türev almaya dayalı kenar belirleme yöntemleri

- Bir görüntü içerisindeki kenarları belirlemek için uygulanabilecek en verimli yöntemlerden birisi, ani gri seviye değişimlerini tespit etmektir.
- Bu amaç için birçok kenar belirleme yönteminin kullandığı temel yaklaşım, bölgesel türev hesabına dayanır. Bölgesel olarak, görüntünün 1.türevi kenar bölgelerinde en büyük değere sahip olur (local maximum).
- Görüntünün 2.türevi ise kenar bölgelerinde sıfır değerini üretir. Bölgesel olarak görüntüye ilişkin 1. ve 2. türevleri hesaplayarak elde edilen lokal maksimum ve sıfır geçiş noktaları ile, ilgili görüntü bölgesi için kenarlar belirlenmiş olur.
- Bu derste kenar belirleme için; 1.türevi kullanan gradient yöntemi ve 2.türevi kullanan laplasyen yöntemini inceleyeceğiz.

Gradient (1.türeve dayalı) Kenar belirleme yöntemleri

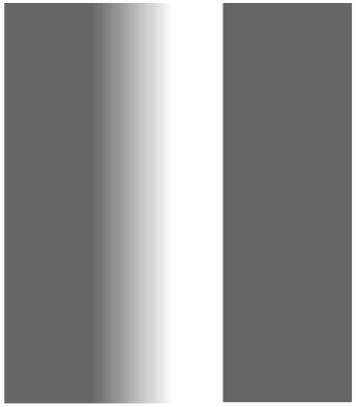
Sayısal görüntüde uygun türev almak için sayısal türev tanımı aşağıdaki denklemlerdeki gibidir.

$$\frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \qquad \lim_{h \to 0} \frac{f(x) - f(x-h)}{h}, \qquad \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$

Eğer h'ın alabileceği en küçük değer 1 olarak düşünülürse sayısal türev hesaplaması aşağıdaki gibi olur.

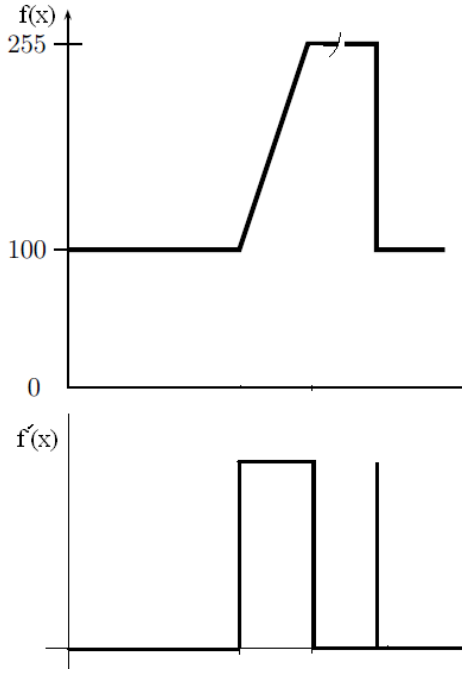
$$f(x+1) - f(x) \qquad f(x) - f(x-1), \qquad (f(x+1) - f(x-1))/2.$$

Bu işlemler tek boyutlu işlemlerdir.



Yumuşak geçişli kenar

İdeal geçişli kenar



1.Türeve dayalı kenar belirleme -2

- Görüntüler iki boyutlu olduğundan, biz kısmi sayısal türev kullanmalıyız. Buradan önemli bir tanımlamaya gradient bağıntısına geçebiliriz. Bir f(x,y) fonksiyonu için; gradiyent bağıntısı

$$\text{(Gradient)}\quad \nabla f \equiv grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$\text{(Gradient'in genliği)}\quad M(x,y) = \text{mag}(\nabla f) = \sqrt{g_x^{\,2} + g_y^{\,2}}$$

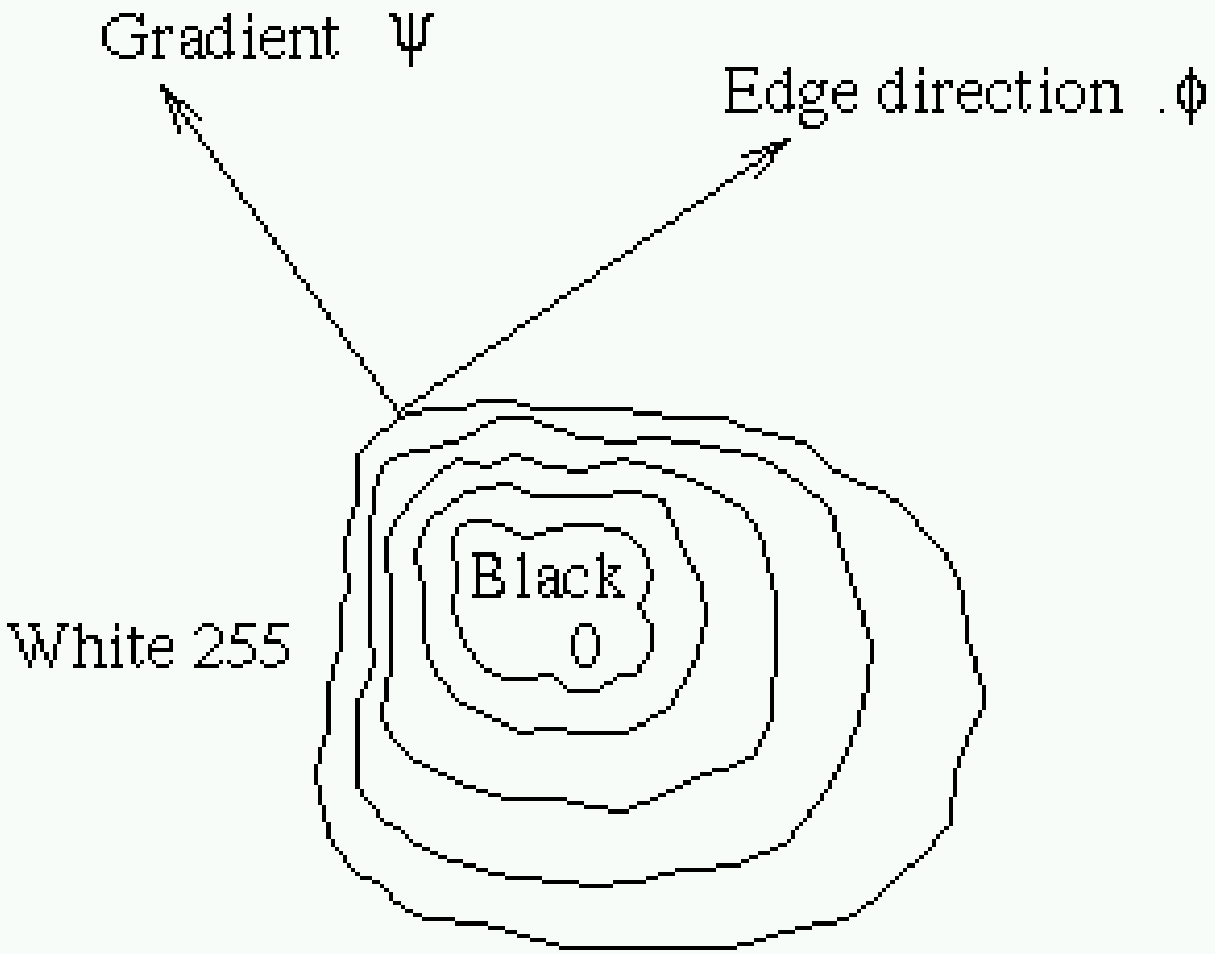
$$\text{Gradient'in yönü}\quad \alpha(x,y) = \tan^{-1}\left[\frac{g_x}{g_y}\right] \qquad \text{Kenarın yönü}\quad \phi = \alpha - 90^\circ$$

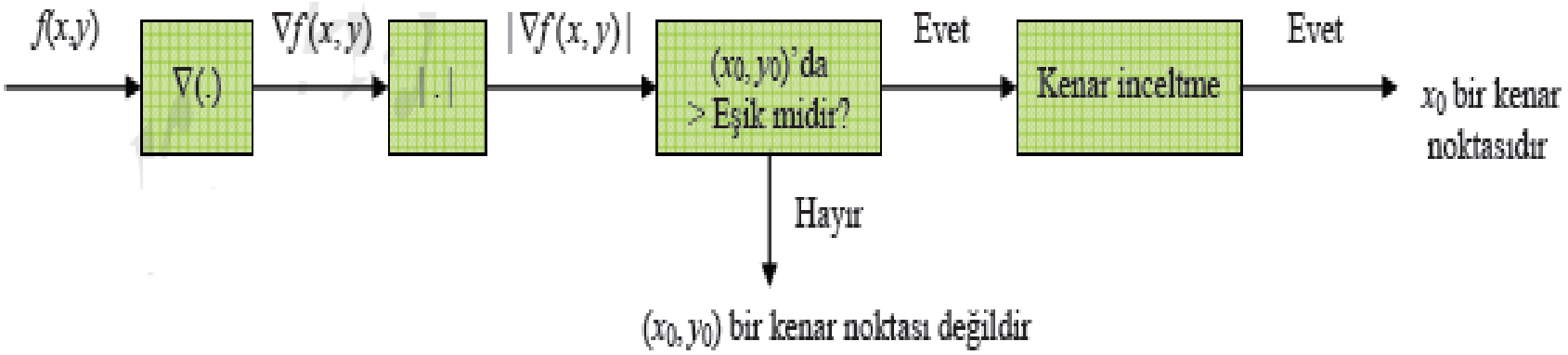
Gradient'in genliği pratikte aşağıda şekillerde hesaplanabilir

$$\text{mag}(\nabla f) = \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| \qquad \text{mag}(\nabla f) = \max\left(\left|\frac{\partial f}{\partial x}\right|, \left|\frac{\partial f}{\partial y}\right|\right)$$

şeklinde değişen bir vektördür. Bu vektör, f(x,y) fonksiyonundaki noktalar için en büyük artış yönünü elde eder. Bu artışın yönü ve genliği aşağıdaki bağıntılardan hesap edilir.

- *f '(x,y) aynı zamanda kenarın* yönünü ve büyüklüğünü kestirmede kullanılabilir. Eğer | *f '(x,y)* | *çok büyük ise, f(x,y) çok hızlı değişir ve bu durum* parlaklıkta hızlı bir değişime karşı düşer. Eğer *f '(x,y) pozitif ise, f(x) artan bir fonksiyondur.*
- Buradan hareketle birçok kenar belirleme algoritmasının temeli,, gradiyenin genliğinin bulunması ve onu bir eşik değerle karşılaştırmaya dayanır.


Gradient Ψ
Edge direction .ϕ
Black
0
White 255



2-B kenar belirleme sisteminin blok diyagramı

- *Şekildeki* sisteme göre, ilk olarak *f(x, y)'nin gradienti olan ∇f (x, y) fonksiyonunun genliği hesaplanır ve bu* genlik, aday kenar noktalarını tanımlamak için bir eşik değeri ile karşılaştırılır. Eğer ∇*f (x, y)* fonksiyonunun genliğinin belirli bir eşik değerinden büyük olduğu bütün (*x, y) noktaları kenar noktası olarak* tanımlanırsa, oluşan kenar görüntüsü ince çizgilerden ziyade kalın şeritler halinde görünecektir.
- Kalın şeritlerden ince çizgiler elde etmek için, kenar görüntüsü inceltme (thinning) işlemine tabi tutulur.

•Gradient tabanlı kenar belirleme sistemleri iki değişik şekilde uygulanabilir. Bunlar, yönlü ve yönsüz kenar kestirimcileri (directional and nondirectional edge detectors) olarak adlandırılır.

| ∇f (x, y) | *fonksiyonunu kullanan* kenar belirleme sistemlerine *yönsüz kenar kestirimcisi adı verilir ki bu sistem herhangi bir doğrultu için ayarlı* değildir ve her yön için eşit ağırlıklı sonuçlar üretir.

Eğer kenar belirleme sistemi herhangi bir yön için ayarlanmışsa bu sistemlere *yönlü kenar kestirimcisi adı verilir. Örneğin, bir önce açıklanan kenar belirleme sisteminde* |∇*f (x, y)*| *fonksiyonu yerine sadece*

| ∂f (x, y) / ∂x | *fonksiyonu kullanılırsa, bu durumda sistem sadece düşey* yöndeki kenarları tanır ve yatay yöndeki kenarlara ise cevap vermez.

SOBEL Kenar detektörü

Sobel kenar detektörü 1. kısmi türevlere dayanır. Aşağıdaki maskeyi kullanır. Başka bir değişle maske ortasına denk gelen pikselin gradienti komşularda gözönüne alınarak aşağıdaki gibi hesaplanır. Matlab fonksiyonu olarak;

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

Burada f, giriş görüntüsü, T eşik değeri, dir, kenar dedektörünün tavsiye edeceği yön indeksidir. 'Horizontal', 'vertical' veya 'both'

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Maske altındaki görüntü pikselleri

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

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

$$G_X = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)$$

$$G_Y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)$$

$$g = \left[G_x^2 + G_y^2\right]^{1/2}$$

$$g = \left[\left[(Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)\right]^2 + \left[(Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)\right]^2\right]^{1/2}$$

$g \geq T$ ise f(x,y) pikseli bir kenar pikselidir.

SOBEL ALGORİTMASI

- Sx=[1 2 1;0 0 0;-1 -2 -1];
- Sy=[-1 0 1;-2 0 2;-1 0 1];
- Gx(i,j) = $\sum_{k=1}^{3}\sum_{l=1}^{3} Sx(k,l) * f(i+k, j+l)$;
- Gy(i,j) = $\sum_{k=1}^{3}\sum_{l=1}^{3} Sy(k,l) * f(i+k, j+l)$;
- Bir piksel için $Gx(i,j)^2 + Gy(i,j)^2 = G(i,j)^2$
- Tüm resim için
- $\sum_{i=1}^{m}\sum_{j=1}^{n} G(i,j) = \sum_{i=1}^{m}\sum_{j=1}^{n} \sqrt{Gx(i,j)^2 + Gy(i,j)^2}$

SOBEL ALGORİTMASI

```
A=imread('peppers.png');
B=rgb2gray(A);
C=double(B);
for i=1:size(C,1)-2
    for j=1:size(C,2)-2
        %Sobel mask for x-direction:
        Gx=((2*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-
            (2*C(i,j+1)+C(i,j)+C(i,j+2)));
        %Sobel mask for y-direction:
        Gy=((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-
            (2*C(i+1,j)+C(i,j)+C(i+2,j)));
              %The gradient of the image
        %B(i,j)=abs(Gx)+abs(Gy);
        B(i,j)=sqrt(Gx.^2+Gy.^2);

    end
end
figure,imshow(B); title('Sobel gradient');
```

Prewitt Kenar detektörü

- Prewitt dedektör Sobel dedektör göre hesaplama uygulamak için biraz daha basit, ama biraz noiser sonuçlar üretme eğilimindedir.
- Matlab fonksiyonu olarak Sobel'in parametreleri ile aynıdır.

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

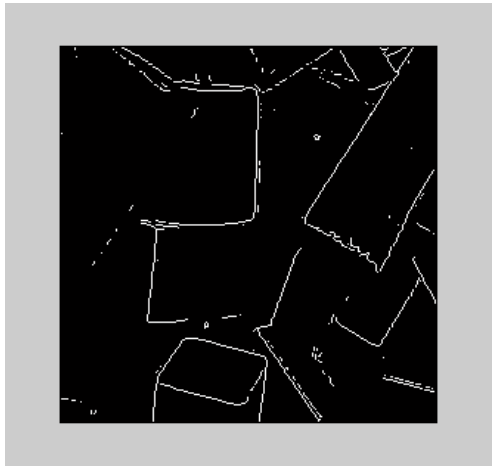
Masks: $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

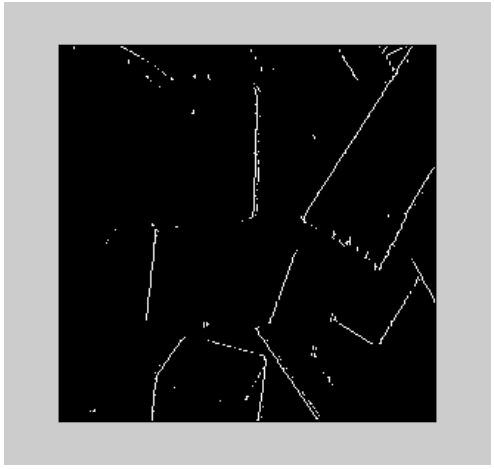
$G_X = (Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3)$

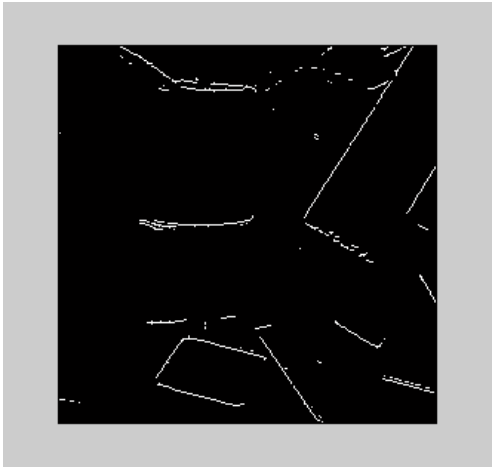
$G_Y = (Z_3 + Z_6 + Z_9) - (Z_1 + Z_4 + Z_7)$

Sobel Örnek, T=0.1

```
>> a=imread('blocks.tiff');
>> imshow(a)
>> s=edge(a,'sobel', 0.1);
>> imshow(s)
>> s=edge(a,'sobel', 0.1, 'vertical');
>> imshow(s)
>> s=edge(a,'sobel', 0.1, 'horizontal');
>> imshow(s)
>> s=edge(a,'sobel', 0.3);
```







Prewitt örneği T=0.05

```
>> a=imread('lena_std.tif');
>> g = rgb2gray(a);
>> imshow(g)
>> s=edge(g,'prewitt', 0.05);
>> imshow(s)
```

Prewitt örneği

```
>> s=edge(g,'prewitt',0.05, 'horizontal');
>> imshow(s)
```

```
>> s=edge(g,'prewitt',0.05, ‘vertical');
>> imshow(s)
```

ÖDEV

- Prewit ve Robert kenar çıkarımlarını gerçekleştiren m-file dosyasını yazınız?
- Ödev Teslim Tarihi: 25.12.201

Roberts Kenar dedektörleri

Roberts detektörü görüntü işlemede en eski ve en basit detektörlerdir. Bu detektör yalnız yatay veya düşey kenarları detekte edebilir. Hızlı ve basit olduğundan gerçek zamanlı uygulamalarda kullanılmaktadır.

Matlab fonksiyonu aşağıdaki gibidir.

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

Masks: $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

$G_X = Z_9 - Z_5$ $G_Y = Z_8 - Z_6$

2.Türeve dayalı kenar belirleme (Laplasian yöntemi)

Herhangi bir işarete ilişkin ani geçiş noktasının belirlenmesi için başvurulan yöntemlerden diğeri, fonksiyona ilişkin 2.türevi kullanır. İşarete ilişkin 1.türevin en büyük veya en küçük olduğu noktada işaretin 2.türevi sıfıra eşittir. Bu gerçeğe dayanarak, bir görüntü fonksiyonuna ilişkin 2.türev alınıp sıfır geçiş noktalarının tespit edilmesiyle görüntüye ilişkin kenar görüntüsüne ulaşmak mümkündür.

2-Boyutlu bir f(x, y) fonksiyonunun x ve y değişkenlerine göre 2.türevi, f(x, y)'nin ayrık (Diskrate) laplasyeni olarak adlandırılır . 1.türeve göre önemli bir avantaja sahiptir. Bu da rotasyon altında değişmez olduğudur. Aşağıdaki ifadelerle verilir.

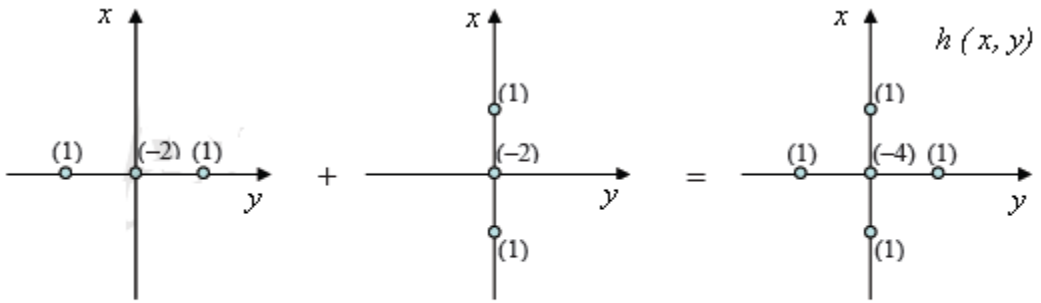
Bu durum 3*3'lük bir filtre matrisle de ifade edilebilir.

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

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

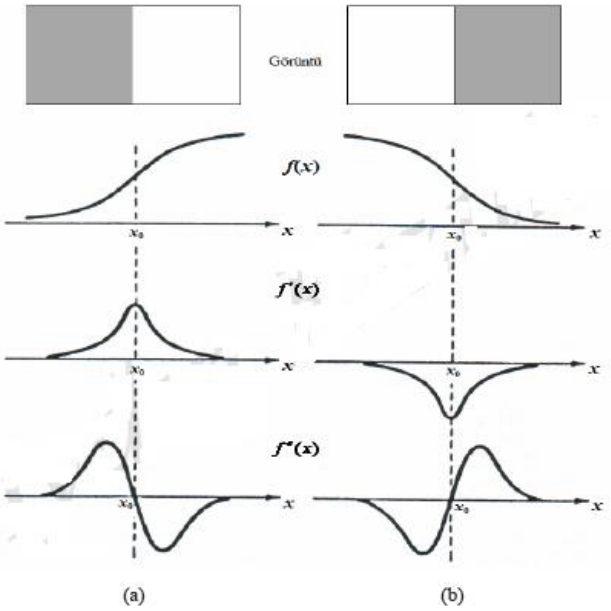
$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

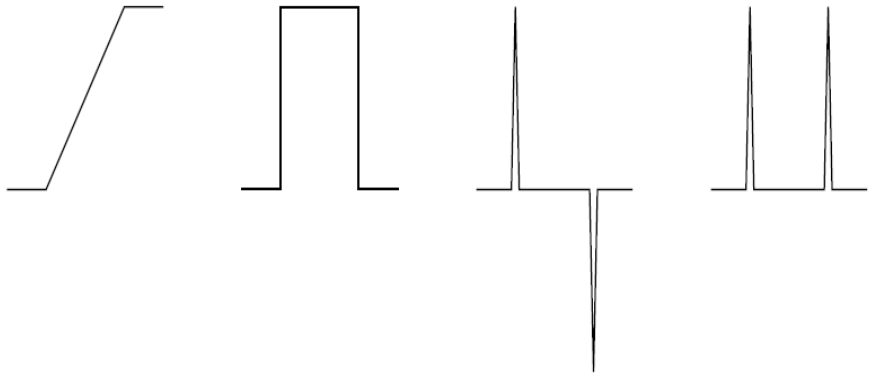
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$



Satır boyunca 2.türev | Sütun boyunca 2.türev | Laplasyen maskesi

$f(x,y)$ 'nin laplasyenini, $f(x,y) \otimes h(x,y)$ ile hesaplamada kullanılan süzgeç impuls cevabı





The edge | First derivative | Second derivative | Absolute values

Sonuç olarak, laplasyen kenar tanıma yönteminde, görüntü fonksiyonu *f(x, y)'yi verilen* laplasyen kenar maskeleri ile konvolüsyona tabi tuttuktan sonra elde edilen laplasyen görüntüsünün mutlak değerinin bir eşik değeri ile karşılaştırılması sonucunda kenar görüntüsüne ulaşılır.

Gradient tabanlı yöntemde de aynı işlemler yapılmasına rağmen laplasyen tabanlı olarak elde edilen kenar görüntüsünde bazı farklar vardır. Bu farklar aşağıdaki gibi sıralanabilir:

• Laplasyen hesabında ikinci dereceden türevler söz konusu olduğundan dolayı, gradientine nazaran bir görüntünün laplasyeni kabul edilemez derecede gürültüye duyarlı olacaktır.

• Görüntünün laplasyeninin mutlak değeri çift kenarların oluşmasına neden olacaktır ki bu durumda kenar yönünün belirlenmesi mümkün olmayacaktır. Bu durum, bir pikselin, kenarın koyu bölgesinde mi yoksa açık bölgesinde mi olduğunu saptamada zorluk teşkil edecektir.

• Sıfır geçişleri, görüntü içerisindeki cisimlere ilişkin sınırları temsil ettiği için kenar görüntüsü sürekli çizgiler halinde olacaktır. Bundan dolayı, gradient-tabanlı yöntemlerde gerekli olan kenar inceltme işlemine laplasyen-tabanlı yöntemlerde ihtiyaç duyulmaz.

• Görüntü içerisindeki en küçük ani değişimler bile bir sıfır geçişi oluşturacakları için, laplasyen-tabanlı yöntem ile elde edilen kenar görüntülerinde çok fazla sayıda yanlış kenar noktalarının oluşması kaçınılmazdır.

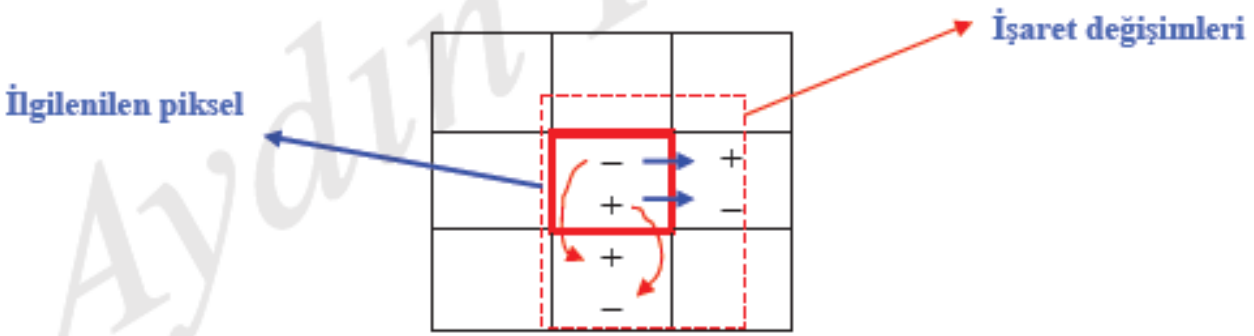
Buna göre, bir görüntünün kenarlarını elde etmek için gerçekleştirilecek laplasyen-tabanlı kenar belirleme algoritması aşağıdaki aşamalardan oluşacaktır:

1. Gri-ton görüntünün laplasyenini elde etmek için, görüntü fonksiyonu *f(x, y)* *laplasyen* maskesi ile konvolüsyona tabi tutulur.

2. Süzgeçlenmiş görüntüdeki (Laplasyen görüntüsü) tüm sıfır geçiş noktaları (zero-crossing points) elde edilir.

Genel olarak laplasyen görüntüsünde sıfır geçiş noktasına karşı düşen pikseller tam olarak sıfıra eşit olmayabilir. Bu durumda, ilgilenilen pikselin bir sıfır geçiş noktasına karşı düşüp düşmediğini belirlemek için komşuluktaki piksel değerlerine bakılır ve {-,+} veya {+,-} şeklinde işaret değişimlerinin olup olmadığı belirlenir. Eğer ki işaret değişimi varsa, sıfır geçişin eğimi (piksel değerlerinin farkının mutlak değeri) bir eşik değeri ile karşılaştırılır. Eşik değerinden büyük olan eğimler kenar noktası olarak belirlenir.

Genellikle eşik değeri, laplasyen görüntüsünün mutlak değerinin ortalamasının %75'i alınarak belirlenmektedir.



Laplasyen görüntüsünde sıfır geçiş noktalarının belirlenmesi için basit bir yaklaşım

Sıfır geçiş belirlemeye örnek

Laplasian filtresinden geçirilmiş görüntü matrisinde; aşağıdaki özelliklerden birini sağlayan piksel sıfır geçiş pikselidir ve dolayısıyla kenar üzerinde bir noktadır.

1- Negatif bir gri değerine sahip olmalı ve bağlantılı komşu piksellerinden en az birisinin gri değeri pozitif olmalı.

2- Pikselin değeri sıfır ise, onun yatay veya dikey komşu piksellerinin değeri farklı işaretlerde olmalı.

50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50
50	50	200	200	200	200	200	200	50	50
50	50	200	200	200	200	200	200	50	50
50	50	200	200	200	200	200	200	50	50
50	50	200	200	200	200	200	200	50	50
50	50	50	50	200	200	200	200	50	50
50	50	50	50	200	200	200	200	50	50
50	50	50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50	50	50

(a) A simple image

−100	−50	−50	−50	−50	−50	−50	−50	−50	−100
−50	0	150	150	150	150	150	150	0	−50
−50	150	−300	−150	−150	−150	−150	−300	150	−50
−50	150	−150	0	0	0	0	−150	150	−50
−50	150	−150	0	0	0	0	−150	150	−50
−50	150	−300	−150	0	0	0	−150	150	−50
−50	0	150	300	−150	0	0	−150	150	−50
−50	0	0	150	−300	−150	−150	−300	150	−50
−50	0	0	0	150	150	150	150	0	−50
−100	−50	−50	−50	−50	−50	−50	−50	−50	−100

(b) After laplace filtering

Laplasian yönteminde, görüntüdeki en küçük gri seviyelerdeki değişimlerde bile sıfır geçiş noktası üretebileceği ve Görüntünün küçük seviyelerde de olsa gürültü ile etkilenmiş olduğu göz önüne alınırsa, laplasyen-tabanlı yöntemlerin doğrudan doğruya görüntü üzerine uygulanması pek sık kullanılan bir yol değildir. Bu durum aşağıdaki şekillerden de görülmektedir.

```
>> a=imread('cameraman.tif');
>> imshow(a)
```

```
>> a=imread('cameraman.tif');
>> l=fspecial('laplacian',1);
>> icz=edge(a,'zerocross',l);
>> imshow(icz)
```

Dolayısıyla, laplasian metodunun doğrudan kullanımı yerine, daha genel olan Laplasyen-Gauss (LoG) yöntemi kullanılır. İlk olarak Marr-Hildreth tarafından önerildiğinden dolayı, bu yönteme aynı zamanda *Marr-Hildreth kenar belirleme yöntemi adı da verilir.*

Marr-Hildreth, görüntüyü değişik kesim frekansları ile sınırlandırdıktan sonra her bir sınırlandırılmış görüntü üzerine kenar tanıma işlemleri uygulayarak değişik kenar görüntüleri elde etmeyi hedeflemişlerdir.

LoG yönteminde, görüntüyü değişik kesim frekanslarında sınırlandırmak için Gaussian alçak geçiren süzgeç kullanılmaktadır (Bu şekilde görüntü geçişleri yumuşaklaştırılır). Görüntünün doğrudan laplasyeninin alınması gürültüyü daha da kuvvetlendireceği için arzu edilmeyen birçok yapay kenar noktasının elde edilmesi olasıdır. LoG yöntemi ile doğrudan doğruya görüntünün laplasyenini almaktansa görüntünün bir Gaussian fonksiyonu ile konvolüsyona tabi tutup daha sonra laplasyenini hesaplamak daha mantıklı bir yol olarak görünmektedir.

Bu yöntemin MATLAB'da uygulaması ise;

Laplasian of Gaussian (lOG)

```
>>a=imread('cameraman.tif');
>> l=fspecial('laplacian',1);
>> icz=edge(a,'zerocross',l);
>> imshow(icz)
```

```
> log=fspecial('log',13,2);
>> icz=edge(a,'zerocross',log);
>> imshow(icz)
```