

Lineer Filtreleme

Kenar Belirleme

Kenar Belirleme

- Kenar, görüntü içerisinde parlaklığın sıçrama yaptığı bölgedir.
- İki farklı alan arasında çerçeve olarak düşünülebilir.
- Kenar belirlemenin temel amacı, görüntü içerisinde istenilen detayları ortaya çıkarmak, istenmeyen detayları ise ortadan kaldırmaktır.
- Görüntüdeki renk geçişlerini keskinleştirmek ve böylece resim içindeki nesneleri ya da farklılıkları elde etmek mümkündür.

Kenar Belirleme

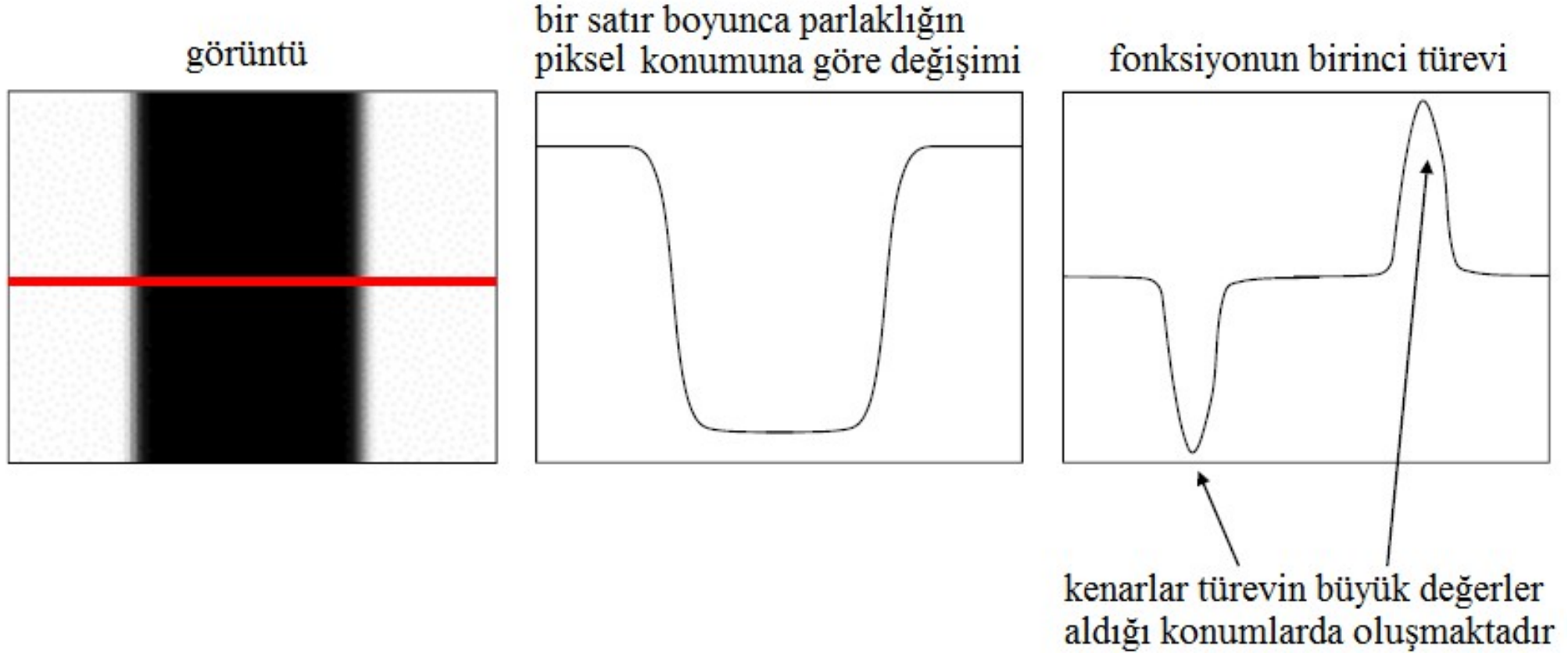
İyi bir kenar kestirimci;

- kenarları iyi bir biçimde sezebilmelidir,
- kenarları doğru konumlarda belirleyebilmelidir,
- Bir kenar için 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 ve kenarların çok ölçekli yapısı, kenar belirleme aşamasında karşılaşılabilecek önemli sorunlardır.

Kenar Belirleme



Türevin büyük değer aldığı pikseller, kenar bölgelerini oluşturur.

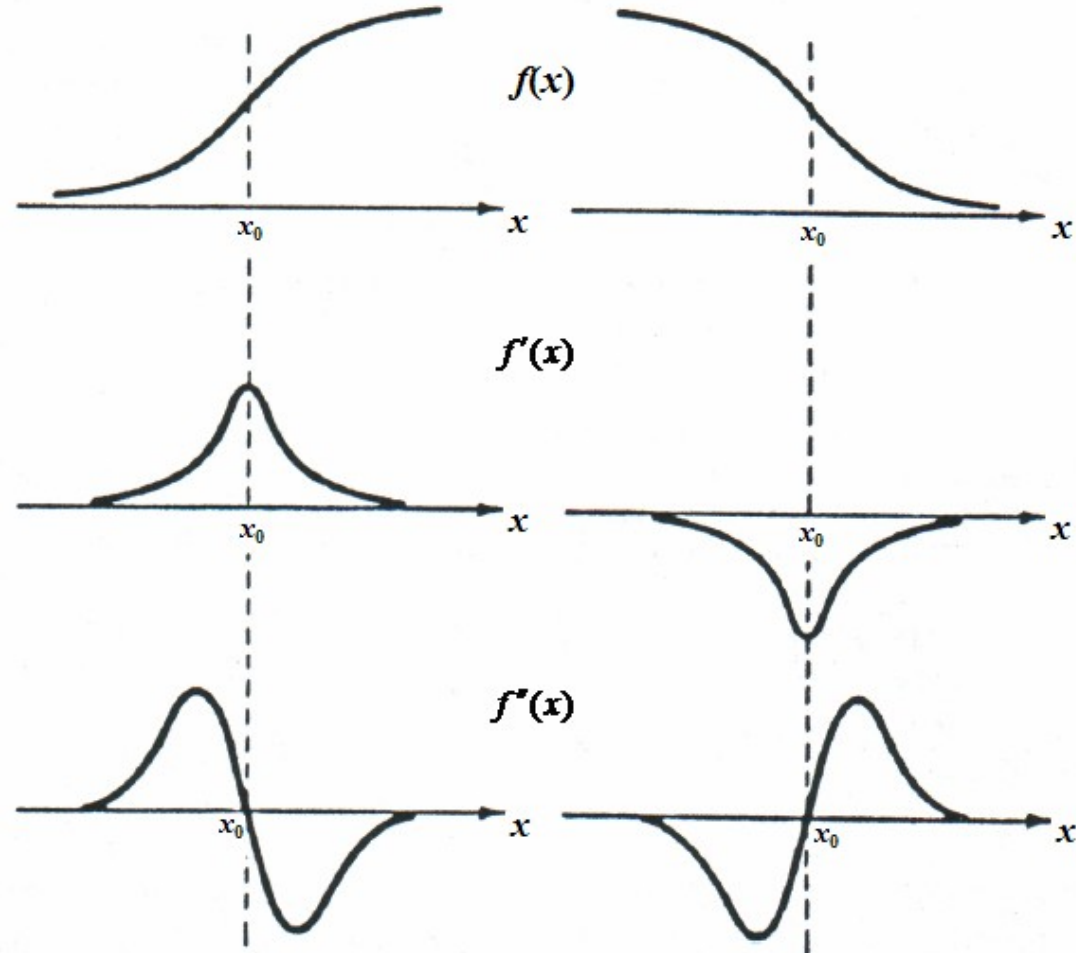
Kenar Belirleme

- Kenar belirleme yöntemleri şu şekilde sınıflandırılabilir:
 - Gradyan tabanlı : Görüntünün birinci türevindeki maksimum ve minimum değerlere bakarak kenarları belirler.
 - Roberts
 - Prewitt
 - Sobel
 - Laplasyan tabanlı : İkinci türevdeki sıfır geçişlerini arar.

Kenar Belirleme



Görüntü



ORJİNAL GÖRÜNTÜ ROBERTS



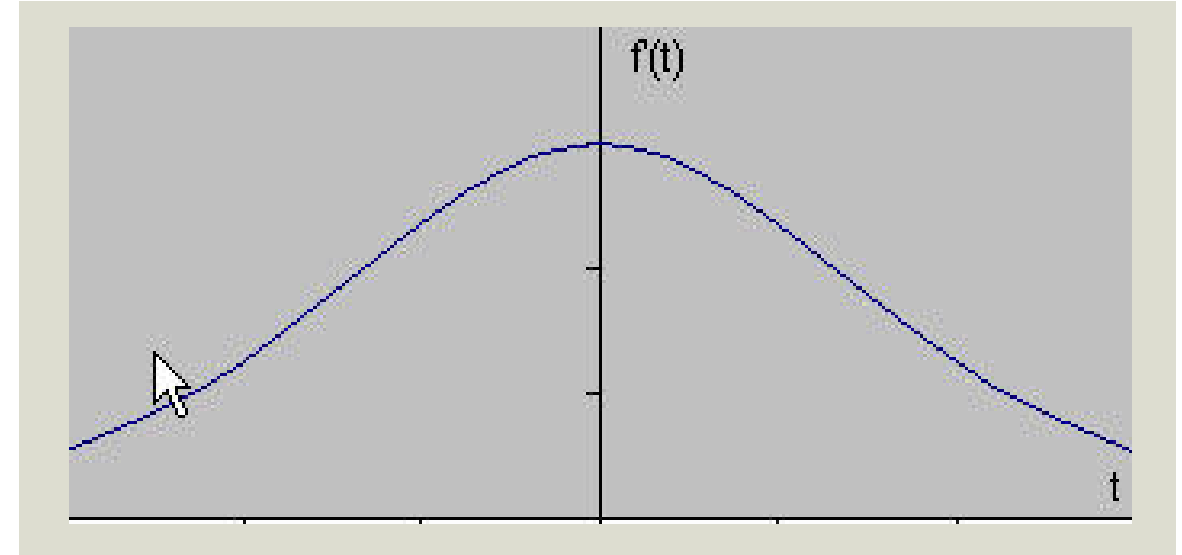
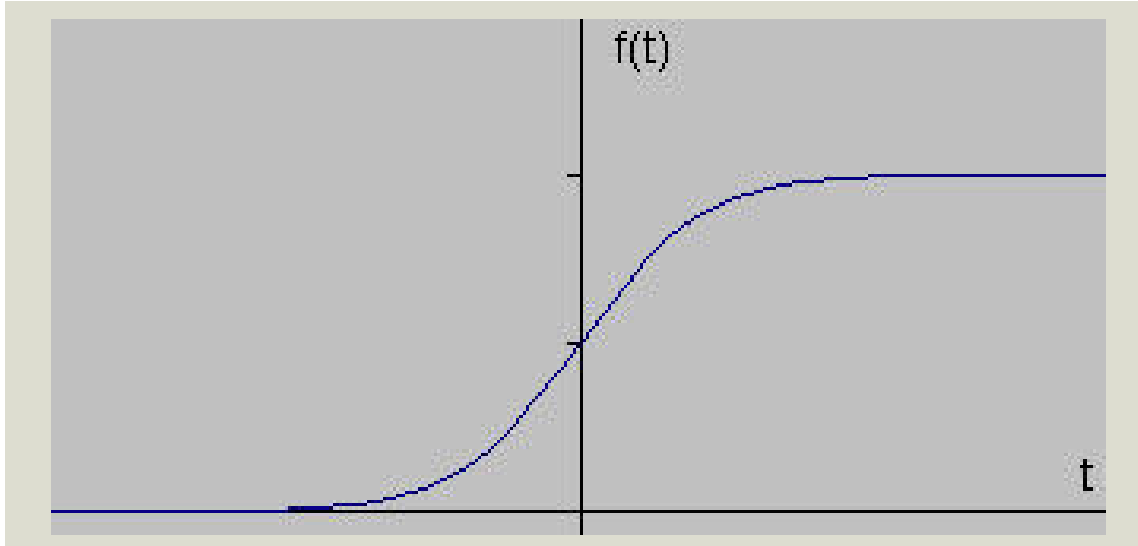
PREWITT LAPLACE



Gradyan Tabanlı Kenar Belirleme

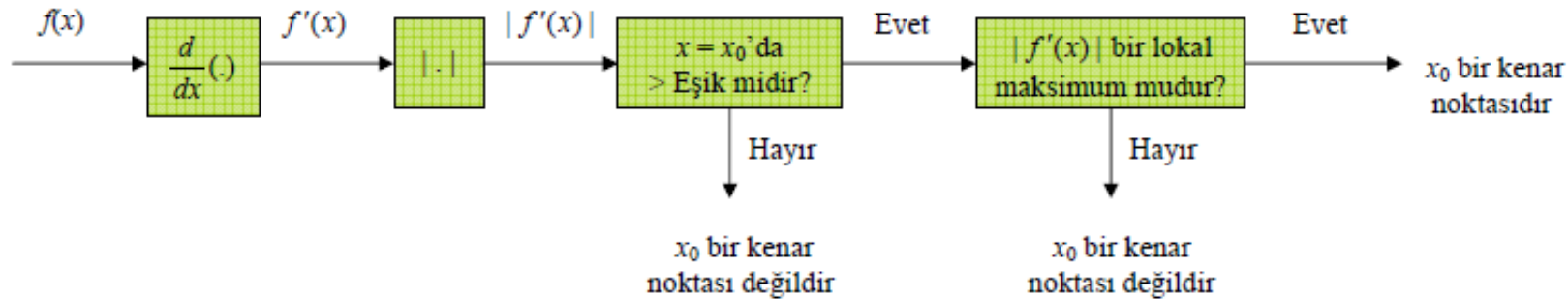
Yandaki sinyali göz önünde bulunduralım. Yoğunluklar arasındaki sıçrama kenarı verir.

Soldaki işaretin türevi (t 'ye göre türevi alınır) sağdaki işareti verir.



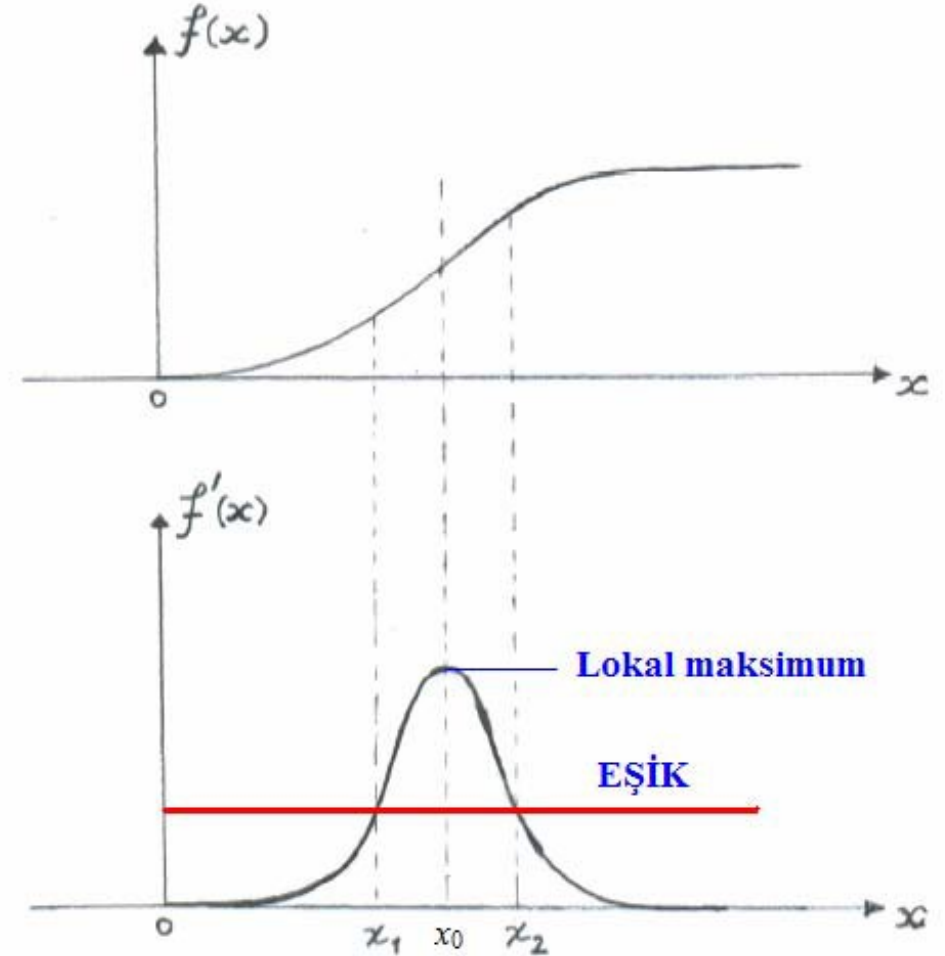
Gradyan Tabanlı Kenar Belirleme

- İlk olarak $f(x)$ 'den $|f'(x)|$ hesaplanır. $|f'(x)|$, belli bir eşik değerinden büyükse bu görüntü pikseli, bir kenar adayı olacaktır. İlgilenilen kenar noktasında eğer bu şart birden fazla x değerleri için sağlanırsa bu durumda bir kenar, noktadan ziyade çizgi olarak görünecektir ve kalın kenarların oluşmasına neden olacaktır. Bu sorunun üstesinden gelmek için en iyi yaklaşım, sadece $|f'(x)|$ değerlerinde lokal maksimuma sahip olan noktaları bulmak olacaktır ve bu noktalar kenar noktaları olarak belirlenir.



Gradyan Tabanlı Kenar Belirleme

- Kenar belirleme işleminde kullanılan eşik değerinin seçimi, uygulamaya bağlı olarak değişiklik arz eder ve kenar görüntüsünün performansı seçilen eşik değeri ile yakından ilgilidir. Eşik değeri, $|f'(x)|$ değerlerinin büyük olanlarını küçüklerinden ayıracak şekilde seçilmelidir. Uygun eşik değeri, $|f'(x)|$ değerlerinin histogramına bakılarak belirlenebilir. Ayrıca uyarlamalı olarak da eşik değerini belirlemek mümkündür. Genellikle eşik değeri, en büyük gradyanın %5 ile %10'u olarak seçilebilir.



Gradyan Tabanlı Kenar Belirleme

- Sayısal bir görüntünün herhangi bir pikseldeki türevi, yatay ve dikey yöndeki türevlerden oluşan bir vektördür. Görüntünün türevine genelde gradyan denir ve $\nabla f(x,y)$ notasyonu ile gösterilir:

$$\nabla f(x, y) = \left[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right]$$

Gradyan Tabanlı Kenar Belirleme

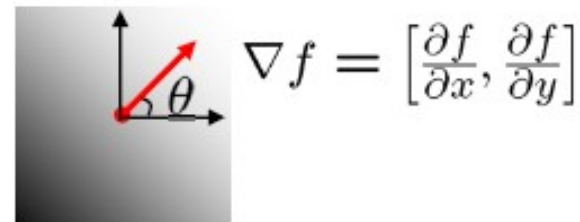
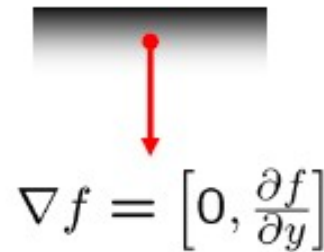
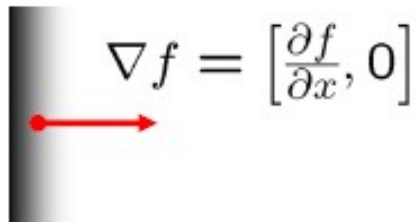
- Gradyan, boyu ve belirli bir referans noktasına göre ölçülen açıyla belirtilebilir. Gradyan açısı kenarın yönü, gradyan genliği kenarın kalınlığı hakkında bilgi verir:

Gradyan genliği: $\|\nabla f(x, y)\| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$

Gradyan açısı: $\theta(x, y) = \tan^{-1}\left(\frac{\partial f(x, y)}{\partial y} / \frac{\partial f(x, y)}{\partial x}\right)$

Gradyan Tabanlı Kenar Belirleme

- Gradyan, parlaklık seviyesindeki değişimin en yüksek olduğu yönü belirtmektedir.
- Bu nedenle, gradyan kenar yönüne diktir.



Gradyan Tabanlı Kenar Belirleme

- Sürekli bir $h(x,y)$ fonksiyonunun x ve y yönlerindeki türevleri şöyle hesaplanır:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \rightarrow 0} \left(\frac{f(x + \varepsilon, y) - f(x, y)}{\varepsilon} \right), \quad \frac{\partial f(x,y)}{\partial y} = \lim_{\varepsilon \rightarrow 0} \left(\frac{f(x, y + \varepsilon) - f(x, y)}{\varepsilon} \right)$$

- Bir yöndeki türev, o yönde fark almaya karşılık geldiğinden bir görüntünün x ve y yönlerindeki türevleri, ilgili yönde fark almayla yaklaşık olarak hesaplanabilir:

$$\frac{\partial f(x,y)}{\partial x} \approx f(x, y) - f(x - 1, y), \quad \frac{\partial f(x,y)}{\partial y} \approx f(x, y) - f(x, y - 1)$$

Gradyan Tabanlı Kenar Belirleme

- Türev almayı, bir sistem gibi düşünebiliriz. Örneğin, x-yönünde türev alma impuls yanıtı $h(x,y)$ olan doğrusal ve ötelemeden bağımsız bir sistemdir:

$$f(x,y) \text{ — } \boxed{h(x,y)} \text{ — } f(x,y) - f(x-1,y)$$

- x-yönünde türev alma, görüntünün $h(x,y)$ ile konvolüsyonunu hesaplamaya eşdeğerdir. Benzer durum, y-yönünde türev alma için de söylenebilir.

Gradyan Tabanlı Kenar Belirleme

- Fark almalar, görüntünün verilen maskeler ile filtrelenmesiyle hesaplanabilir.
- Türevin doğruluğunu arttırmak için çeşitli fark alma yöntemleri önerilmiştir. Prewitt, Sobel ve Roberts, en iyi bilinen türev hesaplama maskeleridir.

Prewitt:	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-1	0	1	-1	0	1	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	1	1	1	0	0	0	-1	-1	-1
-1	0	1																		
-1	0	1																		
-1	0	1																		
1	1	1																		
0	0	0																		
-1	-1	-1																		
Sobel:	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1	<table><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-2</td><td>-1</td></tr></table>	1	2	1	0	0	0	-1	-2	-1
-1	0	1																		
-2	0	2																		
-1	0	1																		
1	2	1																		
0	0	0																		
-1	-2	-1																		
Roberts:	<table><tr><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td></tr></table>	0	1	-1	0	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>-1</td></tr></table>	1	0	0	-1										
0	1																			
-1	0																			
1	0																			
0	-1																			
	x-yönünde maske	y-yönünde maske																		

Gradyan Tabanlı Kenar Belirleme

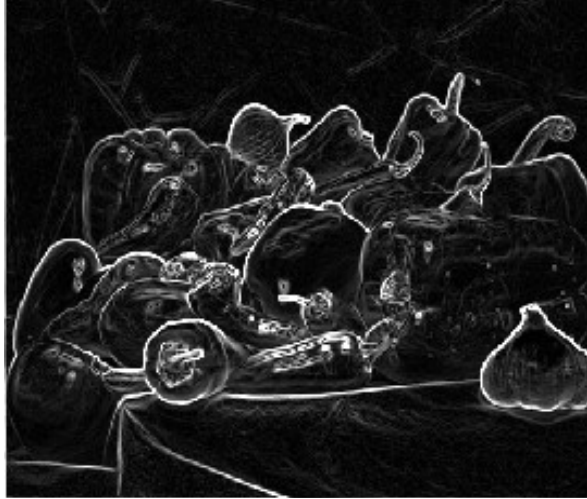
- Her bir maske ile elde edilen gradyan görüntüsünden kenar görüntüsüne ulaşmak için kullanılan eşik değerleri, her bir gradyan görüntüsünün en büyük genlik değerinin %10'u alınarak belirlenmiştir. Bu eşik değerinden büyük gri seviyeler 255, küçük olanlar ise 0 alınarak kenar görüntülerine ulaşılmıştır.

Gri-ton görüntü

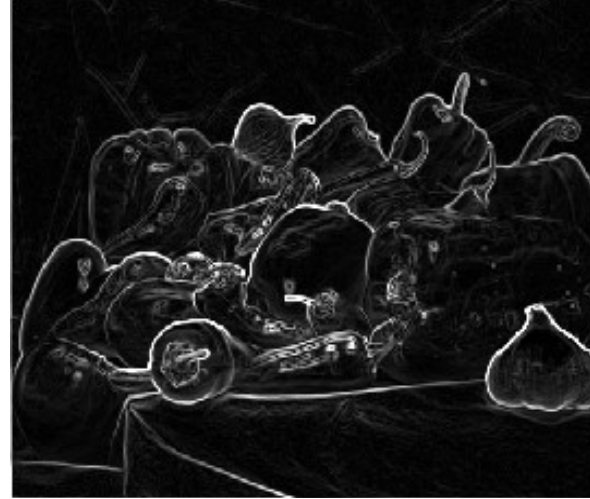


Gradyan Tabanlı Kenar Belirleme

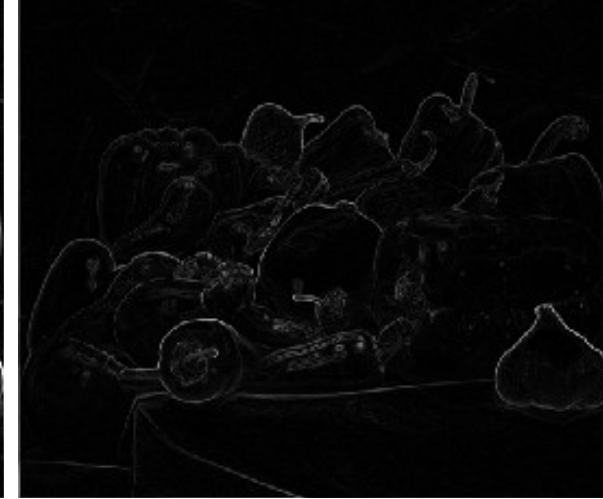
Sobel maskesi ile edilen görüntü



Prewitt maskesi ile edilen görüntü



Robert maskesi ile edilen görüntü



eşik = 77



eşik = 55



eşik = 18



Sobel Kenar Belirleme

- 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 Belirleme

- 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 Belirleme

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

G_x

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

G_y

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Sobel Kenar Belirleme

- 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 Belirleme

- 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 Belirleme

a_{11}	a_{12}	a_{13}	...	a_{1n}
a_{21}	a_{22}	a_{23}	...	a_{2n}
a_{31}	a_{32}	a_{33}	...	a_{3n}
\vdots	\vdots	\vdots		\vdots

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}
\vdots	\vdots	\vdots		\vdots

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

Sobel Kenar Belirleme

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

- Gx maskesi yatay yöndeki kenarları, Gy maskesi dikey yöndeki kenarları vurgular.
- Bunların toplanması ile her iki yöndeki kenarlar tespit edilir.

Sobel Algoritması

```
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');
```

Robert Kenar Belirleme

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

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{veya} \quad \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

- Bu maskeler, düzlemsel bir yüzeye oturtulmuş 2x2 bir pencerenin en küçük kare hatasının eğimini verecek şekilde türetilmişlerdir.
- Hızlı ve basit olduğundan gerçek zamanlı uygulamalarda kullanılmaktadır.

Prewitt Kenar Belirleme

- Prewitt, Sobele göre hesaplama uygulamak için biraz daha basit, ama biraz gürültülü sonuçlar üretme eğilimindedir.
- Prewitt maskeleri aşağıdaki gibidir;

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

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

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

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

Prewitt Kenar Belirleme

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



Prewitt Kenar Belirleme

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

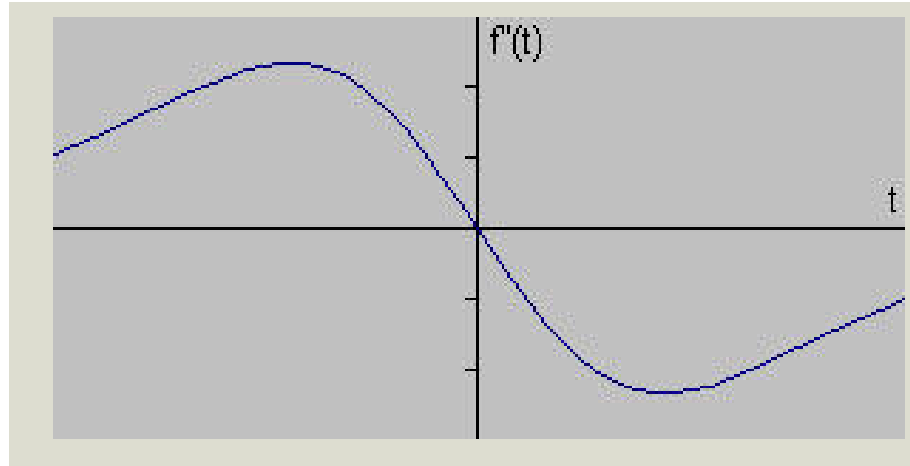


```
>> s=edge(g,'prewitt',0.05, 'vertical');  
>> imshow(s)
```



Laplasyan Tabanlı Kenar Belirleme

- İlk 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.



Laplasyan Tabanlı Kenar Belirleme

- Laplasyan filtreler görüntüdeki keskin geçişleri bulmak için kullanılan ikinci türev tabanlı filtrelerdir. İki boyutlu bir $f(x,y)$ fonksiyonunun ikinci derece türevi veya laplasyanı aşağıdaki gibidir:

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

Laplasyan Tabanlı Kenar Belirleme

- Bir görüntünün Laplasyanı bir konvolüsyon filtresi kullanılarak da bulunabilir. Bu değere yaklaşık olarak da olsa ulaşmak için kullanılan konvolüsyon maskelerinden bazıları şöyledir:

0	-1	0
-1	4	-1
0	-1	0

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

Marr-Hildreth Kenar Belirleme

- Marr-Hildreth 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.

```
>>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)
```



Canny Kenar Belirleme Yöntemi

Canny algoritması, en sık kullanılan kenar bulma yöntemidir:

- Görüntü, Gaussian filtresiyle yumuşatılır.
- Gradyan büyüklüğü ve yönü belirlenir. Yandaki maskeler ile
- Birden fazla piksel kalınlıktaki kenarlar, inceltme ile bir piksel kalınlığa düşürülür (yerel minimum ve maksimumlar yok edilir).
- Biri büyük, diğeri küçük iki eşik değer tanımlanır. Yüksek eşik değerinden yararlanılarak kalın kenar eğrileri belirlenir; düşük eşik değerinden istifade edilerek eğriler devam ettirilir.

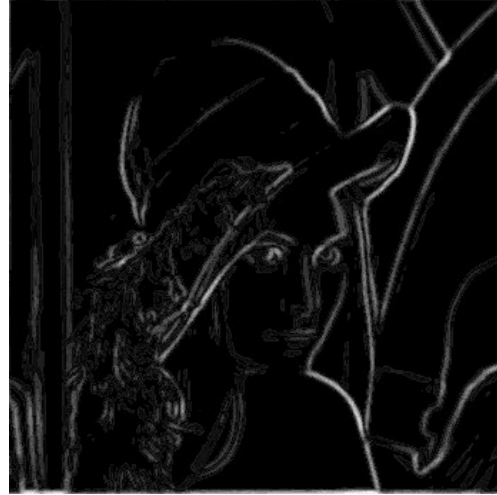
$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

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

Canny Kenar Belirleme Yöntemi



Gradyan genliđi



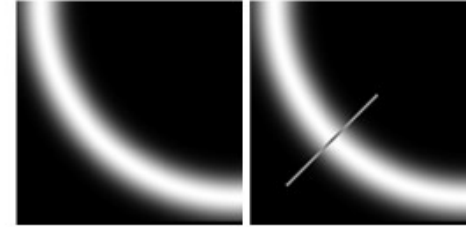
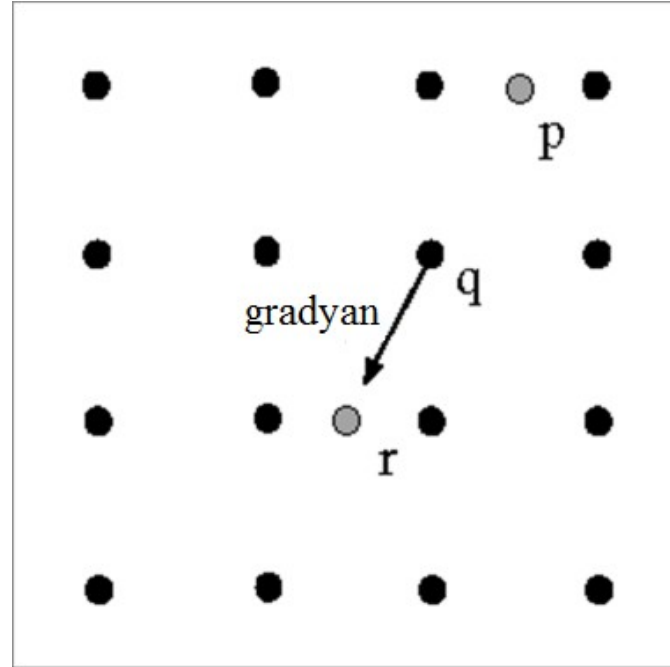
eřikleme



inceltme

Canny Kenar Belirleme Yöntemi

- İnceltme, piksel ile komşularının parlaklık seviyeleri karşılaştırılarak yapılır. q bir kenarsa gradyan yönündeki iki komşusu p ve r 'den daha büyük değer almalıdır.



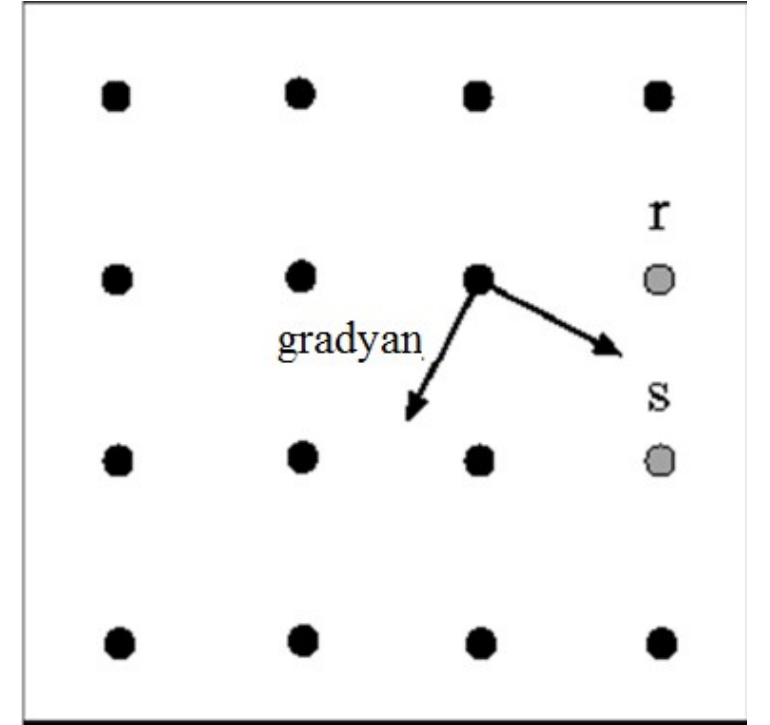
Canny Kenar Belirleme Yöntemi

- Görüntüde ince ve kalın kenarlar birlikte bulunabilir.
- Eşik değeri büyük seçilirse kalın kenarlar; düşük seçilirse ince kenarlar ve gürültü tespit edilir.
- İki tür kenarı da kaçırmamak için eşik değeri ilk önce büyük seçilerek kalın kenarlar belirlenir. Daha sonra eşik değeri küçültülerek bulunan kalın kenarlar devam ettirilir.



Canny Kenar Belirleme Yöntemi

- Kenar bağlama, piksel ile komşularının parlaklık seviyeleri karşılaştırılarak yapılır. Kenar yönündeki piksellerin, gradyan genliği ve açısı yakın değerler almalıdır.
- Bir kenar pikseli için gradyan yönüne dik yöndeki en yakın komşu (s veya r) bulunur. Piksel ile komşusunun gradyan genlikleri ve açıları yakın değerler alıyorsa, komşusunun da bir kenar pikseli olduğuna karar verilir.



Canny Kenar Belirleme Yöntemi



yüksek eşik
(kuvvetli kenarlar)



düşük eşik
(zayıf kenarlar)



histerisis (çift) eşikleme

MATLAB ile Kenar Belirleme

```
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



Prewitt

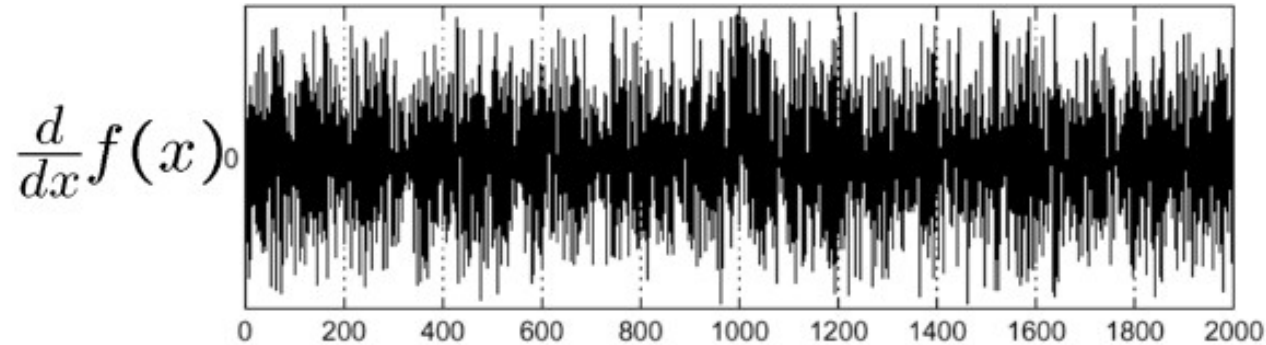
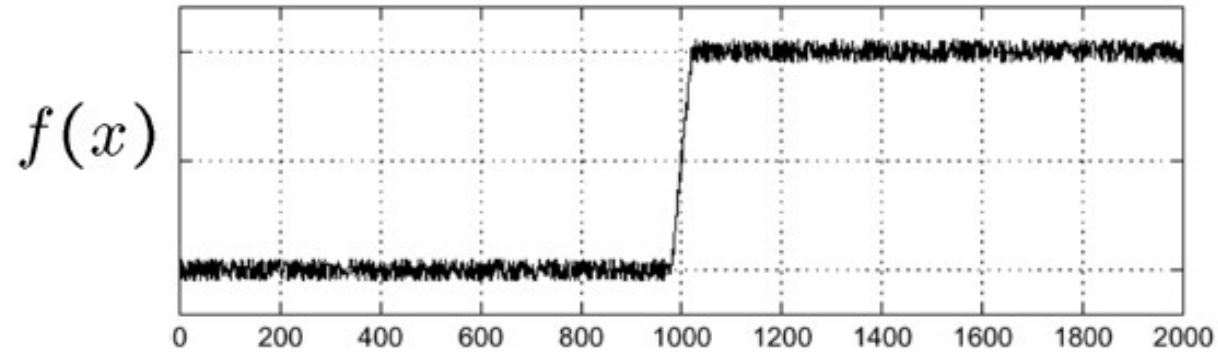


Canny



Gürültü

- Gürültülü bir görüntünün bir satırının parlaklık seviyesinin türevinden kenar piksellerinin konumlarının belirlenemeyeceği görülmektedir.



Gürültü

- **Tuz ve biber gürültüsü:** Rastgele siyah ve beyaz piksellerin oluşması
- **İmpuls gürültüsü:** Rastgele beyaz piksellerin oluşması
- **Gauss gürültüsü:** Parlaklık seviyelerinde Gauss dağılımına uyan değişimlerin oluşması



Orijinal



Tuz ve biber gürültüsü



İmpuls gürültüsü



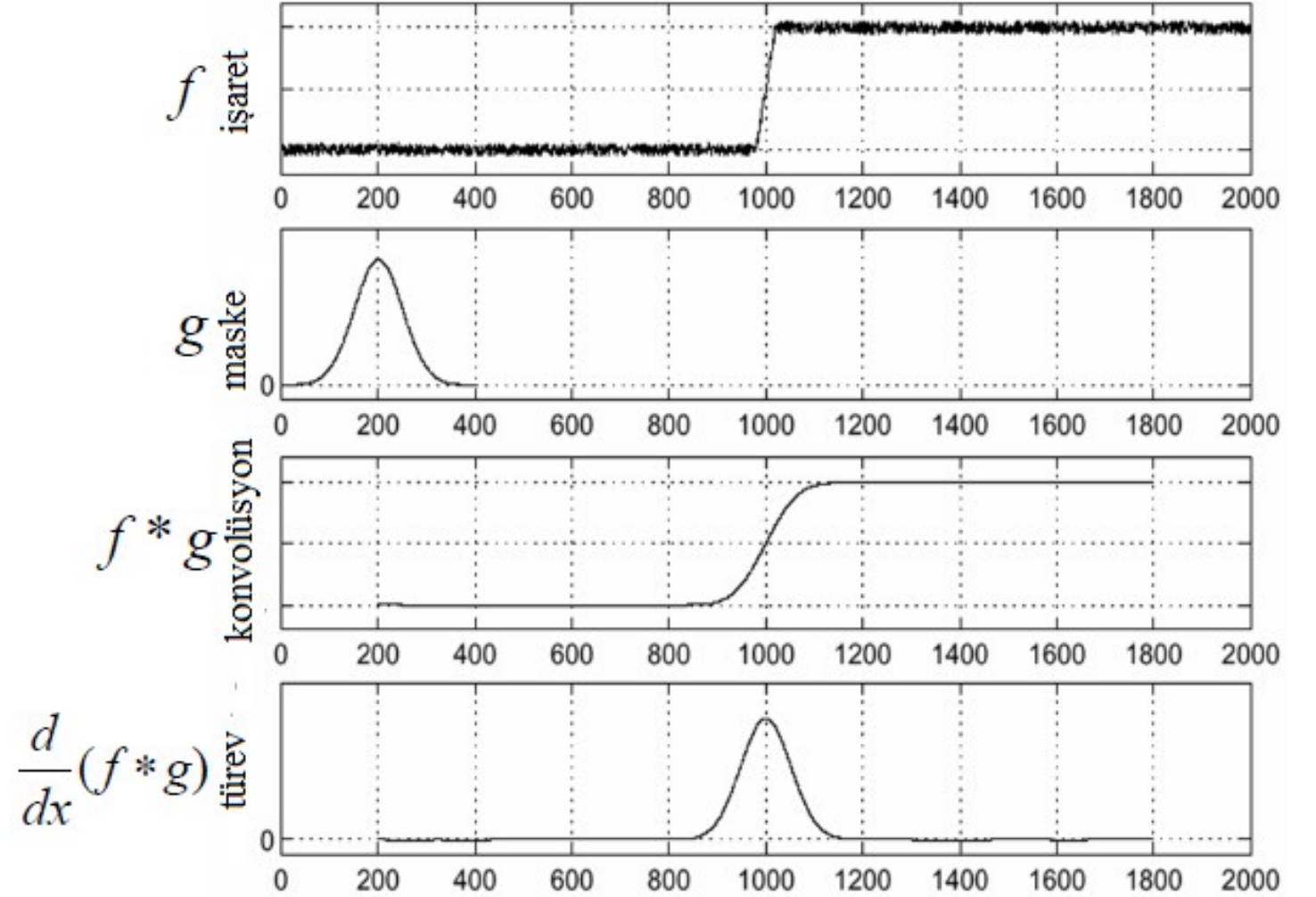
Gauss gürültüsü

Gürültü

- Gürültü nedeniyle, bir pikselin parlaklık seviyesi komşularınıninkine göre oldukça farklı değerler alabilir.
- Türev alan filtreler, parlaklık seviyesinde değişme olan yerleri belirlemek amacıyla tasarlanmıştır.
- Gürültü değişime neden olduğundan, fark alma filtreleri gürültüye duyarlı olacaktır. Gürültü ne kadar büyük olursa duyarlılık da o derecede fazla olacaktır.
- Gürültülü durumda, türev hesaplanmadan önce görüntüyü alçak geçiren (yumuşatan) bir filtreden geçirmek faydalı olabilir.
- Alçak geçiren filtreleme, bir pikselin değerini komşularınıninkine yakın olmaya zorladığından gürültüyü yok edecektir.

Gürültü

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



Gürültü

- 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.
- $f * g$ 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.

Frekans Kavramı

- Bir görüntüdeki filtrenin etkisini standart bir formda ifade edebilmek için en önemli kriterlerden birisi görüntünün frekans bileşenleridir.
- Kabaca bir görüntünün frekans bileşenleri; mesafeye göre gri seviye değişiminin miktarı olarak ifade edilir.

Frekans Kavramı

- Görüntünün Yüksek frekanslı bileşenleri; küçük mesafelerde piksellerin gri değerlerin büyük miktarda değişikliklerini karakterize eder. Yüksek frekans bileşenlerine örnek olarak; resmin kenarları (en büyük gri seviye değişimleri kenarlarda olur) ve gürültüler verilebilir.
- Görüntünün Düşük frekanslı bileşenleri: Resimdeki piksellerin gri değerlerinin mesafeye göre pek az değiştiği görüntü parçaları ile karakterize edilir. Bunlara örnek arka planlar (gri seviyeleri çok az değişen yüzeyler), cilt dokuları verilebilir.

Frekans Kavramı

- Bu tanımlamalar göre filtreleri;
- Yüksek geçiren Filtreler: Yüksek frekanslı bileşenleri geçirir. Düşük frekanslı bileşenleri yok eder. Örnek kenar çıkarma işlemleri.
- Alçak geçiren Filtreler: Görüntüdeki Alçak frekans bileşenlerini geçirir veya kuvvetlendirir. Yüksek frekans bileşenlerin yok eder. Örneğin 3x3Lük average filtresi bir alçak geçiren filtredir. Çünkü resmin kısa mesafede değişen (Özellikle kenarlar - Gürültüler) piksellerini yok eder. Belirli mesafede az değişen piksellerini muhafaza eder. Dolayısıyla resim bulanıklaşır (Blurring).

Lineer Filtreler

- Alçak geçiren (Low-pass) Filtreler:
 - Gürültüyü yok eder
 - Görüntüyü yumuşatır
 - Kenarları bulanıklaştırır
- Yüksek Geçiren (High-pass) Filtreler:
 - Görüntü içerisindeki detayları, kenarları ve gürültüyü ön plana çıkarır

Gaussian Filtre

- Alçak geçiren bir filtredir.
- Aynı zamanda fourier dönüşümüdür.

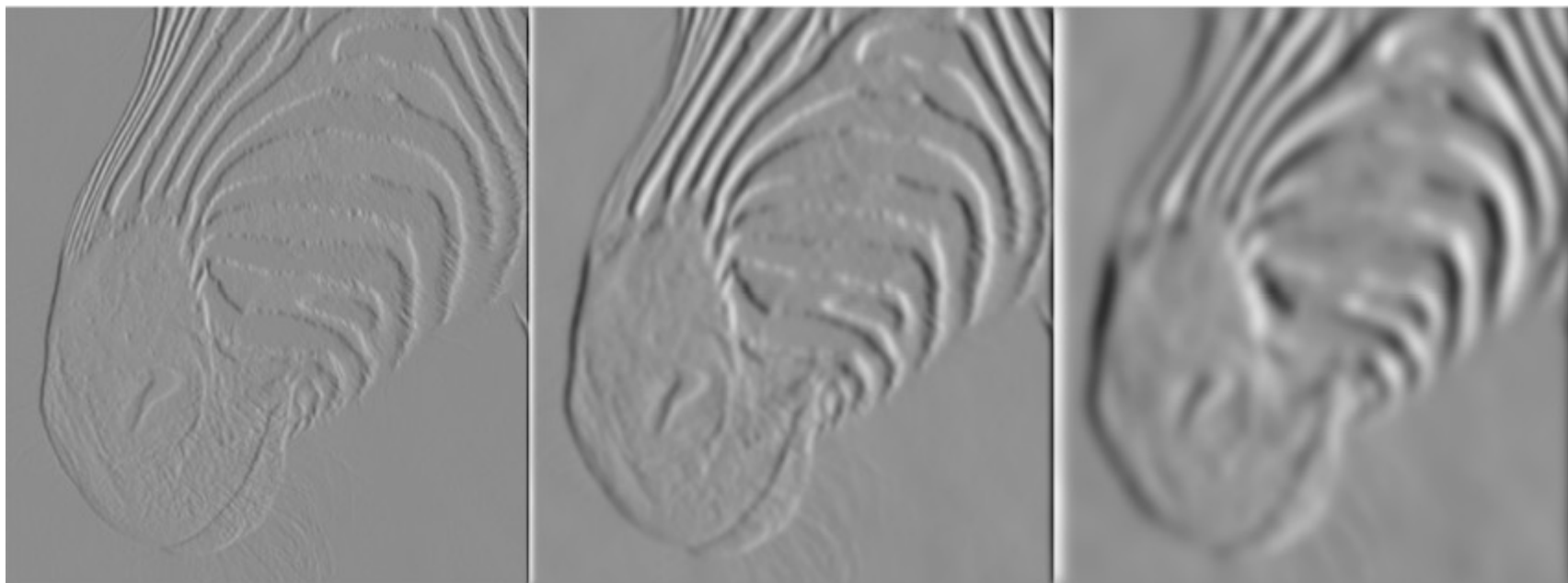
$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-(u^2 + v^2)/(2\sigma^2)}$$

- Gaussian filtresi için kullanılacak matrisin oluşturulmasında gaussian dağılımındaki (normal dağılım) gibi bir yol izlendiği söylenebilir.

Gaussian Filtre

- Maske için komşuluk belirlenir. Eğer 1 seçilirse 3x3 bir matris elde edilir.
- Maske uygulaması, sert ton değişimlerini azaltmakta ve görüntünün daha yumuşak olmasını sağlamaktadır.
- Maske boyutu arttıkça bulanıklaşma artacak ve kenarlar kalınlaşacaktır.

Gaussian Filtre



1 piksel

3 piksel

7 piksel

Laplasyan Filtre

- 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.
- Buradan hareketle bir boyutlu $f(x)$ fonksiyonun birinci türevi

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

denklemlerle verilir.

Laplasyan Filtre

- İkinci türev ise

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

olacaktır.

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

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

denklemlerle verilir.

Laplasyan Filtre

- 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 Filtre

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

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

LoG (Laplasyan of Gaussian)

- Laplasyen maskeleri görüntünün ikinci türev değerine yaklaştığı için gürültüye oldukça duyarlıdırlar. Gürültüyü azaltmak için genellikle görüntü laplasyen filtresinden önce Gaussian filtresi gibi bir filtreden geçirilir. Böylece yüksek frekanslı gürültü bileşenleri azaltılarak görüntü yumuşatılır. İki aşamalı bu filtreleme işlemine LoG (Laplacien of Gaussian) denir.
- Sıfır merkezli olmak üzere iki boyutlu (2-D) bir LoG fonksiyonu aşağıdaki gibi tanımlanır:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

LoG (Laplasyan of Gaussian)

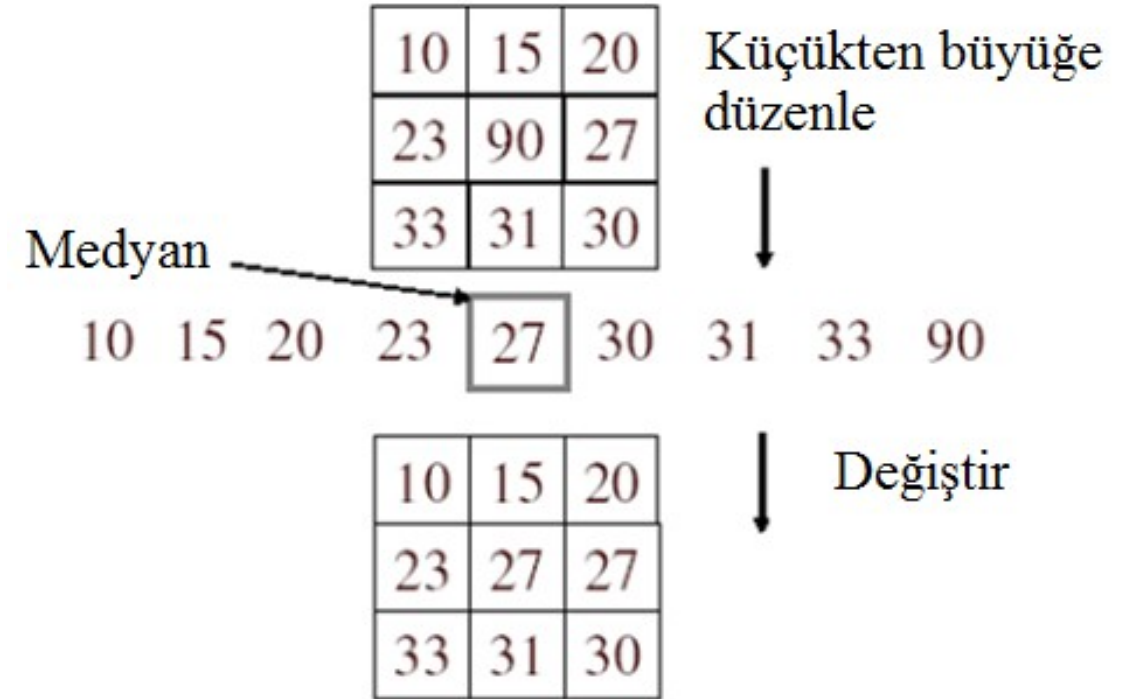
- σ , Gaussian standart sapması'dır. Aynı şekilde maske kullanılarak da bu işlem gerçekleştirilebilir.

0	-1	0
-1	5	-1
0	-1	0

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

Medyan Filtre

- Gauss filtresi gürültüyü giderirken görüntüyü bulanıklaştırmaktadır. Görüntüyü bulanıklaştırmadan gürültü giderilmek isteniyorsa Medyan filtre kullanılabilir.
- Bir görüntüyü medyan filtrelemek için bir pikselin değeri, komşuluğundaki piksel değerlerinin medyanı ile değiştirilir.



Medyan Filtre



Bozulmuş görüntü



3x3 medyan filtresi uygulanmış

Medyan Filtre



7x7 medyan filtresi uygulanmış



Üç kere 3x3 medyan filtresi uygulanmış

Görüntünün okunması ve gösterilmesi

```
I = imread('eight.tif');  
imshow(I)
```



Gürültü eklenmesi

```
J = imnoise(I,'salt & pepper',0.02);  
Figure(),  
imshow(J);
```



Gürültülü görüntünün filtrelenmesi

```
K = filter2(fspecial('average',3),J)/255;  
Figure(),  
imshow(K)
```

Gürültülü görüntüyü filtrelemek için medyan filtresi kullanılması

Medfilt2 fonksiyonunun gürültüyü kaldırmakta daha iyi sonuç verdiği ve kenarları daha az bulanıklaştırdığı görülmektedir.

```
L = medfilt2(J,[3 3]);
```

```
Figure(), imshow(K)
```

```
Figure(), imshow(L)
```



Bulanıklaştırma

Bulanıklaştırma, geçerli piksel ve bu pikselin 4 komşusunun ortalamasının alınması ile yapılır. Geçerli piksel ve 4 komşusunun toplamı alınarak 5'e bölünür veya filtrede beş kere 0.2 ile çarpılır:

```
double filter[filterWidth][filterHeight] =  
    { 0.0, 0.2, 0.0,  
      0.2, 0.2, 0.2,  
      0.0, 0.2, 0.0 };
```

```
veya, double factor = 1.0/5;  
double bias = 0.0;
```

Böyle küçük filtre matrisleri ile çok yumuşak bir bulanıklaştırma sağlanır:

Bulanıklaştırma



Orijinal görüntü



Bulanıklaştırılmış görüntü

Keskinleştirme

Görüntüyü keskinleştirmek kenar bulmaya çok benzer. Orijinal görüntü ve kenarları bulunmuş görüntü toplanır ve sonuçta kenarları kuvvetlendirilmiş yeni bir görüntü oluşur. Bu görüntüyü eklemek bir önceki örnekteki kenar bulma filtresinin merkez değeri bir arttırılarak yapılır. Filtre elemanlarının toplamı 1'dir ve sonuç orijinal görüntü ile aynı parlaklıkta fakat daha keskin olacaktır.

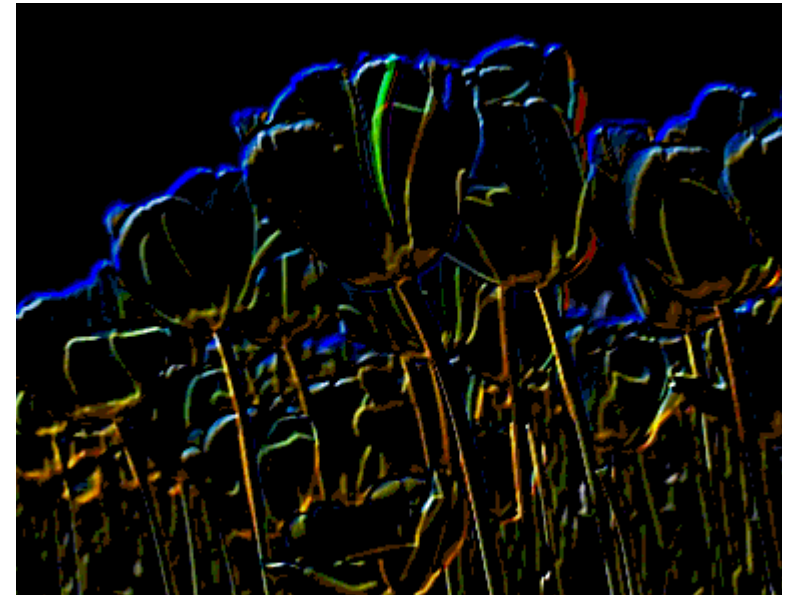
```
A=imread('original_image.bmp');  
A=rgb2gray(A);  
H=[-1 -2 -1;-2 13 -2;-1 -2 -1];  
C=conv2(A,H);  
C=uint8(C);  
imshow(C)
```



Kabartma

Bir kabartma filtresi bir görüntüye 3-boyutlu gölge efekti verir. Sonuç, görüntünün tümseklik haritası elde edilir. Bu, merkezin bir tarafındaki piksellerin alınarak diğer taraftakilerden çıkarılması ile yapılabilir. Pikseller pozitif veya negatif sonuç değerleri alabilir. Negatif pikselleri gölge ve pozitif olanları aydınlık yerleri temsil eder. Bu durumda görüntünün çoğu gri renge ve kenarlar koyu gri/siyah veya açık gri/beyaz'a dönüşecektir.

```
A=imread('original_image.bmp');  
s=[-1 -1 0;-1 0 1;0 1 1];  
f=imfilter(A,s,'conv');  
imshow(f)
```



Kabartma Uygulama

```
h = fspecial('sobel');  
I2 = filter2(h,I);  
imshow(mat2gray(I2));
```



Örnek Filtrelemeler

```
I = imread('C:\lena512.jpg');  
subplot(2,2,1);  
imshow(I); title('Original Image');
```

```
H = fspecial('motion',20,45);  
MotionBlur =  
imfilter(I,H,'replicate');  
subplot(2,2,2);  
imshow(MotionBlur);title('Motion  
Blurred Image');
```

```
H = fspecial('disk',10);  
blurred = imfilter(I,H,'replicate');  
subplot(2,2,3);  
imshow(blurred); title('Blurred Image');
```

```
H = fspecial('unsharp');  
sharpened = imfilter(I,H,'replicate');  
subplot(2,2,4);  
imshow(sharpened); title('Sharpened  
Image');
```

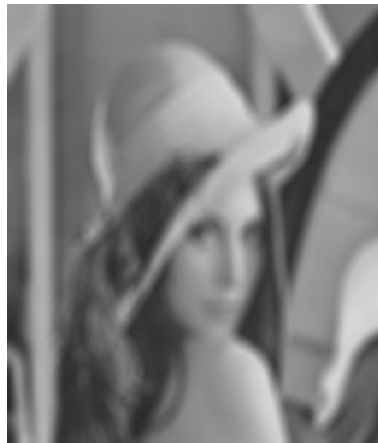
Original Image



Motion Blurred Image



Blurred Image



Sharpened Image



Ödev

- Prewit ve Robert kenar belirleme gerçekleştiren m-file dosyasını yazınız?
- Teslim tarihi : 18 Nisan 2019

Ödev

- lena512.png görüntüsü için,
 - Histogramı çizilecek
 - Histogram germe uygulanacak
 - Salt and pepper gürültüsü eklenecek
 - Median filtresi uygulanacak
-
- Teslim Tarihi : 18 Nisan 2019