

# Görüntü İşleme Teknikleri - 1

Oğuzhan Öztaş

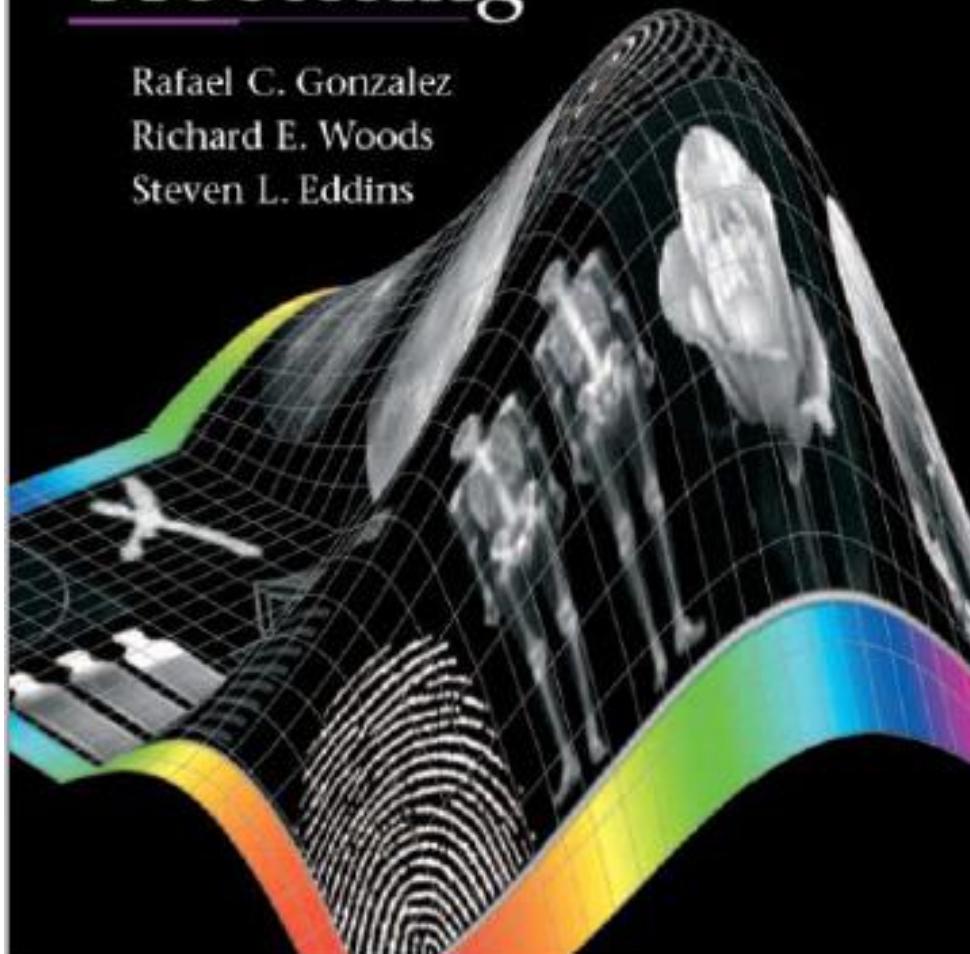
# Digital Image Processing

USING MATLAB®

Rafael C. Gonzalez

Richard E. Woods

Steven L. Eddins



# İçerik

- Görüntü
- İmge
- İmge işleme ve uygulama alanları
- İmgenin oluşturulması
- Video kavramı
- Sayısal imge kavramı
- Sayısala dönüştürme
- Sayısal imgede pixel kavramı

# İçerik

- Çözünürlük
- İmge dosyaları
- İmge ters çevirme
- İmge aynalama
- Matlab ile imge döndürme işlemleri
- İmge öteleme
- Yakınlaştırma
- Uzaklaştırma
- Matlab ile imge boyut değiştirme işlemleri

# İçerik

- İmge pekiştirme
- Parlaklık ayarı
- Kontrast ayarı
- Eşikleme, olumsuzlama
- Histogram
- Histogram eşitleme
- Karşılık yayma
- Konvolüsyon

# İçerik

- Alçak geçiren sözgeç
- Yüksek geçiren sözgeç
- Kenar bulma
- Korelasyon
- Ortanca sözgeç

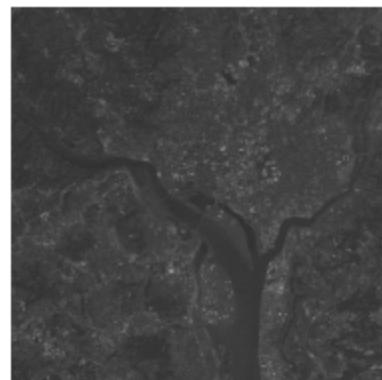
## İmge Kavramı

İmge (image), çeşitli yollarla elde edilen bilgilerin görsel olarak saklanması ve gösterimine olanak sağlayan resimlerdir. Her türlü iki boyutlu bilgi imge olarak ele alınabilir.



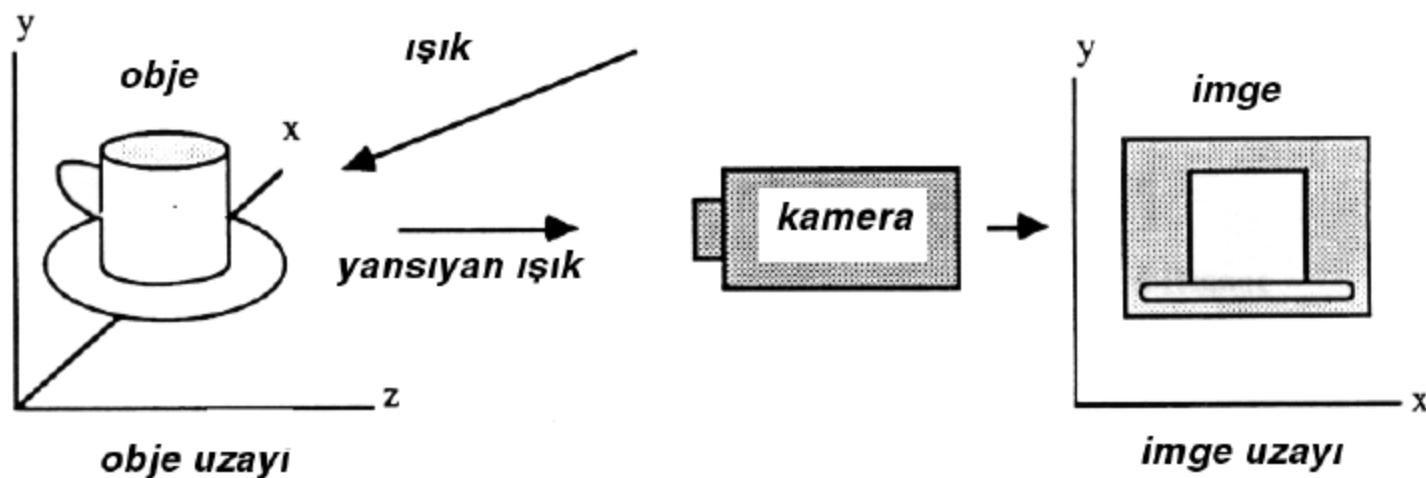
# İmge İşleme ve Uygulama Alanları

- Tıp ve biyoloji (x-ışınları, biyomedikal görüntüler...)
- Coğrafî bilimler (jeodezi, hava ve uydu görüntülerinden hava tahmini)
- Restorasyon (eski, hasar görmüş fotoğrafların onarılması,
- Fizik (elektron miroskobu görüntüler, spektrometreler)
- Uzay bilimleri (uydu, mikrodalga radar görüntüleri...)
- Savunma sanayi (gece görüş, akıllı roket sistemleri...)
- Endüstriyel uygulamalar (sureç, ürün denetimi...)
- Tüketicî elektroniği (VCD, DVD, video kayıt cihazları, cep telefonları...)
- Güvenlik sistemleri (iris-parmak izi tanıma, güvenlik-kamera uygulamaları)



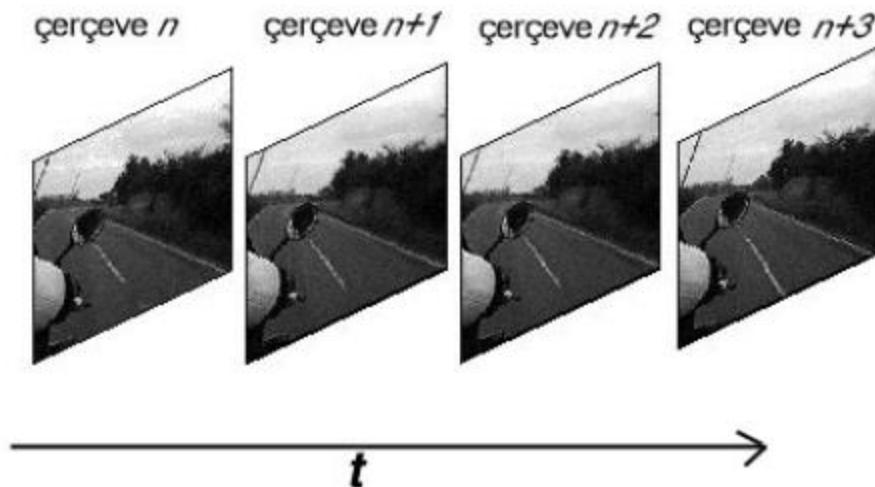
## İmgenin Oluşturulması

İmgeler, 3-Boyutlu gerçek nesne uzayından sadece 2-Boyutun kullanımı ile oluşturulur. Kamera benzeri cihazların ışığa duyarlı 2-B yüzeyine nesneden yansıyan ışık kullanılarak imgé uzayına geçilir.



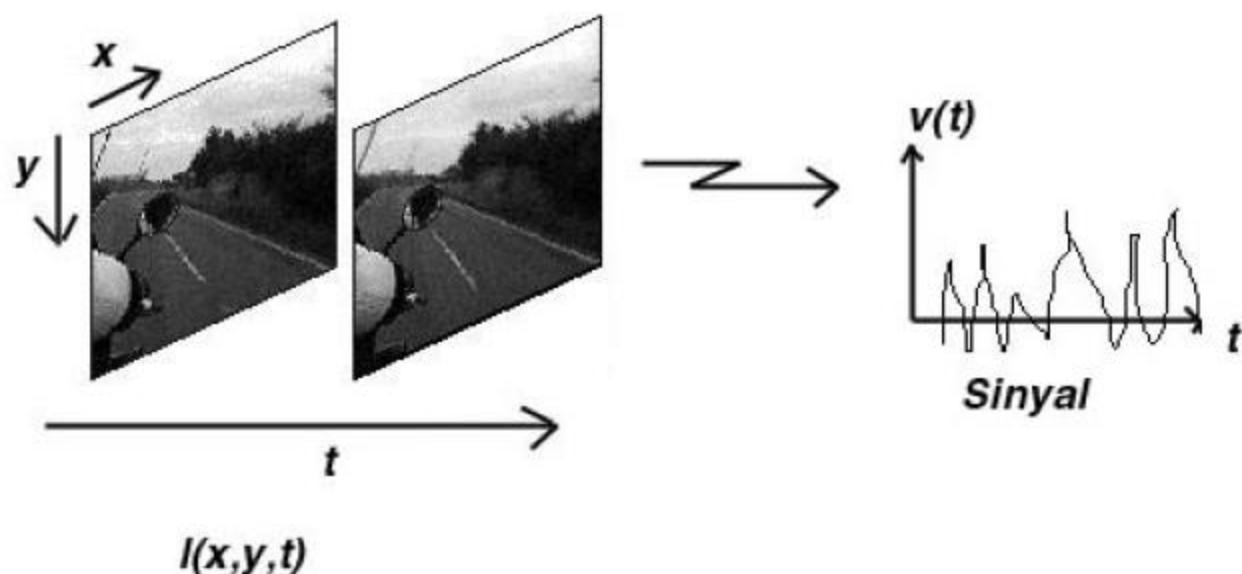
## Video Kavramı

İnsan beyni, belirli şartlar altında, gözün retinasına odaklanmış hareketli bir imge ile çok hızlı değişen bir durağan imge dizisi arasındaki farkı algılayamaz. Psikogörsel araştırmalar saniyede gösterilen durağan en az 50 resim hareketli görüntüden ayırt edilemediğini göstermiştir.



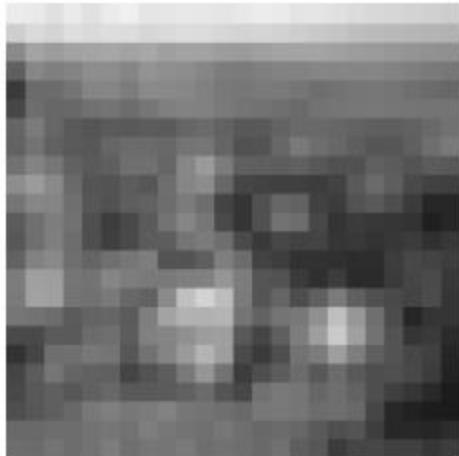
## Televizyon (Video'nun İletimi)

Günümüzde video kayıt, saklama ve iletimi sistemlerinin bir kısmı halen analog biçimde çalışmaktadır (örn; analog TV iletimi). Analog video işaretini tek boyutlu ve zamana bağlı elektriksel işaret ettir. 2-B olan video bilgisinin tek boyutlu zamanla değişen bir işareteye çevrilmesi ile elde edilir. Bu işlem tarama adı verilen yöntemlerle yapılır.



## Analog Videoda Çözünürlük

Her çerçevede taranan satır sayısına bağlıdır.



25 çizgi



250 çizgi

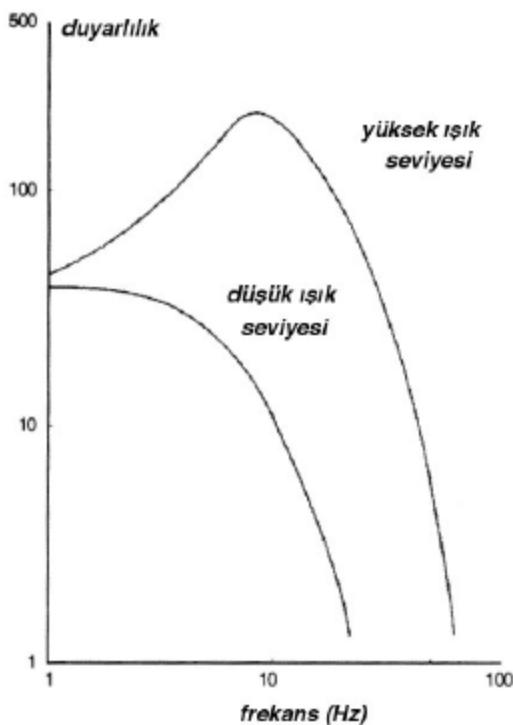


500 çizgi

Avrupa'da 625 satır (çizgi), Amerika'da ise 525 satır (çizgi) kullanılmaktadır.

## Görsel Kırışma Duyarlılığı

Önceki slaytlarda insan gözünün hareketsiz resimleri hareketli olarak algılaması için saniyede en az 50 resim gösterilmesi gerektiğinden bahsedilmişti.



TV'de ekran tazeleme hızı:

Avrupa'da 50Hz,  
Amerika'da 60Hz dir.

## Analog Video - Standartlar

Günümüzde kullanılan analog video standartları 3 farklı biçimde gruplandırılabilir:

Component Analog Video (CAV),  
Composite Video (CV),  
S-Video (Y/C Video)

**CAV:** ışıklılık:  $Y=0.30R+0.59G+0.11B$   
renk:  $I=0.60R+0.28G-0.32B$  yada  $Cr=R-Y$   
 $Q=0.21R-0.52G+0.31B$   $Cb=B-Y$

Bu sistemin sağladığı renk kalitesi iyidir fakat üç renk bileşeni için kusursuz senkronizasyon gerektirmesi ve kullandığı band genişliğinin fazlalığı dezavantajlarıdır.

## Analog Video - Standartlar (Devamı)

**CV:** Renk bileşenleri ayrı ayrı değil, ıshıklılık işaretinin üzerine bindirilir. Renk bileşenleri, ıshıklılık bileşenleri ile aynı spektrumu paylaşır.

- NTSC (National Television Systems Committee)  
Çerçeve başına 525 çizgi (485'i görüntü için kullanıyor)  
Saniyede 30 çerçeve  
I-Q formatında renk işaretİ  
6MHz bant genişliği, 4:3 En boy oranı
- PAL (Phase Alternation Line)  
Çerçeve başına 625 çizgi (576)  
Saniyede 25 çerçeve  
Cr-Cb formatında renk işaretİ, 5.5MHz bant genişliği, 4:3 En boy oranı

**S-Video:** Sayısal kameralar gibi yüksek görüntü kalitesi gerektiren sistemlerde kullanılmaktadır.

## Televizyon Bant Genişliği

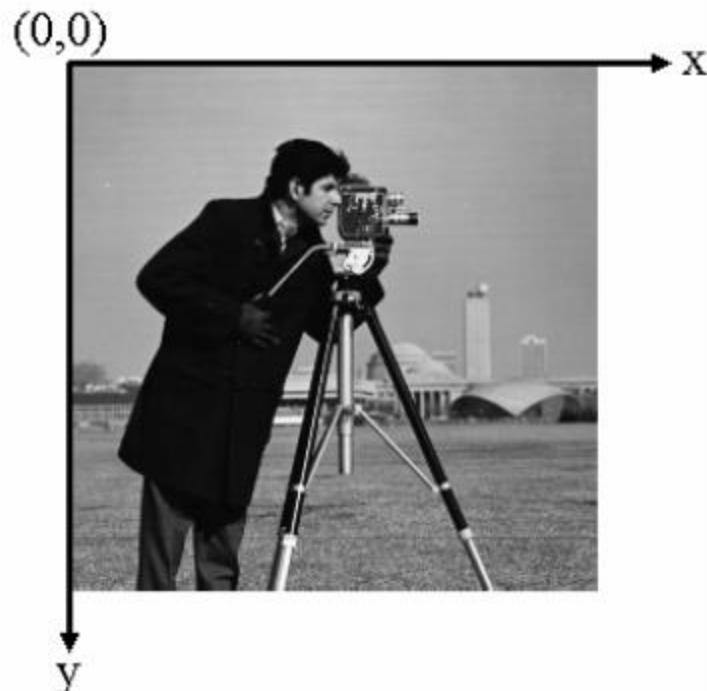
625 satır tarama sisteminde bir satırın  $1/(625*25)=64\mu s$ 'de taranaması gerekmektedir. Ancak geri dönüş (fly back) sırasında kaybedilen süre nedeniyle bir satır  $52\mu s$ 'de taranmalıdır.

Geri dönüş yüzünden kaybedilen süre nedeniyle 625 satırdan ancak 576'sı TV ekranına basılabilmektektir. Televizyonda en boy oranı (aspect ratio) 4:3 olduğundan  $576*(4/3)=768$  nokta/satır basılması gerekmektedir.

Bir satır için 768 piksel basılması gerekiyorsa 1 sn için 14.76 Milyon piksel basılmalıdır. İşaretin sadece 0 ve 1'lerden olduğu varsayılsa bant genişliği  $B=7.38MHz$  olacaktır. Ancak normalde pikseller arası geçişler keskin olmadığından Kell faktörü dikkate alınarak bant genişliği  $B=7.78*0.75=5.5MHz$  olarak bulunur.

## Sayısal İmge Kavramı

İmge Modeli: İmge iki boyutlu ışık-yoğunluk fonksiyonu olarak adlandırılabilir ve  $f(x,y)$  ile gösterilir.  $(x,y)$  koordinatları belirtmekte,  $f(x,y)$  fonksiyonunun değeri ise  $(x,y)$  noktasındaki pikselin ışıklılık (intensity) değerini ifade etmektedir.



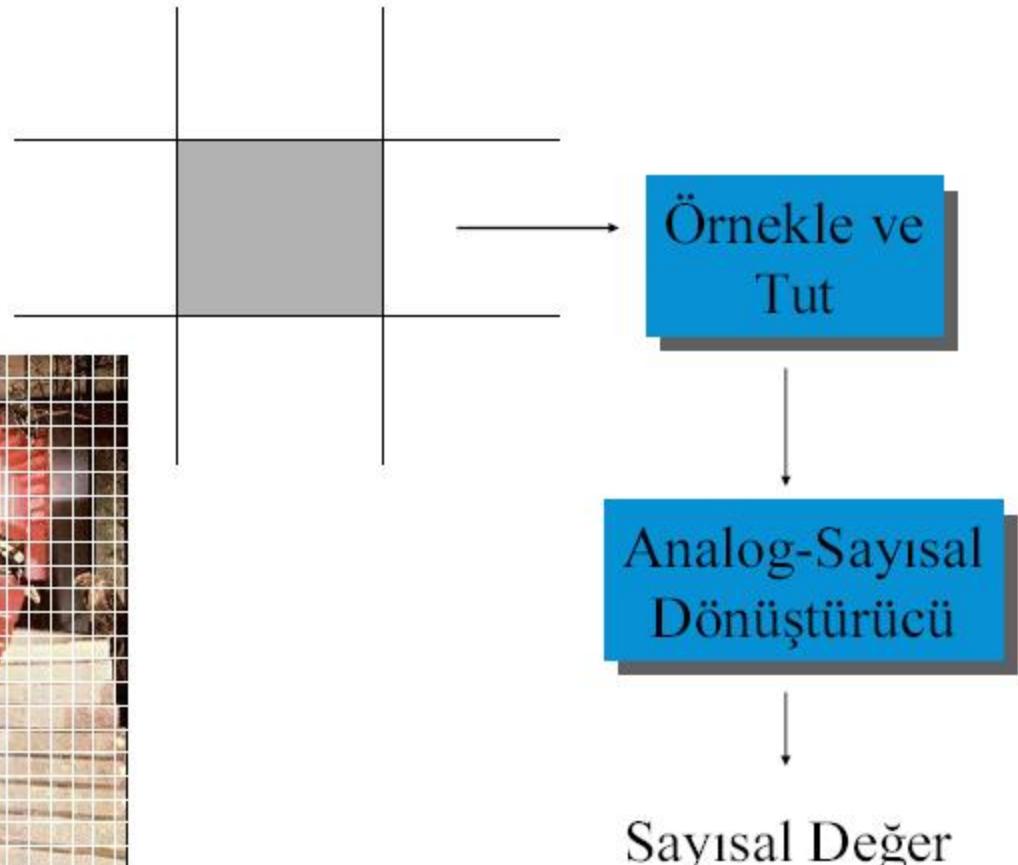
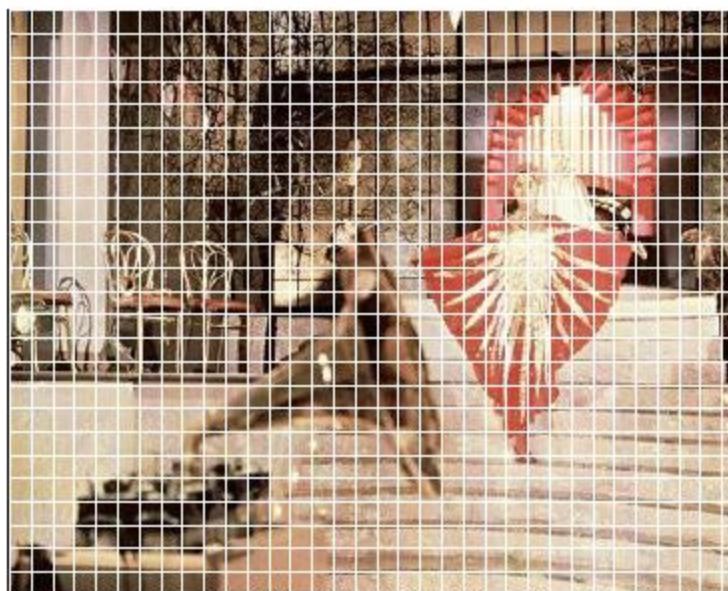
## Sayısal İmge : Sayısala Dönüştürme

İmgelerin sayısal imge işleme teknikleri ile işlenebilmesi analog biçimden sayısala çevrilmesi gerekmektedir. Bilindiği gibi analog bir işaretin sayısala çevrilmesi iki aşamada gerçekleşir.

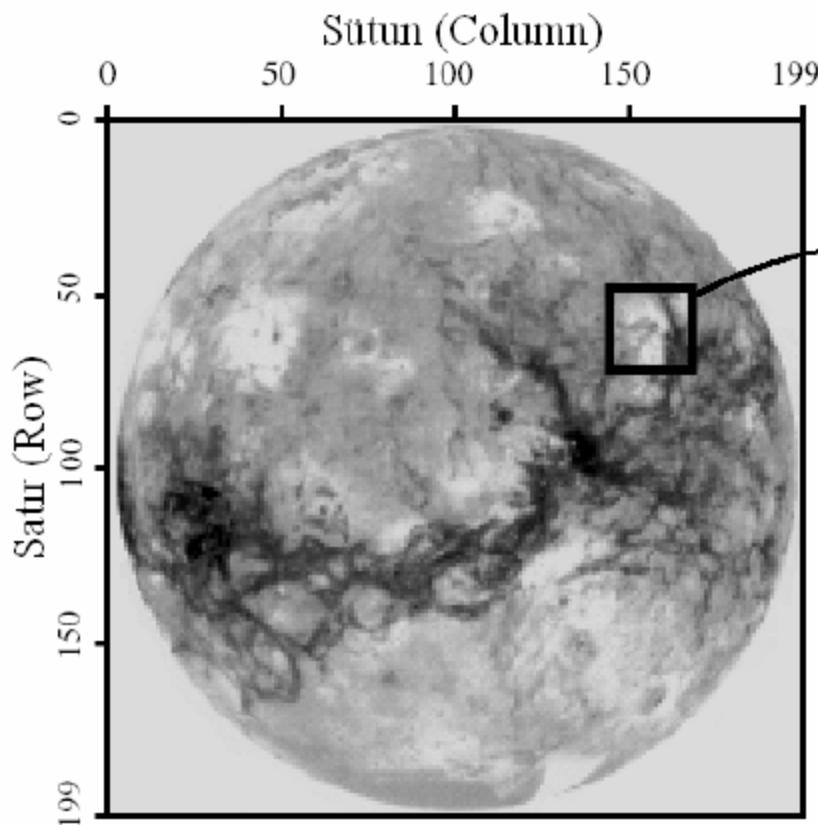
- Örnekleme : İşaretten belirli zaman anlarında örnekler alınmasına anlamına gelir. İki boyutlu imgelerde uzamsal düzlemin sayısallaştırılmasına karşılık gelir.
- Nicemleme (Kuantalama) : Genlik seviyelerinin sadece belirli değerleri alması işlemidir. Genlik değerlerinin sayısallaştırılmasına karşılık gelir.

$$\text{SAYISALLAŞTIRMA} = \text{ÖRNEKLEME} + \text{NİCEMLEME}$$

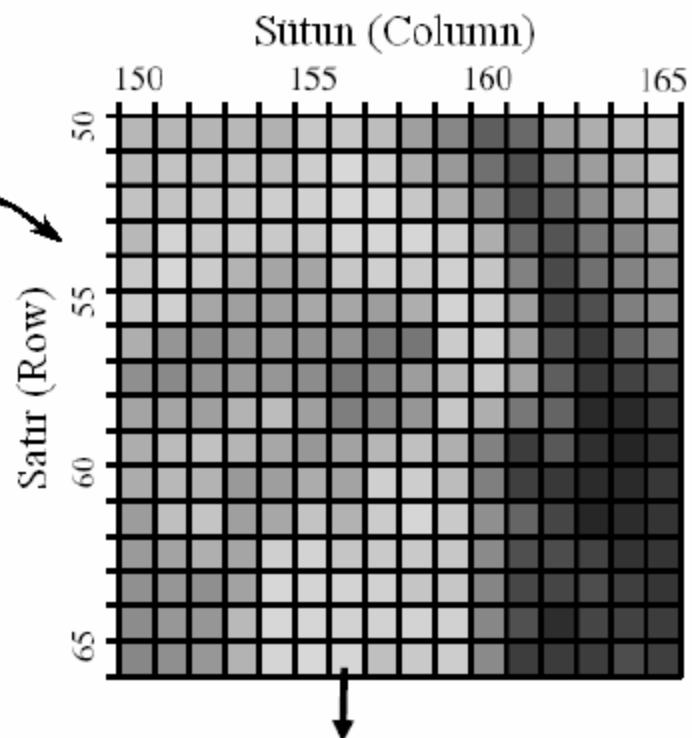
## Sayısal İmge : Sayısal Dönüştürme (Devami)



## Sayısal İmgede Piksel Kavramı

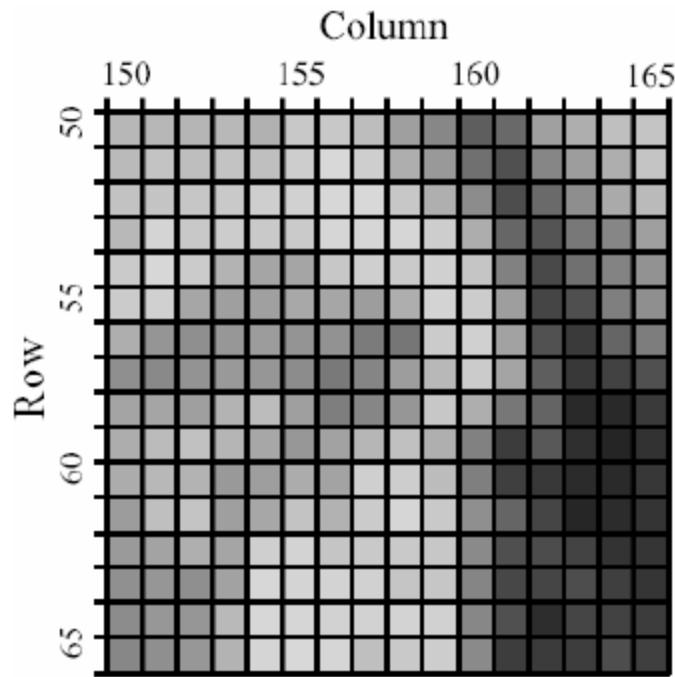


200x200 piksel boyunda bir imge

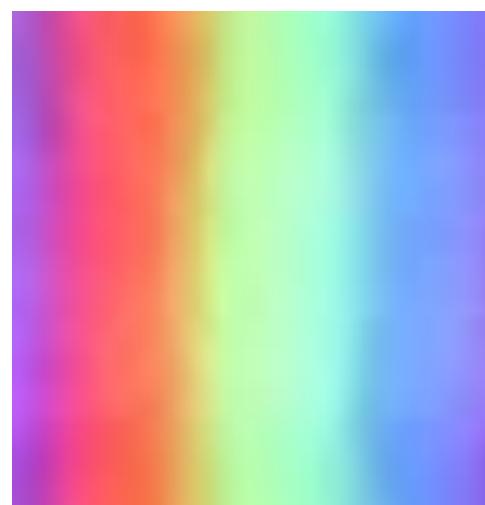
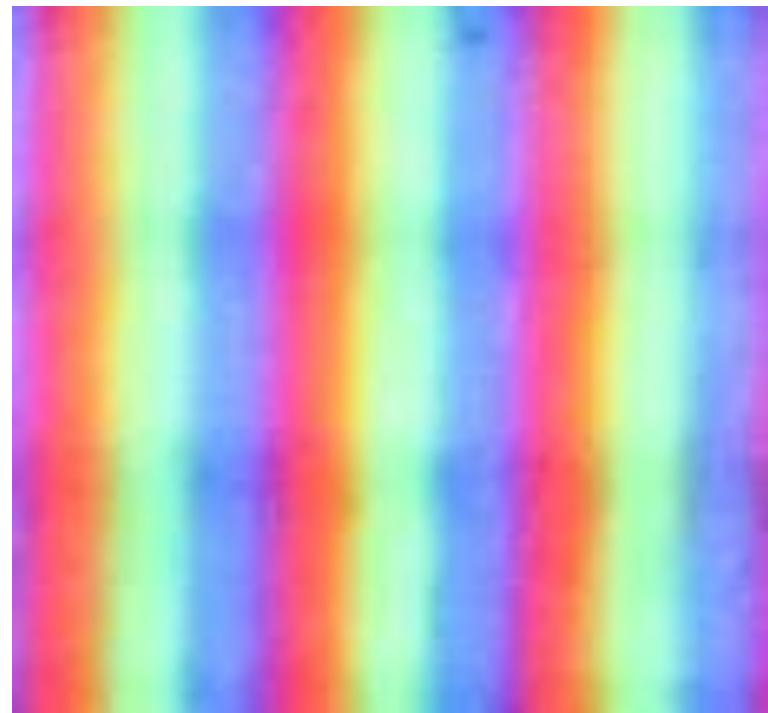
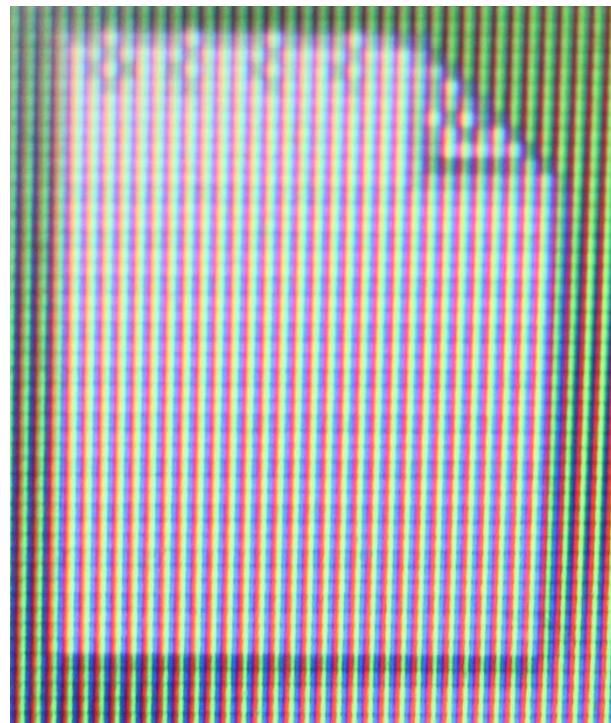
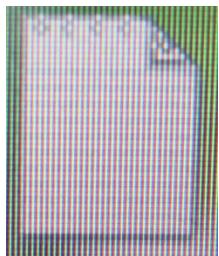


İmge/resim elemani  
(picture element=pixel/pel)

# Sayısal İmgede Piksel Kavramı



Column	150	155	160	165													
Row	50	183 183 181 184 177 200 200 189 159 135 94 105 160 174 191 196	186 195 190 195 191 205 216 206 174 153 112 80 134 157 174 196	194 196 198 201 206 209 215 216 199 175 140 77 106 142 170 186	184 212 200 204 201 202 214 214 214 205 173 102 84 120 134 159	202 215 203 179 165 165 199 207 202 208 197 129 73 112 131 146	203 208 166 159 160 168 166 157 174 211 204 158 69 79 127 143	174 149 143 151 156 148 146 123 118 203 208 162 81 58 101 125	143 137 147 153 150 140 121 133 157 184 203 164 94 56 66 80	164 165 159 179 188 159 126 134 150 199 174 119 100 41 41 58	173 187 193 181 167 151 162 182 192 175 129 60 88 47 37 50	172 184 179 153 158 172 163 207 205 188 127 63 56 43 42 55	156 191 196 159 167 195 178 203 214 201 143 101 69 38 44 52	154 163 175 165 207 211 197 201 201 199 138 79 76 67 51 53	144 150 143 162 215 212 211 209 197 198 133 71 69 77 63 53	140 151 150 185 215 214 210 210 211 209 135 80 45 69 66 60	135 143 151 179 213 216 214 191 201 205 138 61 59 61 77 63



## Sayısal İmgede Örnekleme (Uzamsal Çözünürlük)

256x256 piksel



128x128 piksel



64x64 piksel



32x32 piksel



72ppi

36ppi

18ppi

Fiziksel boyutlar aynı ( $3.6'' \times 3.6''$ )

Çözünürlükler farklı

Çözünürlük imgenin piksel sayısı/ birim fiziksel boyut ifadesi ile bulunur. Birimi genelde ppi (pixels per inch) ya da dpi (dot per inch) ifadeleri ile verilir.

## Sayısal İmgede Örnekleme (Uzamsal Çözünürlük)



L=8 bit



L=7 bit



L=6 bit



L=5 bit



L=4 bit



L=3 bit



L=2 bit



L=1 bit

## İmge Dosyaları (Tipleri)

Sayısal imgeler, manyetik, optik, flash disk gibi ortamlarda dosyalar halinde saklanmaktadır. İmge dosyalarının bilgisayarda saklanması için genellikle bmp veya jpg dosya tipleri tercih edilmektedir.

Örneğin gri-tonlu (gray-scale) tonlu 8 bit/pixel bit derinliğinde 200x200 piksel boyutunda bir imgenin bellekte kaplayacağı alanı hesaplayalım.

İmgede  $200 \times 200 = 40000$  piksel bulunacaktır. 8 bit/pixel bit derinliği kullanıldığından her piksel  $8\text{bit} = 1\text{ bayt}$  ile ifade edilecektir. Sonuç olarak bu imge için  $40000\text{ bayt} = 39\text{KB}$  bellek kullanılmalıdır. Renkli imgeler için ise bu sayı 3 ile çarpılmalıdır. Çünkü RGB renk uzaylarının her biri için  $40000\text{bayt}$  kullanılmalıdır.

Yukarıda anlatılan durum herhangi bir sıkıştırma algoritması kullanılmaması durumunda geçerlidir.

## İmge Dosyaları (Tipleri) : RAW

Genellikle sadece piksel ışıklılık değerlerinin bulunduğu dosya tipidir. İmgenin piksel boyutunu gösteren herhangi bir başlık bilgisi içermez (genelde). Dosyadaki imgeyi açmak için piksel boyutunu bilmek gereklidir.

Matlab ile raw dosyaları açmak için standart dosya okuma işlemleri kullanılır;

```
clear all; close all; clc;          % değişkenleri temizle, pencereleri kapat, ekranı sil  
w=256;                            % imgenin önceden bilinen yatay piksel sayısı  
h=256;                            % imgenin önceden bilinen düşey piksel sayısı  
  
image=['cm.raw'];                  % dosya adı  
FILE=fopen(image,'r');  
I1=fread(FILE);                   % dosya I1 değişkenine okundu  
status=fclose(FILE);  
  
Im=reshape(I1,w,h);              % I1 değişkeni wxh boyutuna getirilip, devriği  
figure;imshow(uint8(Im))         % (transpoze) alınıp Im'e atıldı ve ekranada gösterildi
```

## İmge Dosyaları (Tipleri) : RAW

Dosyada herhangi bir başlık bilgisi bulunmadığı için dosya boyutu  $256 \times 256 = 65536$  bayt'tır. Bu bilgi dosya özelliklerine girilerek veya Matlab'da Im değişkeninin boyutu kontrol edilerek doğrulanabilir.

The image shows two MATLAB windows side-by-side. On the left is the 'cm.raw Özellikleri' (Properties) dialog box for the file 'cm.raw'. It displays basic file information: Type: RAW Dosyası, Location: C:\MATLAB6p5\work, and Size: 64,0 KB (65.536 bayt). The size value is highlighted with a red box. On the right is the 'Workspace' browser window, which lists variables and their properties. The variable 'Im' is highlighted with a blue selection bar, showing its size as 256x256, which matches the file size.

Name	Size	Bytes	Class
FILE	1x1	8	doubl
Im	65536x1	524288	doubl
h	1x1	8	doubl
abc	image	1x6	12 char
status	1x1	8	doubl
w	1x1	8	doubl

## İmge Dosyaları (Tipleri) : YUV

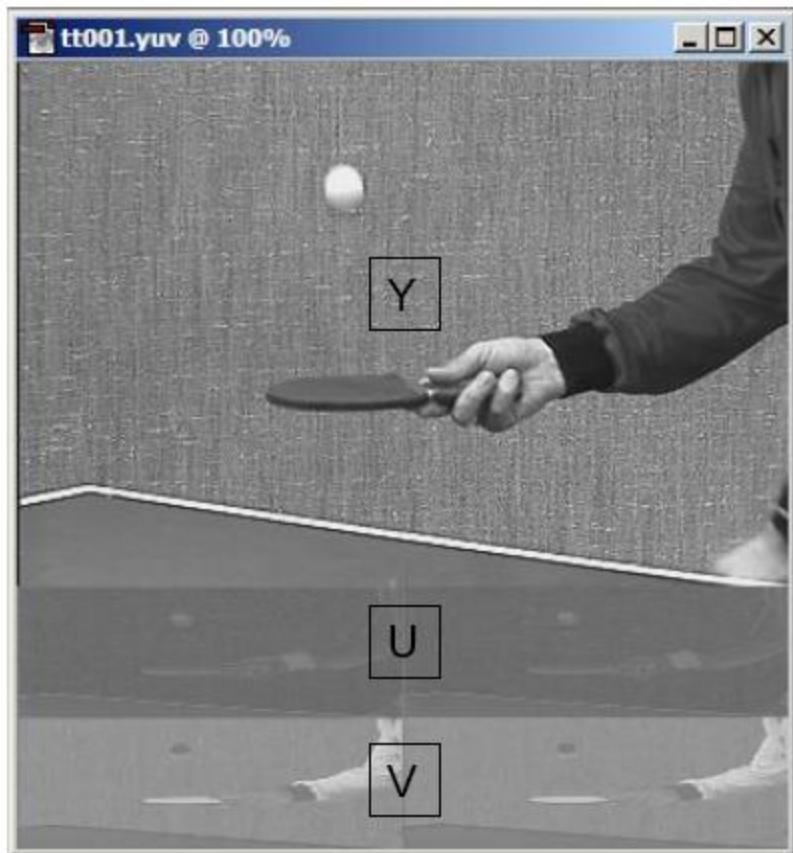
Hem ışıklılık (Y), hem de renk bileşenlerinin (U,V) olduğu dosya tipidir. Raw tipinde olduğu gibi genellikle imgenin piksel boyutu dosya içerisinde olmayıp önceden bilinmesi gereklidir.

Örneğin SIF formatındaki (352x240 piksel) tt001.yuv dosyasını Photoshop'ta açmak için imge piksel boyutları 352x360 olarak belirtmek gereklidir. Bunun nedeni ise ışıklılık bileşeninin 352x240 piksel yer kaplaması, renk bileşenlerinin ise bunun yarısı kadar yani 352x120 piksel yer kaplamasıdır.

Genellikle ışıklılık bileşenleri ile çalışacağımız için YUV tipi dosyalardan Y (ışıklılık) bileşeni çekilerek farklı bir dosya biçiminde (raw, bmp) saklanır.

**Not:** Renkli imgeler konusu işlenirken YUV formatına geri dönülüp, renk bilgilerinin kullanımına deðinilecektir.

## İmge Dosyaları (Tipleri) : YUV



YUV uzantılı imge dosyaları Photoshop programı ile raw uzantılı dosyaların açıldığı şekilde açılır. İmgenin piksel boyutu önceden bilinmelidir.

Yandaki şekilde tt001.yuv dosyasının ışıklılık (Y) ve renk bilgisini (U,V) kısımları görülmektedir.

## İmge Dosyaları (Tipleri) : YUV

YUV tipi dosyalardan ıshıklılık bileşenin (Y) çekilerek Matlab ile kaydedilmesi;

```
clear all; close all;clc; % değişkenleri temizle, pencereleri kapat, ekranı sil  
w=352; % imgenin önceden bilinen yatay piksel sayısı  
h=240; % imgenin önceden bilinen yatay piksel sayısı  
  
image=['tt001.yuv']; % dosya adı  
FILE=fopen(image,'r');  
I1=fread(FILE); % dosya I1 değişkenine okundu  
status=fclose(FILE);  
  
I2=I1(1:w*h); % I1'den sadece ıshıklılık bilgisi I2 değişkenine atıldı  
Im=reshape(I2,w,h)'; % I2 imgesi yeniden boyutlandırılmış, devriği alındı  
figure;imshow(uint8(Im)); % Elde edilen imge gösterildi  
  
imwrite(uint8(Im),'tt001.bmp'); % Elde edilen imge tt001.bmp olarak kaydedildi
```

## İmge Dosyaları (Tipleri) : BMP

Yaygın olarak kullanılan sıkıştırılmamış bir imge dosya biçimidir. Genellikle her renk bileşeninin 8bit/piksel bit derinliği ile ifade edildiği 24-bitlik format kullanılır. Gri tonlu imgeler bmp formatında saklandığı zaman 8bit/piksel bit derinliği kullanılır. Raw tipinden farklı olarak ıshıklılık (ve varsa renk) bilgisinin dışında imgenin piksel boyutu, bit/derinliği gibi bilgileri tutan başlık kısmı mevcuttur.

Matlab ile bmp uzantılı dosyaların açılması (gri-tonlu imgeler 2 boyutlu, renkli imgeler 3 boyutlu matris oluşturur);

```
clear all; close all; clc; % değişkenleri temizle, pencereleri kapat, ekranı sil
```

```
Im=imread('cm.bmp'); % cm.bmp adlı dosyayı oku, ıshıklılık (ve varsa renk)  
% değerlerini Im değişkenine yükle
```

```
figure;imshow(Im) % İmgeyi göster
```

## İmge Dosyaları (Tipleri) : TIFF, JPG ve diğerleri

JPEG formatı özellikle sıkıştırma verimliliği nedeni ile Internet'te kullanımıyla oldukça yaygınlaşmıştır. Genellikle kayıplı sıkıştırma ile elde edilen jpeg dosyaları kullanılmaktadır.

TIFF dosya biçimi kayıplı ve kayıpsız türde sıkıştırmaya ve sıkıştırma kullanmadan dosya saklamaya izin verir.

Bu dosya tipleri de bmp gibi Matlab'ın “imread” komutu ile okunabilir. imread komutunun desteklediği imge dosya tipleri aşağıda verilmektedir ( Matlab 6.5 sürümü için).

- JPEG, TIFF, GIF, BMP, PNG, HDF, PCX, XWD, ICO, CUR, RAS, PBM, PGM, PPM.

## Standart İmge Piksel Boyutları

İmge işlemede kullanılan bazı standart imge boyutu formatları bulunmaktadır. Bu formatlardan bazıları ve ilgili piksel boyutları aşağıda verilmektedir.

CIF (Common Input Format): 352x288 piksel

QCIF (Quadrat CIF) : 176x144 piksel

SIF (Source Input Format) : 352x240 piksel

## İmge Ters Çevirme

İmgelerle yapılacak işlemlerde, imgeleri  $w \times h$  boyutunda bir matris olarak düşünmek oldukça kolaylık sağlayacaktır. Ters çevirme işleminde, en üstteki satır en alta, en alttaki satır ise en üste gelecek şekilde matrisi (imgenin) bütün elemanları (ışıklılık değerleri) yer değiştirmektedir.



Orjinal imgé



Ters çevrilmiş imgé

## İmge Aynalama

Aynalama işlemi ters çevirme işleminin düşey eksende yapılmış şekli olarak düşünülebilir. Bu durumda, örneğin; [1,1] noktasındaki piksel (sol üst köşedeki) [w,1] noktasına (sağ üst köşeye), (w,1) noktasındaki piksel (sağ üst köşedeki) ise [1,1] noktasına (sol üst köşeye) gidecektir. Burada  $w$  imgenin yatay piksel uzunluğunu göstermektedir.



Orjinal imge



Aynalananmış imge

## MATLAB İle İmge Döndürme İşlemleri

MATLAB'ın hazır fonksiyonları arasında imge döndürmeye olanak sağlayan imrotate komutu bulunmaktadır.

**I2=imrotate(I1,Açı,Yöntem);**

I1 : Döndürülecek imge

Açı : Saat yönünün tersinde kaç derece döndürüleceği

Yöntem: Döndürme işlemi sonrası imgelerin yeni piksel değerlerinin hangi aradeğerleme yöntemi ile belirleneceğini belirtmede kullanılır. ‘nearest’, ‘bilinear’, ‘bicubic’ değerlerini alabilir. Belirtilmemesi durumunda varsayılan yöntem ‘nearest’ dir.

Örnek : **Im=imrotate(I,65,'bicubic');**

## İmge Öteleme

Öteleme işlemi, yatay ve düşey eksenlerde belirlenen piksel miktarı kadar imgenin yatay ve düşey eksende kaydırılması ile gerçekleştirilir.

30 piksellik yatay, 10 piksellik düşey öteleme için orjinal ve ötelenmiş ”Akiyo” imgesi aşağıda verilmektedir.



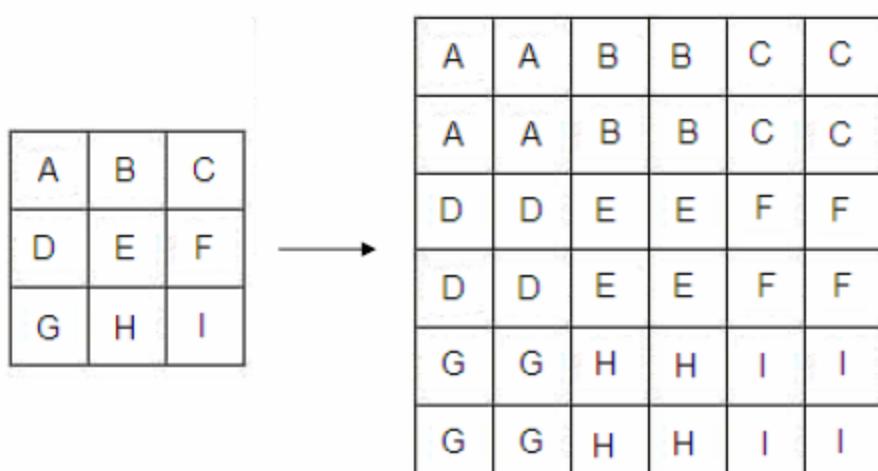
Orjinal imge



Ötelenmiş imge

## Boyut Değişikliği (Yakınlaştırma / Zoom In)

Bilindiği üzere yakınlaştırma etkisi bir imge veya görüntü içerisinde ilgilenilen bölgenin daha büyük göstermeyi ve daha detaylı bilgi edinmeyi sağlamaktadır. Bu işlemin optik olarak fotoğraf makinesi veya kamera ile yapılması yakınlaştırma sonrası elde edilen görüntüde kalite düşmesine neden olmazken, yazılımsal olarak (*digital zoom*) yapılması kalitede düşmeye neden olur.



3x3'lük bir imgenin yakınlaştırma öncesi ve sonrasında piksel değerleri harflerle gösterilmiştir. Burada yapılan sadece ilgili piksel değerinin imge boyutunu iki katına çıkaracak şekilde yeni imgeye kopyalanmasıdır. (nearest neighbor interpolation)

## Boyut Değişikliği (Yakınlaştırma / Zoom In)

Yakınlaştırma işleminde ilgilenilen pikselin değerini kopyalamak yerine komşu piksel değerleri de kullanılabilir. Böylelikle daha yumuşak geçişler elde edilir.

The diagram illustrates a zoom-in operation (nearest neighbor interpolation) from a 3x3 input matrix to a 5x5 output matrix. An arrow points from the smaller input matrix to the larger output matrix.

A	$\frac{A+B}{2}$	B	$\frac{B+C}{2}$	C
D	$\frac{A+D}{2}$	$\frac{A+B+D+E}{4}$	$\frac{B+E}{2}$	$\frac{B+C+E+F}{4}$
G	$\frac{D+G}{2}$	$\frac{D+E+G+H}{4}$	$\frac{E+H}{2}$	$\frac{E+F+H+I}{4}$
	H	$\frac{G+H}{2}$	I	$\frac{H+I}{2}$

## Boyut Değişikliği (Yakınlaştırma / Zoom In)

Doğrudan kopyalama ve komşu piksel değerlerini kullanma yöntemlerinden hangisi daha iyi sonuç verir ? (Tabii ki sağdaki ☺)



Kenar bölgeleri incelendiğinde, doğrudan piksel değerlerini kopyalama yönteminin (nearest neighbor interpolation) özellikle kenarlarda sert geçişlere neden olduğu ve sonuç olarak görsel kaliteyi düşürdüğü görülmektedir.

## Boyut Değişikliği (Uzaklaştırma / Zoom Out)

Uzaklaştırma işleminde temel olarak birden fazla pikselin değeri çeşitli işlemlerden (ortalama gibi) geçirilerek bir piksele atanır. En basit uzaklaştırma yöntemi B,E,F değerini göz ardı edip 2x2'lik blok için A pikselinin değerini kullanmaktadır. Bu yöntemden biraz daha iyi sonuç veren aradeğerleme kullanılmıştır. Uzaklaştırma işlemi uygulanan imgenin piksel değerleri 4 pikselin ışıklılık değerlerinin ortalaması ile bulunmaktadır.

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

$\frac{A+B+E+F}{4}$	$\frac{C+D+G+H}{4}$
$\frac{I+J+M+N}{4}$	$\frac{K+L+O+P}{4}$

# MATLAB İle İmge Boyut Değiştirme İşlemleri

MATLAB'ın hazır fonksiyonları arasında imgenin piksel boyutunu değiştirmeye olanak sağlayan imresize komutu bulunmaktadır.

**I2=imresize(I1,Oran,Yöntem);**

**I1 :** Boyutu değiştirilecek imge

**Oran :** Giriş imgesinin kaç kat büyütüleceği belirlenir. 1'den büyük olması durumunda yakınlaştırma, 1'den küçük olması durumunda uzaklaştırma işlemi yapılır.

**Yöntem:** Boyut değiştirme işlemi sonrası imgelerin yeni piksel değerlerinin hangi aradeğerleme yöntemi ile belirleneceğini belirtmede kullanılır. 'nearest', 'bilinear', 'bicubic' değerlerini alabilir. Belirtilmemesi durumunda varsayılan yöntem 'nearest' dir.

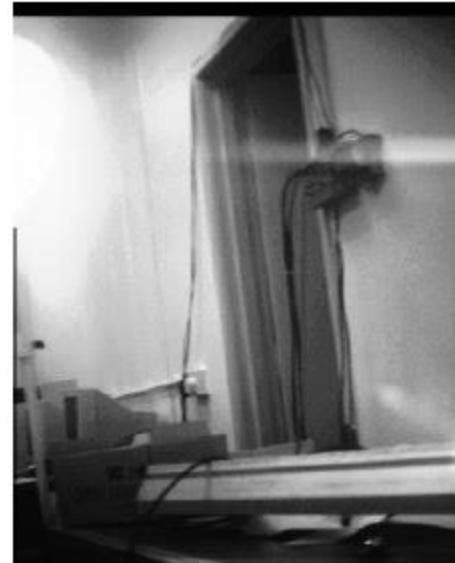
Örnek : **I2=imresize(Ix,1.25,'bilinear');**

## İmge Pekiştirme

İmge pekiştirmede amaçlanan çeşitli nedenlerden dolayı kullanışız olan (yeterli bilgi vermeyen) imgenin işlenerek görsel veya işlevsel anlamda kullanılabilir bir imge elde etmektir.



Pozlandırma süresinden dolayı  
oluşan karanlık imge



Pekiştirme işleminden sonra  
elde edilen imge

## İmge Pekştirme : Parlaklık Ayarı

Parlaklık değişimi imgenin ışıklılık değerlerinin belirli bir sayıyla toplanması veya çıkarılması ile gerçekleştirilir.  $f(x,y)$  orjinal imgeyi göstermek üzere, parlaklık ayarı yapılmış  $g(x,y)$  aşağıdaki şekilde elde edilir.

$$g(x, y) = f(x, y) + b$$

$b > 0$  ise parlaklık artırılır  
 $b < 0$  ise parlaklık azaltılır



Orjinal



$b = -50$



$b = +50$

## İmge Pekştirme : Karşılık (Kontrast) Ayarı



Parlaklık değişimi imgenin ışıklılık değerlerinin belirli bir sayıyla çarpılması ile gerçekleştirilir.

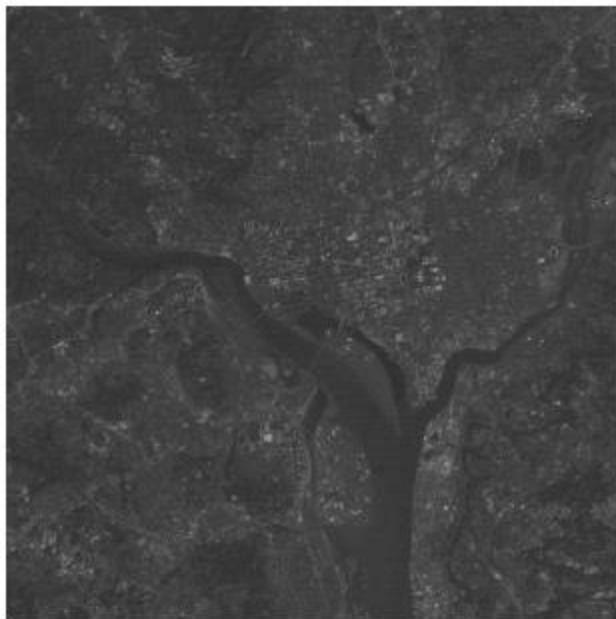
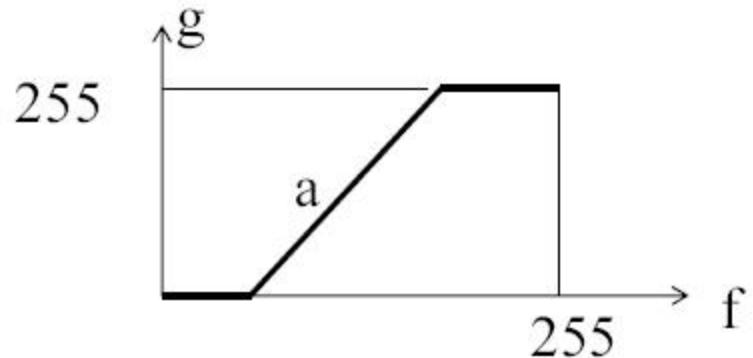
$$g(x, y) = af(x, y)$$

$a > 0$  ise karşılık arttırılır  
 $a < 0$  ise karşılık azaltılır

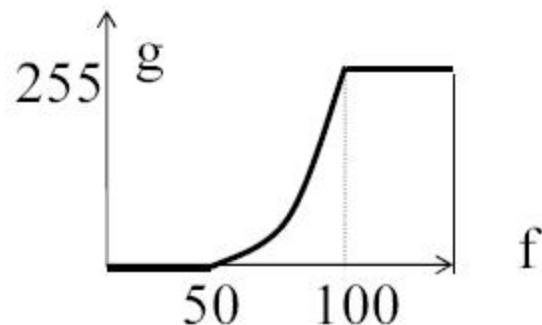


## İmge Pekştirme : Parlaklık & Karşılık Ayarı

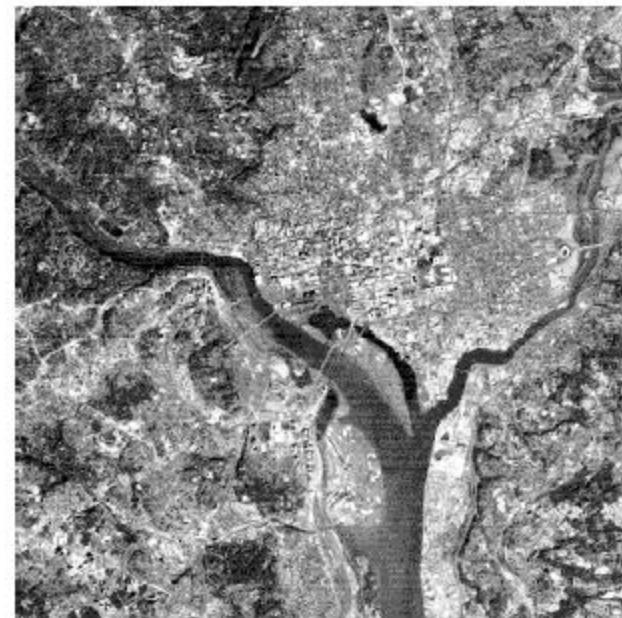
$$g(x, y) = af(x, y) + b$$



## İmge Pekştirme : Parlaklık & Karşılık Ayarı

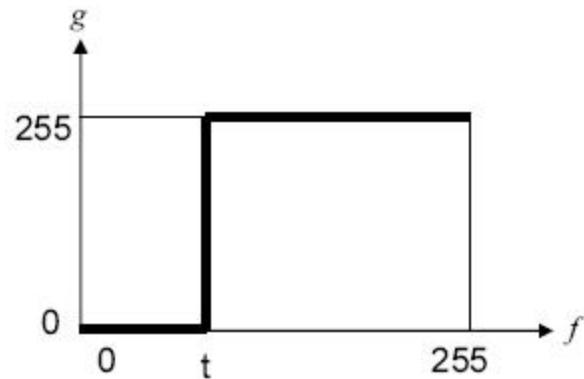


Matlab'da **imadjust** komutu kullanılarak parlaklık ve karşılık ayarı yapılabilir.



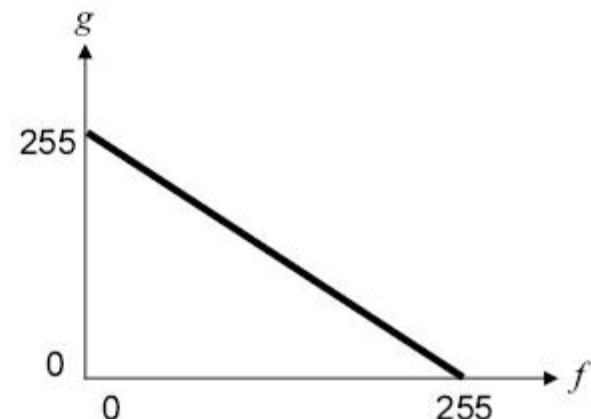
## İmge Pekştirme : Eşikleme (Thresholding)

İmge ışıklılık değerlerinin belirli bir değerden büyük/küçük olmasına göre ikili (binary) bir imge oluşturulur. Bu ikili imgedeki renkler siyah ve beyazdır.



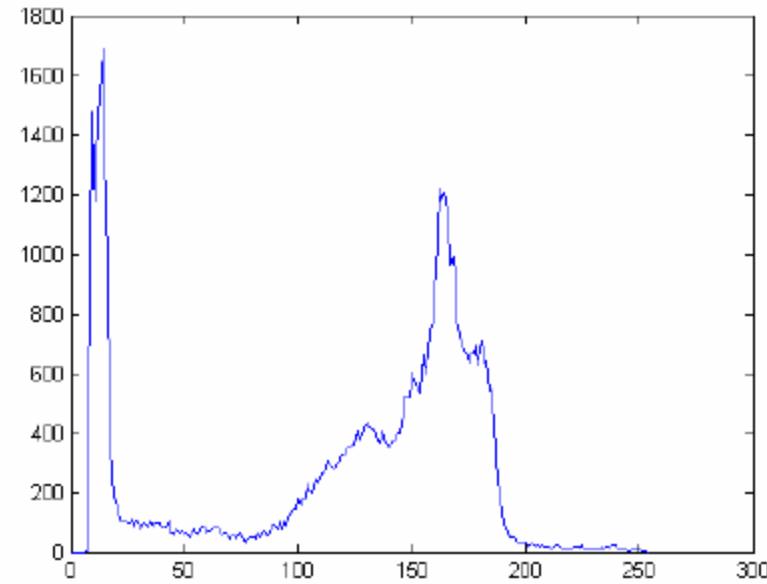
## İmge Pekiştirme : Olumsuzlama (Negation)

İmge ıshıklılık değerlerinin doğrusal biçimde ters çevrilmesi ise gerçekleştirilir.

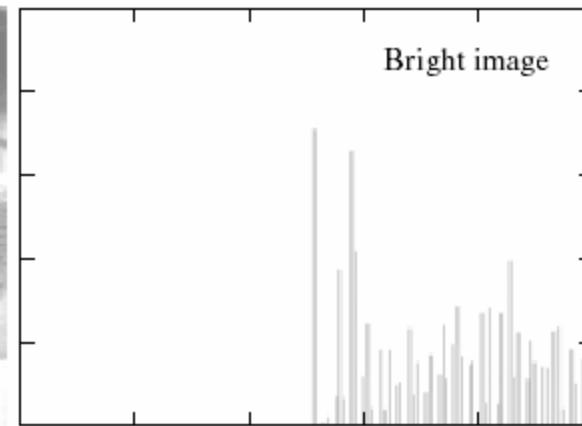
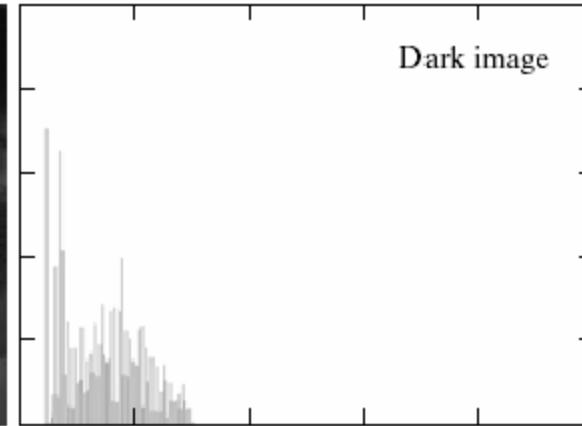
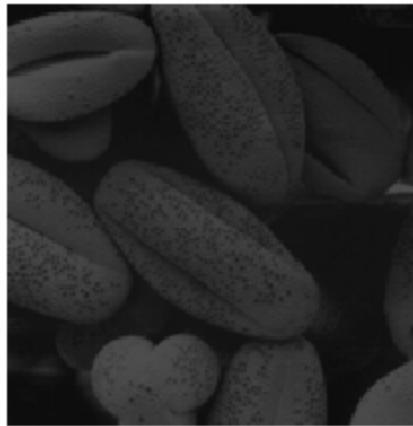


# Histogram

Histogram: Her piksel seviyesinin imgedeki miktarını (frekansını) gösterir. [h(i)] Örneğin camereman imagesinde, paltonun rengi siyaha yakın olduğu için histogramda 20 civarlı piksel seviyesinde bir tepe değeri oluşmuştur.



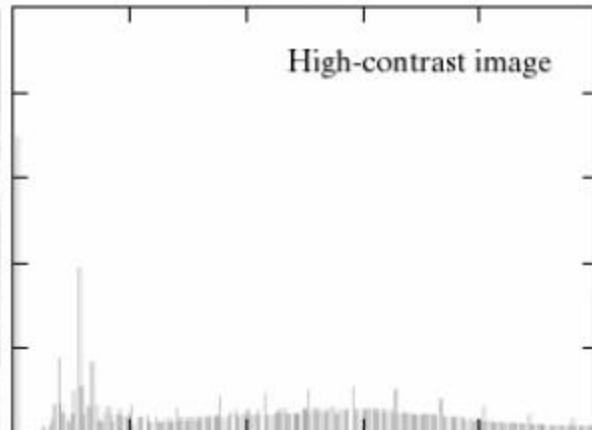
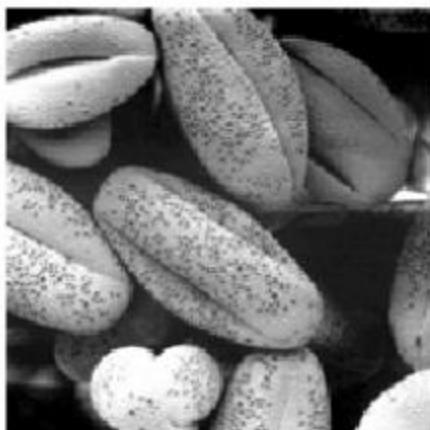
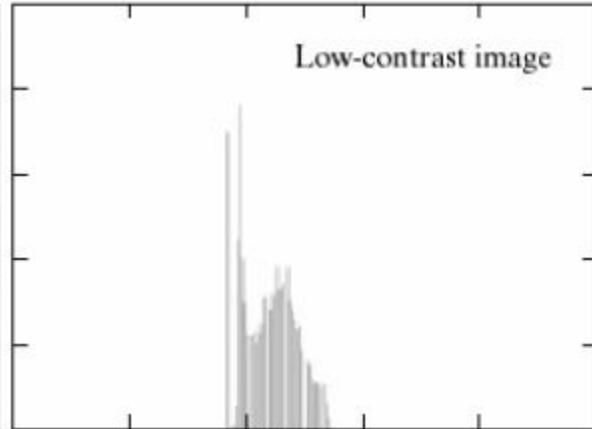
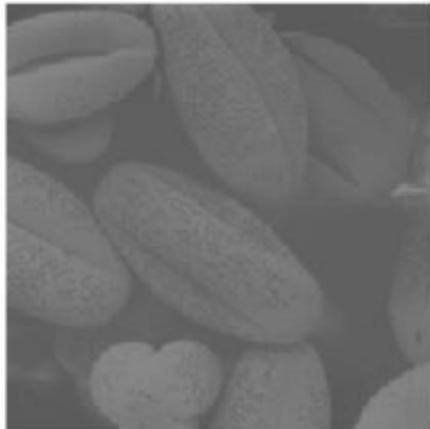
# Histogram



İmgenin histogramı sol tarafta yoğunlaştığı için imgenin koyu bir imgे olduğunu söylenebilir.

İmgenin histogramı sağ tarafta yoğunlaştığı için imgenin açık tonlu bir imge olduğunu söylenebilir.

# Histogram



İmgenin histogramı ortada küçük bir alan kapladığı için karşıtlığı düşük bir imge olduğu söylenebilir.

İmgenin histogramı geniş bir alan kapladığı için karşıtlığı bu imgenin karşıtlığı yüksek bir imge olduğu söylenebilir.

## İmge Pekştirme : Histogram Eşitleme

- İmgedeki düşük karşılığı iyileştirmek amacı ile kullanılır. Histogramdaki frekans bilgisine bağlı olarak yapılan doğrusal olmayan eşlemedir.
- Yüksek frekanslı piksel seviyesi geniş piksel alanına, düşük frekanslı piksel seviyesi ise dar piksel alanına yerleştirilmektedir.
- Bu sayede çok kullanılan piksel seviyeleri belirgin hale dönüştürülmektedir.

Algoritma :

$d=255/\text{imgenin toplam piksel sayısı}$

//imgenin histogramını hesapla

$c(0)=d*\text{histogram}(0)$

//geri kalan seviyeler için

$c(i)=c(i-1)+d*\text{histogram}(i)$

#imgenin tüm pikselleri için

$g(x,y)=c(f(x,y))$

## Histogram eşitlemenin özeti

$$s_k = \sum_{j=0}^k \frac{n_j}{n} * (L-1) \quad k = 0, 1, 2, \dots, L-1$$

- **Uygulanışı:**

1-Resmin histogramı bulunur (**her gri seviye için piksel sayısı grafiği**).

2-Histogramdan yararlanılarak kümülatif histogram bulunur. Kümülatif histogram, histogramın her değerinin kendisinden öncekiler ve kendisinin toplamı ile elde edilen değerleri içeren büyülüktür.

3-Kümülatif histogram değerleri normalize edilip (toplam piksel sayısına bölünerek), yeni resimde olmasını istediğimiz max. renk değerleri ile çarpılır, çıkan değer tam sayıya yuvarlanır. Böylelikle yeni gri seviye değerleri elde edilmiş olur.

4- Eski (Orijinal) gri seviye değerleri ile; 3.adımda elde edilen gri seviye değerleri biribirine karşılık düşürülür ve yeni histogram grafiği çizilir.

**n:** giriş görüntüsündeki toplam piksel sayısı ( $n_0 + n_1 + \dots + n_{L-1} = n$ )

**$n_j$  ( $n_k$ ):** j. gri seviyedeki piksel sayısı,

**L:** mümkün olan (veya istenilen) toplam gri seviye sayısı( 8 bit renk derinliğinde 255 v.b)

**$s_k$ :** Daha iyi kontraslı bir görüntü elde etmek için gri seviye dönüşüm değeri.

## Histogram Eşitleme

- Amaç: İmgedeki düşük görünürlüğü iyileştirmek.
- Olasılık dağılımına bağlı olarak doğrusal olmayan dönüşüm gerçekleştirilir.
- Bu sayede, bulunma olasılığı yüksek pikseller arası fazlaca açılırken, düşük olasılıklı seviyeler birbirine daha yakın hale gelir.

$$cdf(v) = round\left(\frac{cdf(v) - cdf_{\min}}{(M \times N) - cdf_{\min}} \times (L - 1)\right)$$



**Örnek:**  
Görüntü  
bloğu

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	86	2	113	1
65	3	66	3	73	2	87	1	122	1
68	2	66	2	75	1	88	1	126	1
59	3	57	1	76	1	90	1	144	1
60	1	66	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Value	cdf	Value	cdf	Value	cdf	Value	cdf	Value	cdf
52	1	64	19	72	40	85	51	113	60
65	4	66	22	73	42	87	52	122	61
58	6	66	24	75	43	88	53	126	62
59	9	67	25	76	44	90	54	144	63
60	10	68	30	77	45	94	55	154	64
61	14	69	33	78	46	104	57		
62	15	70	37	79	48	106	58		
63	17	71	39	83	49	109	59		

cdf

$$cdf(v) = round\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

$$cdf(v) = round\left(\frac{cdf(v)-1}{64-1} \times 255\right)$$

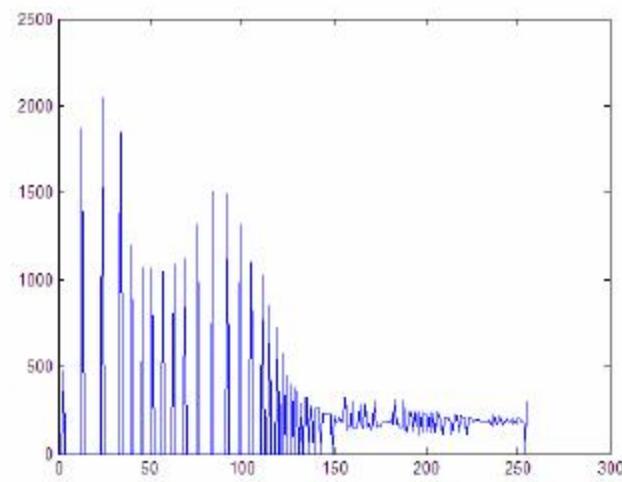
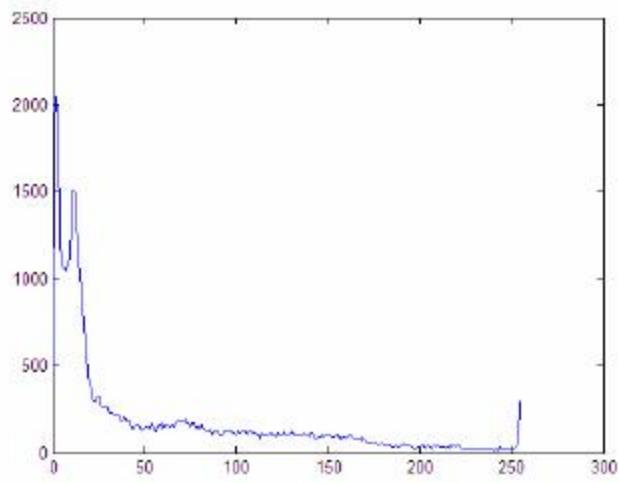
$$cdf(78) = round\left(\frac{46-1}{63} \times 255\right) = 182$$

histogram

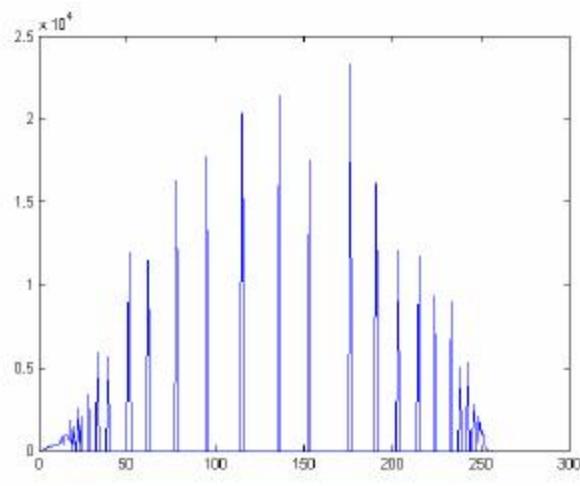
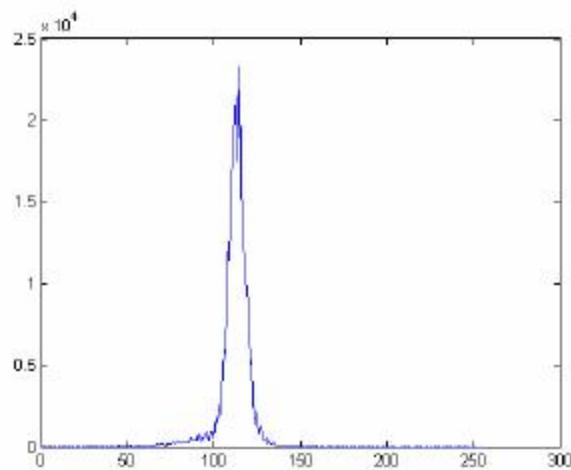
$$cdf(154) = round\left(\frac{64-1}{63} \times 255\right) = 255$$

0	12	53	93	146	53	73	166
65	32	12	215	235	202	130	158
57	32	117	239	251	227	93	166
65	20	154	243	255	231	146	130
97	53	117	227	247	210	117	146
190	85	36	146	178	117	20	170
202	154	73	32	12	53	85	194
206	190	130	117	85	174	182	219

## İmge Pekştirme : Histogram Eşitleme

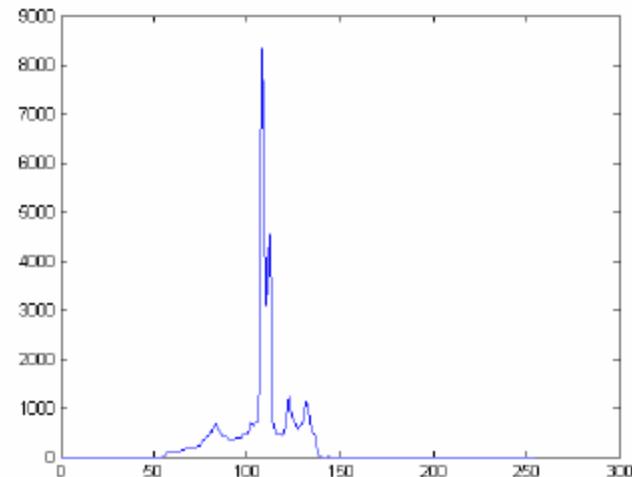


## İmge Pekştirme : Histogram Eşitleme



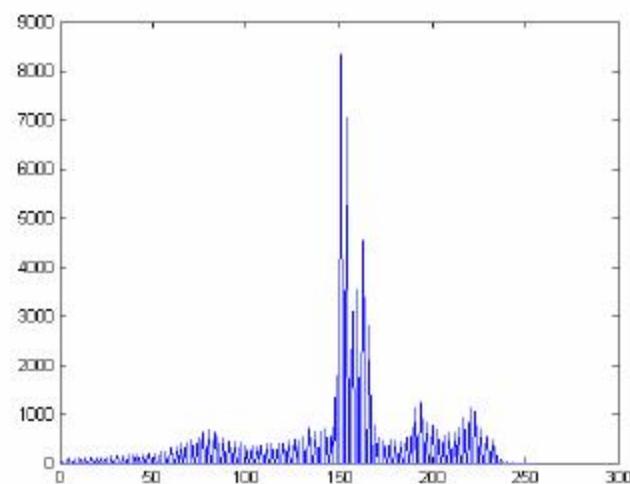
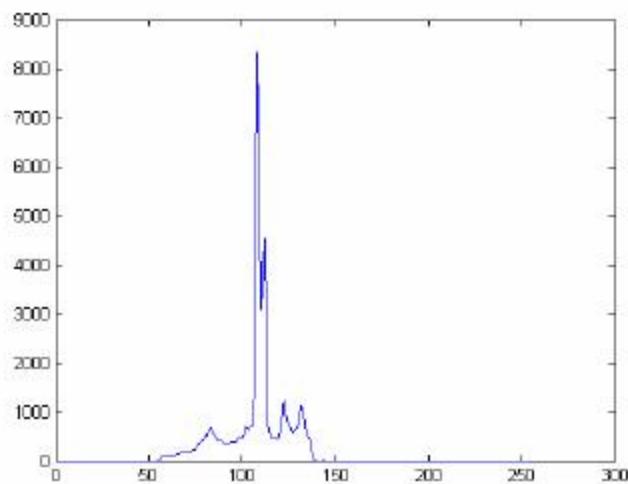
## İmge Pekştirme : Karşılık Yayma

$$y\_imge[i, j] = \frac{imge[i, j] - \min\_piksel\_deg}{\max\_piksel\_deg - \min\_piksel\_deg} \times \max\_gri\_ton\_deg$$

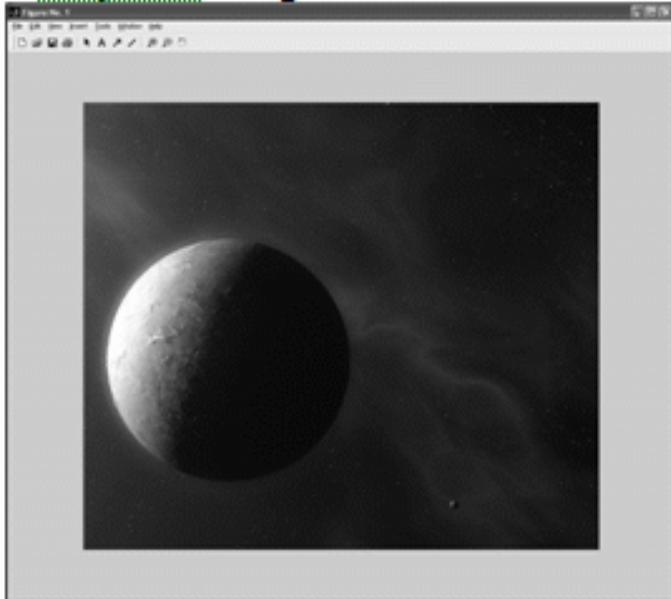


- Dikkat edilirse yapılan yayma işlemi imgedeki en düşük ve en yüksek ışıklılık seviyelerine bağlıdır. Yapılan işlem doğrusaldır.

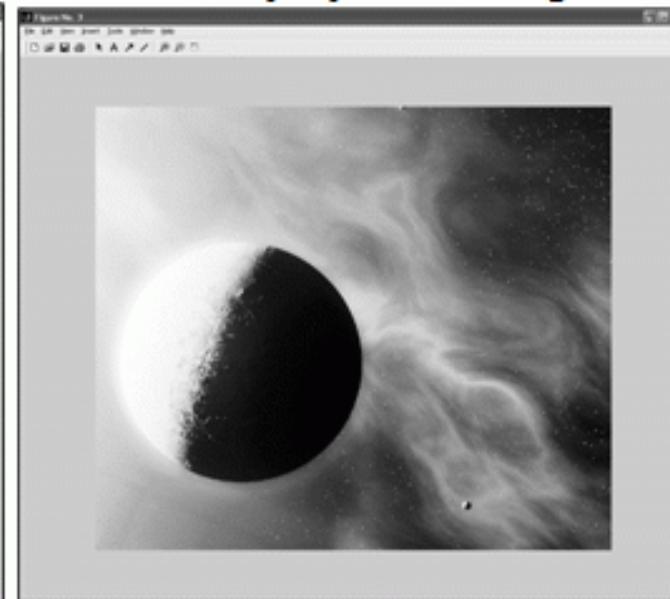
## İmge Pekştirme : Karşılık Yayma



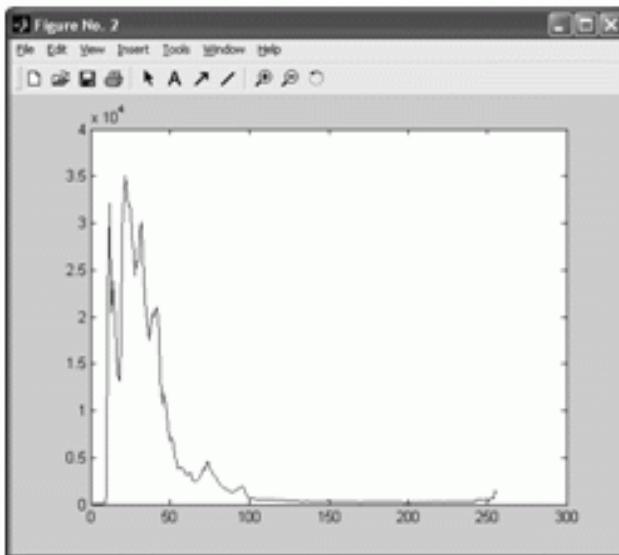
orijinal imge



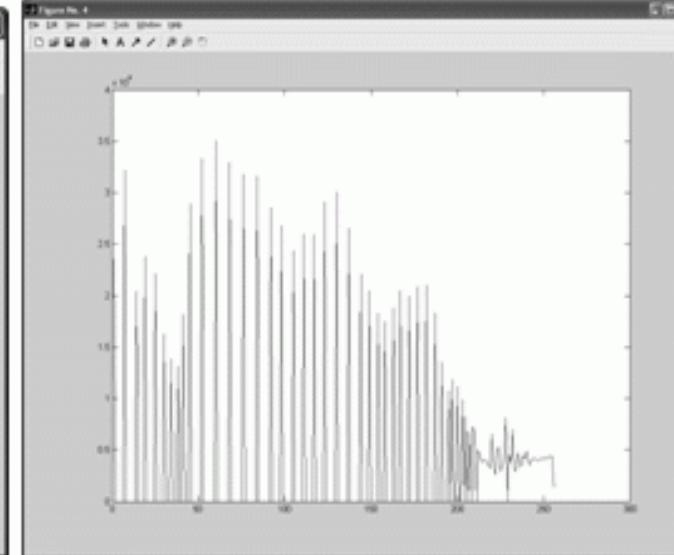
karsitlik yayilmis imge



Orijinal histogram



karsitlik yayilmis histogram



## İmge Pekiştirme : Piksel Komşuluk İşlemleri

Parlaklık, karşılık gibi noktasal pekiştirme işlemlerinde ilgili pikselin değeri sadece kendisine bağlı olarak değiştirilmekteydi

Piksel temelli komşuluk işlemlerinde ise ilgili pikselin yeni değeri, komşu piksellerin değerleri de dikkate alınarak hesaplanır. Hangi komşuların nasıl kullanılacağı seçilen piksel komşuluk işleme bağlı olarak belirlenir.

İmge üzerinde yapılacak uzamsal işlemlerde genellikle komşuluk ilişkileri dikkate alınır. Bunun nedeni ise tek başına fazla bir anlam ifade etmeyen bir pikselin, komşuları ile birlikte daha fazla bilgi içermesidir (örneğin; kenar bilgisi gibi).

Piksel komşuluk işlemlerinin hesapsal yükü, noktasal pekiştirme işlemlerinde daha fazladır.

## İmge Pekiştirme : Konvolüsyon

Konvolüsyonda bir pikselin değeri çevresindeki piksellerin ağırlıklı ortalaması ile bulunmaktadır. Komşuların gri seviyeleri konvolüsyon kerneli olarak adlandırılan bir matrisin katsayılarına göre ağırlıklandırılır.

Konvolüsyon işlemi aşağıda şekilde ifade edilmektedir:

$$g(x, y) = \sum_{j=-n}^n \sum_{i=-m}^m k(i, j) f(x-i, y-j) = k * f$$

$k$  : konvolüsyon kerneli

$f$  : islenecek imge

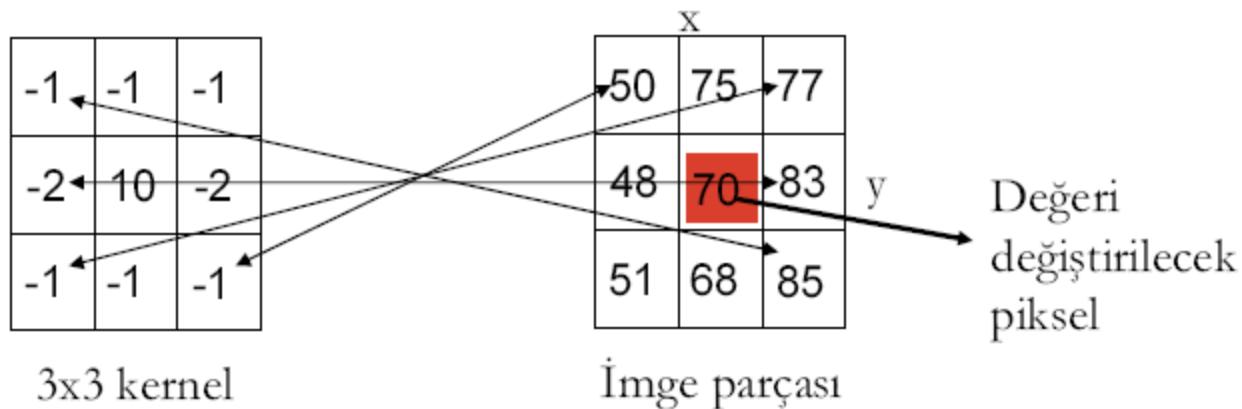
$w, h$  : imge piksel boyutu (tek sayı)

$$m = \frac{(w-1)}{2}$$

$$n = \frac{(h-1)}{2}$$

## İmge Pekştirme : Konvolüsyon

Konvolüsyonda kerneli uygulamaya göre  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  gibi boyutlarda seçilebilir.



$$\begin{aligned} g(x,y) = & k(-1,-1)f(x+1,y+1) + k(0,-1)f(x,y+1) + k(1,-1)f(x-1,y+1) + k(-1,0)f(x+1,y) \\ & + k(0,0)f(x,y) + k(1,0)f(x-1,y) + k(-1,1)f(x+1,y-1) + k(0,1)f(x,y-1) \\ & + k(1,1)f(x-1,y-1) \end{aligned}$$

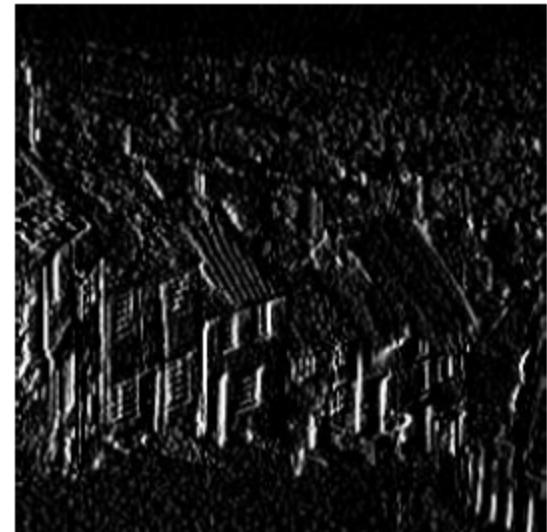
$$\begin{aligned} g(x,y) = & (-1 \times 85) + (-1 \times 68) + (-1 \times 51) + (-2 \times 83) + (10 \times 70) + (-2 \times 48) + (-1 \times 77) \\ & + (-1 \times 75) + (-1 \times 50) = 32 \end{aligned}$$

## İmge Pekiştirme : Konvolüsyon Örneği

$f(x,y)$



$g(x,y)$



$k(i,j)$

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

\*

=

Not: Konvolüsyon işlemi Matlab ile conv2 komutu ile yapılabilir.

## Konvolüsyon: Hesapsal Zorluklar

Konvolüsyon işleminin sonucu imgé ıshklılık degerinin alt ve üst sınırlarını geçebilir. Bu noktada 8-bit derinliğindeki imgeler için piksel ıshklılık degerlerinin 0-255 aralığında kalmasına dikkat edilmelidir.

İşlenecek imgenin kenar kısımlarına dikkat edilmelidir:

- Kenar bölgelerinin işlememe
- Kenar bölgelerini aynen kopyalama
- Kenar bölgelerini kesme
- Kenar bölgelerinde konvolüsyon kernelini kes
- Yansıtılmış endeksleme kullan...

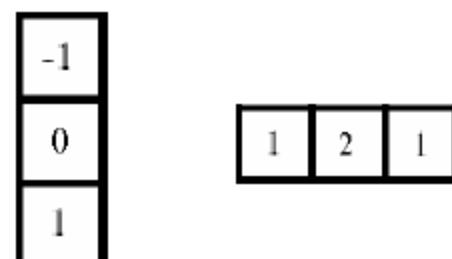
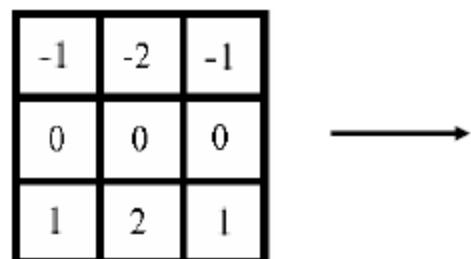
## Konvolüsyon: Ayrılabilir Kerneller

$m \times m$  boyutlu bir kernel için  $m^2$  çarpma ve  $m^2$  toplama işlemi gerekmektedir.

-1/8	-1/8	-1/8
-1/8	1	-1/8
-1/8	-1/8	-1/8

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

Ayrıştırılabilir kernelde ise  $2m$  çarpma ve  $2m$  toplama işlemi gerekmektedir.



# Konvolüsyon : Örnekler

0	0	0
0	1	0
0	0	0

Delta fonksiyonu



0	0	0
0	1	0
0	0	-1

Kaydır ve çıkart



## Konvolüsyon : Örnekler

$$\begin{bmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 1 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{bmatrix}$$

Kenar bulma



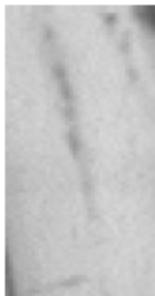
$$\begin{bmatrix} -k/8 & -k/8 & -k/8 \\ -k/8 & k+1 & -k/8 \\ -k/8 & -k/8 & -k/8 \end{bmatrix}$$

Kenar pekiştirme

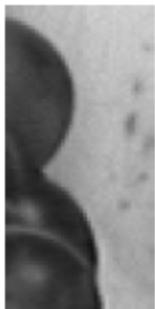


## İmgede Uzamsal Frekans Kavramı

İmgede pikseller arasındaki yumuşak geçiş *uzamsal düşük frekanslara* karşılık gelir.



İmgede pikseller arasındaki sert geçiş (kenarlar, nesne sınırları...) *uzamsal yüksek frekanslara* karşılık gelir.



## İmge Pekiştirme: Alçak Geçiren Süzgeç

Eleman değerleri pozitif olan kerneller imge ile konvolüsyona sokulduğunda alçak geçiren süzgeç işlevi görürler.

İşlem sonrası elde edilen imge, işlenen imgeye oranla daha yumuşak geçişlere sahiptir.



İmgedeki aşırı ışıklılık artışlarını önlemek için normalizasyon gereklidir.

## İmge Pekiştirme : Alçak Geçiren Süzgeç

Bu konvolüsyon işlemi aynı zamanda ortalama alma işlemine karşılık gelir. Bu yüzden ortalama alan süzgeç (mean filter) olarak da adlandırılırlar. Konvolüsyon kernelinin boyutu büyükçe imge daha da yumuşak geçişlere sahip olur.

$$(1/9) \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline 1/9 & 1/9 & 1/9 \\ \hline \end{array}, \quad (1/25) \times$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

## İmge Pekiştirme : Alçak Geçiren Süzgeç

Alçak geçen süzgeç işlemi, aşağıda verilen doğrusal olmayan kerneller kullanılarak da gerçekleştirilebilir.

0	1	0
1	0	1
0	1	0

0	2	0
2	1	2
0	2	0

,

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

## İmge Pekiştirme : Alçak Geçiren Sözgeç

İmge işleme yaygın olarak kullanılan kernellerden birisi de Gauss kernelidir. Aşağıda farklı katsayı ve boyutlarda Gauss kerneli örnekleri verilmiştir.

0	1	0
1	2	1
0	1	0

1	4	1
4	16	4
1	4	1

1	2	1
2	4	2
1	2	1

1	2	4	2	1
2	4	8	4	2
4	8	16	8	4
2	4	8	4	2
1	2	4	2	1

## İmge Pekiştirme : Yüksek Geçiren Süzgeç

Yüksek geçen süzgeçlerde kernel değerleri hem pozitif hem de negatif değerler içerir. Genellikle kenar bulma ve imgedeki yüksek frekansları (kenar bilgilerini) pekiştirme de kullanılır.

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

-1	-2	-1
-2	12	-2
-1	-2	-1

-1	-1	-1
-1	C	-1
-1	-1	-1

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	2	4	2	-1

Kenar bulma

Yüksek geçen süzgeç

Negatif kernel elemanlarından dolayı sonuç negatif çıkabilir. Bu durumda, oluşan imgeye ön değer eklenir.

## İmge Pekştirme : Yüksek Geçiren Süzgeç



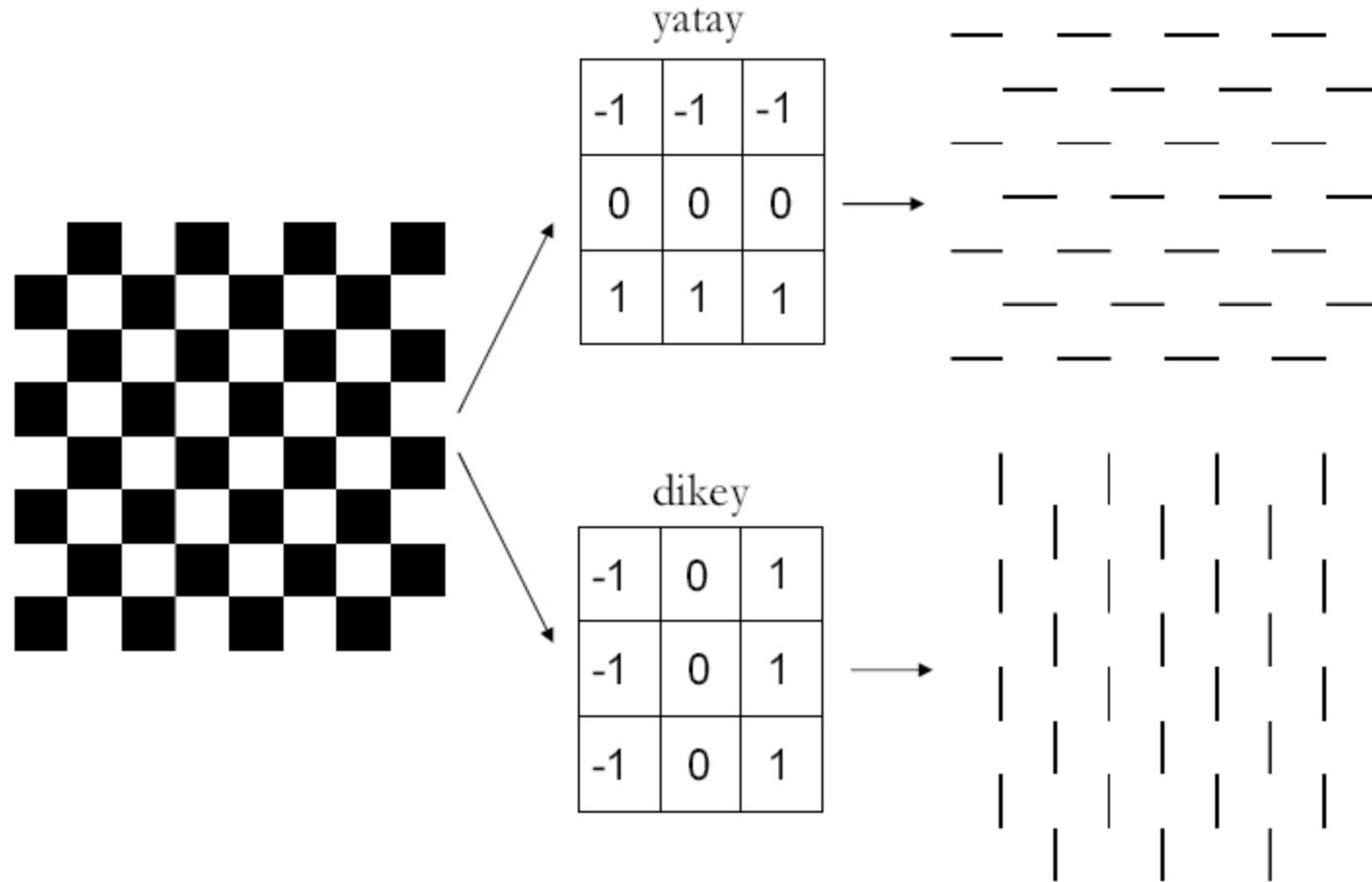
## İmge Pekiştirme : Kenar Bulma

**Kenar:** Gri-seviyesi veya renklerde meydana gelen ani değişimler olarak değerlendirilir. Kenar bulma işlemi imge işleme uygulamalarında sıkılıkla kullanılır.

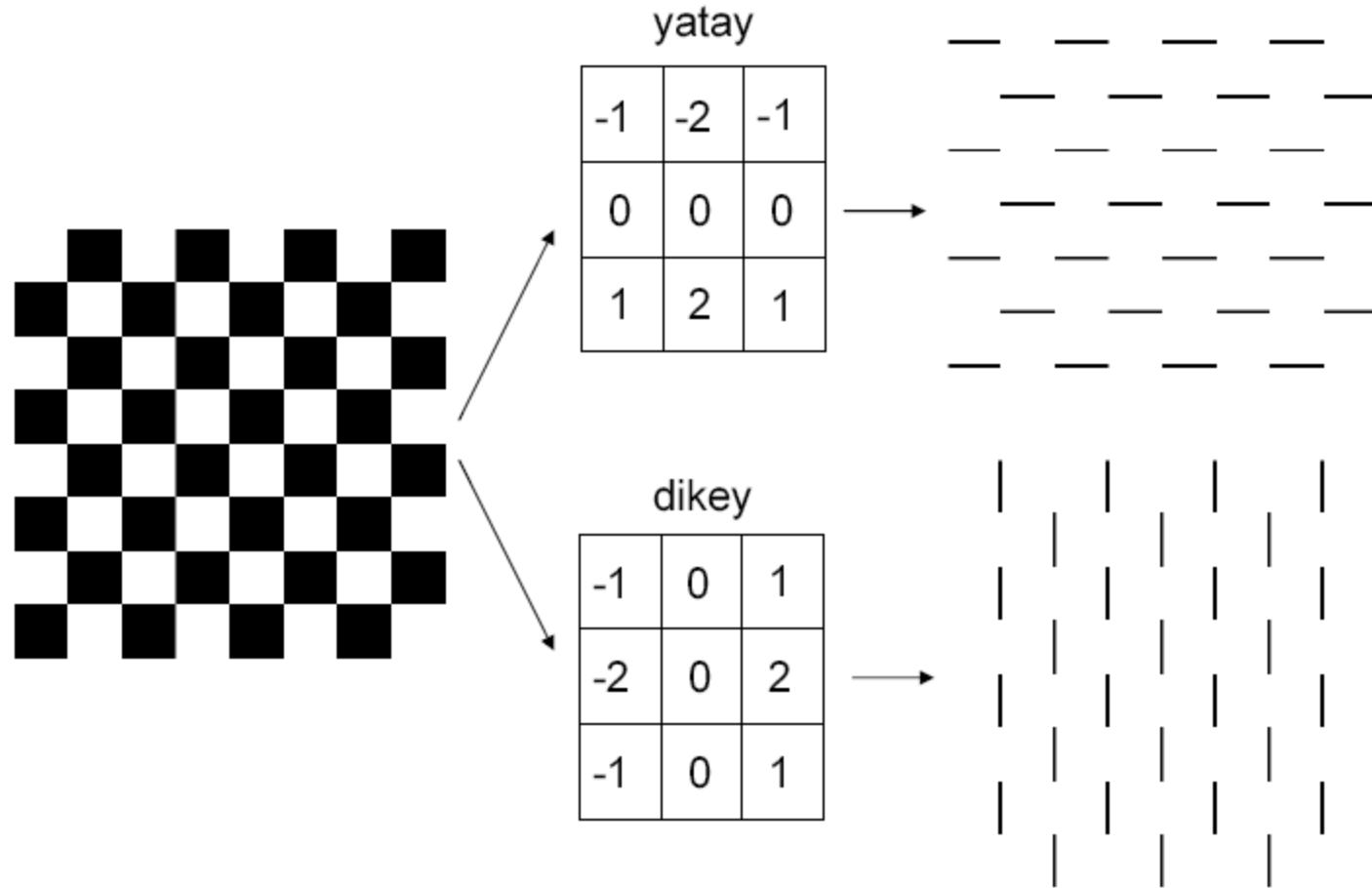
Kenar bulma işlemi 3 ana aşamada incelebilir:

- 1- Gürültü Azaltma :Anlamlı kenarları etkilemeden mümkün olduğu kadar gürültünün bastırılması
- 2- Kenar Pekiştirme: Kenarlarda daha yüksek tepki veren bir filtrenin kullanılması.
- 3- Kenar Tespiti :Filtre çıktılarından hangilerin anlamlı kenar hangilerin gürültü olduğuna karar verilmesi.

## Kenar Bulma: Prewitt Kernelleri



## Kenar Bulma: Sobel Kernelleri

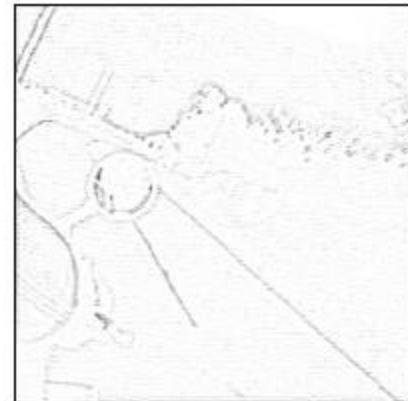
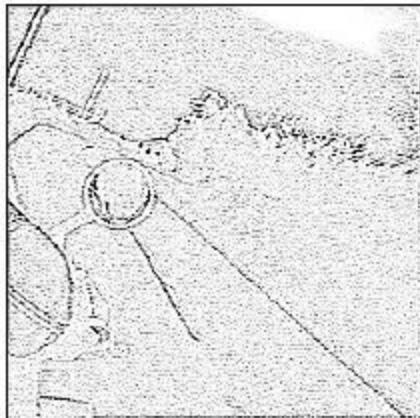


## Kenar Bulma



Kenar Bulma

AGS+Kenar Bulma



## Kenar Bulma: Matlab

Kenar bulma işlemleri konvolüsyon işleminin yukarıda anlatılan kerneller kullanılarak yapılabileceği gibi, Matlab'da bulunan **edge** komutu kullanılarak da yapılabilir.

**edge** komutu ile aşağıda verilen kenar bulma yöntemleri gerçekleştirilebilir:

- Prewitt
- Sobel
- Laplacian of Gaussian (LoG)
- Roberts
- Zero-cross
- Canny\* : Diğer yöntemlere göre genellikle daha iyi sonuç verir.

## İmge Pekiştirme : Korelasyon

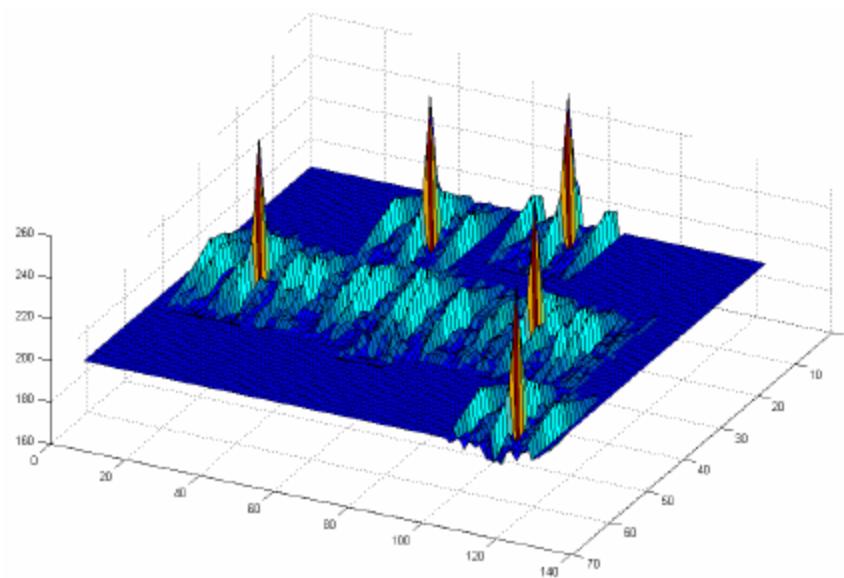
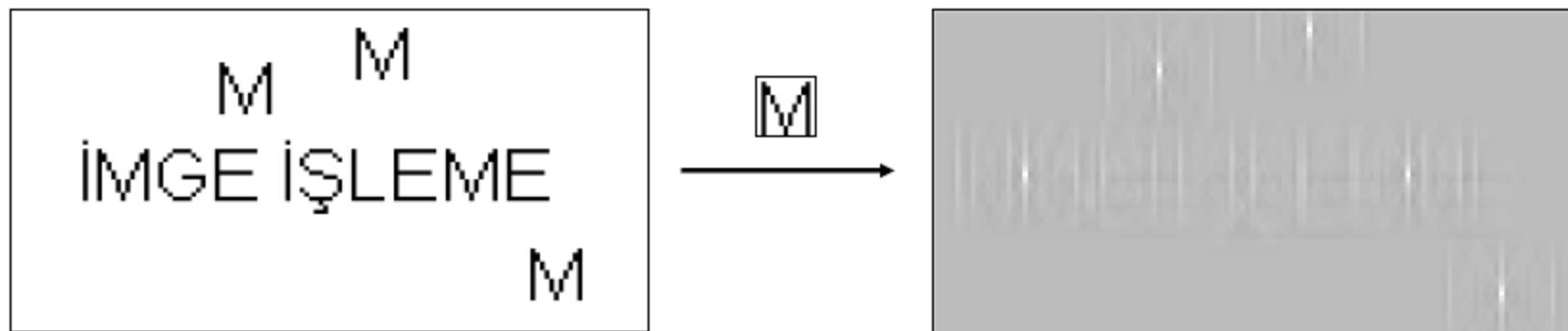
İki imge arasındaki ilişkinin bulunmasında ya da bir imgenin içerisinde önemli bir bilginin aranmasında kullanılır. Konvolüsyondan farklı olarak kerneldeki katsayılar doğrudan kendi konumlarına karşılık gelen piksel değerleri ile çarpılmaktadır.

$$g(x, y) = \sum_{k=-n}^n \sum_{j=-m}^m h(j, k) f(x+j, y+k)$$

Normalizasyon için:

$$g'(x, y) = \frac{\sum_{k=-n}^n \sum_{j=-m}^m h(j, k) f(x+j, y+k)}{\sum_{k=-n}^n \sum_{j=-m}^m f(x+j, y+k)}$$

## Korelasyon : Örnek Uygulama



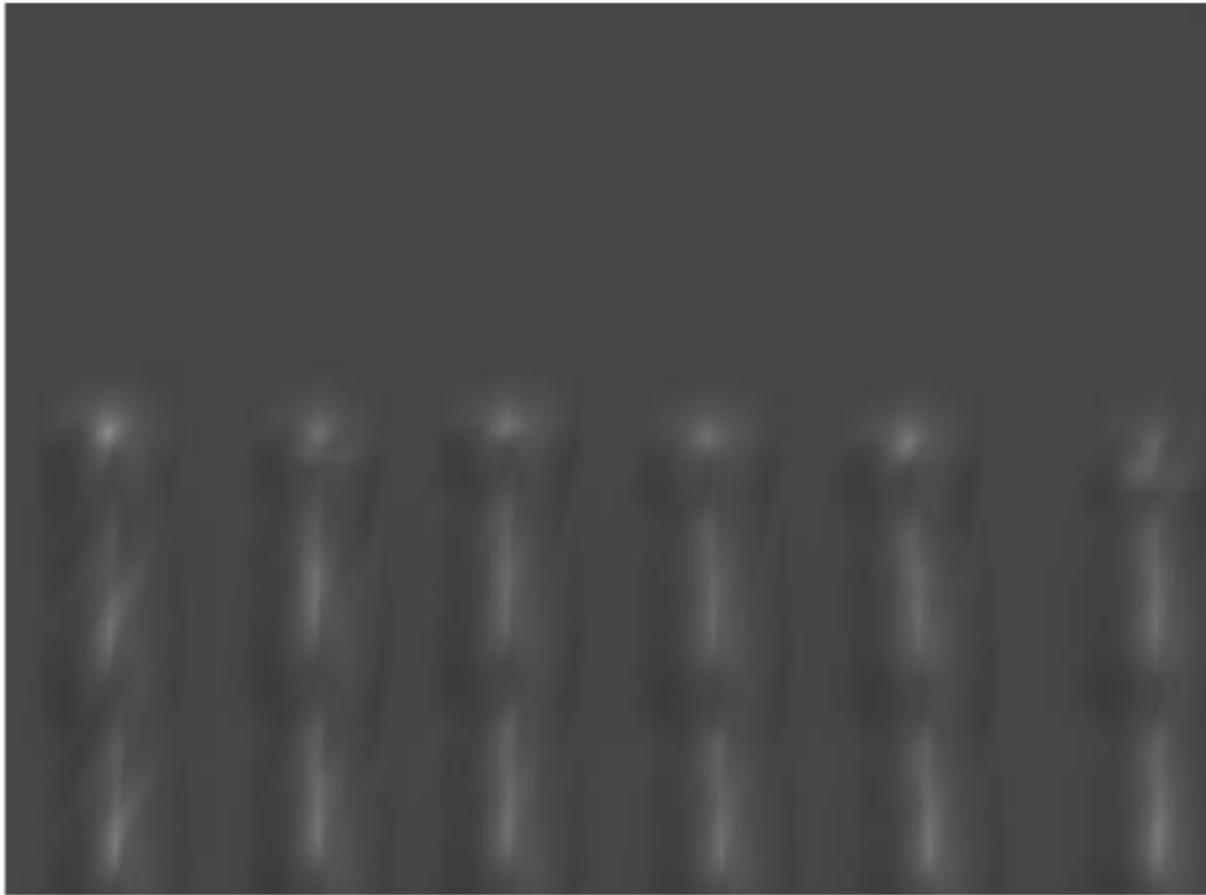
## Korelasyon: Örnek Uygulama



Sağlam  
matkap ucu

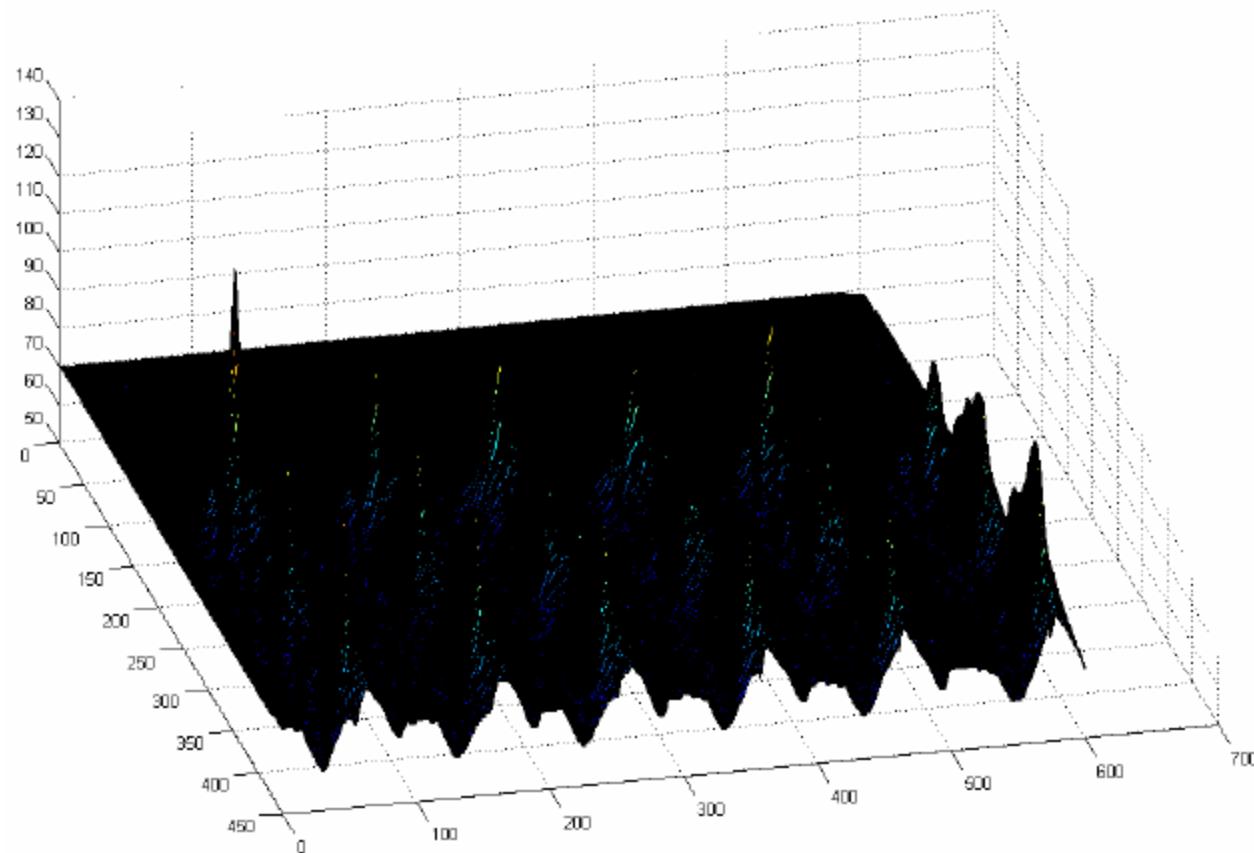


## Korelasyon: Örnek Uygulama



Elde edilen  
korelasyon  
imgesi

## Korelasyon: Örnek Uygulama



Elde edilen  
korelasyon  
yüzeyi

## İmge Pekiştirme : Ortanca Süzgeci (Median Filter)

Pikselin yeni ıshıklılık değeri herhangi bir matematiksel işlem yapılmadan, belirlenen çember (blok) içerisindeki ıshıklılık değerlerinin ortanca değeri olarak seçilir. Aşağıdaki örnekte 3x3 bloklar kullanılarak yapılan bir ortanca süzgeci işlemi görülmektedir.

25	28	34	x	x
45	41	56	x	x
38	46	29	x	x
x	x	x	x	x
x	x	x	x	x

25 28 29 34 38 41 45 46 56



Yeni piksel değeri

## İmge Pekiştirme : Ortanca Süzgeci

