Bilgisayar Görmesi

Ders 6:KENAR BULMA

Dr. Öğr. Üyesi Serap ÇAKAR

Kenar Bulma

Kenarlar sınırları karakterize eder. Bir görüntüdeki kenarlar güçlü yoğunluk zıtlığı olan bölgelerdedir. Bir yoğunluktan diğerine sıçrama noktasıdır.

Bir görüntünün kenarlarını bulma görüntüdeki önemli yapısal özellikleri korurken verinin büyük bir kısmını önemli ölçüde azaltır ve gereksiz bilgiyi filtreler.

Kenar bulmanın birçok yolu vardır. Bu metotlar iki gruba ayrılır;

Gradyan (Eğim)

Laplasian metotları.

Kenar Bulma

Gradyan metodu; görüntünün birinci türevindeki maksimum ve minimum değerlere bakma ile kenarları tespit eder.

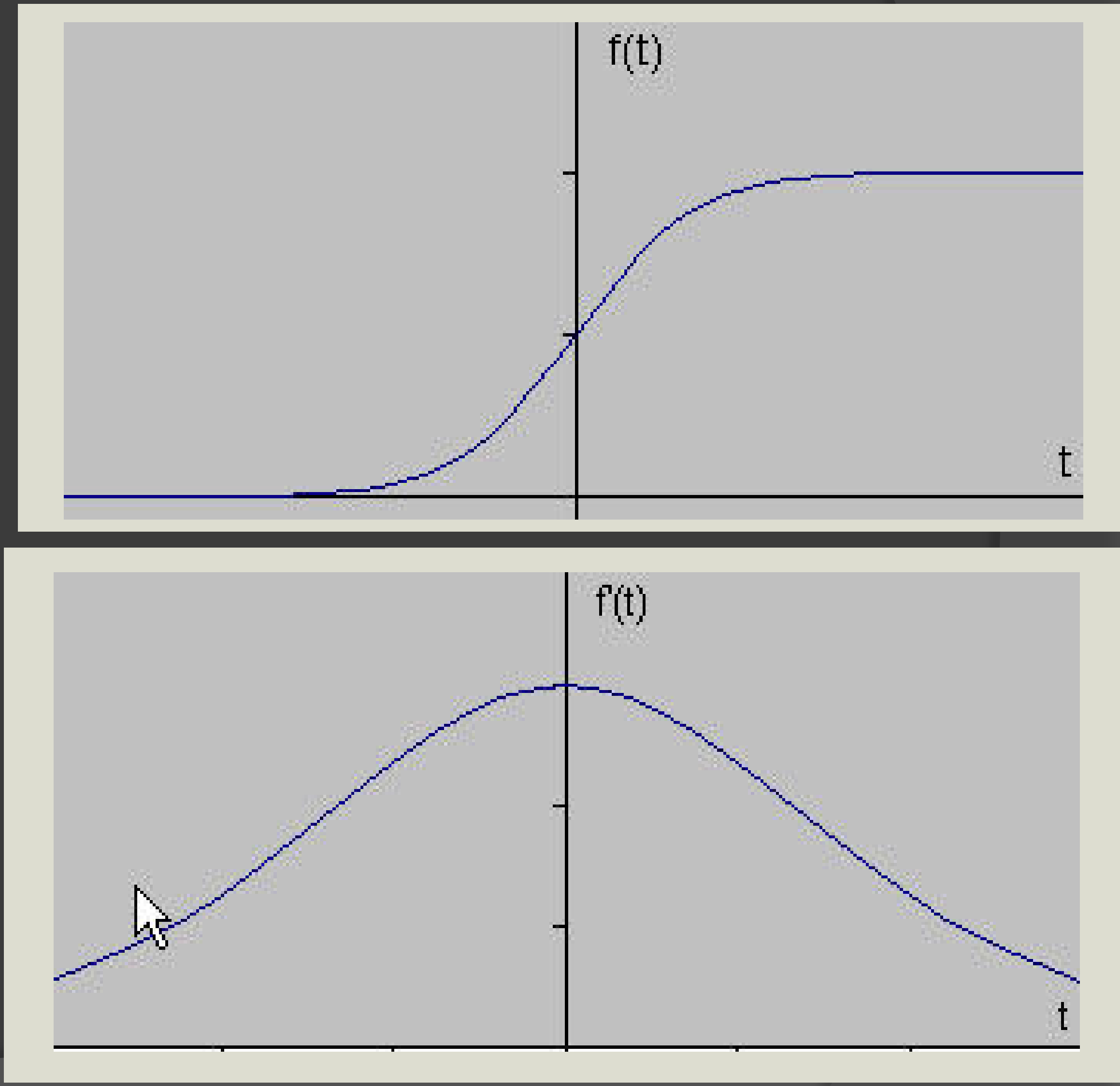
Laplasyan metodu; kenarları bulmak için ikinci türevdeki sıfır geçişlerini arar.

Bir kenar, rampa fonksiyonunun bir boyutlu şekline sahiptir.

Kenar Bulma

Yandaki sinyali göz önünde bulunduralım. Yoğunluklar arasındaki sıçrama kenarı verir.

Üstteki işaretin türevi (t'ye göre türevi alınırsa) aşağıdaki işareti verir.



Kenar Bulma

Türev işlemi orijinal sinyaldeki kenarın merkezindeki maksimum değeri gösterir.

Birinci türeve dayanan bu kenar tespiti yöntemi “gradyan filtre” metodunun karakteristik özelliğidir. Gradyan filtre metodunu kullanan kenar bulmaya örnek Sobel verilebilir.

Gradyan değeri eşik değerini aşarsa pikselin yeri kenarın yerini gösterir.

Kenar Bulma

Kenarları belirten piksellerin renk değerleri kenarın dışındaki piksellerden daha büyüktür.

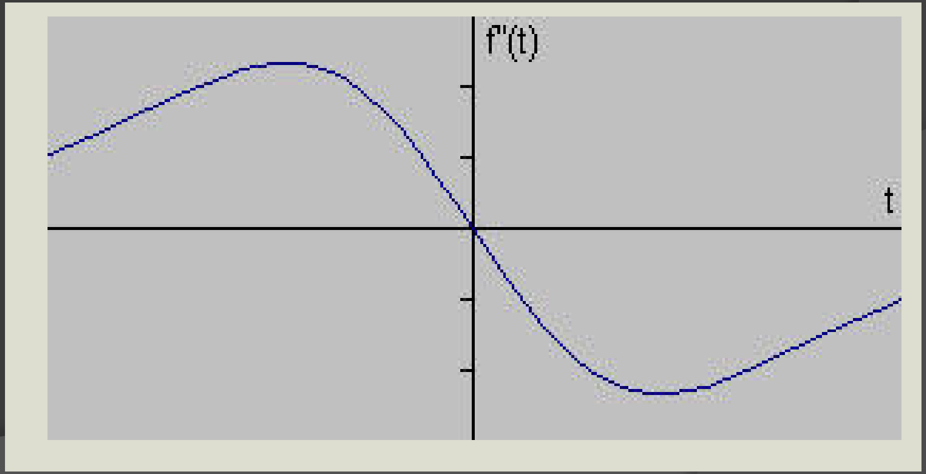
Böylece bir kere eşik değeri ayarlandığında gradyan değerini eşik değeri ile karşılaştırabiliriz ve eşik değeri aşıldığında kenarı tespit edebiliriz.

Kenar Bulma

Ayrıca ilk türev maksimumda iken ikinci türev sıfırda olur.

Sonuç olarak kenar bulmanın alternatif bir yolu ikinci türevde sıfır noktasını tespit etmektir.

Bu metot Laplasyan olarak bilinir ve işaretin ikinci türevi aşağıda gösterilmiştir. Örnek olarak Marr-Hilderth metodu verilebilir.



Kenar Bulma

Sayısal fonksiyonlarda birinci türev;

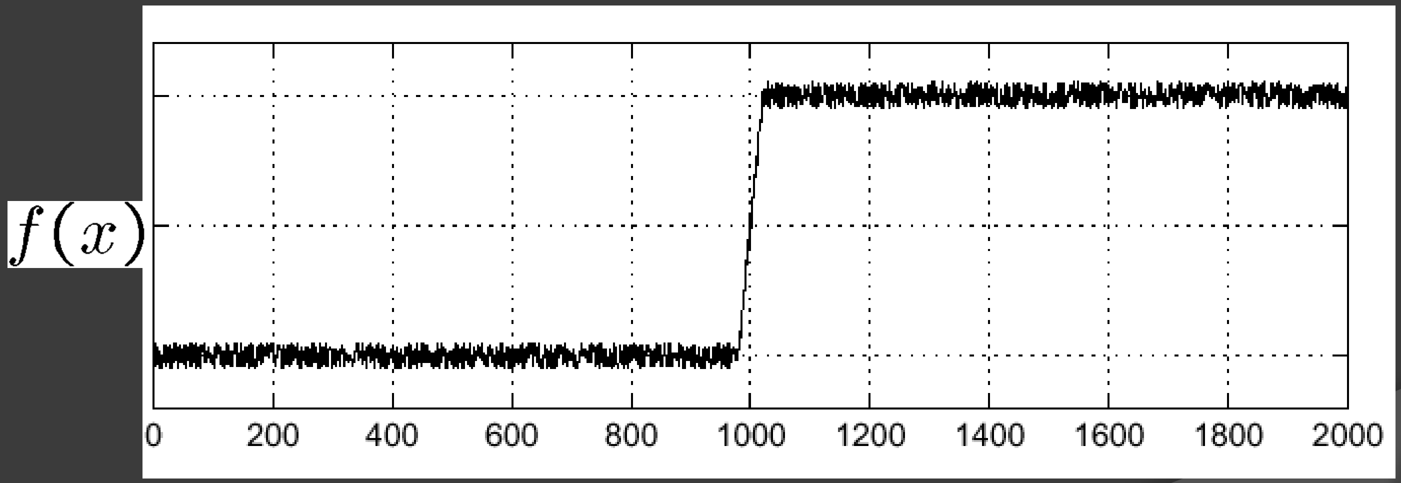
- Düz zeminde sıfır olmalıdır (görüntüdeki tek renk olan bölümler),
- Grilik seviyesindeki ani değişimlerde veya rampa geçişlerinde sıfırdan farklı olmalıdır,
- Rampa boyunca sıfırdan farklı olmalıdır.

İkinci türev ise,

- Düz zeminde sıfır olmalıdır,
- Rampa değişiminin başında ve sonunda sıfırdan farklı olmalıdır,
- Sabit eğimli rampa boyunca sıfır olmalıdır

Kenar Bulma

- Bir görüntüye ait satır veya sütun vektörünü göz önüne alırsak,
  - Vektörü oluşturan piksellerin pozisyonlarına göre parlaklık çizdirildiğinde aşağıdaki *f(x)* fonksiyonu elde edilir

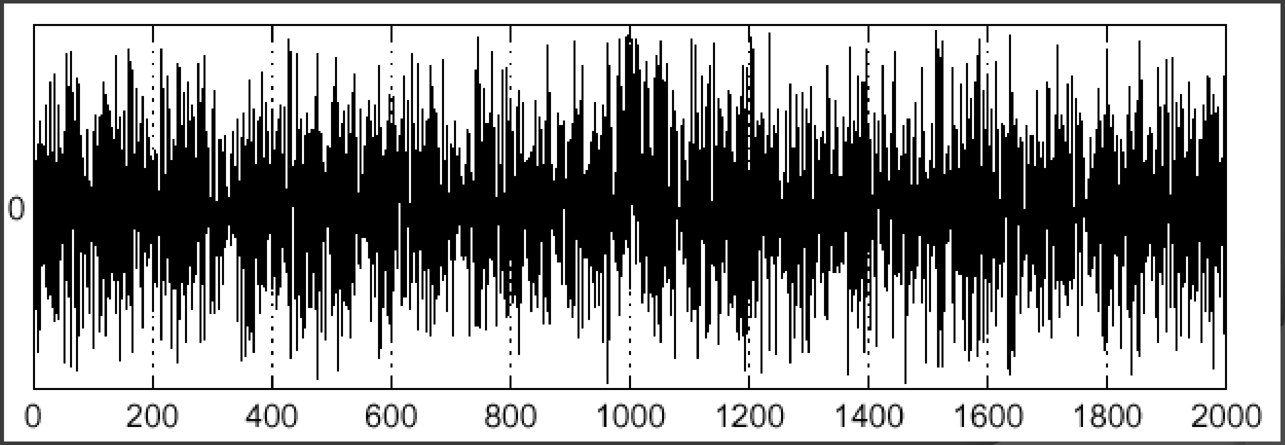


Kenar Bulma

- Bu *f(x)* fonksiyonunun birinci dereceden türevi alındığında aşağıdaki sinyal elde edilir

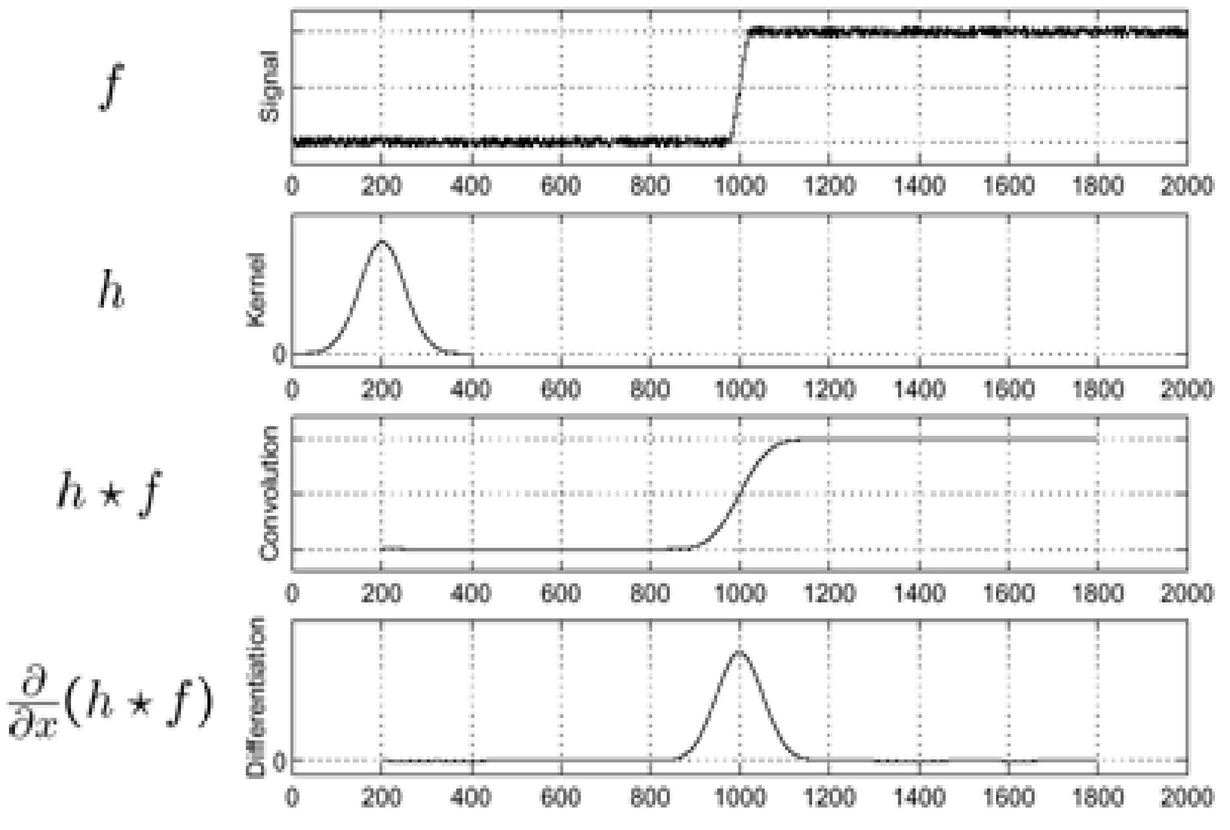
- Bu sinyale bakıldığında görüntüdeki kenar noktası belirgin değildir

$$\frac{d}{dx}f(x)$$



Kenar Bulma

- Sinyaldeki kenarı gösteren noktanın bulunması için, sinyalin bir *h* Gauss fonksiyonu ile çarpılması çözüm olabilir;



Kenar Bulma

- *f* fonksiyonunda siyahtan beyaza yumuşak bir geçiş vardır.
- İdeal bir kenar aslında basamak fonksiyonudur ancak görüntüyü elde eden sistemler ve örnekleme işlemlerinden dolayı kenarlar daha çok rampa fonksiyonlarına benzemektedir.
- $h*f$ konvolüsyonu sonucunda gürültü yok edilmiştir.
- Sonucun birinci dereceden türevi alındığında piksel değerlerinin sabit kaldığı yerlerde değişim sıfır olduğu için sonuç sıfır olmuş, değişimin olduğu yerde (kenar) ise sonuç sıfırdan farklı olmuştur.

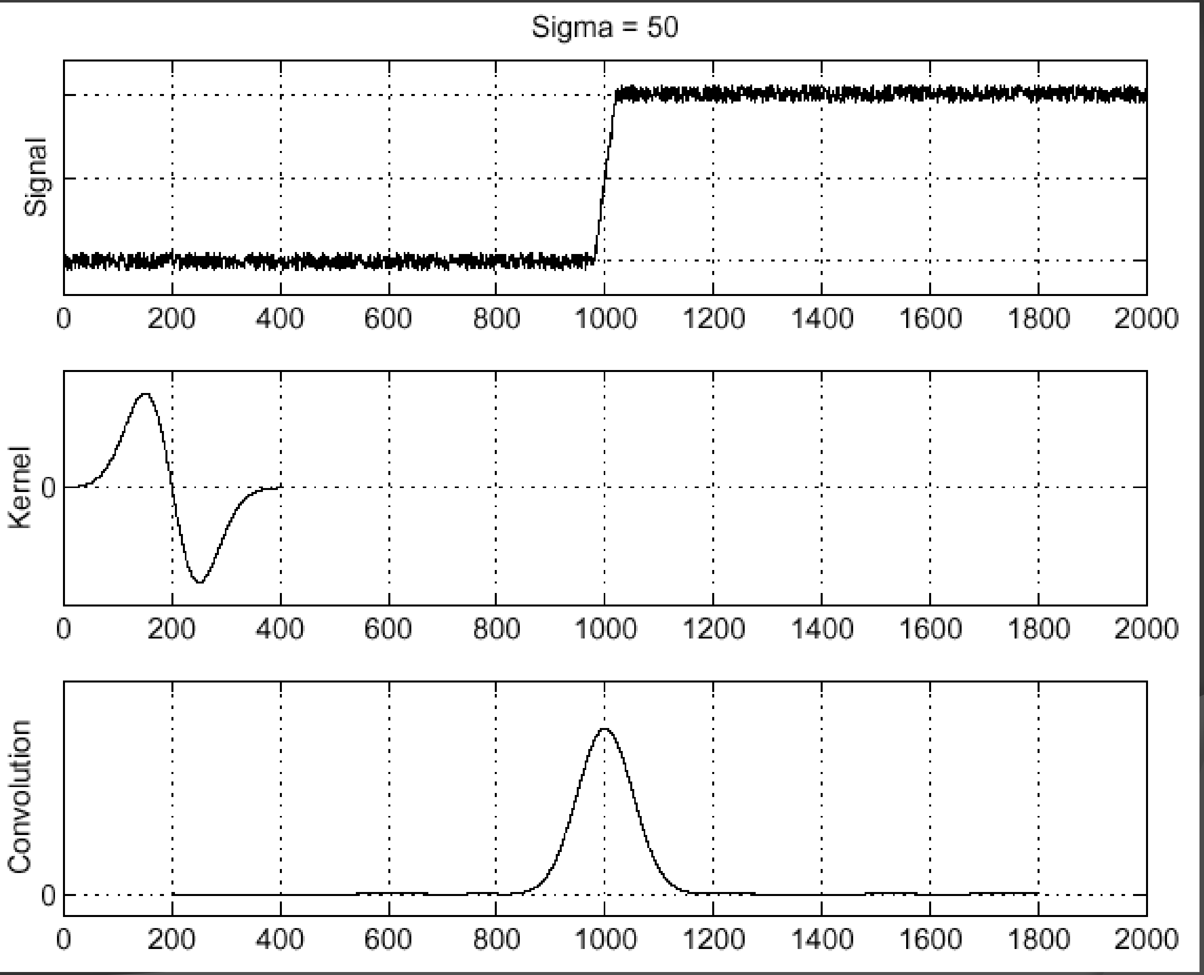
Konvolüsyonun Birleşme Özelliği

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$



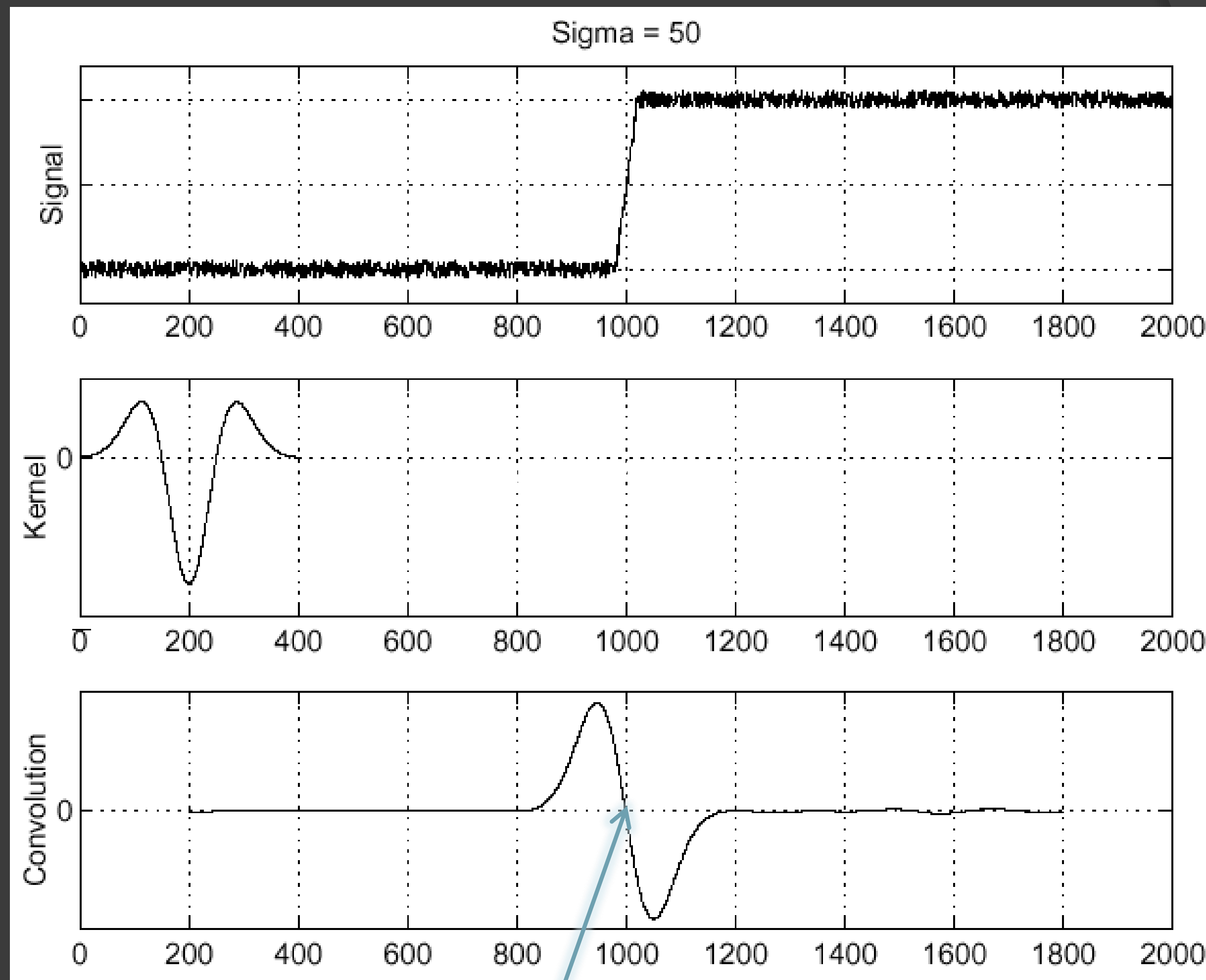
Laplasyan

$$\frac{\partial^2}{\partial x^2}(h \star f)$$

f

$\frac{\partial^2}{\partial x^2}h$

$(\frac{\partial^2}{\partial x^2}h) \star f$



Sıfır noktası Kenarı temsil eder

Laplasyan

- Sayısal görüntüler sonlu sayısal nicelikler olduğu için ve maksimum grilik seviyesi değişimleri de sonlu olduğu için en yakın mesafedeki değişim birbirine komşu iki piksel değeri arasındadır.

Laplasyan

- Buradan hareketle bir boyutlu *f(x)* fonksiyonunun birinci türevi

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

denklemiyle verilir.

- İkinci türev ise

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

olacaktır.

Laplasyan

- Buradan da iki boyutlu *f(x,y)* fonksiyonun türevi

$$\Delta^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

denklemiyle verilir.

Laplasyan

- *x* -yönünde kısmi ikinci türevi göz önüne alırsak,

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

- *y* -yönünde kısmi ikinci türev ise,

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

Laplasyan

- İki boyutlu Laplasyan operatörünün sayısal uygulaması *x* ve *y* yönündeki bu iki bileşenin toplanmasıyla elde edilir;

$$\Delta^2 f = \left[f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1) \right] - 4f(x,y)$$

Laplasyan

- Laplasyan türev operatörü, görüntüdeki sürekli olmayan grilik seviyelerini belirginleştirecek, grilik seviyesinin çok yavaş değiştiği bölümleri ise azaltacaktır.

- Dolayısıyla Laplasyan operatörü bir görüntüye uygulandığında çıkışta kenar çizgileri, sürekli olmayan grimsi noktalar ve siyah arka plana sahip bir görüntü elde edilir.

Laplasyan

- Laplasyan operatörünün dezavantajı görüntüdeki gürültüye çok duyarlı olmasıdır.
- Diğer bir dezavantajı ise her bir kenar için iki değer üretmesidir.
- Gürültünün meydana getireceği bozulmaları engellemek için Laplasyan'dan önce görüntüye Gauss kerneli uygulanır.
- Gauss kerneli bir alçak geçiren filtredir. İşlem sonucunda görüntü bulanıklaşır.
- Ardından Laplasyan uygulanarak sıfır kesişleri tespit edilir.

1. ve 2. Türev Fonksiyonları

- Bir fonksiyona ait birinci türev; **(Gradyan Metot)**
  - Görüntüdeki kalın kenarları,

- Bir fonksiyona ait ikinci türev; **(Laplasyan Metot)**
  - Görüntüdeki ince kenar, gürültü noktaları gibi ince detayları,

belirtir.

- Bu ifadelere göre ikinci türev birinci türeve göre görüntüdeki ince detayları daha iyi yakalamamızı sağlar.

SOBEL KENAR BULMA

- Doğru bir yaklaşımla, bir boyutlu analiz temeline dayanarak iki boyutlu bir görüntünün türevi hesaplanabilir.

- Sobel işlemi bir görüntüde 2-boyutlu özel bir eğim (gradyan) ölçümü gerektirir.

- Bu işlem görüntüde her bir noktada kesin eğim genliğini bulmak için kullanılır.

SOBEL KENAR BULMA

- Sobel kenar algılayıcı; biri x-yönündeki eğimi, diğeri y-yönündeki eğimi hesaplamak üzere 3x3'lük bir konvolüsyon maske çifti kullanır.

- Konvolüsyon maskesi görüntü boyutundan daha küçük olur.

- Mantık olarak bir kerede bir pikseli işleyerek maske görüntüde hareket eder.

SOBEL KENAR BULMA

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

Gx

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

Gy

$$|G| = \sqrt{Gx^2 + Gy^2}$$

SOBEL Kenar Bulma İşlem Adımları

- Görüntünün üst köşesine yerleştirilen maske, merkezine denk gelen pikselin değerini değiştirerek bir sonraki piksele geçer.

- Aşağıdaki örnek görüntüde, maskenin merkezi değiştirilecek pikselin etrafına yerleştirilir.

- Satır, sütun değerlerini ifade eden (I, J) değerlerinin değiştirilmesi ile maske hareket ettirilir.

SOBEL KENAR BULMA

- 3x3'lük maske ile ilk ve son satır ve ilk ve son sütundaki piksellerin işlenemeyeceği açıktır.

- Eğer maske ilk satırdaki piksele yerleştirilseydi, görüntü sınırlarının dışına taşardı.

- Bu durumda görüntüye birer satır ve sütun eklenerek bu problem ortadan kaldırılabilir.

SOBEL KENAR BULMA

a_{11}	a_{12}	a_{13}	…	a_{1n}
a_{21}	a_{22}	a_{23}	…	a_{2n}
a_{31}	a_{32}	a_{33}	…	a_{3n}
⋮	⋮	⋮		⋮

m_{11}	m_{12}	m_{13}
m_{21}	m_{22}	m_{23}
m_{31}	m_{32}	m_{33}

b_{11}	b_{12}	b_{13}	…	b_{1n}
b_{21}	b_{22}	b_{23}	…	b_{2n}
b_{31}	b_{32}	b_{33}	…	b_{3n}
⋮	⋮	⋮		⋮

$$b_{22} = (a_{11}*m_{11}) + (a_{12}*m_{12}) + (a_{13}*m_{13}) + (a_{21}*m_{21}) + (a_{21}*m_{21}) +$$

$$(a22*m_{22}) + (a_{23}*m_{23}) + (a_{31}*m_{31}) + (a_{32}*m_{32}) + (a_{33}*m_{33})$$

SOBEL KENAR BULMA

$$b_{22} = (a_{11} * m_{11}) + (a_{12} * m_{12}) + (a_{13} * m_{13}) + (a_{21} * m_{21}) + (a_{21} * m_{21}) +$$
$$(a22 * m_{22}) + (a_{23} * m_{23}) + (a_{31} * m_{31}) + (a_{32} * m_{32}) + (a_{33} * m_{33})$$

Gx maskesi yatay yöndeki kenarları, Gy maskesi dikey yöndeki kenarları vurgular.

Bunların toplanması ile her iki yöndeki kenarlar tespit edilir.

SOBEL KENAR BULMA

- Sobel kerneli aşağıdaki matrislerden elde edilir;

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Yüzey eğimi mean filtresi ile yumuşatılır.

LAPLASYAN

Laplasyan x ve y yönlerindeki ikinci türev için 5x5'lik bir tane konvolüsyon maskesi kullanır.

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

Gradyan vs. Laplasyan

- H = fspecial('laplacian');
- L = imfilter(I,H,'replicate');
- imshow(L);

- H=fspecial('sobel');
- S = imfilter(I,H ,'replicate');
- imshow(S);

Robert Kenar Bulma

- En basit eğim operatörü, köşeden köşeye çapraz geçiş yapan Robert operatörüdür;

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

veya

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

  - Bu kerneller, düzlemsel bir yüzeye oturtulmuş 2x2 bir pencerenin en küçük kare hatasının eğimini verecek şekilde türetilmişlerdir.

Prewitt Kenar Bulma

- Prewitt kernelleri aşağıdaki gibidir;

$$\nabla_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \nabla_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\nabla_x = \begin{bmatrix} -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \\ -3 & -1 & 1 & 3 \end{bmatrix} \qquad \nabla_y = \begin{bmatrix} 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -3 & -3 & -3 & -3 \end{bmatrix}$$

Marr-Hill Kenar Bulma

- Marr-Hilderth yöntemi LoG (Laplacian of Gaussian) operatörünün sıfır kesişlerine dayanmaktadır.
- Sıfır kesişi, bir fonksiyonun sıfır eksenini kestiği yer olarak tanımlanır.
- Bu yöntemde, sıfır kesişleri boyunca kenarları bulmak için Laplasyan ön işlem olarak uygulanan yumuşatma operatörüyle (Gauss filtre) birlikte kullanılır.

Matlab ile kenar bulma

```
BW = edge(I,'sobel')
BW = edge(I,'sobel',thresh)
BW = edge(I,'sobel',thresh,direction)
[BW,thresh] = edge(I,'sobel',...)

BW = edge(I,'prewitt')
BW = edge(I,'prewitt',thresh)
BW = edge(I,'prewitt',thresh,direction)
[BW,thresh] = edge(I,'prewitt',...)

BW = edge(I,'roberts')
BW = edge(I,'roberts',thresh)
[BW,thresh] = edge(I,'roberts',...)

BW = edge(I,'canny')
BW = edge(I,'canny',thresh)
BW = edge(I,'canny',thresh,sigma)
[BW,threshold] = edge(I,'canny',...)
```

```
BW1 = edge(I,'sobel');
subplot(2,2,1);
imshow(BW1);title('Sobel');

BW2 = edge(I,'prewitt');
subplot(2,2,2);
imshow(BW2); title('Prewit');

BW3 = edge(I,'roberts');
subplot(2,2,3);
imshow(BW3); title('Roberts');

BW4 = edge(I,'canny');
subplot(2,2,4);
imshow(BW4); title('Canny');
```

Sobel

Roberts

Prewit

Canny

```
a=imread('c:\cameraman512.jpg');
imshow(a);
 %sobel kenar bulma algoritması
[m, n]=size(a);
 for i=2:m-1
  for j=2:n-1
      b(i,j)=abs((a(i-1,j-1)+2*a(i-1,j)+a(i-1,j+1))-(a(i+1,j- 1)+...
          2*a(i+1,j)+a(i+1,j+1)))+abs((a(i-1,j-1)+2*a(i,j-1)+...
          a(i+1,j-1))-(a(i-1,j+1)+2*a(i,j+1)+a(i+1,j+1)));
  end
end
figure;imshow(b);
%eşikleme algoritması
for i=2:m-1
  for j=2:n-1
      if(b(i,j)>100)
          b(i,j)=255;
      else
          b(i,j)=0;
      end
  end
end
figure;imshow(b);
```