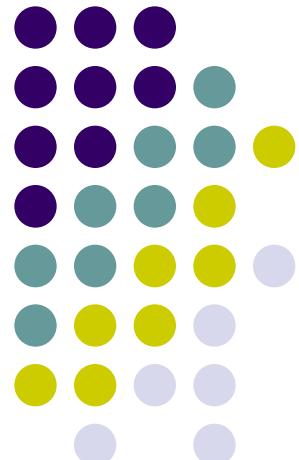


İMGE İŞLEME



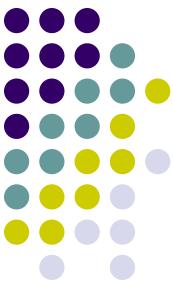
KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İşaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)



Değerlendirme

- Ara dönem notu: %20 ödev, %80 sınav
- Final notu: %30 sınav, %70 dönem ödevi
- Geçme Notu: %40 ara dönem + %60 final



İçerik

- Temel Kavramlar
- Dosya Tipleri ve Temel İşlemler
- İmge Pekiştirme
- İmge Nicemleme (Lloyd Max)
- Piksel Komşuluk İşlemleri
- İmge Bölütleme
- Renkli İmge İşleme
- Morfolojik İşlemler
- Frekans Uzayı İşlemleri
- İmge Sıkıştırma



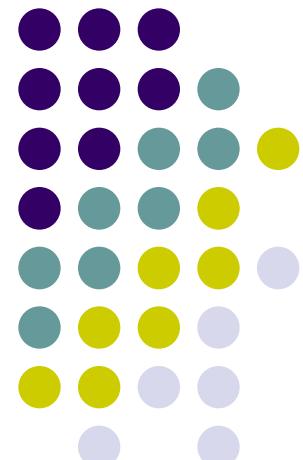
Kaynaklar

- Digital Image Processing (2nd Edition)
Rafael C. Gonzalez, Richard E. Woods , 2002
- Digital Image Processing Using MATLAB(R)
Rafael C. Gonzalez , 2003
- Image Processing Handbook The: Second Edition
John C. Russ , 2006

iMGE İŞLEME

Ders-1

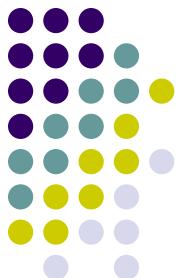
Temel Kavramlar



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İşaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

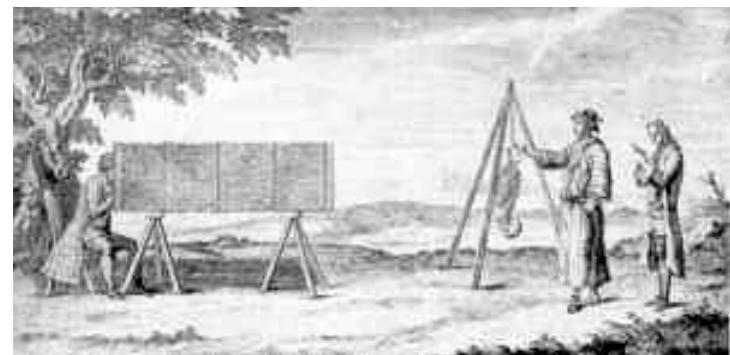
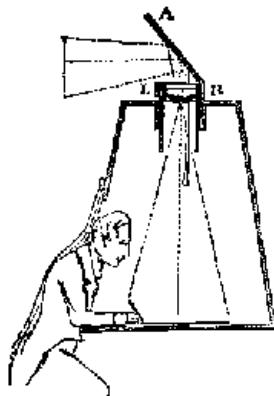
Camera Obscura: A Pinhole Camera



- History
 - Chinese philosopher Mo-Ti (5th century BC) was the earliest to report such a device
 - Aristotle (384-322 BC) understood the optical principle of pinhole projection
 - 11th century, Al-Haytham wrote a book on optics
 - 1490 Leonardo Da Vinci gave a well defined description
 - Lenses were used in 16th century allowing more light
 - Common use as a drawing tool in 17th century



Aristotle's pinhole camera



Drawing tool used for recording human anatomy



İlk Fotoğraf

- Inspired by the newly-invented art of lithography (a printing technique), after several years of work Joseph Nicéphore Niépce succeeded in recording an image captured by a camera obscura (1826)





Fotoğrafın tarihçesi

- The word photography was first used in the year 1839, "the year the invention of the photographic process was made public".
- Eastman Kodak establishes his company (at age 24) in 1880. After roll film is introduced in 1889 Photographic process becomes widely-used.
- Louis Lumiere invents the first motion picture camera (Cinematographe) in 1895
- 1936: development of Kodachrome, the first color multi-layered color film.
- In 1971 C-41 color negative process introduced
- 199X- Digital age

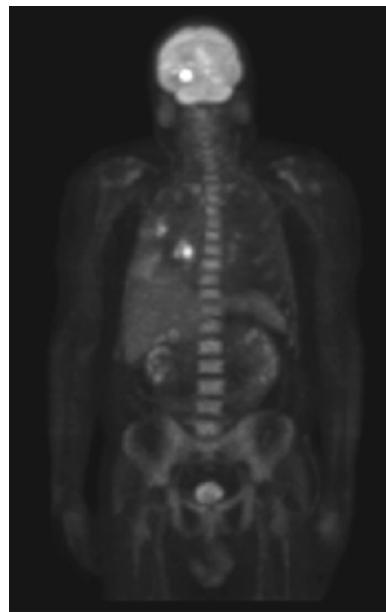
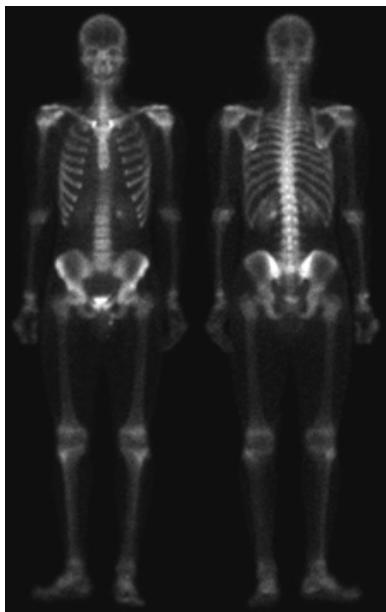


- Sayısal İmge İşleme (Digital Image Processing):
İmgeyi iyileştirmek, dönüştürmek ya da bilgi
çıkartmak
- Bilgisayarla Görü (Computer Vision):
İmgeleri işleyerek gerçek dünya ile ilgili bilgi
çıkartmak.

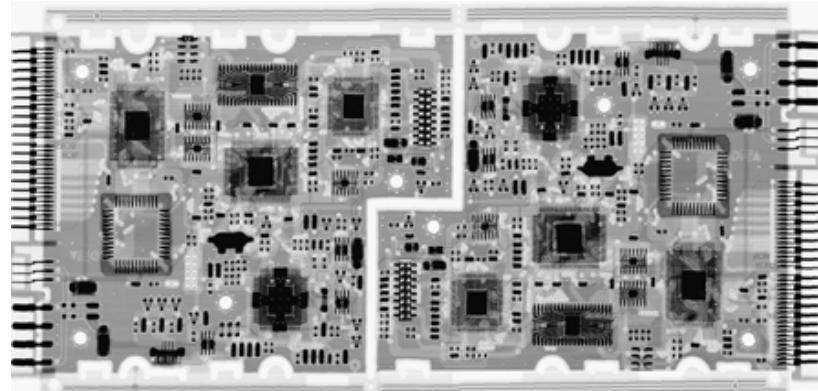
Uygulama Alanları



Gama ışını görüntüleme:



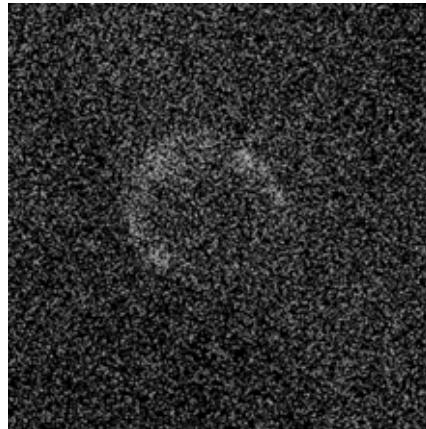
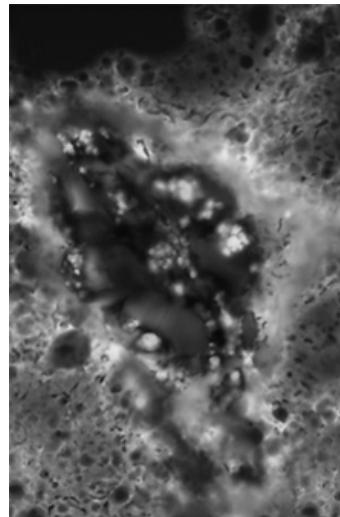
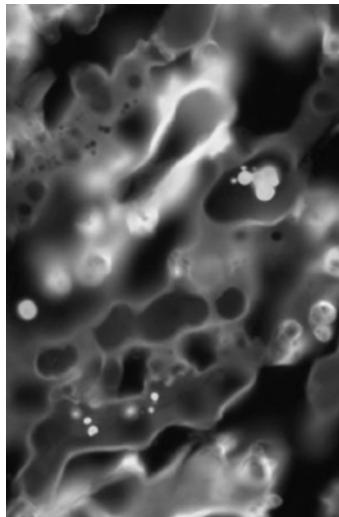
X ışını görüntüleme:



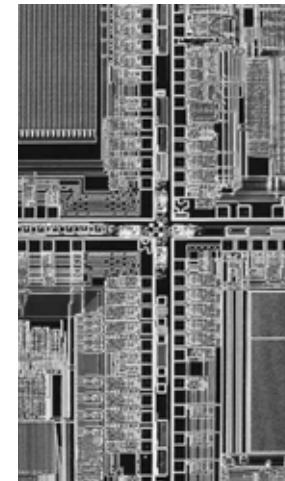
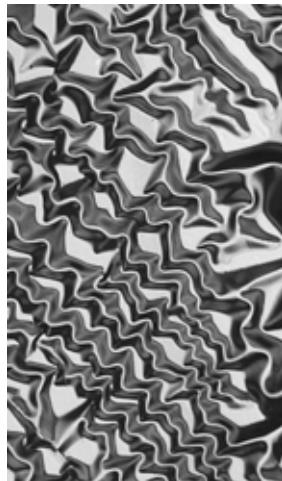
Uygulama Alanları



Mor ötesi bandı görüntüleme:



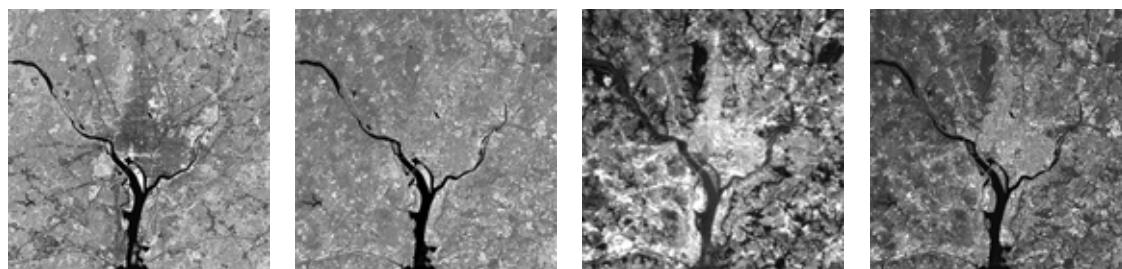
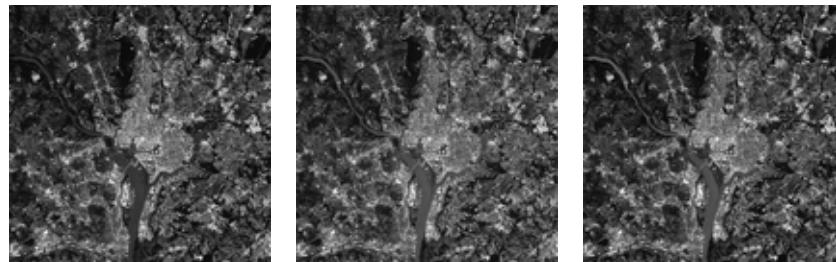
Görünür ve kızılberisi görüntüleme:



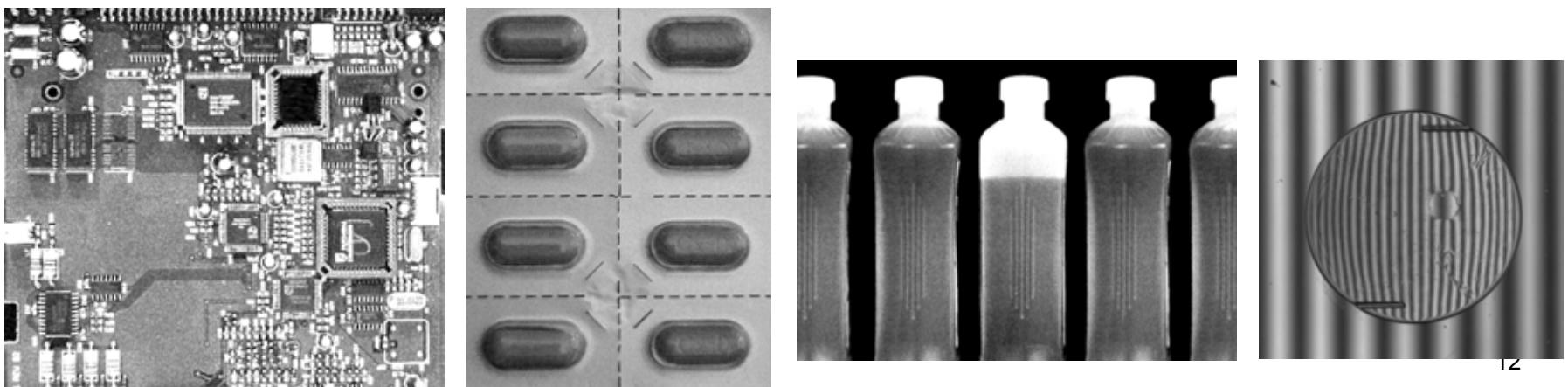
Uygulama Alanları



Çok bantlı görüntüleme:



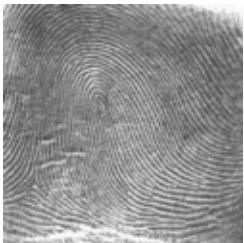
Kalite kontrol:



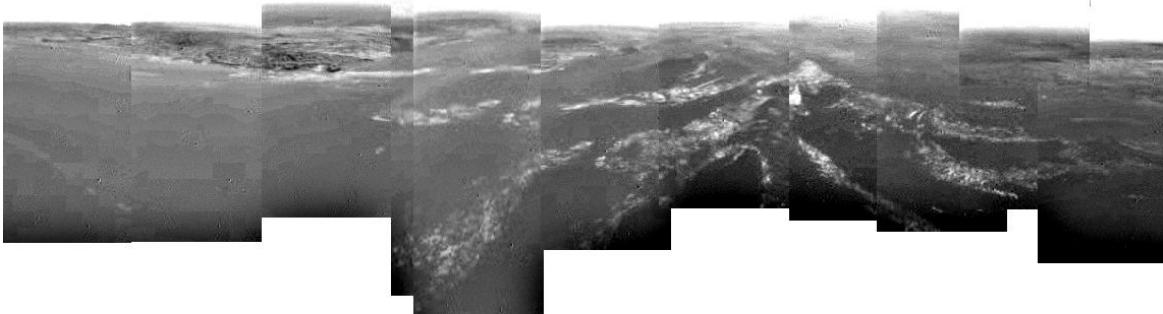
Uygulama Alanları



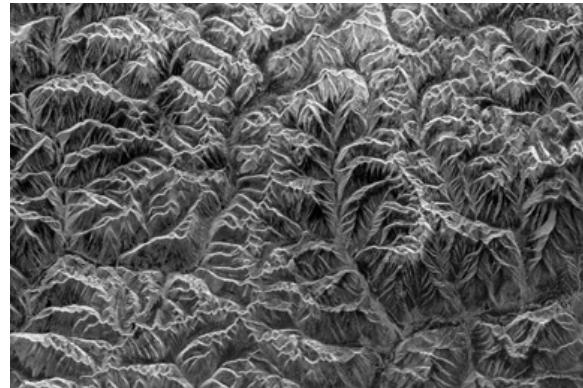
Örütü tanıma:



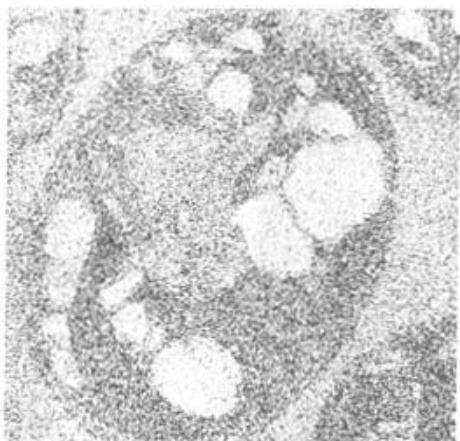
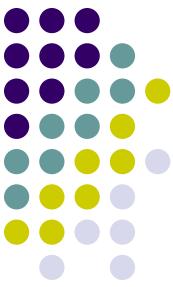
Radar görüntüleme:



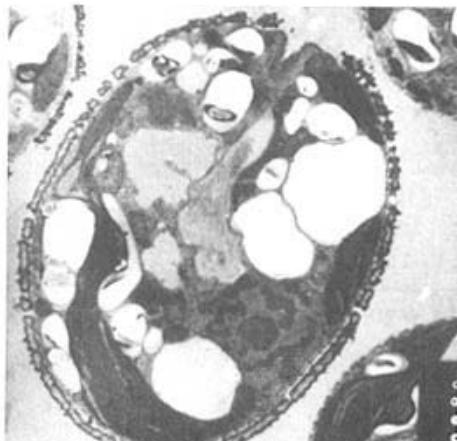
3/16/2012



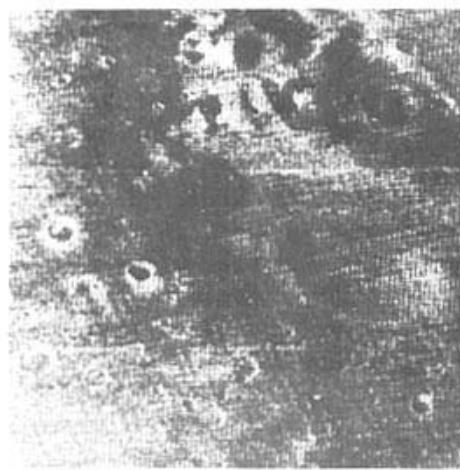
13



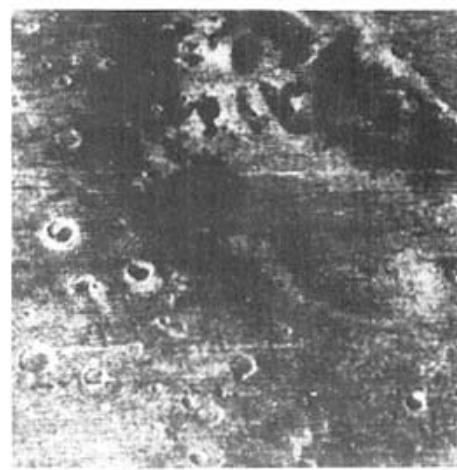
(a)



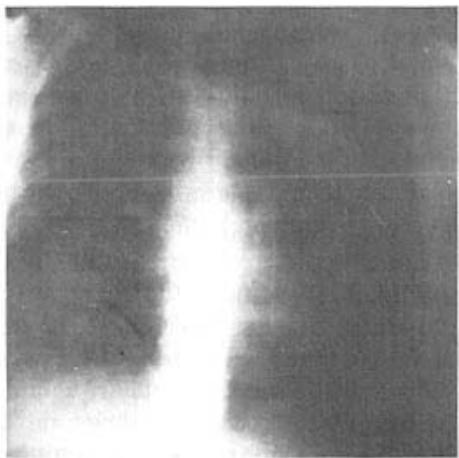
(b)



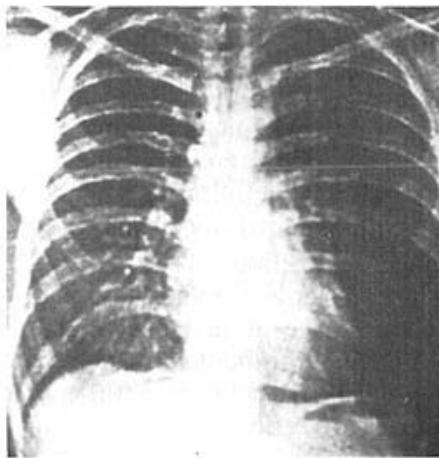
(c)



(d)



(e)



(f)



(g)

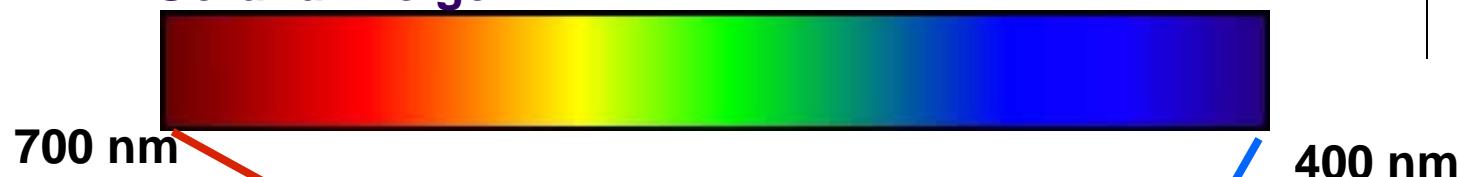


(h)

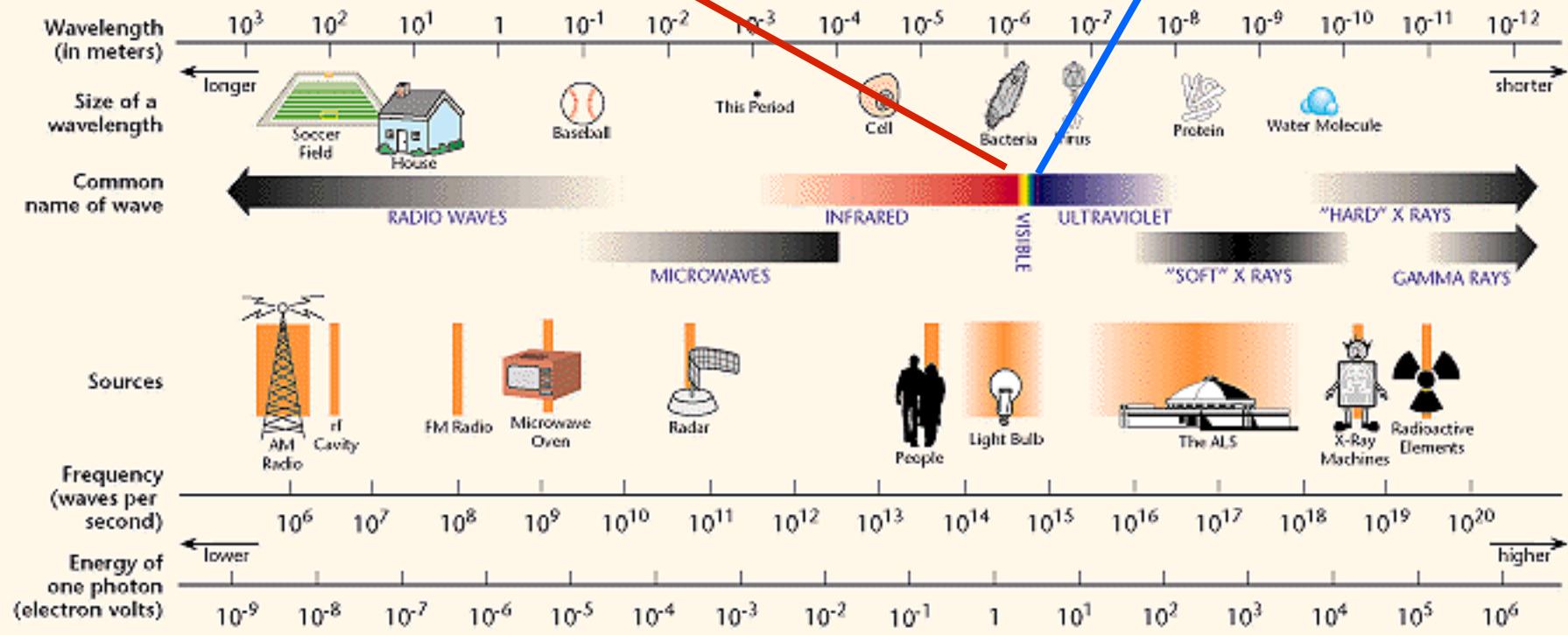
Elektromanyetik Spektrum



Görünür Bölge



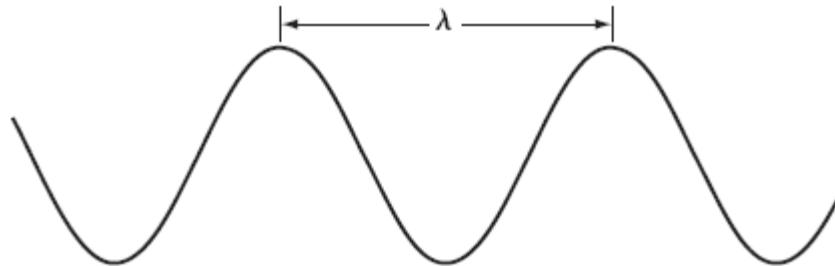
THE ELECTROMAGNETIC SPECTRUM



İşık ve Elektromanyetik Spektrum



- Cam prizmadan geçirilen güneş ışığının sürekli renk spektrumu oluşturduğunu kim bulmuştur?
(Isaac Newton-1666)



Dalgaboyu:

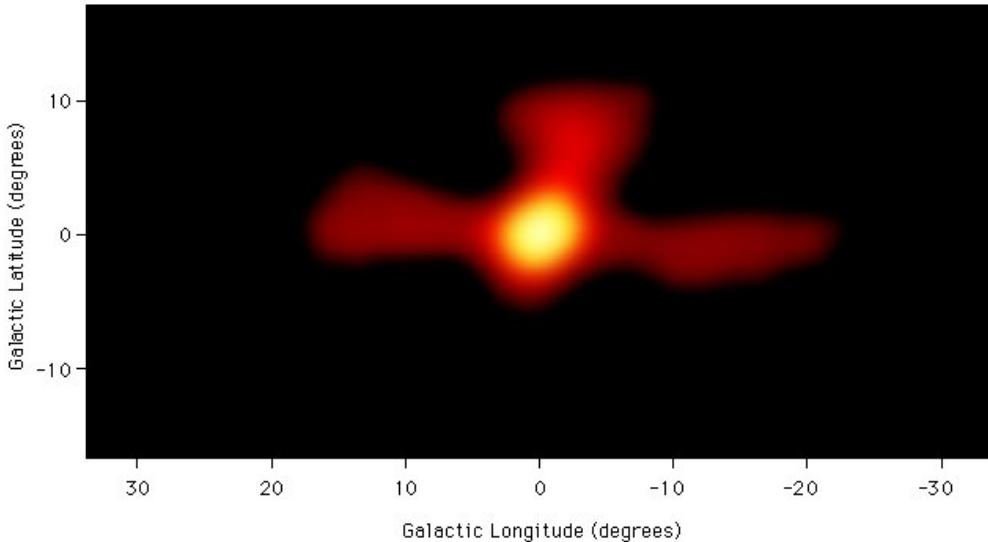
$$\lambda = \frac{c}{f}, c = 2.998 \times 10^8 \text{ m/s}$$

Enerji:

$$E = hv, h : \text{plank sabiti}$$

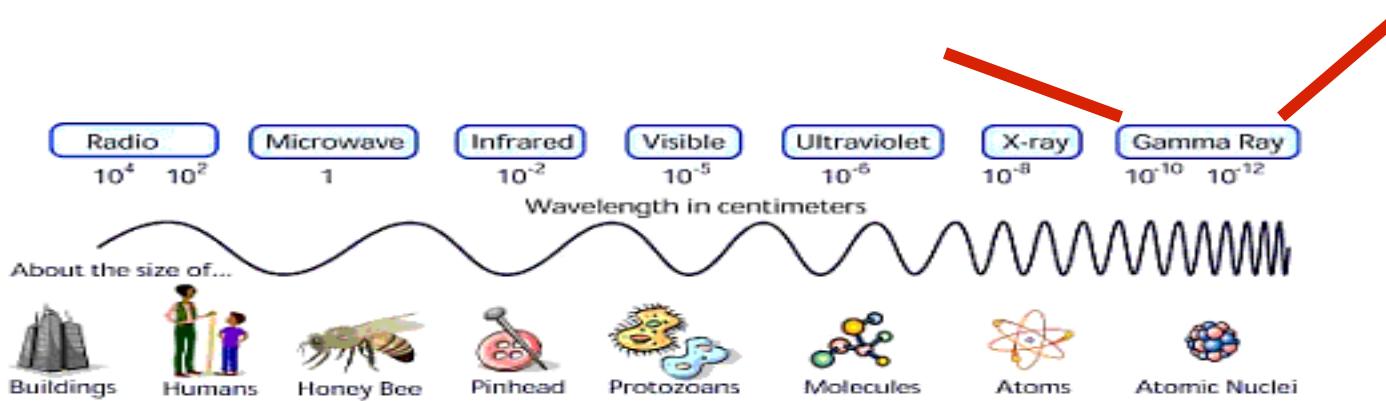
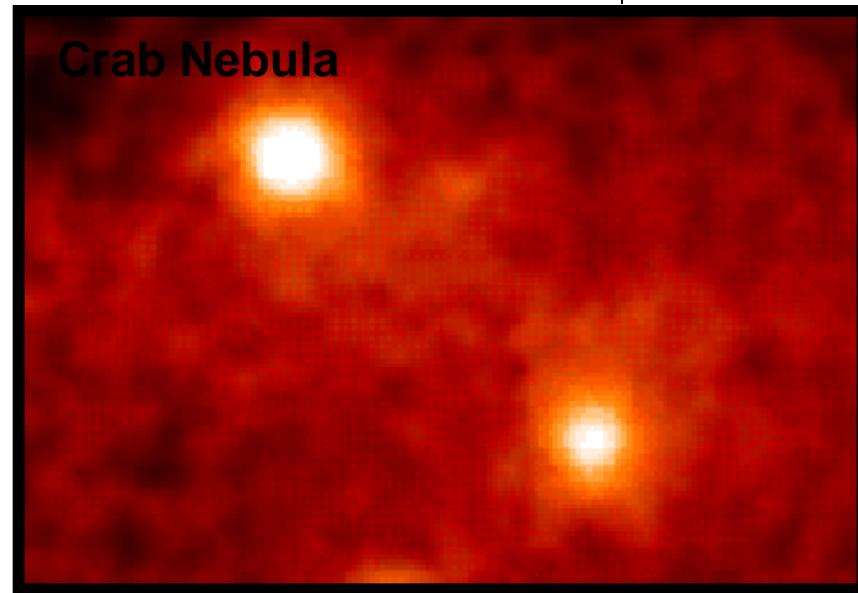


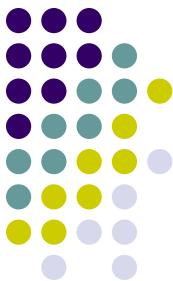
Across the EM Spectrum



Caption: Map of the distribution of positrons towards the center of the Milky Way Galaxy, including the newly discovered antimatter "cloud". The brightest feature corresponds to the nucleus of the Galaxy. The horizontal structure lies along the plane of the Galaxy. The antimatter "cloud" is located above the Galactic center.

Courtesy of D. D. Dixon (University of California, Riverside) and W. R. Purcell (Northwestern University)

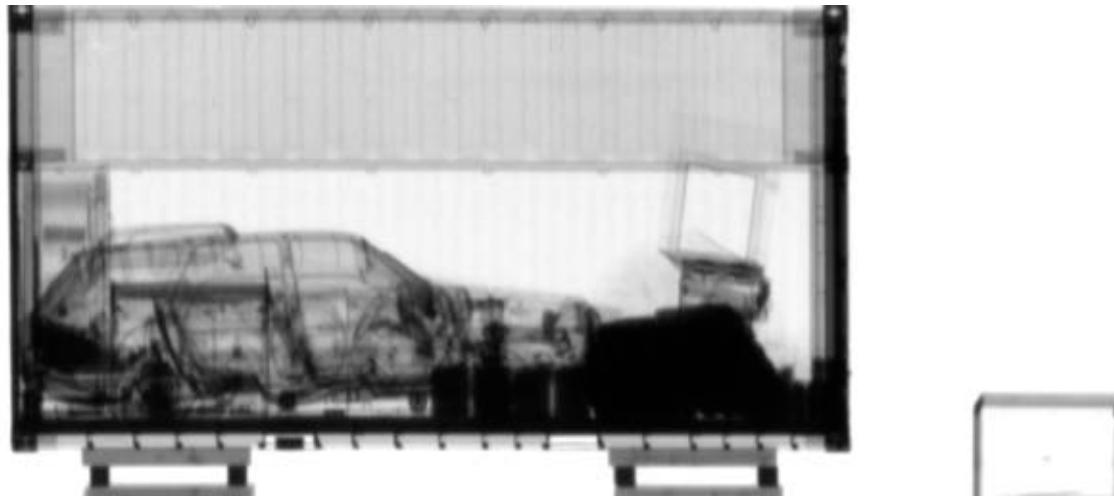




Across the EM Spectrum

Cargo inspection using Gamma Rays

Mobile Vehicle and Cargo Inspection System (VACIS®)



Gamma rays are typically waves of frequencies greater than 10^{19} Hz

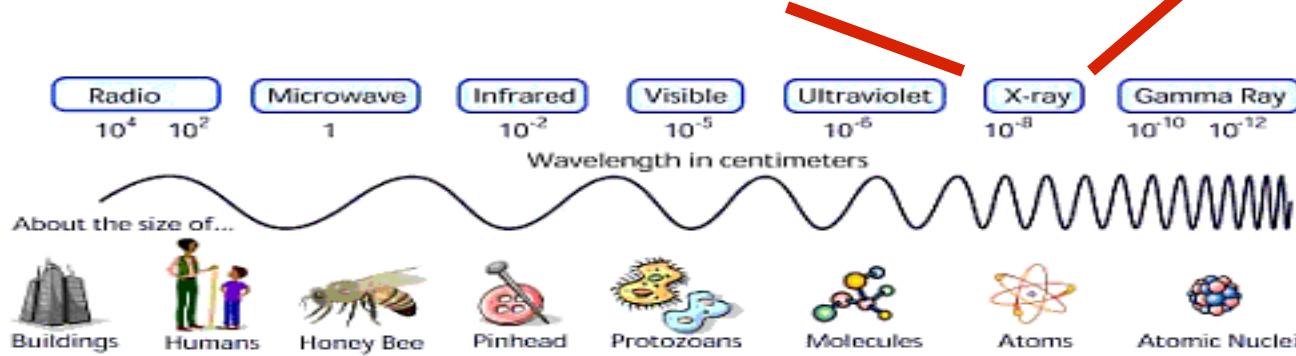
Gamma rays can penetrate nearly all materials and are therefore difficult to detect

Courtesy: Science Applications International Corporation (SAIC)¹⁹

Across the EM Spectrum



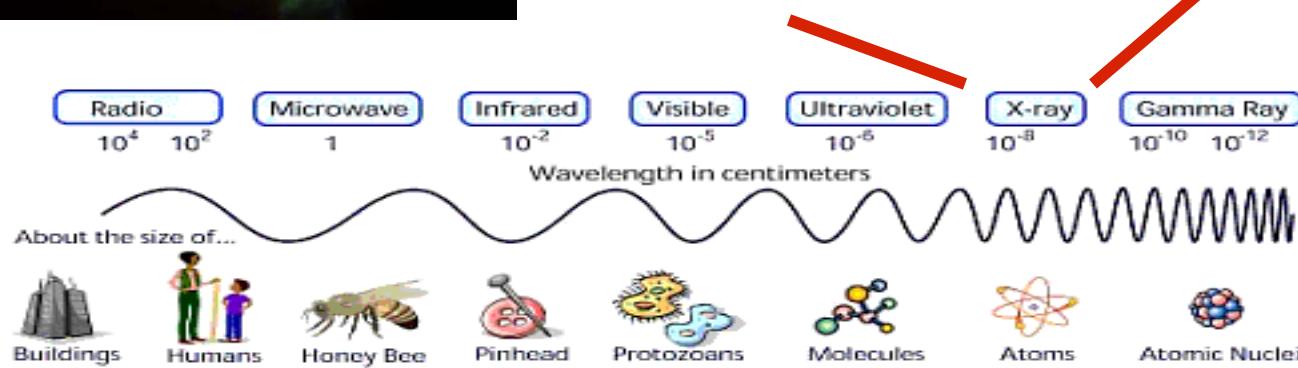
- Medical X-Rays





Across the EM Spectrum

- Chandra X-Ray Satellite

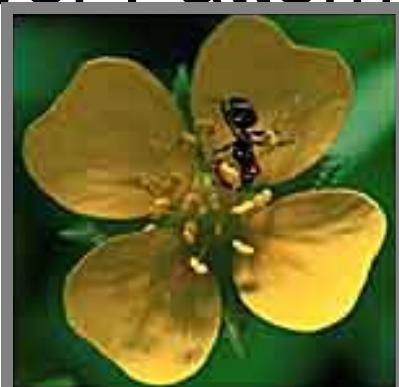




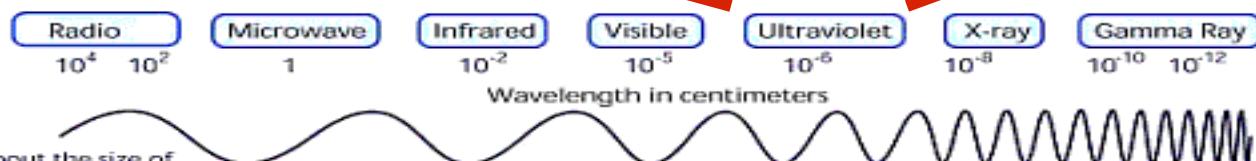
Across the EM Spectrum

- Flower Patterns in Ultraviolet

Potentilla



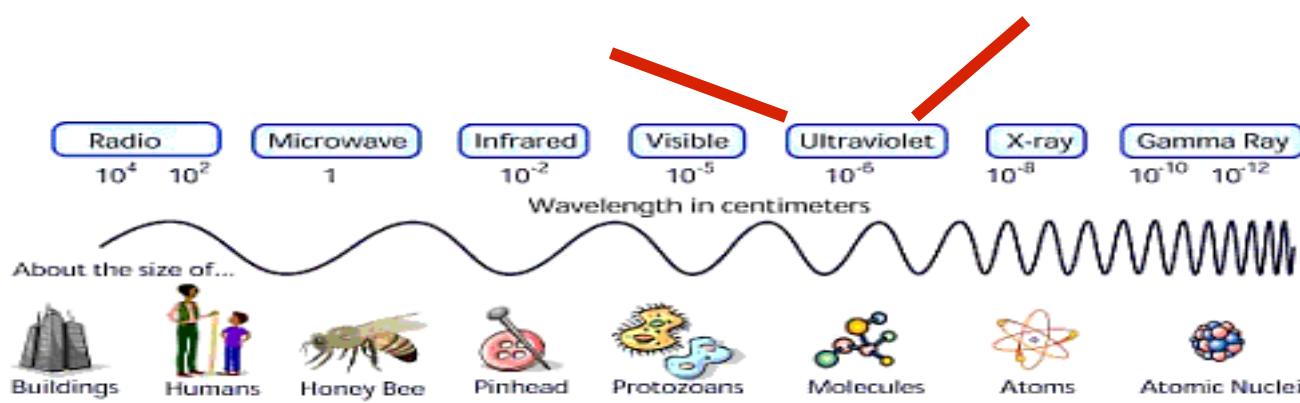
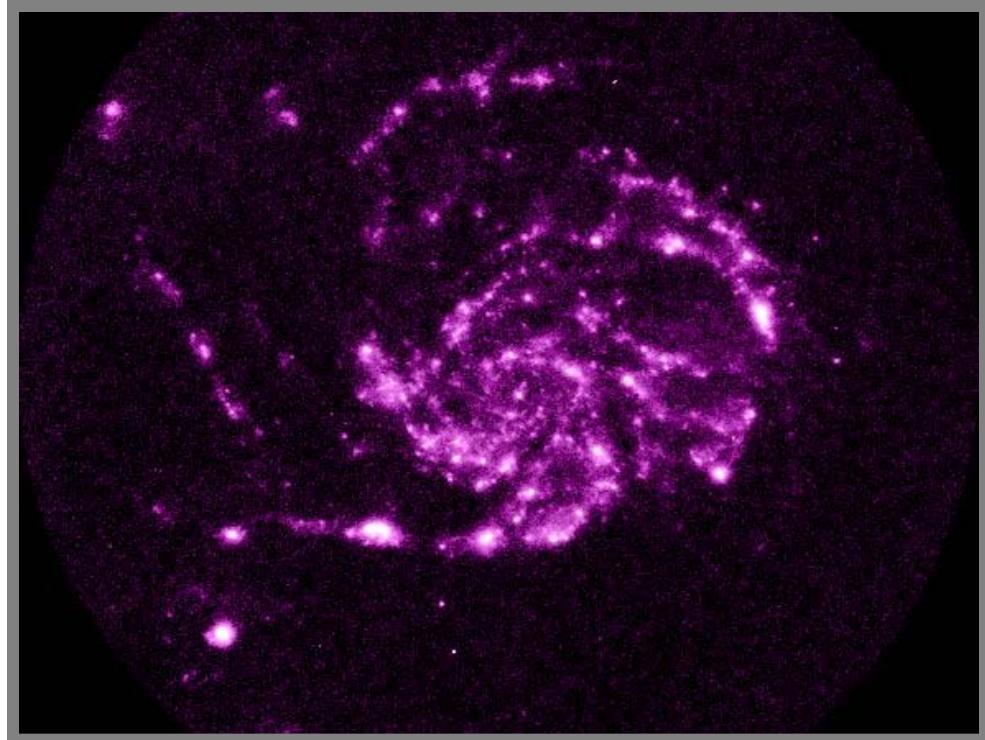
Dandelion - UV





Across the EM Spectrum

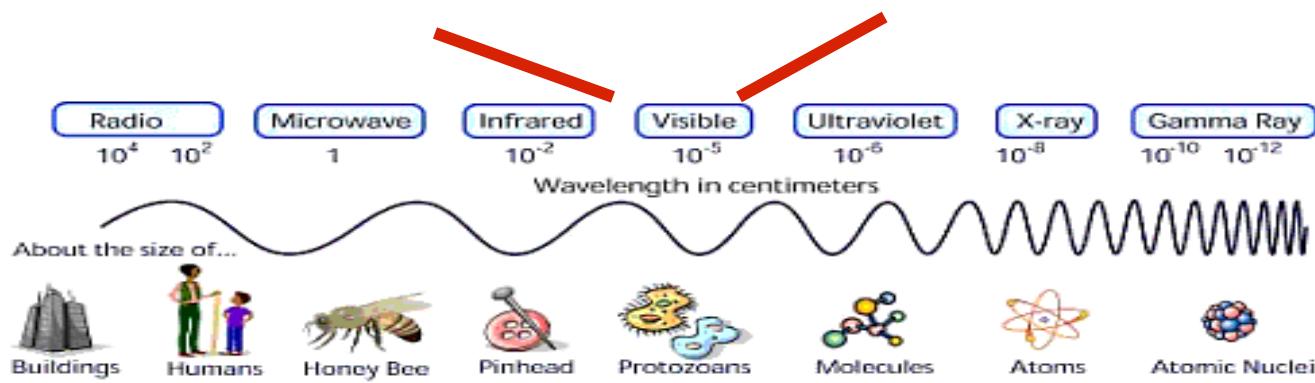
- Messier 101 in Ultraviolet





Across the EM Spectrum

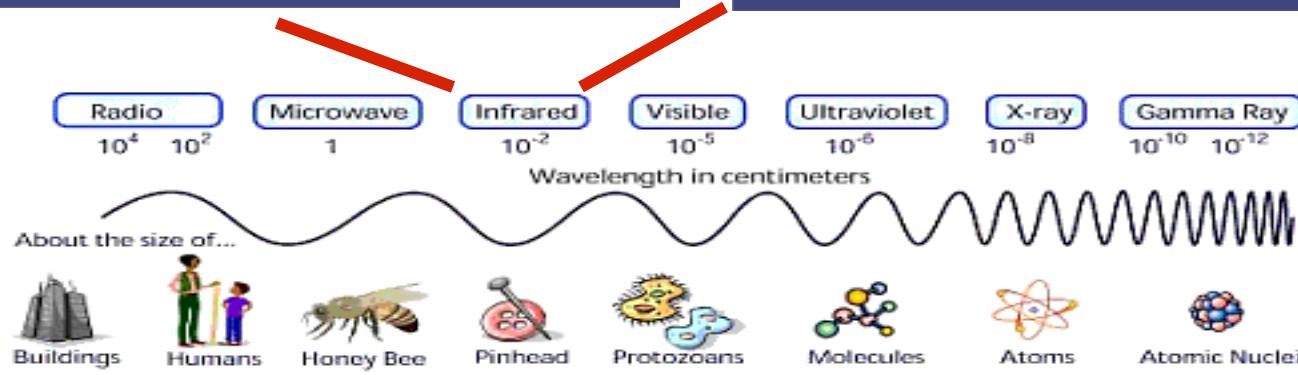
- Traditional images





Across the EM Spectrum

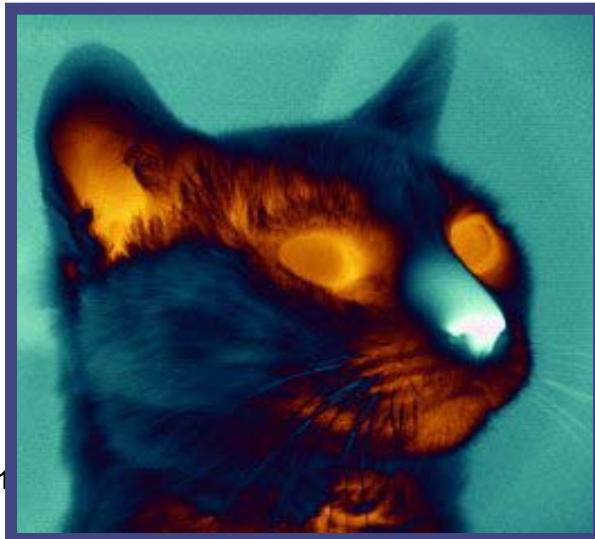
- IR: **Near**, Medium, Far (~heat)





Across the EM Spectrum

- IR: Near, Medium, Far (~heat)





Across the EM Spectrum

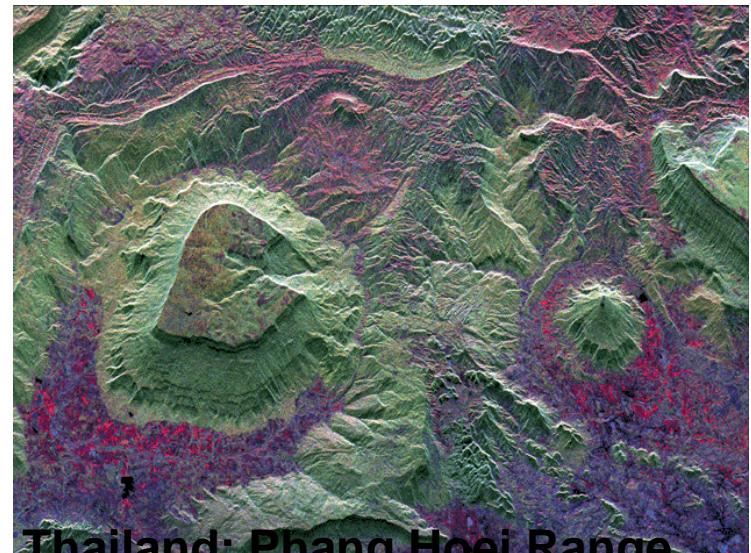
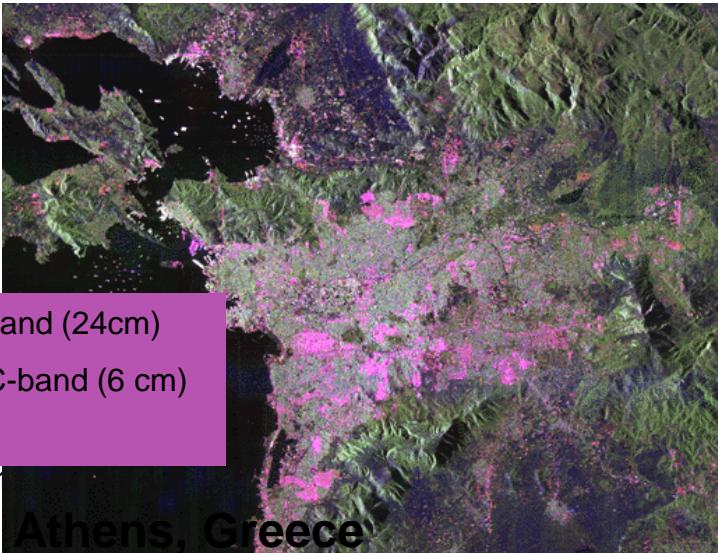
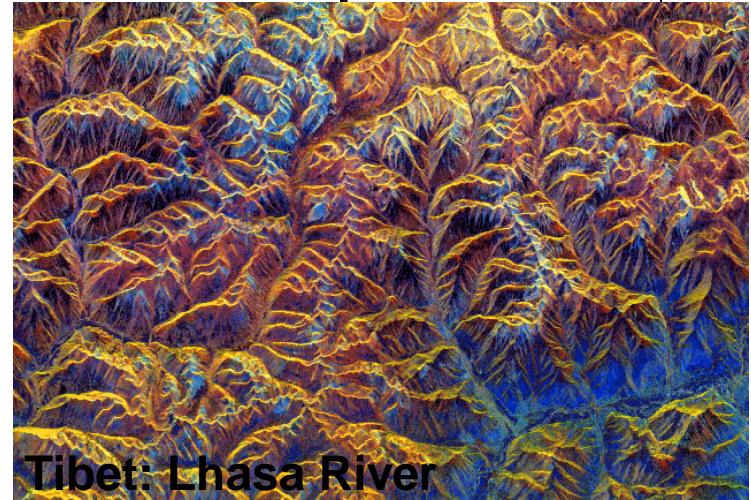
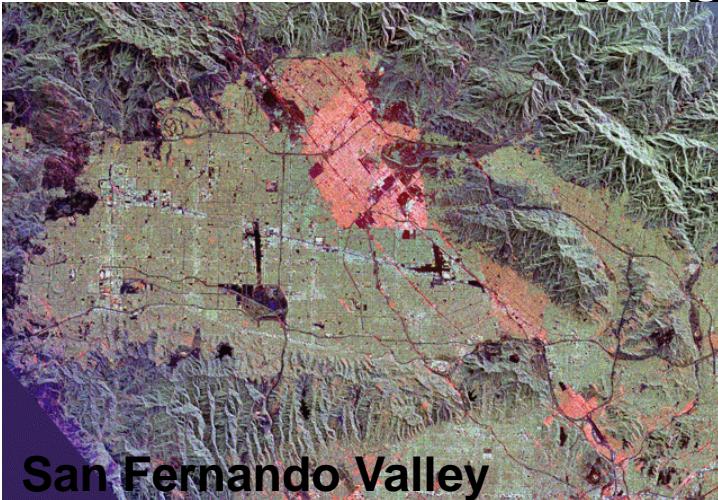
- IR: Finding chlorophyll -the green coloring matter of plants that functions in photosynthesis





Across the EM Spectrum

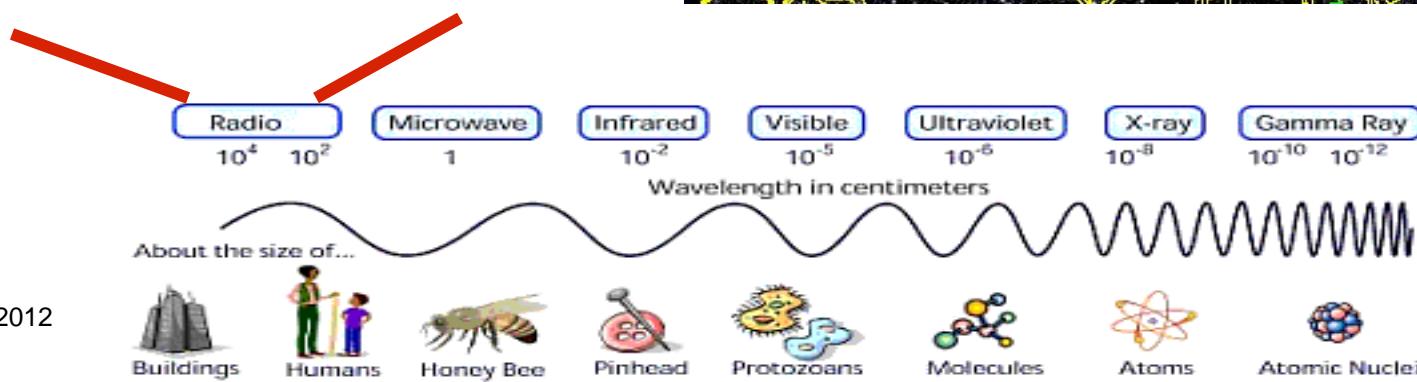
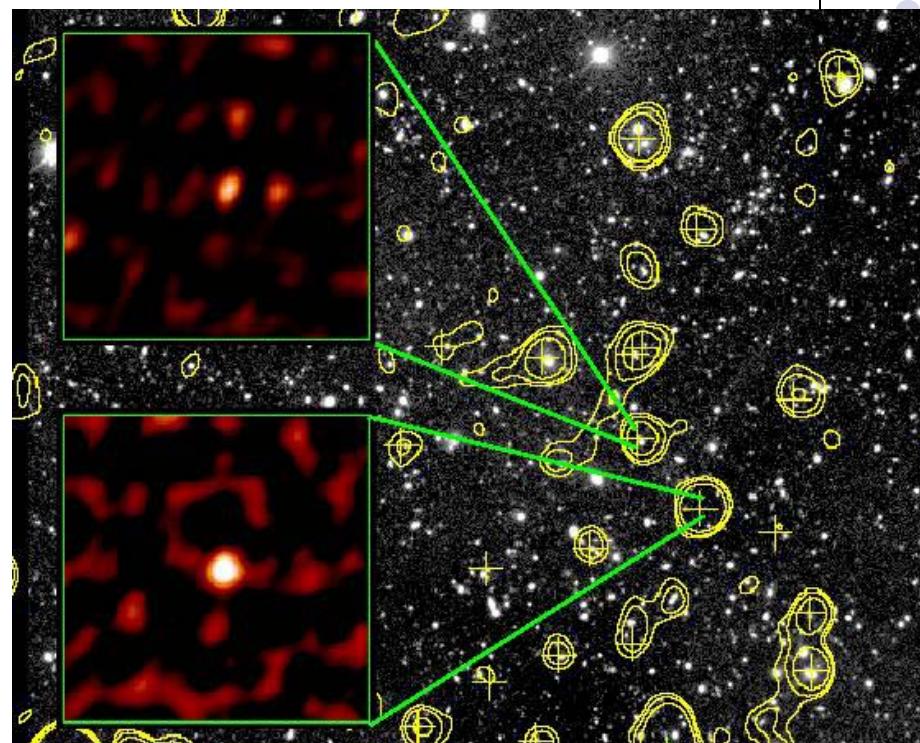
- Microwave Imaging: Synthetic Aperture





Across the EM Spectrum

- Radio Waves
(images of cosmos from radio telescopes)



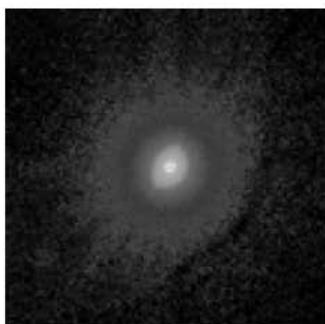


Dalgaboyları

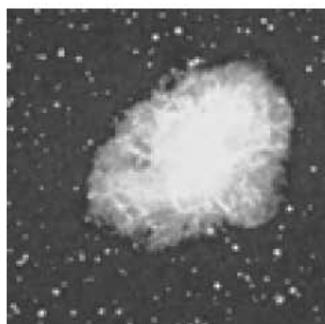
Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.52–0.60	Good for measuring plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.76–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Middle infrared	2.08–2.35	Mineral mapping



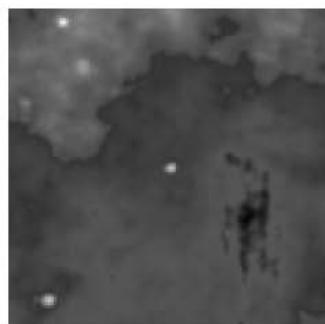
Gamma



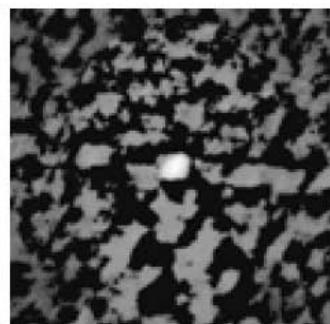
X-ray



Optical



Infrared



Radio

Yengeç gök cismi



İmge İşleme ve İlgili Alanlar

- Image Processing: image to image
- Computer Vision: Image to model
- Computer Graphics: model to image

***All three are
interrelated!***

- Pattern Recognition: image to class
 - image data mining/ video mining
- Artificial Intelligence: machine smarts

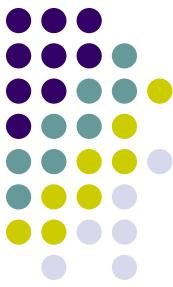
AI

- Photogrammetry: camera geometry, 3D reconstruction
- Medical Imaging: CAT, MRI, 3D reconstruction (2nd meaning)
- Video Coding: encoding/decoding, compression, transmission

Applications

- Physics: basics
- Mathematics: basics
- Computer Science: programming skills

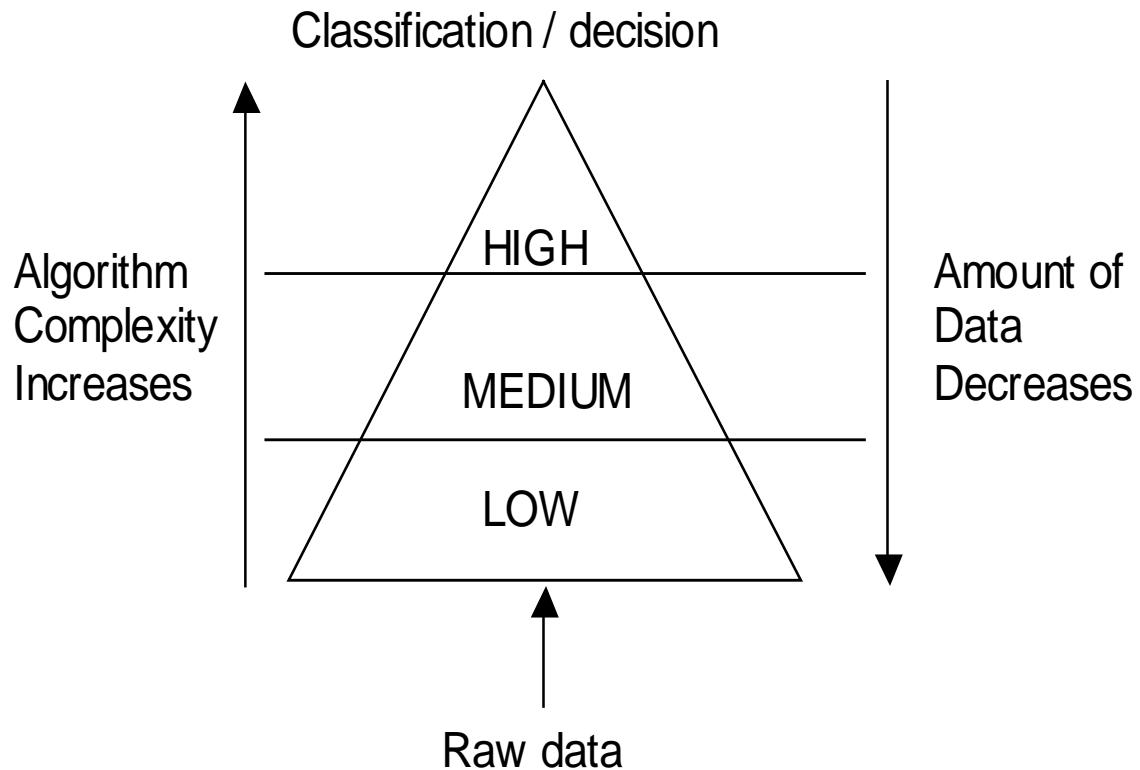
Fundamentals



Uygulama Alanları

- Tıp ve biyoloji (x-ışınları, biyomedikal görüntüler...)
- Coğrafi bilimler (hava ve uydu görüntülerinden hava tahmini)
- Eski, hasar görmüş fotoğrafların onarılması, GPR-mayın tarama, arkeolojik kalıntıların tespiti
- Oyun Programlama (bilgisayarda görü, 3-B modelleme)
- Fizik (spektrometreler, elektron mikroskopu görüntüleri)
- Uzay bilimleri (uydu, mikrodalga radar görüntüleri...)
- Savunma sanayi (gece görüş, akıllı roket sistemleri...)
- Endüstriyel uygulamalar (sureç, ürün denetimi...)
- Tüketiciler elektroniği (Video kayıt cihazları, cep telefonları...)
- Biyometrik tanıma ve güvenlik sistemleri (iris-parmak izi tanıma, güvenlik-kamera uygulamaları)
- Uzaktan algılama

Different Layers of Image Processing

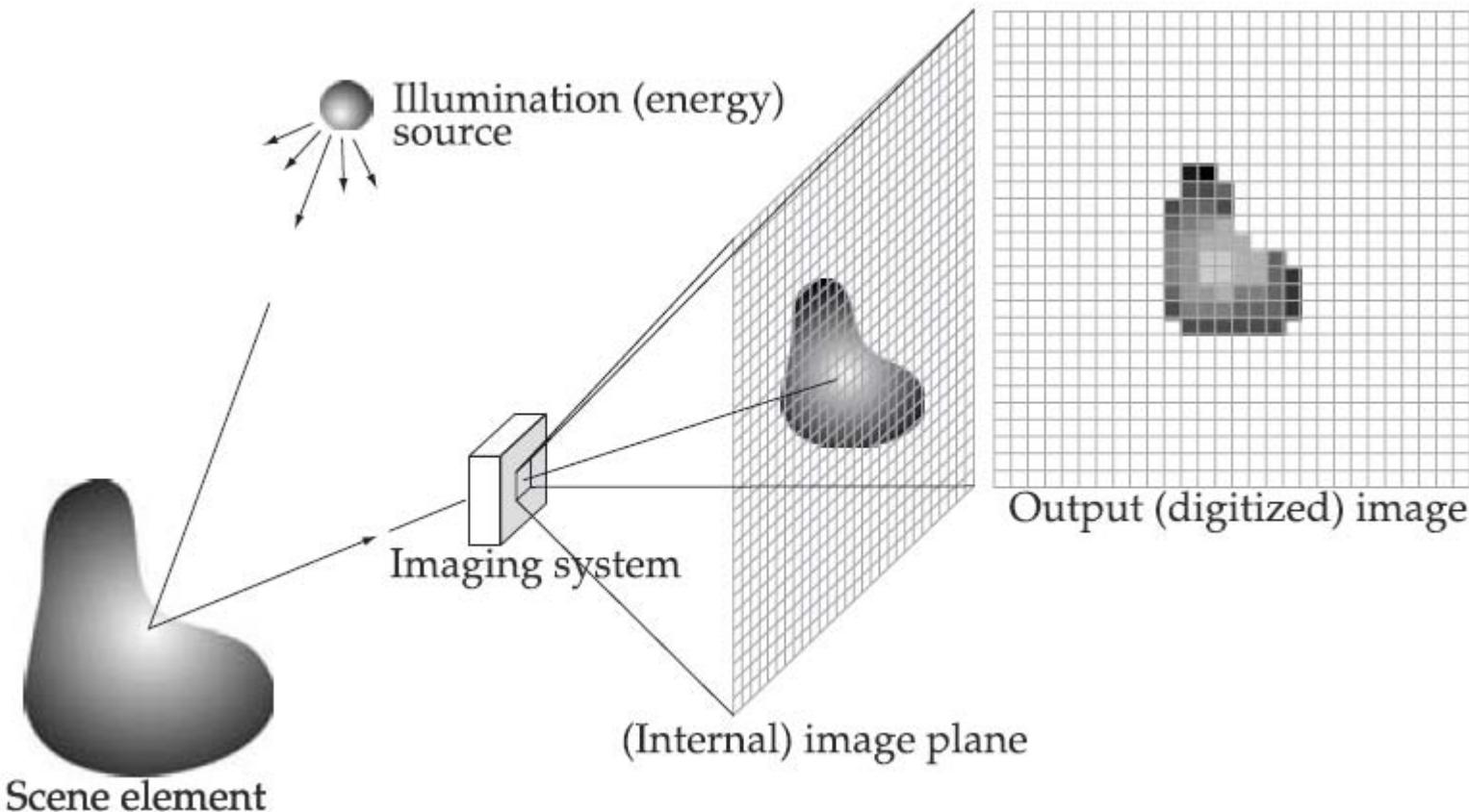


- Acquisition, preprocessing
 - no intelligence
- Extraction, edge joining
- Recognition, interpretation
 - intelligent

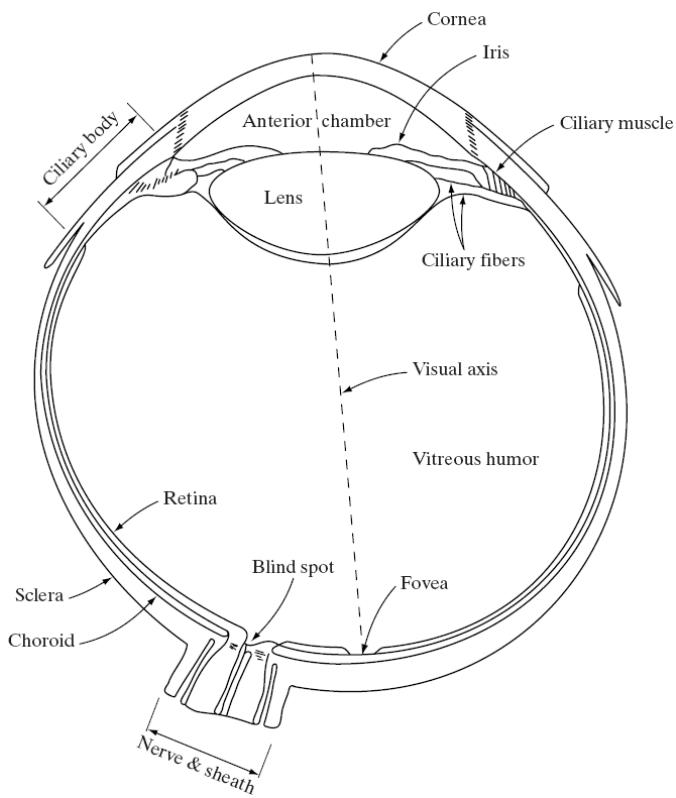
İmge Kavramı



- Sözlük anlamı: Işık etkisi veya optik mercek ile meydana gelen suret, hayal ($3B \rightarrow 2B$)

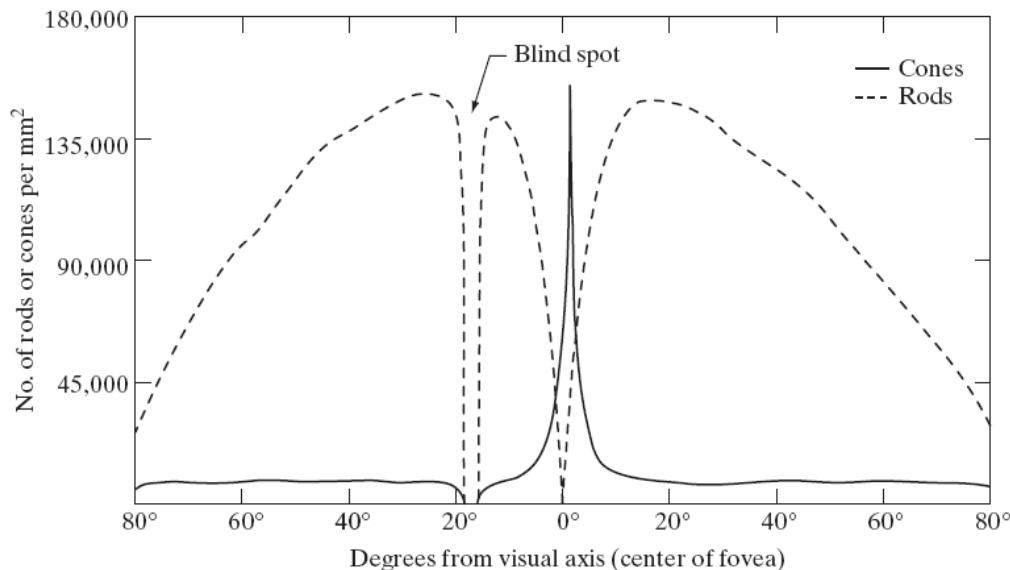


Görsel Algı Elemanları



Gözün retina bölgesinde iki tip algılayıcı vardır:

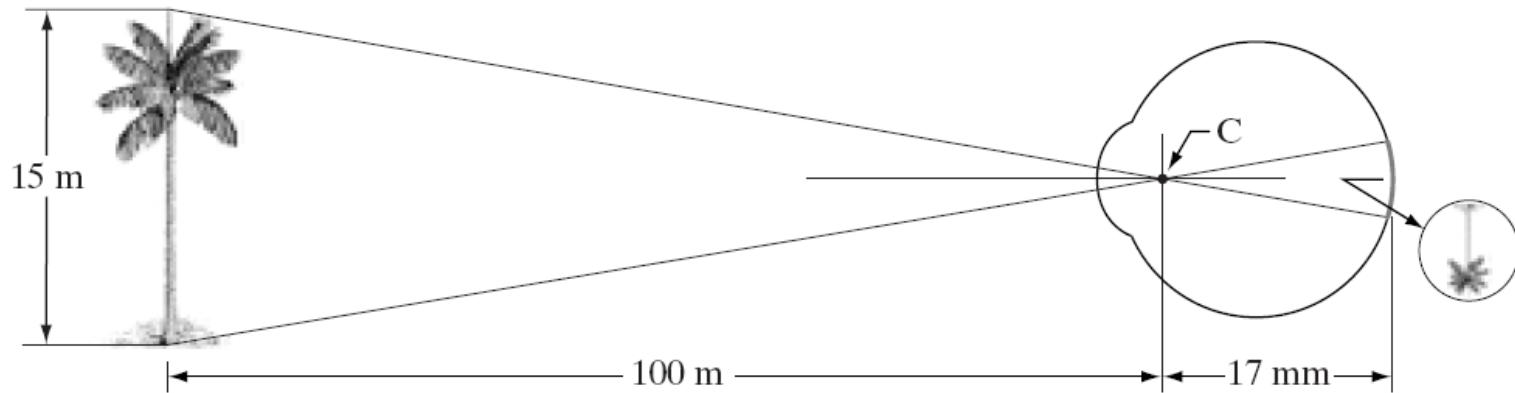
- Koni (6-7 milyon), renge duyarlı.
- Çubuk (75-150 milyon), ışığa duyarlı.



Görsel Algı Elemanları



- İmgenin gözde oluşumu:

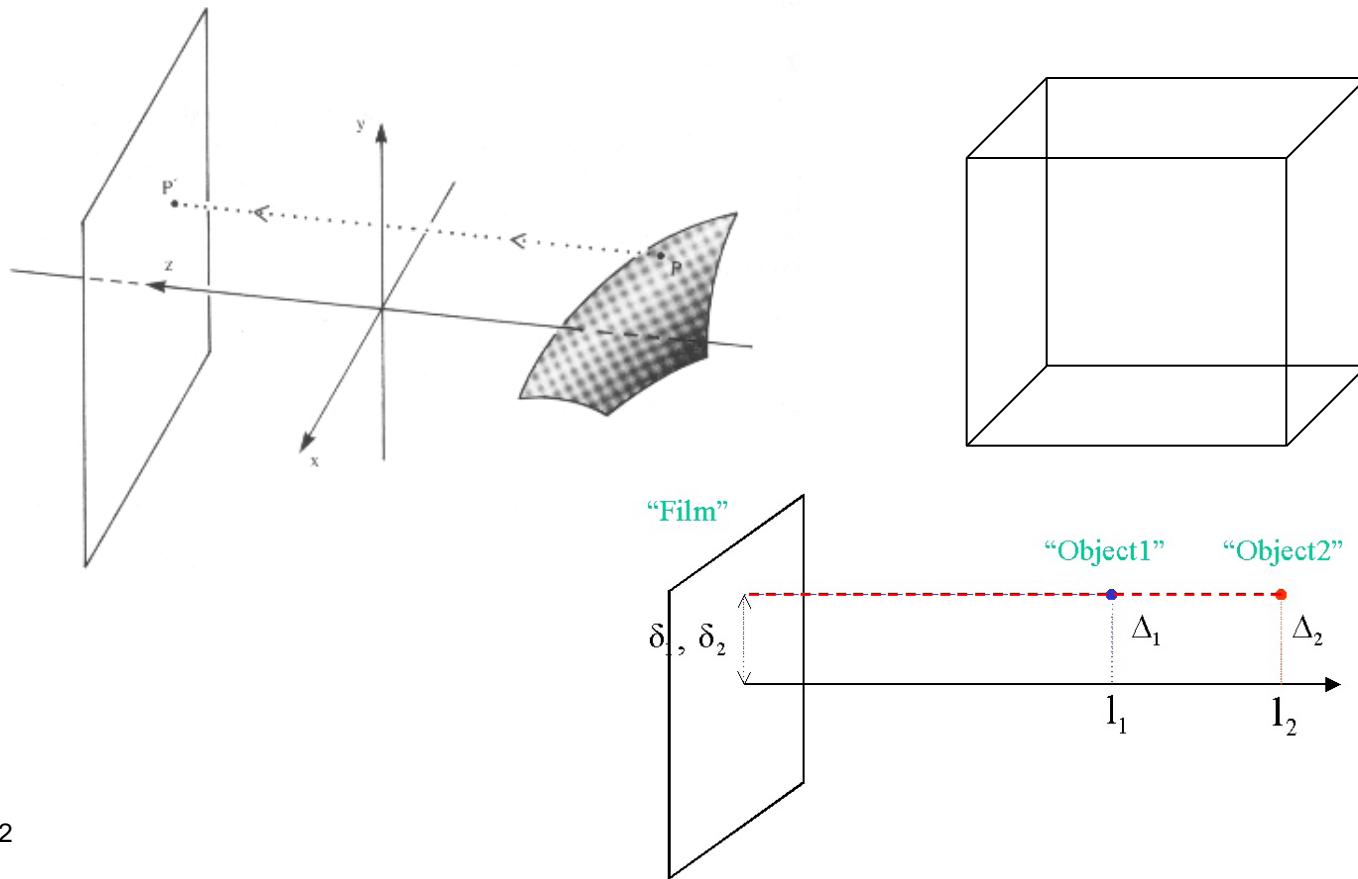


Gözde oluşan imgenin uzunluğu nedir?



İmge nasıl oluşturulur?

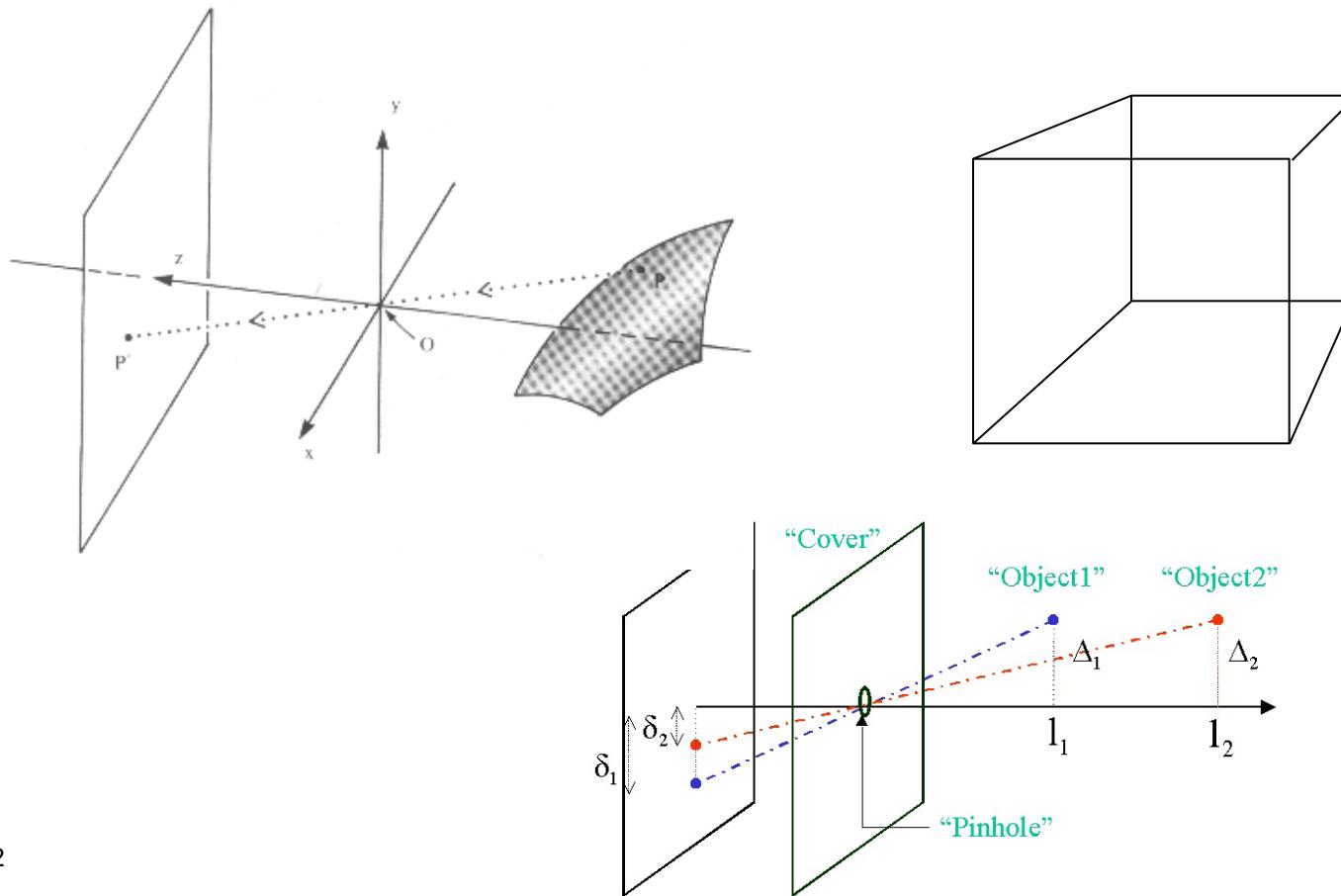
- Paralel izdüşüm (orthographic projection)





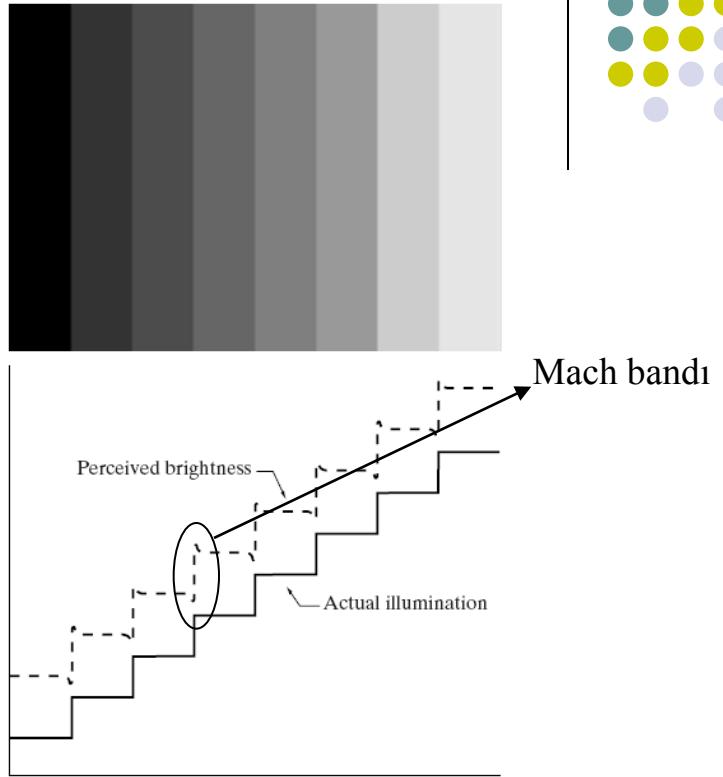
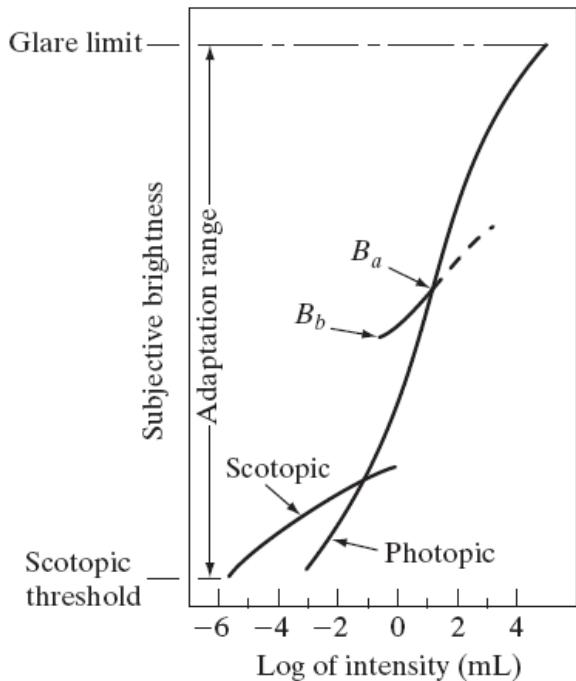
İmge nasıl oluşturulur?

- Perspektif izdüşüm (perspective projection)



Görsel Algı Elemanları

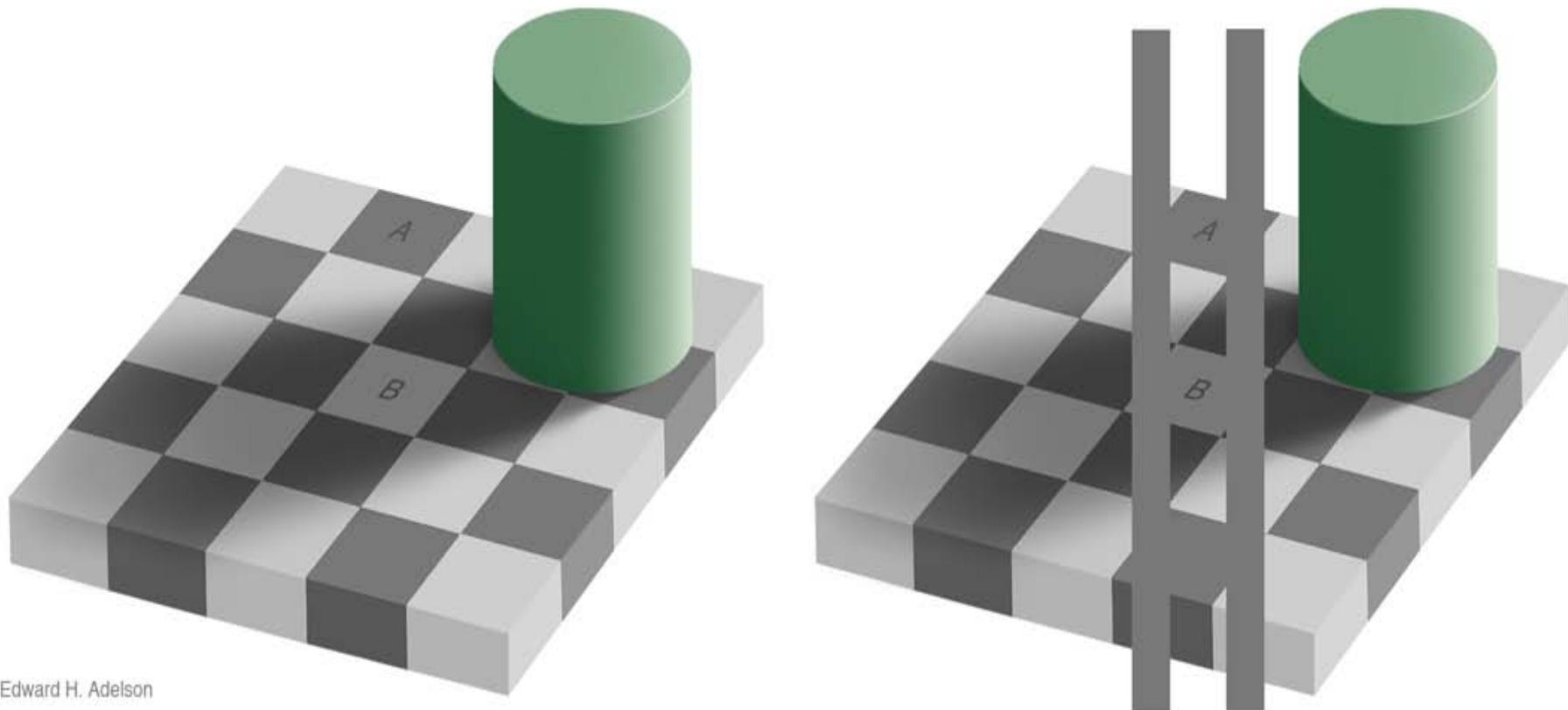
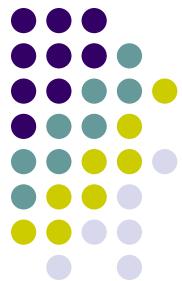
- Işıklılık uyumu ve ayırmsama:



İnsan görme sistemi tarafından alınan ışık (öznel ışık), göze gelen ışık yoğunluğunun logaritmik bir fonksiyonudur.



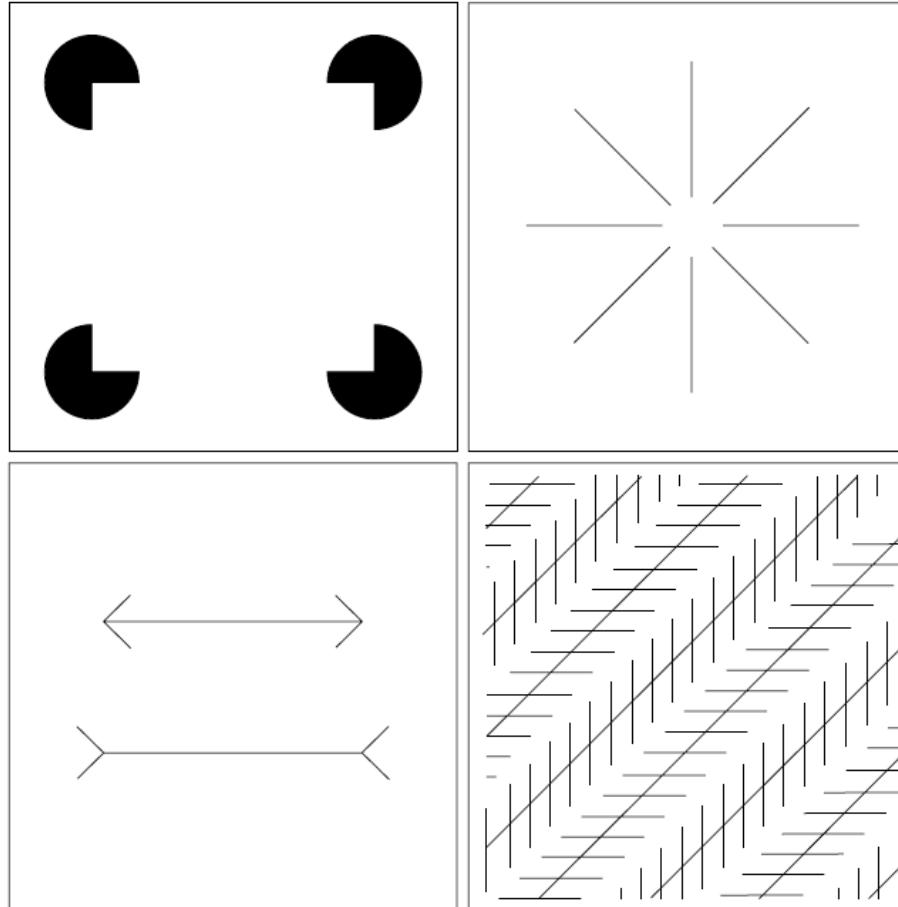
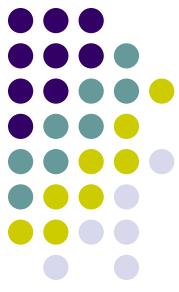
Görsel Algı Elemanları



Edward H. Adelson

Görsel Algı Elemanları

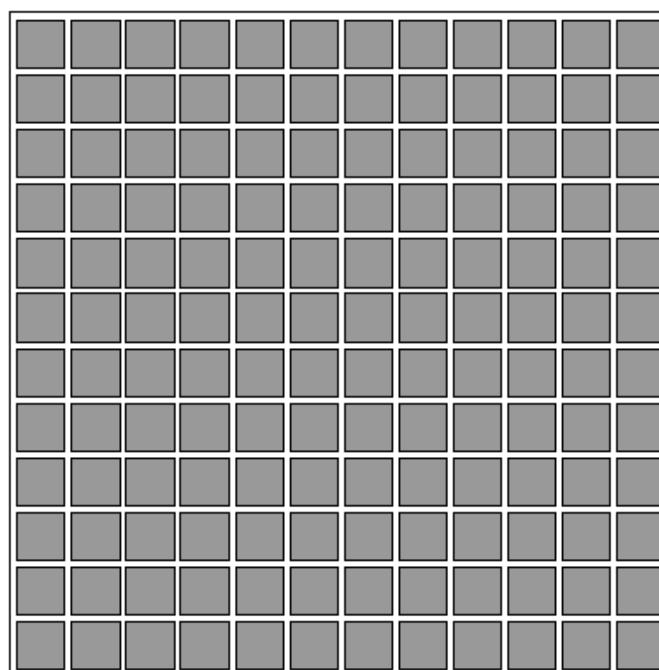
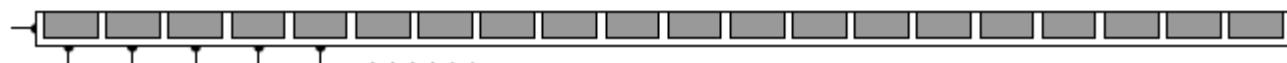
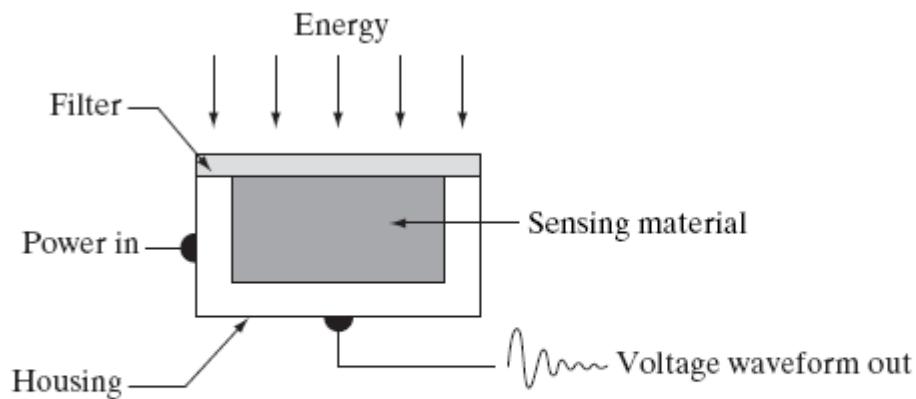
- Optik yanlışları:



Görsel Algı Elemanları



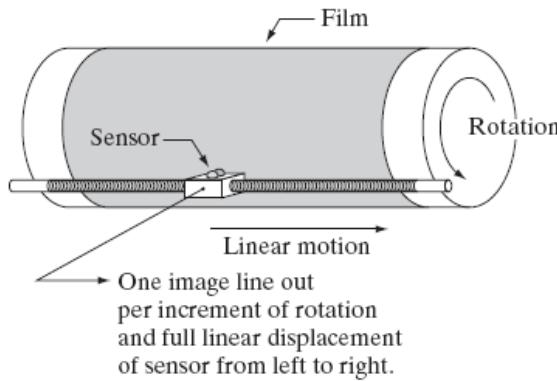
İmge Algılama ve Alma



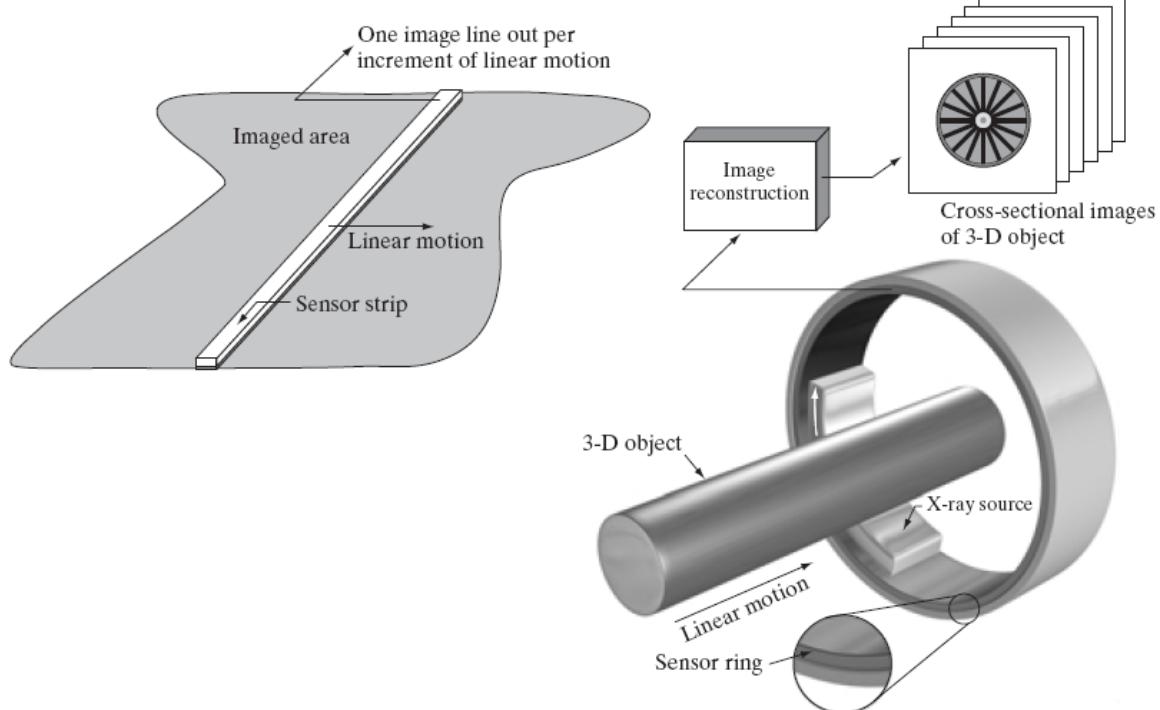
İmge Algılama ve Alma



- Tek algılayıcı ile imge alma:

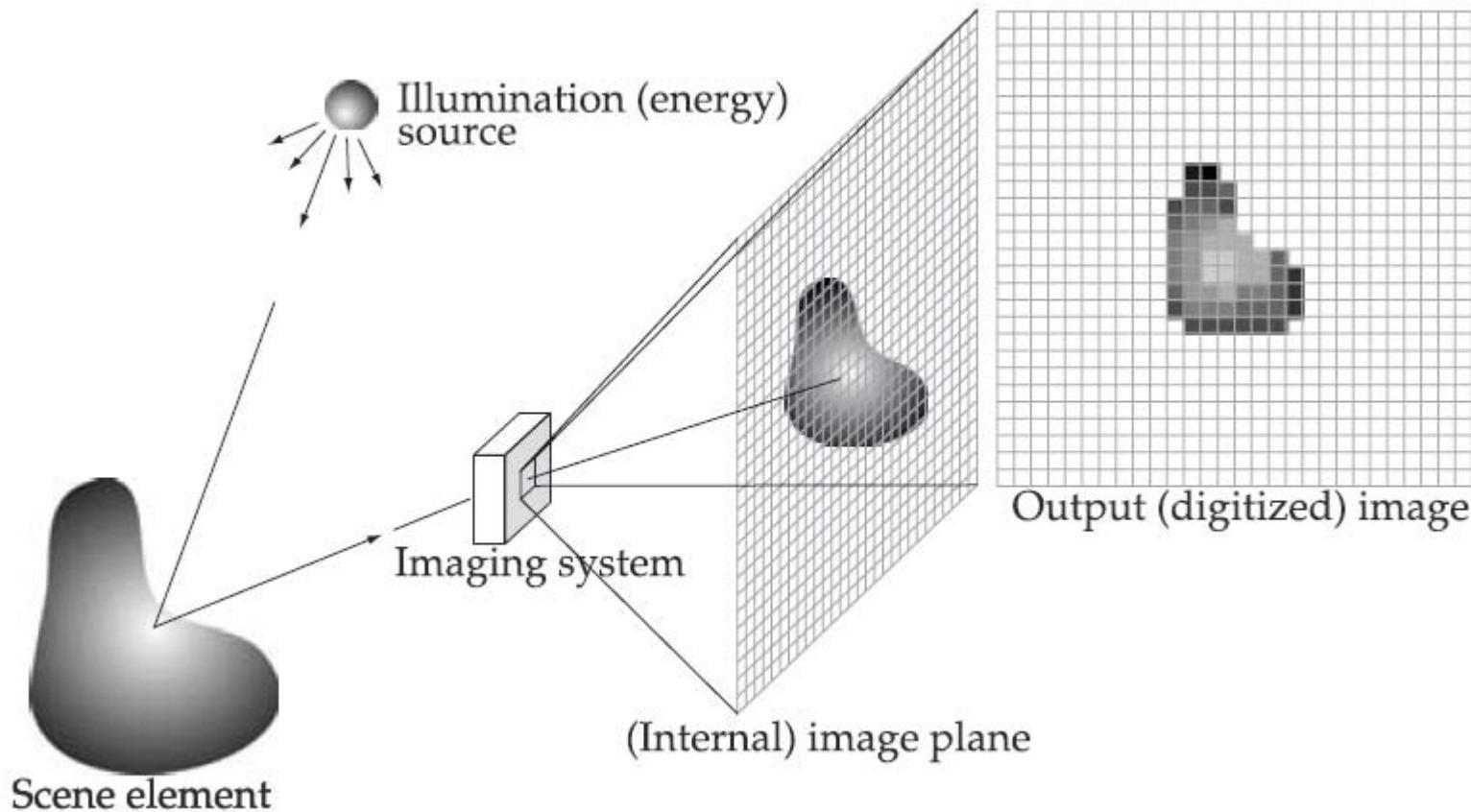


- Algılayıcı dizisi kullanarak imge alma:

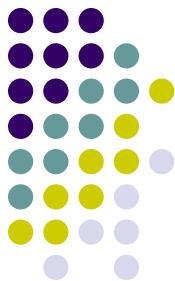


İmge Algılama ve Alma

- Algılayıcı dizileri ile imge alma:



İmge Gösterim Modeli



- Temel olarak bir imge iki boyutlu bir fonksiyon ile ifade edilir:

$$f(x, y)$$

$(x, y) \rightarrow$ yatay x ve y koordinatları belirtir.

- Fiziksel işlemlerle oluşan bir imgede elde edilen ışık değeri, fiziksel kaynaktan saçılan enerji ile orantılıdır.
- Bu nedenle;

$$0 < f(x, y) < \infty$$

- Sonuç: Mutlaka bir enerji vardır. Bu nedenle $f(x, y)$ kesinlikle sıfır olamaz.



İmge Gösterim Modeli

- $f(x, y)$ fonksiyonu iki bileşen ile tanımlanabilir:
- Sahneyi aydınlatan ışık kaynağının aydınlatması (*illumination*).
- Aydınlatma ışığının sahnedeki nesnelerden yansımıası (*reflectance*).

$$f(x, y) = i(x, y)r(x, y)$$

$$0 < i(x, y) < \infty$$

$$0 < r(x, y) < 1$$

- Gri tonlu imgelerde ışıklılık değeri:

$$\ell = f(x, y)$$

$$L_{\min} < \ell < L_{\max}$$

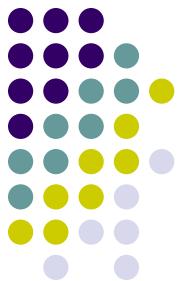
$$L_{\min} = i_{\min} r_{\min} \approx 10$$

$$L_{\max} = i_{\max} r_{\max} \approx 1000$$

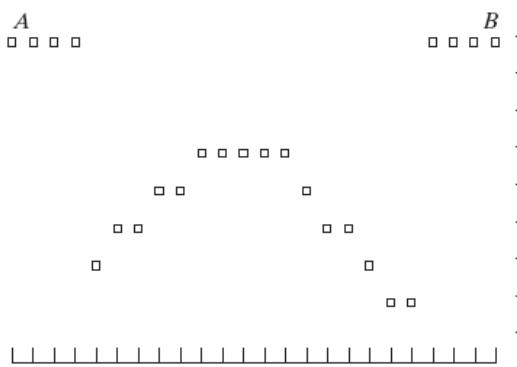
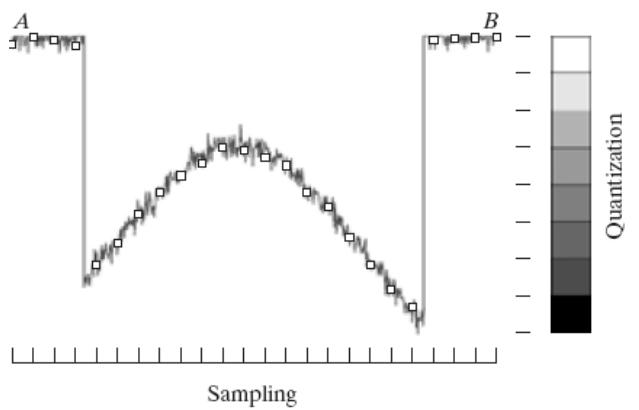
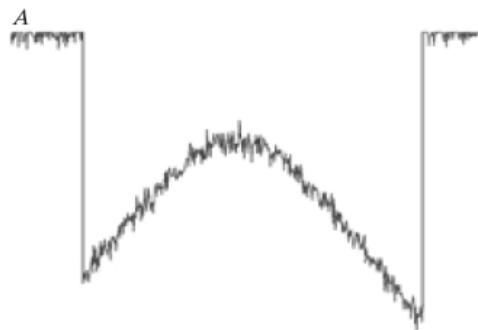
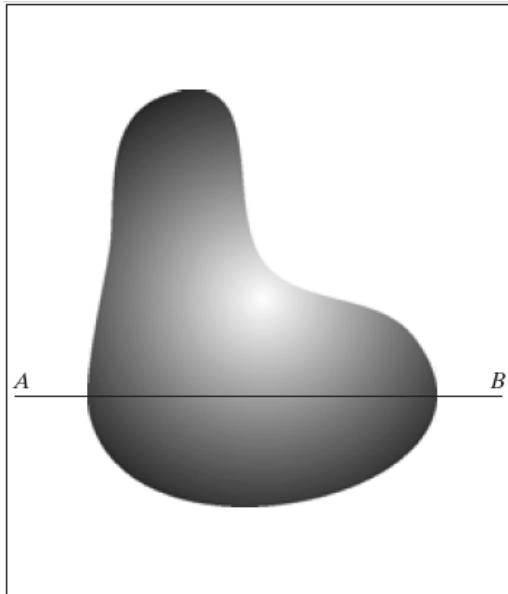
$[L_{\min}, L_{\max}]$: Gri tonlaral

Pratikte: L_{\min} = siyah, L_{\max} = beyaz

İmge Örnekleme ve Nicemleme



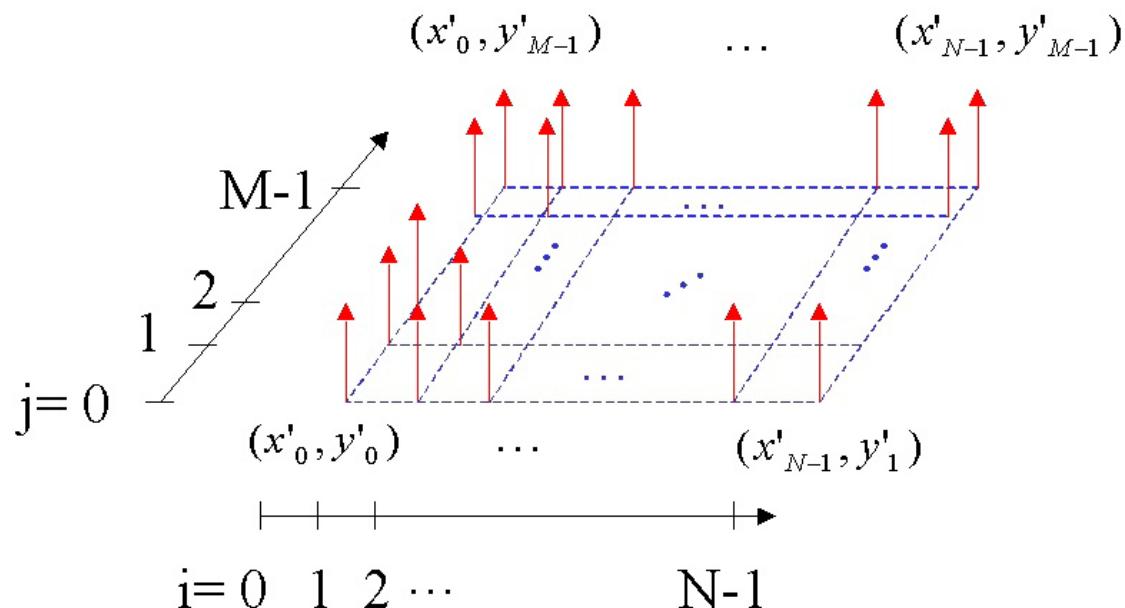
- Koordinat değerlerini sayısallaştırma → *örnekleme*.
- Genlik değerlerini sayısallaştırma → *nicemleme*.





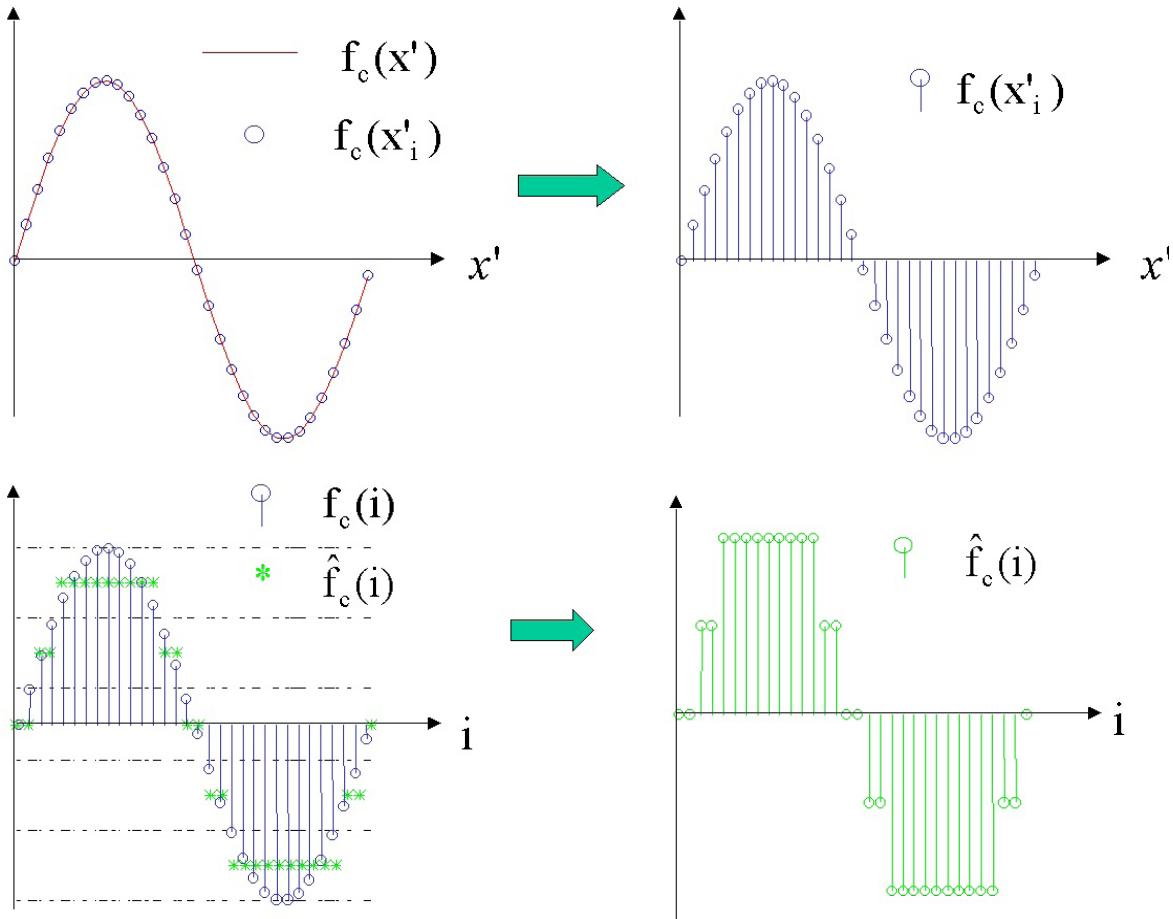
Örnekleme

comb(x' , y')



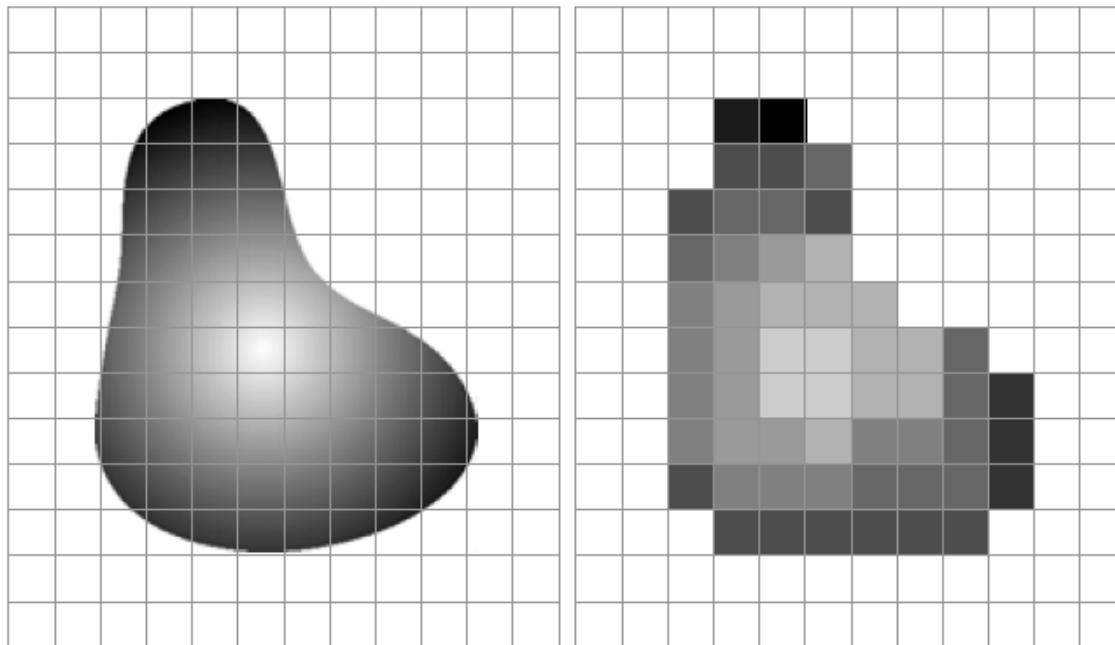
$$comb(x', y') = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \delta(x' - i\Delta_x, y' - j\Delta_y)$$

Örnekleme $\rightarrow f_C(x', y') \times comb(x', y')$





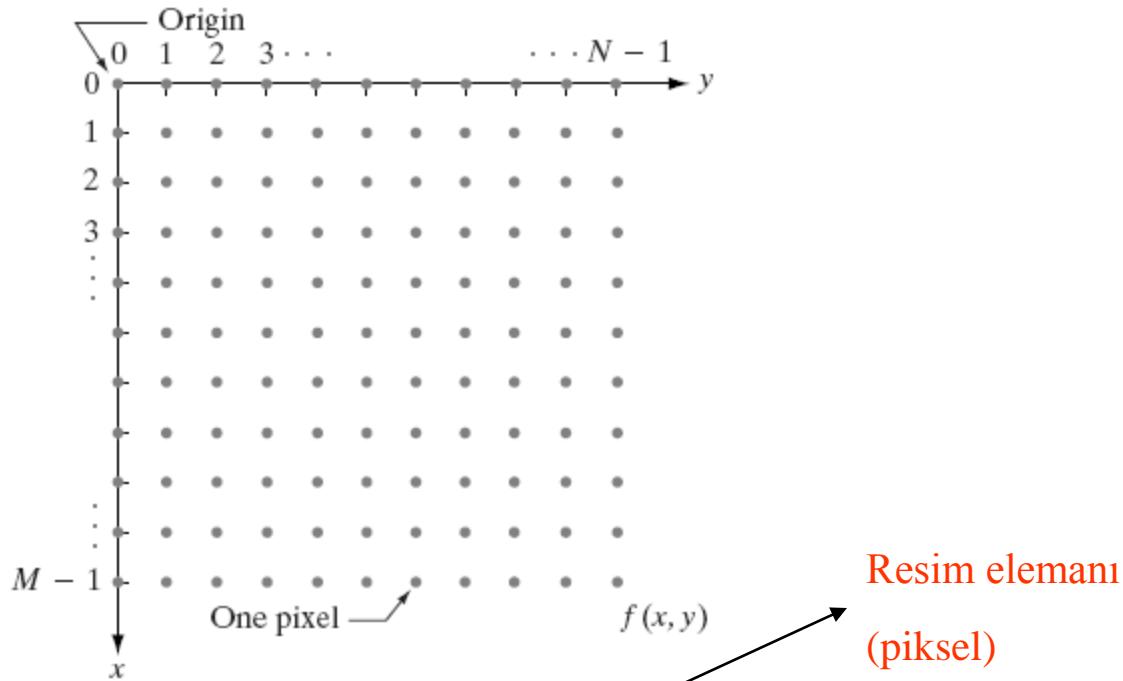
- Genelde seviye sayısı $P=2^8=256$ ve $\log_2(P)=\log_2(2^8)=8$ bit nicemleme.



Sayısal İmge Gösterimi



- $f(x, y)$ imgesi örneklendiğinde M satır ve N sütunluk bir sayısal imge oluşturur.



$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix}$$

Sayısal İmge Gösterimi



- Bazen sayısal imge gösterimi:

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

$$a_{i,j} = f(x=i, y=j) = f(i, j)$$

- Sayısalallaştırmada genellikle uzamsal boyutlar önemli değildir.
- Donanımsız açıdan asıl önemli olan, gri ton seviyesinin 2'nin kuvveti olmasıdır.

$$L = 2^k$$

$[0, L-1]$: imgenin dinamik ~~größe~~ dynamic range).

Sayısal İmge Gösterimi



- Sayısal imgeyi saklamak için gerekli olan bit sayısı:

$$b = M \times N \times k$$

$M = N$:

N/k	1 ($L = 2$)	2 ($L = 4$)	3 ($L = 8$)	4 ($L = 16$)	5 ($L = 32$)	6 ($L = 64$)	7 ($L = 128$)	8 ($L = 256$)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

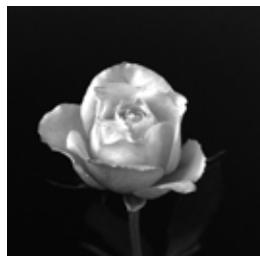
Sayısal İmge



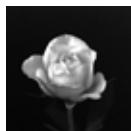
- Uzamsal çözünürlük:



1024x1024



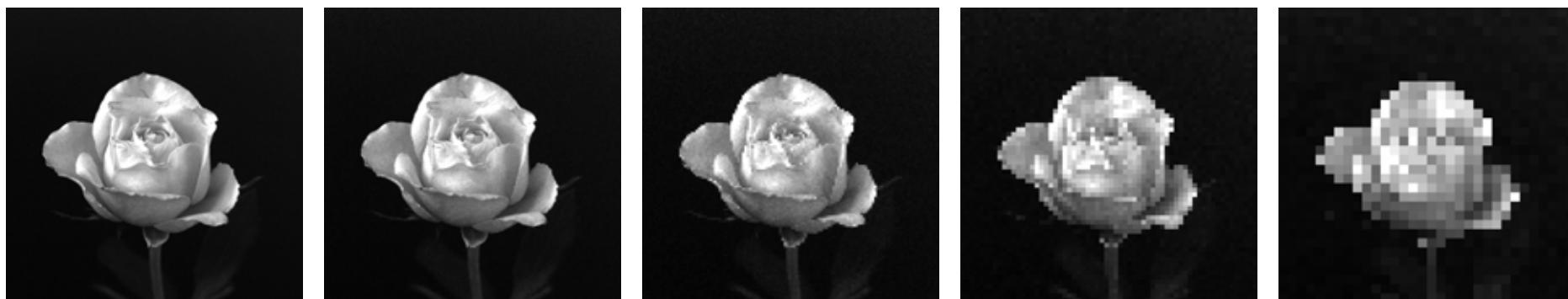
512x512



256x256



64x64



Sayısal İmge



- Bit derinliği:



L=8



L=7



L=6

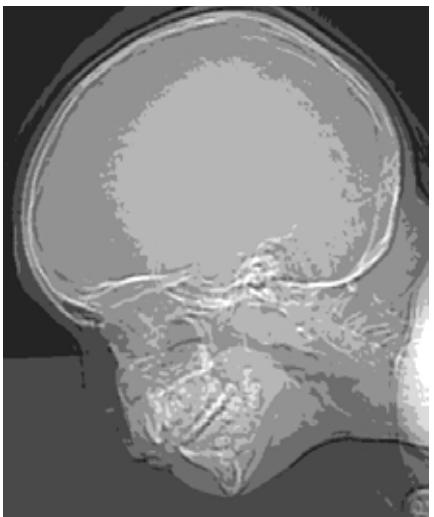


L=5

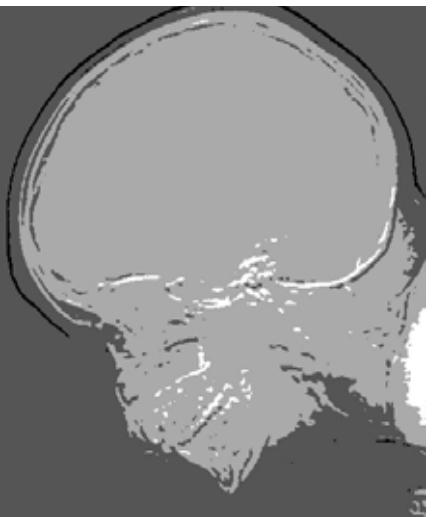


3/10/2012

L=4



L=3



L=2

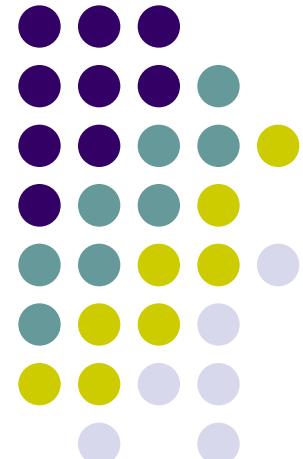


L=1

İMGE İŞLEME

Ders-2

İmge Dosya Tipleri ve Temel İşlemler



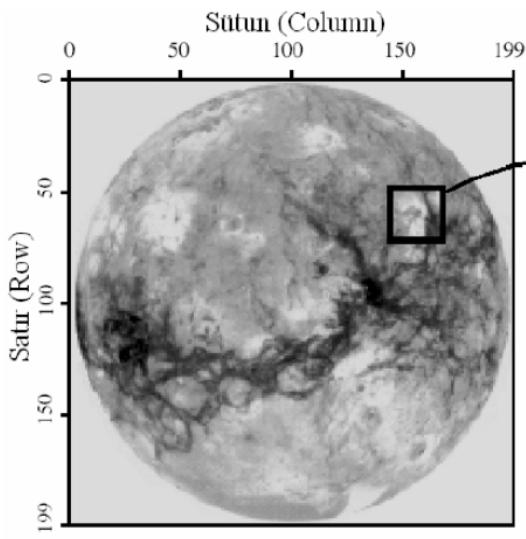
KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İ işaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

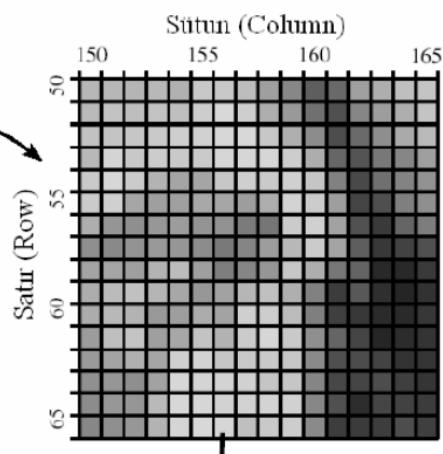
Dersin web sayfası: http://mf.kou.edu.tr/elohab/kemalg/imge_web/odev.htm



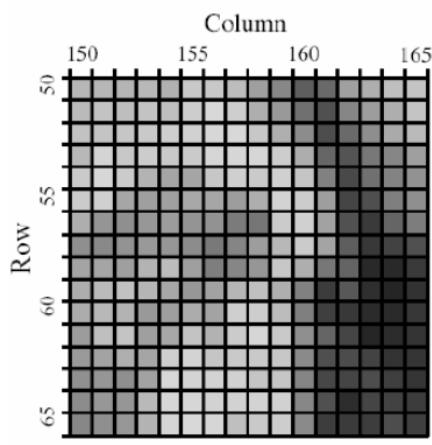
MATLAB temel bilgiler



200x200 piksel boyunda bir img



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



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

Column			
150	155	160	165
183	183	181	184
186	195	190	195
194	196	198	201
184	212	200	204
202	215	203	179
203	208	166	159
174	149	143	151
143	137	147	153
164	165	159	179
173	187	193	181
172	184	179	153
156	191	196	159
154	163	175	165
144	150	143	162
140	151	150	185
135	143	151	179
199	183	184	188
162	182	191	172
132	175	188	163
197	175	199	207
129	129	127	205
60	60	63	188
88	88	56	127
47	47	43	58
37	37	42	50
52	52	44	58
51	51	41	41
53	53	41	41
67	67	41	58
76	76	41	53
77	77	42	53
61	61	44	52
59	59	44	53
66	66	45	53
60	60	45	60



İmge Dosya Tipleri

Sayısal imgeler genellikle bmp, jpg, tiff, raw gibi formatlarda saklanmaktadır.

Gri tonlu, 8bit/piksel bit derinliğinde, 1000x1000 piksel boyutlu bir imgenin bellekte kaplayacağı alan nedir?

Yanıt: $1000000 \text{ bayt} = 977 \text{ kbayt}$

Eğer yukarıdaki imge renkli olsaydı, bu imgenin bellekte kaplayacağı alan:

$977 \text{ kbayt} \times 3 = 2931 \text{ kbayt} = 2.86 \text{ Mbayt}$



İmge Dosya Tipleri (raw)

- Yalnızca piksel ışıklılık değerlerini barındıran dosya tipidir.
- İmgenin piksel boyutunu gösteren herhangi bir başlık bilgisi içermez.
- İmgeyi açmak için piksel boyutunu bilmek gereklidir.
- Bu tip dosyaları Matlab ya da C gibi programlama dillerini kullanarak açmak için bilinen dosya açma işlemlerini yapmak gerekmektedir.

```
w=256; % imgenin yatay boyutu  
h=256; % imgenin düşey boyutu  
f=fopen('C:\Documents\lena.raw','r');  
% açılacak dosyanın konumu ve adı f dosya değişkenine yüklendi
```

```
I=fread(f); % imge I değişkenine dizi olarak alındı  
I=reshape(I,w,h); % 2-boyutlu matrise dönüştürüldü  
status=fclose(f); % dosya kapandı
```

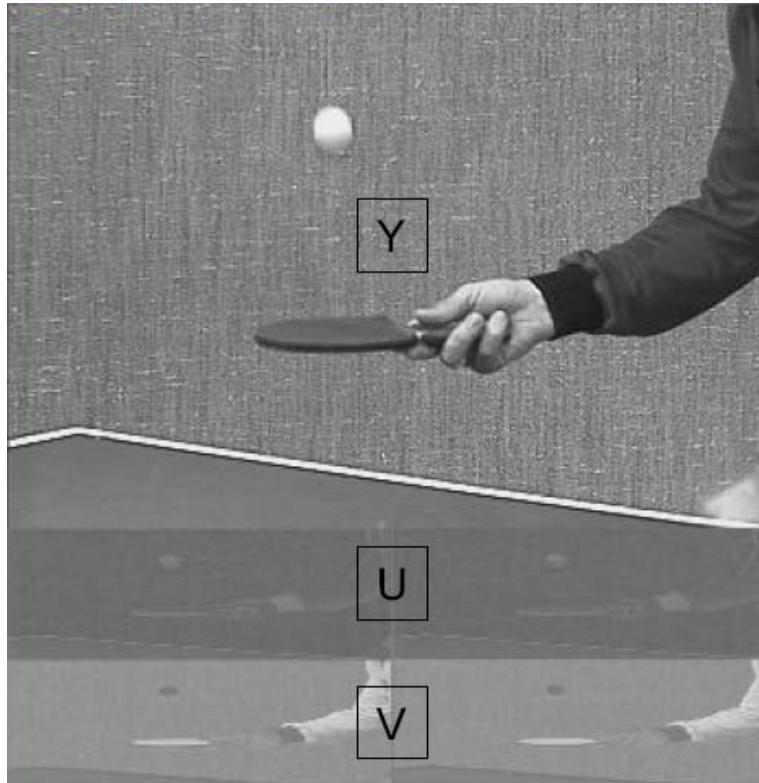
```
figure; imshow(uint8(I)); % Ekranda imge görüntülendi
```





İmge Dosya Tipleri (yuv)

- Renk bileşenleri sıkıştırılmıştır.
- Raw dosya tipine benzer şekilde, bu dosya tipinde de imgenin piksel boyutu dosya içerisinde yoktur.
- Bu nedenle imgenin boyutlarının önceden bilinmesi gerekmektedir.





İmge Dosya Tipleri (bmp)

- Sıkça kullanılan bir imge dosya tipidir.
- Sıkıştırılmamış ve sıkıştırılmış dosya yapısı mevcuttur.
- Sıkıştırılmamış dosya yapısında, raw dosya tipine ek olarak imgenin piksel boyutu, bit derinliği gibi bilgileri turan başlık kısmı vardır.
- Matlab ile .bmp uzatılı bir imgeyi okumak için

```
I=imread('C:\Documents\lena.bmp');
```

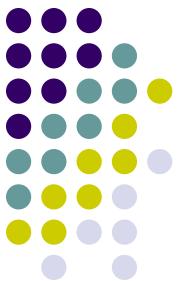
Kodunu yazmak yeterlidir.

- C ve diğer programlama dillerinde ise mevcut kütüphanelerden faydalananarak bu dosya tipi okunabilmektedir.



İmge Dosya Tipleri (diğer)

- Tiff, jpeg, png, gif, pbm, pgm, hdf, pcx... gibi imge dosya tipleri de vardır.
- Tiff genellikle bmp benzeri bir dosya tipi olmakta birlikte kayıplı sıkıştırma modu da vardır.
- Bunun yanında 10 bit/piksel, 16 bit/piksel gibi bit derinliklerinde kayıt olanına sahiptir.
- Matlab “imread” komutu ile okunmaktadır.
- Jpeg kayıplı bir imge dosya tipidir.
- Bu sıkıştırma kullanılarak imge yüksek verimlilikle sıkıştırılabilir.
- Bu dosya tipi de Matlab “imread” komutu ile okunmaktadır.



İmge standart boyutları

- CIF: Common Intermediate Format
 - VGA: Video Graphics Array
 - SIF: Source Intermediate Format

SQCIF 128×96

QCIF 176×144

CIF 352×288

4CIF 704×576

16CIF 1408×1152

VGA 640×480

QVGA 320×240

SCIF 352×240



devirme

$$B = A'$$

$$B(j,i) = A(i,j)$$

$$(i=0,...,N-1, j=0,...,M-1)$$





düşeyde çevirme

$$B(i, M - j - 1) = A(i, j)$$
$$(i = 0, \dots, N - 1, j = 0, \dots, M - 1)$$





döndürme

- $90^\circ, 180^\circ, 270^\circ$ gibi açılarda döndürme işlemlerini kolayca gerçekleştirebiliriz.
- Bu açıların dışındaki değerlerde ise açısal döndürme işlemlerinin ($\sin x$, $\cos x$ değerlerini kullanarak) yapılması gerekmektedir.
- Bunun yerine, Matlab hazır işlevlerinden “imrotate” kullanılabilir.

Ir=imrotate(I,acı,yöntem);

acı: saat yönünün tersi dönülecek açı değeri.

yöntem: döndürme işlemi sonrasında yeni piksel değerlerinin hesaplanacağı aradeğerleme yöntemi.

‘nearest’, ‘bilinear’, ‘bicubic’,

Örn;

Ir=imrotate(I,45, 'bilinear');



kırpma

$$B(i, j) = A(n_1 + i, n_2 + j)$$

$$(i = 0, \dots, m_1 - 1, j = 0, \dots, m_2 - 1)$$

(n_1, n_2) ~~şlanga~~ noktası

(m_1, m_2) → pencere boyutları





Öteleme

$$B(i, j) = A(i - n_1 + 1, j - n_2 + 1)$$

$(i = n_1, \dots, N, j = n_2, \dots, M)$

(n_1, n_2) şlangħaq noktası



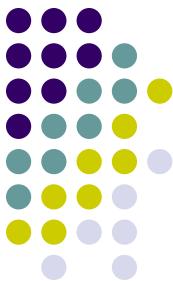


Öteleme

Öteleme işlemi yapan bir Matlab işlevi yazalım:

```
function [B]=my_otele(A,n1,n2)  
[w,h]=size(A);  
  
B=zeros(w,h);  
  
for i=n1:w  
    for j=n2:h  
        B(i,j)=A(i-n1+1,j-n2+1);  
  
    end  
end
```

Burada for döngüleri yerine tek bir satır yazarak aynı işlem yapılabilir.



boyut değiştirme-yakınlaştırma

- Yakınlaştırma, düşük piksel boyutlu bir imgenin piksel boyutunun yazılımsal olarak artırılmasıdır.
- Sayısal yakınlaştırma (digital zoom).

A	B	C
D	E	F
G	H	I



A	A	B	B	C	C
A	A	B	B	C	C
D	D	E	E	F	F
D	D	E	E	F	F
G	G	H	H	I	I
G	G	H	H	I	I



boyut değiştirme-yakınlaştırma

- Boyut büyültmede daha yumuşak geçişler için:

The diagram illustrates a transformation process from a smaller input matrix to a larger output matrix. On the left, a 3x3 input matrix is shown with elements A through I. An arrow points from this matrix to a 5x5 output matrix on the right. The output matrix contains elements calculated using bilinear interpolation formulas.

A	B	C		
D	E	F		
G	H	I		

→

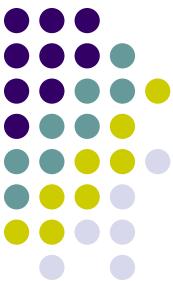
A	$\frac{A+B}{2}$	B	$\frac{B+C}{2}$	C
$\frac{A+D}{2}$	$\frac{A+B+D+E}{4}$	$\frac{B+E}{2}$	$\frac{B+C+E+F}{4}$	$\frac{C+F}{2}$
D	$\frac{D+E}{2}$	E	$\frac{E+F}{2}$	F
$\frac{D+G}{2}$	$\frac{D+E+G+H}{4}$	$\frac{E+H}{2}$	$\frac{E+F+H+I}{4}$	$\frac{F+I}{2}$
G	$\frac{G+H}{2}$	H	$\frac{H+I}{2}$	I



boyut değiştirme-yakınlaştırma

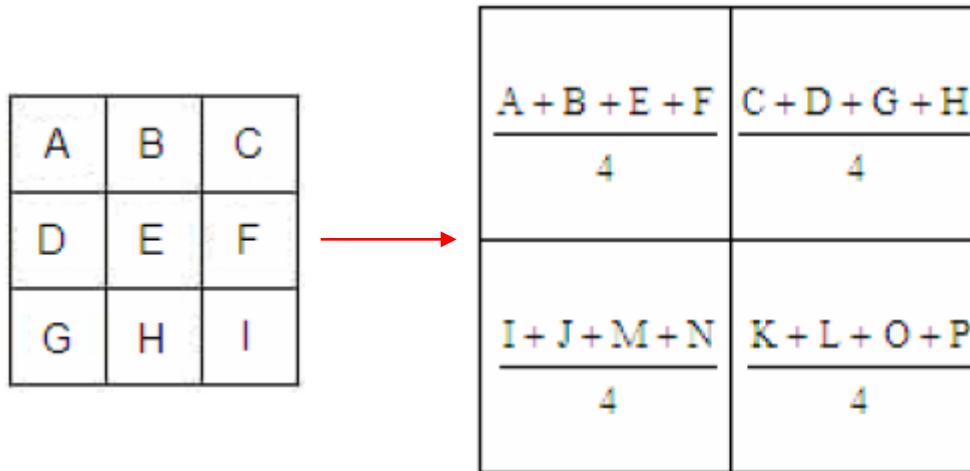
- Hangisi daha görünür?





boyut değiştirme-uzaklaştırma

- Birden fazla pikselin değeri çeşitli matematiksel işlemlerden geçirilerek bir piksele atanır.





boyut değiştirme

- Matlab ile boyut değiştirme için “imresize” adındaki işlev kullanılabilmektedir.

`Is=imresize(I,oran,yöntem);`

oran : giriş imgesinin boyutunun değişme oranını verir. $oran > 1$ (büyütme), $oran < 1$ (küçültme).

yöntem : boyut değiştirmede kullanılacak aradeğerleme yöntemi.

Örn;

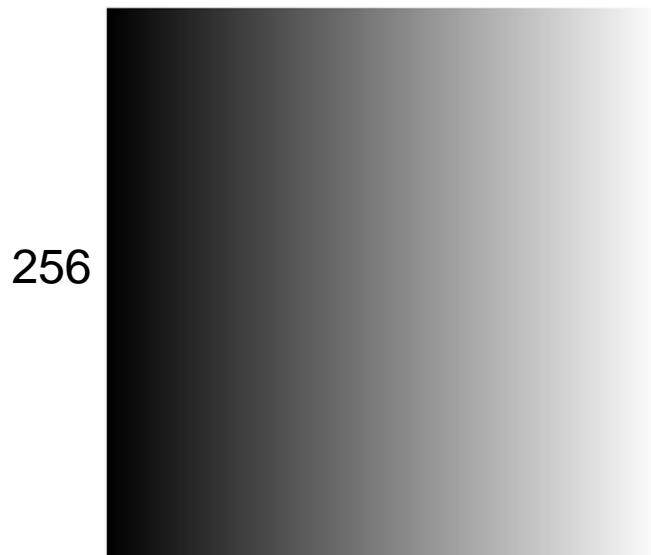
`Is=imresize(I,0.97, 'bicubic');`



İmge oluşturma

$$\mathbf{A} = \left[\begin{array}{ccccc} 0 & 1 & 2 & \dots & 255 \\ 0 & 1 & 2 & \dots & 255 \\ \vdots & & & & \\ 0 & 1 & 2 & \dots & 255 \end{array} \right] \quad \left. \right\} 256 \text{ rows}$$

256



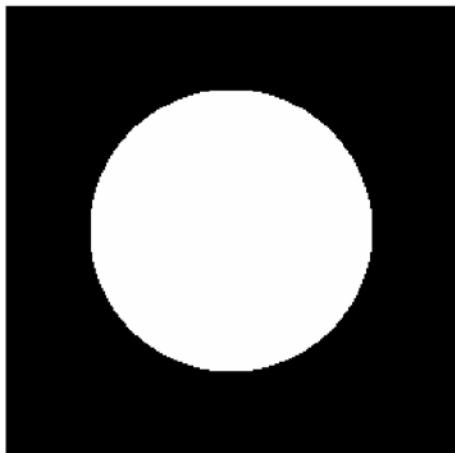
256

```
for i = 1 : 256
    for j = 1 : 256
        A(i, j) = j - 1;
    end
end
```



İmge oluşturma

- (128,128) merkezli, yarıçapı 80 piksel beyaz bir daire



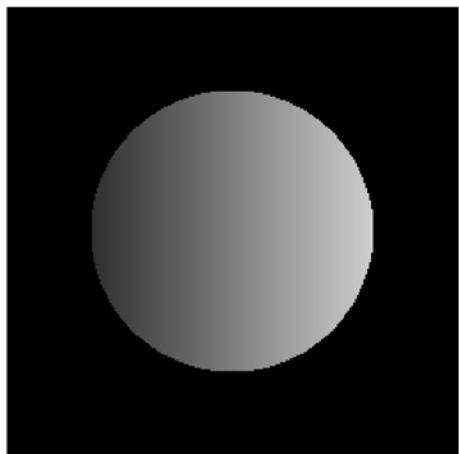
$$B(i, j) = \begin{cases} 255 & \text{if } \sqrt{(i - 128)^2 + (j - 128)^2} < 80 \\ 0 & \text{otherwise} \end{cases}$$

```
for i = 1 : 256
    for j = 1 : 256
        dist = ((i - 128)^2 + (j - 128)^2)^(.5);
        if (dist < 80)
            B(i, j) = 255;
        else
            B(i, j) = 0;
        end
    end
end
```



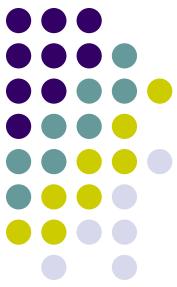
İmge oluşturma

- ???



$$C = \begin{matrix} A \\ \times \\ B \end{matrix} / 255$$

```
for i = 1 : 256  
    for j = 1 : 256  
        C(i, j) = A(i, j) * B(i, j)/255;  
    end  
end
```



Ortalama ve Değişinti

- Bir imgenin örnek ortalaması (sample mean):

$$m_A = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} A(i, j)}{NM}$$

- Örnek değişintisi (sample variance):

$$\sigma_A^2 = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (A(i, j) - m_A)^2}{NM}$$

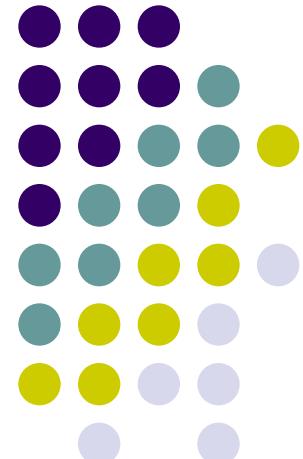
- Örnek standart sapması (örnek st. sapma):

$$\sigma_A = \sqrt{\sigma_A^2}$$

İMGE İŞLEME

Ders-3

İmge Pekiştirme
(Nokta İşlemleri)



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İ işaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/lohab/kemalg/imge_web/odev.htm

Nokta İşlemleri

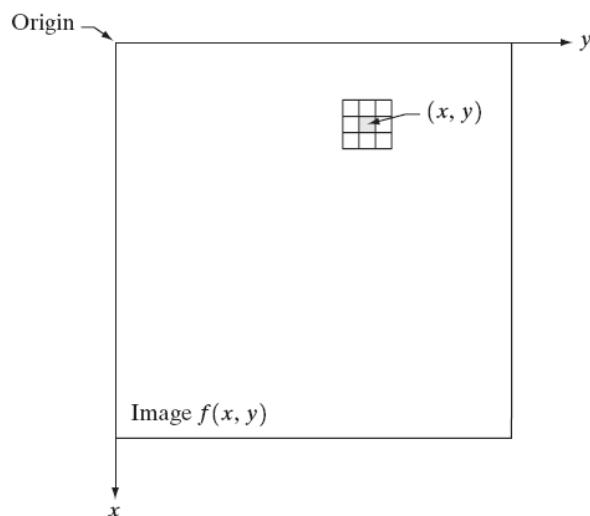


- Piksellerden oluşan imge uzayına uzamsal düzlem (spatial domain) denir.
- Uzamsal düzlem işlemleri aşağıdaki gösterimle ifade edilmektedir.

$$g(x, y) = T[f(x, y)]$$

↓
işlev

- Buradaki T işlevi, doğrudan (x, y) pikselini işleyebileceği gibi, (x, y) pikselinin komşuluklarını da hesaba katabilir.

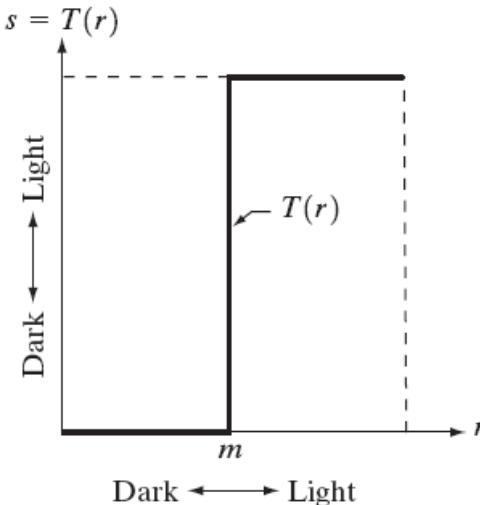
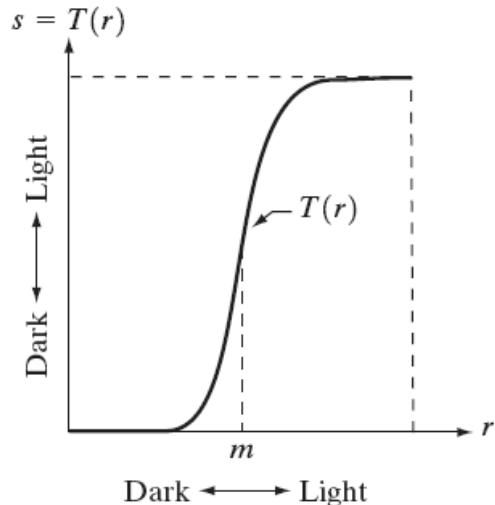


Nokta İşlemleri



- Eğer komşuluk boyutu 1x1 ise (yalnızca (x,y) pikseli alınıyor), bu durumda T *gri-seviye dönüşüm işlevi (grayscale-level transformation function)* olarak adlandırılır.
- Bu tür işlemlere de *nokta işlemleri (point operations)* adı verilir.
- Bu işlem kısaca aşağıdaki şekilde yazılabilir.

$$s = T(r)$$



Parlaklık Ayarı



$$g(x, y) = T[f(x, y)]$$

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

$b > 0$ ise parlaklık artar

$b < 0$ ise parlaklık azalır

$$s = r + b$$



orjinal



$b = -50$



$b = +50$

Karşılık (Kontrast) Ayarı



$$g(x, y) = T[f(x, y)]$$

$$= af(x, y)$$

$a > 1$ ise karşılık artar

$a < 1$ ise karşılık azalır

$$s = ar$$



orjinal



$a = 0.5$



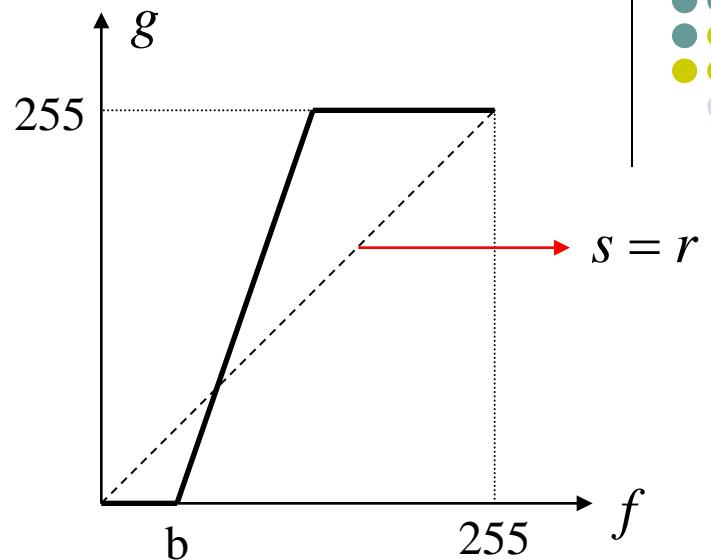
$a = 2$

Parlaklık+Karşılık Ayarı



$$\begin{aligned}g(x, y) &= T[f(x, y)] \\&= af(x, y) + b\end{aligned}$$

$$s = ar + b$$

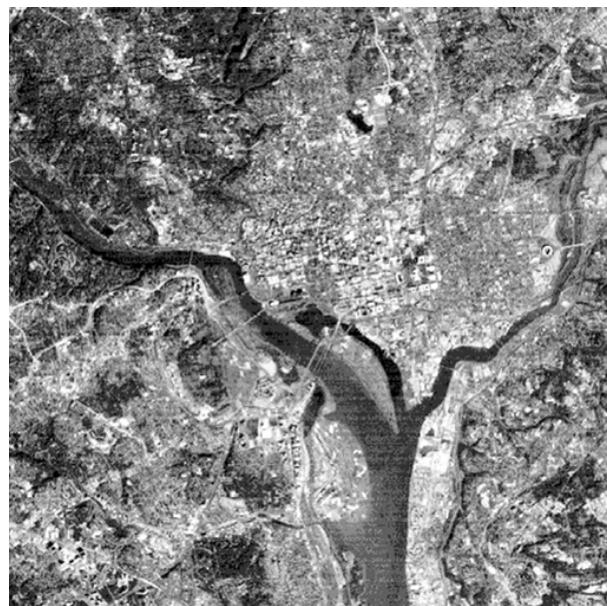
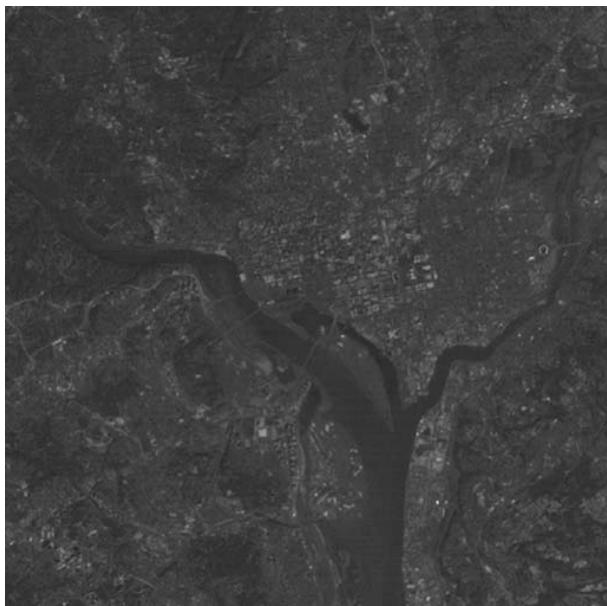
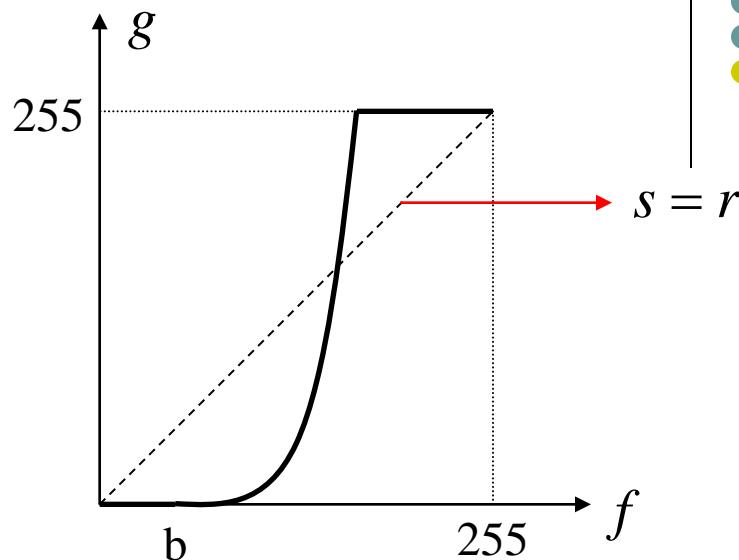


Parlaklık+Karşılık Ayarı

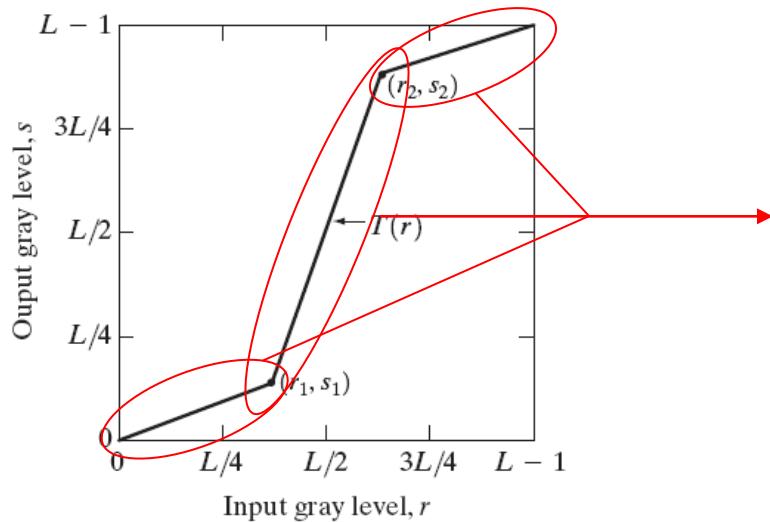


$$g(x, y) = T[f(x, y)]$$

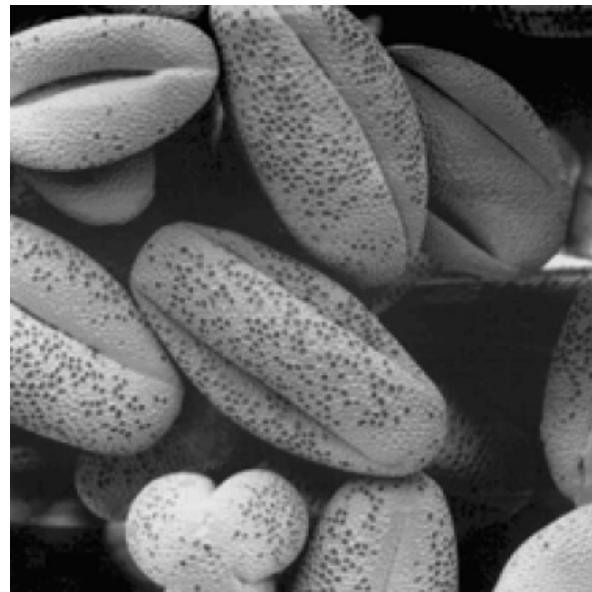
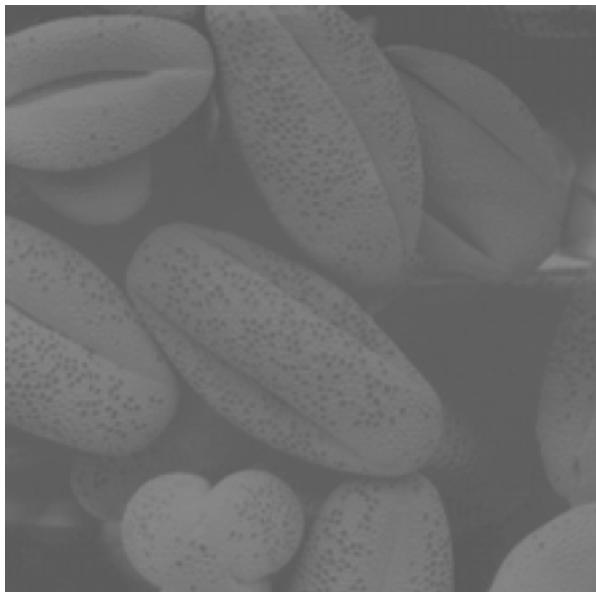
MATLAB **imadjust** işlevi



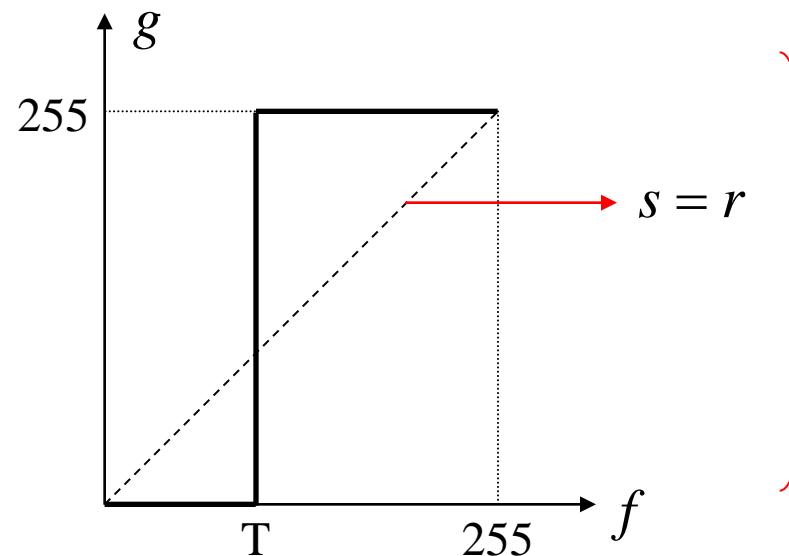
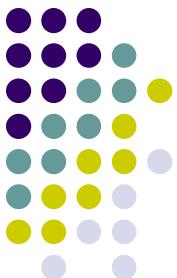
Parlaklık+Karşılık Ayarı



Kısmi-doğrusal dönüşüm



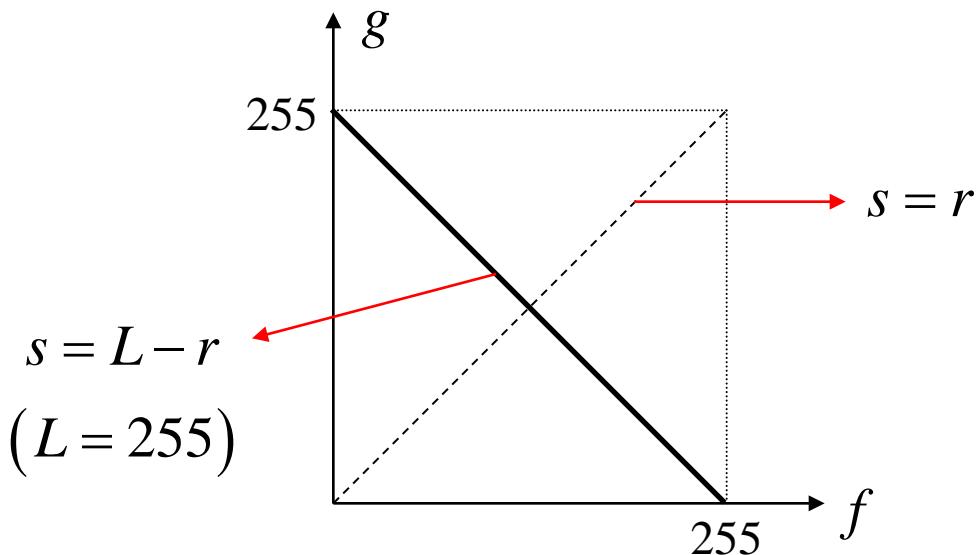
Eşikleme



Sonuçta ikili (binary) imgé oluşuyor.



Olumsuzlama



Histogram



- Her bir gri ton seviyesinin ([0,255]) imgedeki bulunma sıklığını (frekansını) gösterir.
- Yani imgedeki piksellerin dağılımı hakkında bilgi verir.
- İmge pekiştirmede sıkça kullanılmaktadır.

$$h(r_k) = n_k$$

r_k : k . gri seviye

n_k : k . gri seviyedeki toplam piksel sayısı

- Histogram normalize edildiğinde ise gri seviyelerin imge içerisindeki bulunma olasılıklarını verir.

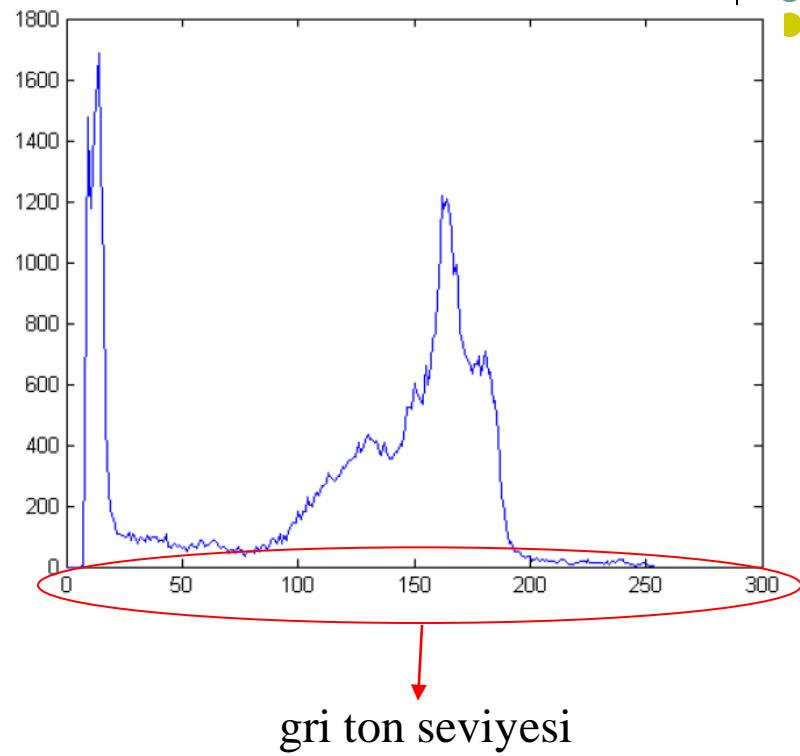
İlgili seviyenin
olasılık değeri

$$p(r_k) = \frac{n_k}{n}$$

$k = 0, 1, \dots, L-1$

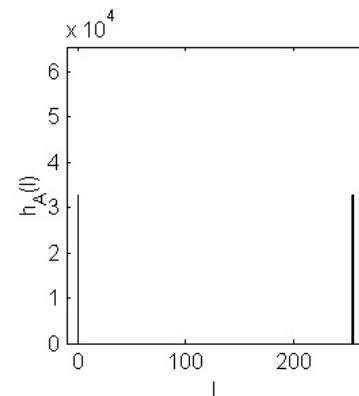
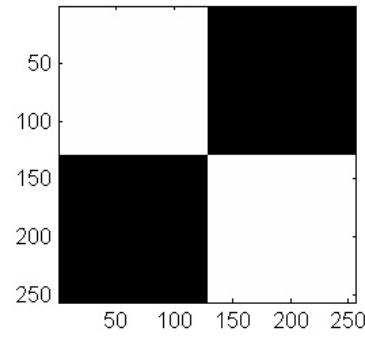
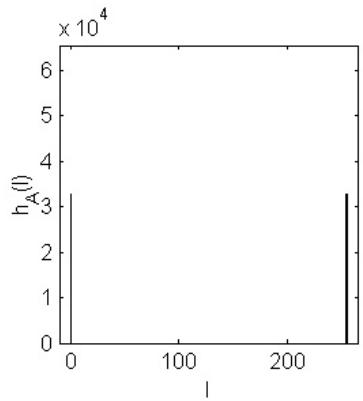
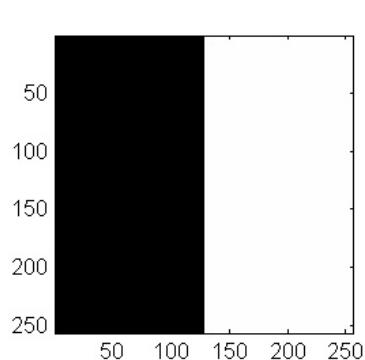
imedeki toplam
piksel sayısı

Histogram



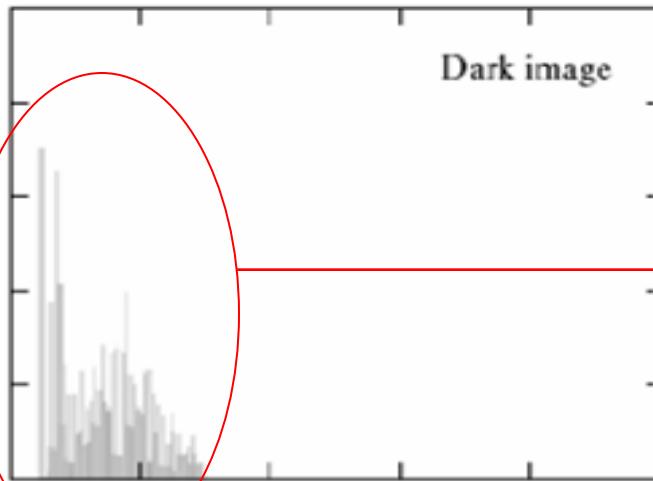
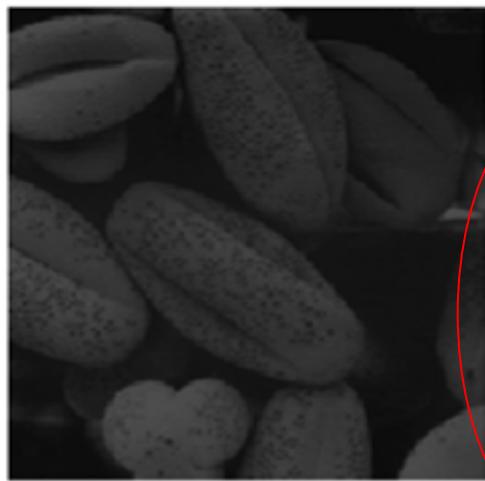
MATLAB **imhist** işlevi

Histogram

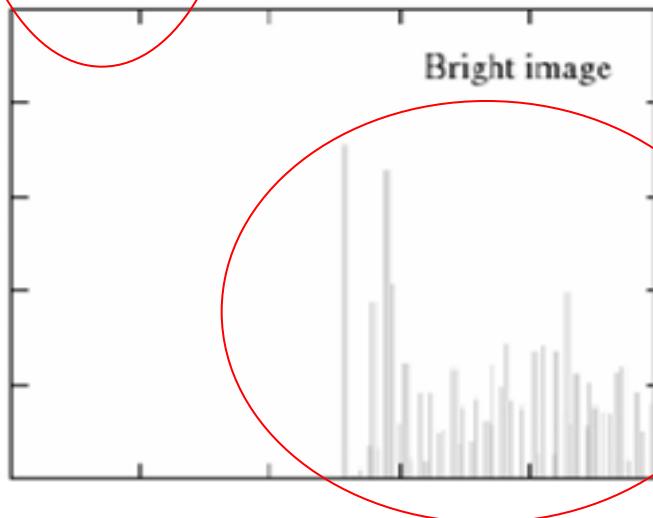
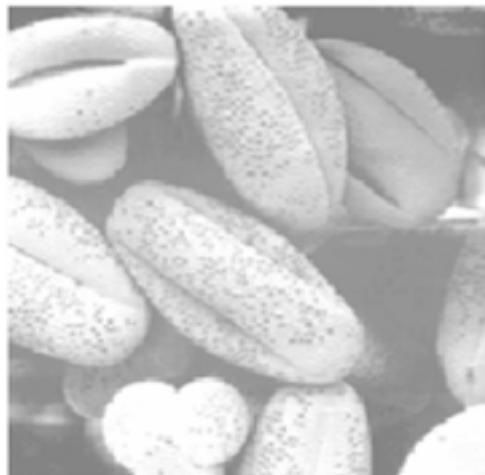


Piksel konum bilgisi bulunmaz!

Histogram

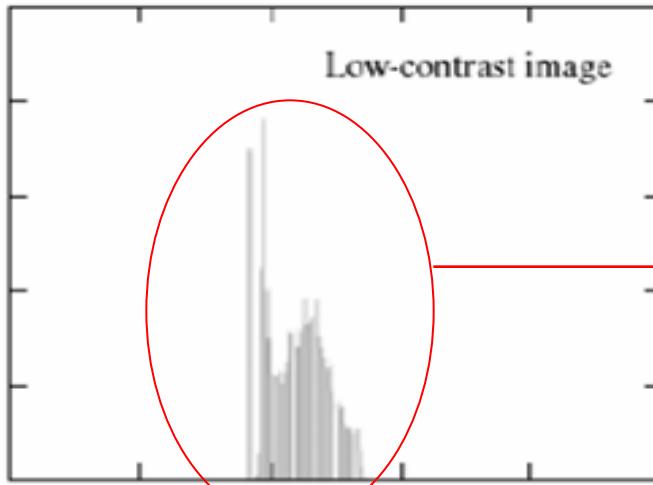
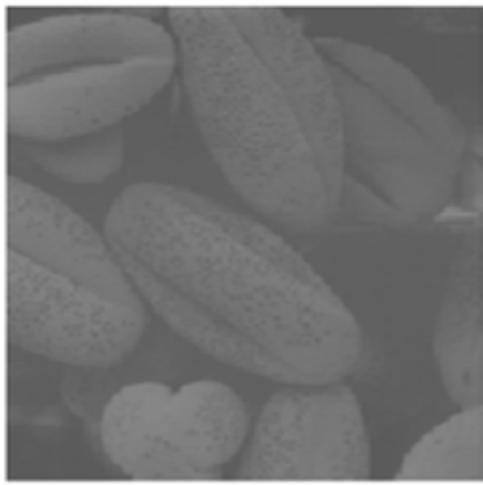


Karanlık imge

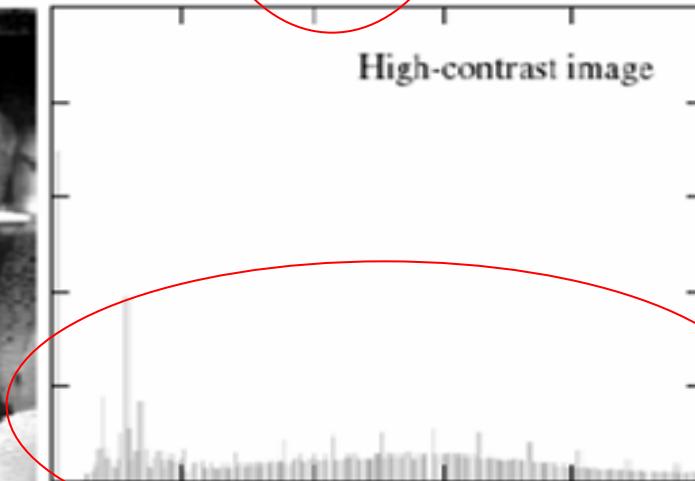


Parlak imge

Histogram



Karşılığının düşük
imge



Karşılığının yüksek
imge

Sürekli Genlik Rastlantı Değişkenleri



- Let χ be a continuous amplitude random variable $\chi \in (-\infty, +\infty)$.

$f_\chi(x)$: the **probability density function** of χ ,

$F_\chi(x)$: the **probability distribution function** of χ .

$$f_\chi(x)dx = \text{Probability}(x \leq \chi < x + dx)$$

$$F_\chi(x) = \text{Probability}(\chi \leq x)$$

- Properties:

$$F_\chi(x) = \int_{-\infty}^x f_\chi(t)dt \Rightarrow \frac{dF_\chi(x)}{dx} = f_\chi(x)$$

$$f_\chi(x) \geq 0 \Rightarrow F_\chi(x) \geq 0, \quad F_\chi(x + dx) - F_\chi(x) \geq 0$$

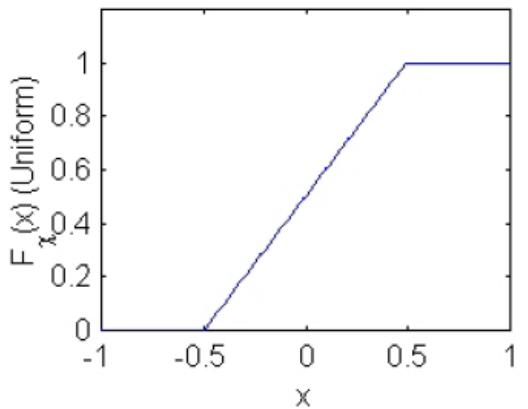
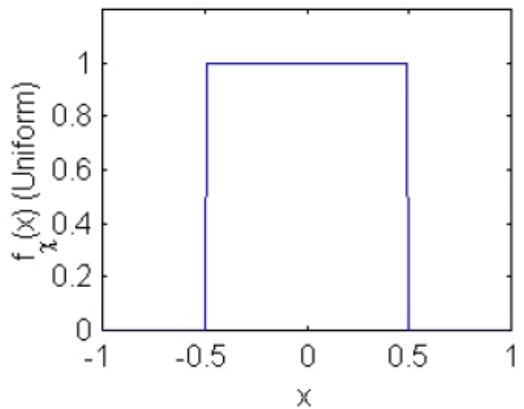
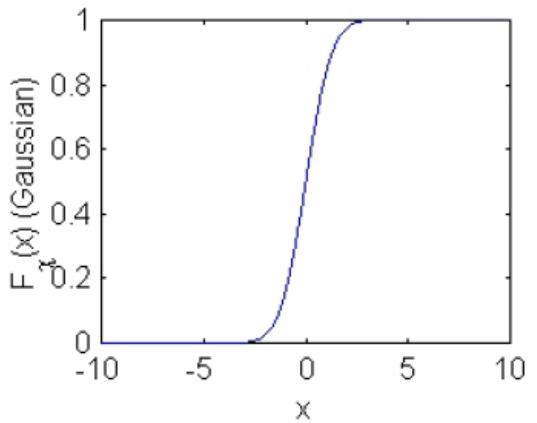
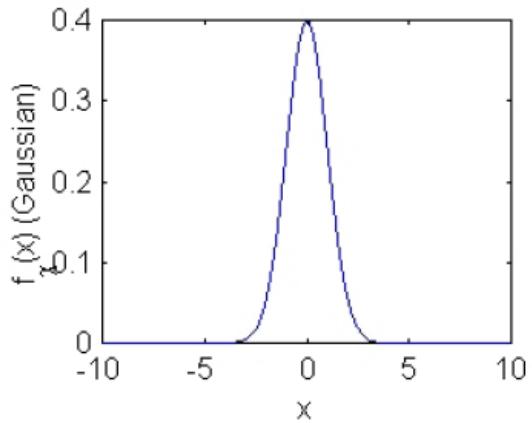
$F_\chi(x)$ is a non-decreasing function.

$$\int_{-\infty}^{+\infty} f_\chi(t)dt = 1 \Rightarrow f_\chi(x)|_{x=+/-\infty} = 0$$

$$F_\chi(x)|_{x=+\infty} = 1$$

$$F_\chi(x)|_{x=-\infty} = 0$$

Sürekli Genlik Rastlantı Değişkenleri



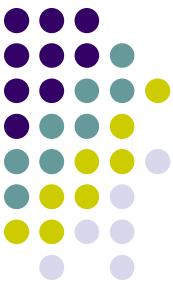
Gaussian:

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Uniform ($a < b$):

$$f_x(x) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & \text{otherwise} \end{cases}$$

Ortalama ve Değişinti



- Mean (μ):

$$\mu = \int_{-\infty}^{+\infty} x f_\chi(x) dx$$

Analogy: Average price of apples

- “I bought $f_\chi(x)dx$ many apples at a price of x , ...”
- “Total price I paid: $P = \int_{-\infty}^{+\infty} x f_\chi(x) dx$.
- “Total number of apples I purchased: $N = \int_{-\infty}^{+\infty} f_\chi(x) dx = 1$.
- “My average price for the overall purchase: $\mu = P/N$.

- Variance (σ^2):

$$\sigma^2 = \int_{-\infty}^{+\infty} (x - \mu)^2 f_\chi(x) dx$$

Ayrık Genlik Rastlantı Değişkenleri



- Let Θ be a discrete amplitude random variable.

$\Theta = x_i$ for some $i, \dots, -1, 0, 1, \dots$

x_i are a sequence of possible values for Θ .

$p_\Theta(x_i)$: the **probability mass function** of Θ ,

$F_\Theta(x_i)$: the **probability distribution function** of Θ .

$$p_\Theta(x_i) = \text{Probability}(\Theta = x_i)$$

$$F_\Theta(x_i) = \text{Probability}(\Theta \leq x_i)$$

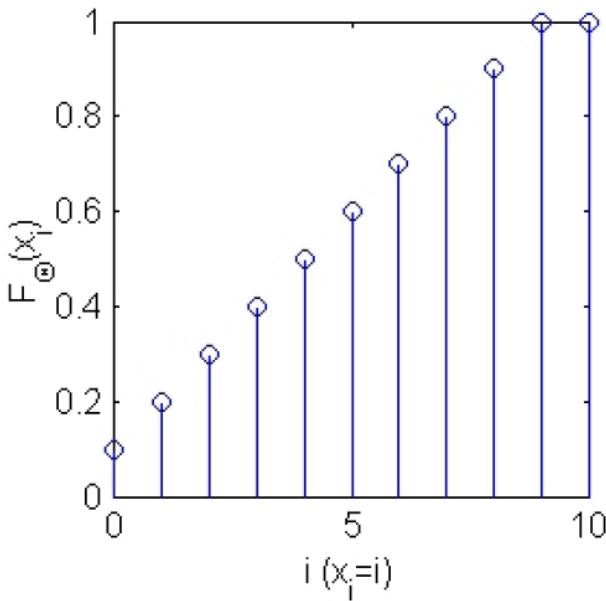
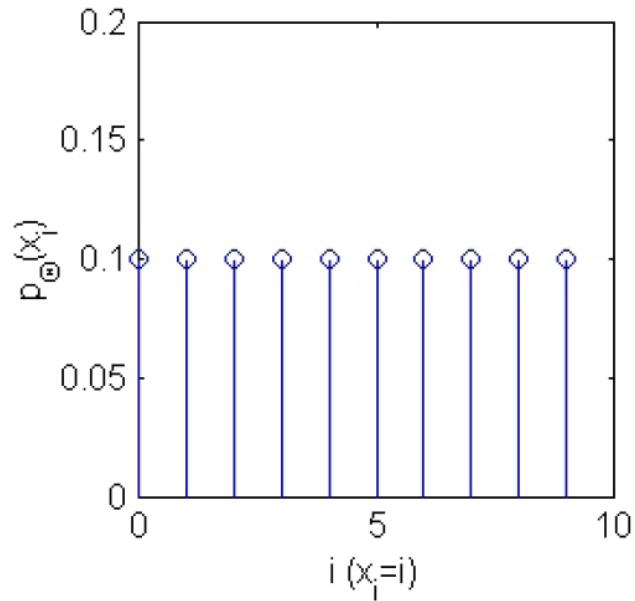
- Properties:

$$F_\Theta(x_i) = \sum_{j=-\infty}^{j=i} p_\Theta(x_j)$$

$$p_\Theta(x_i) = F_\Theta(x_i) - F_\Theta(x_{i-1}) \geq 0$$

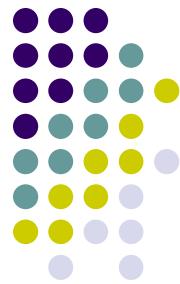
$$\sum_{j=-\infty}^{j=+\infty} p_\Theta(x_j) = 1$$

Ayrık Genlik Rastlantı Değişkenleri



The probability mass and distribution functions for a uniform, discrete amplitude random variable.

Olasılık Yoğun Fonk. Olarak Histogram



- For a given image A , consider the image pixels as the realizations of a discrete amplitude random variable “ A ”.
 - For example suppose we toss a coin (Heads=255 and Tails=0) $N \times M$ times and record the results as an N by M image matrix.
- Define the sample probability mass function $p_A(l)$ as the probability of a randomly chosen pixel having the value l .

$$p_A(l) = \frac{h_A(l)}{NM}$$

- Note that the sample mean and variance we talked about in Lecture 2 can be calculated as:

$$\begin{aligned}m_A &= \sum_{l=0}^{255} l p_A(l) \\ \sigma_A^2 &= \sum_{l=0}^{255} (l - m_A)^2 p_A(l)\end{aligned}$$

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.

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



Histogram Eşitleme



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

blok

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

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

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

histogram

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

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	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
55	4	65	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		

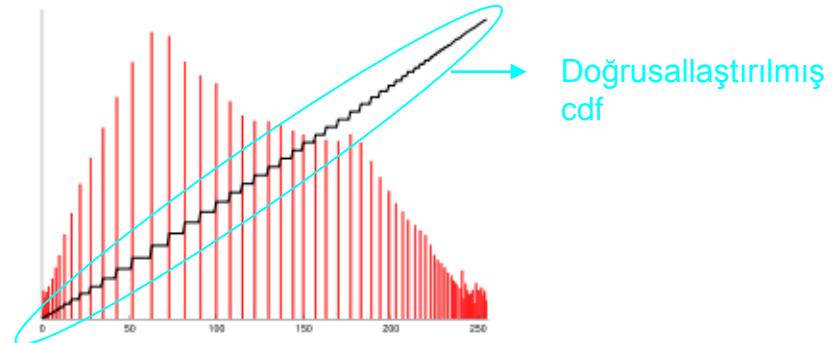
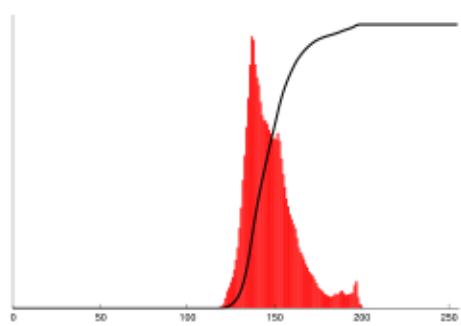
cdf

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

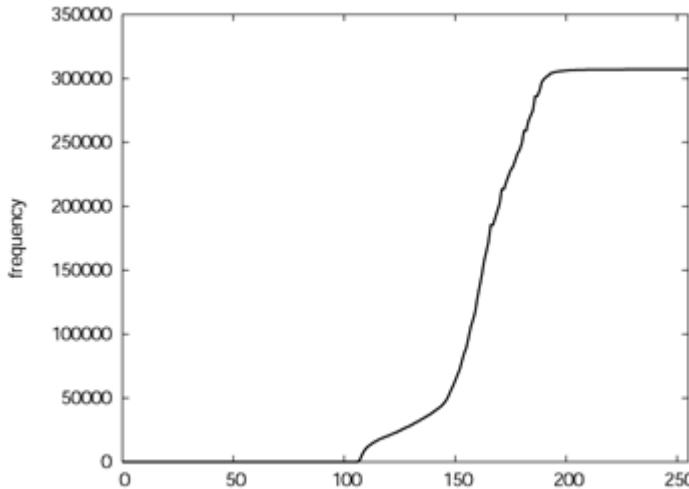
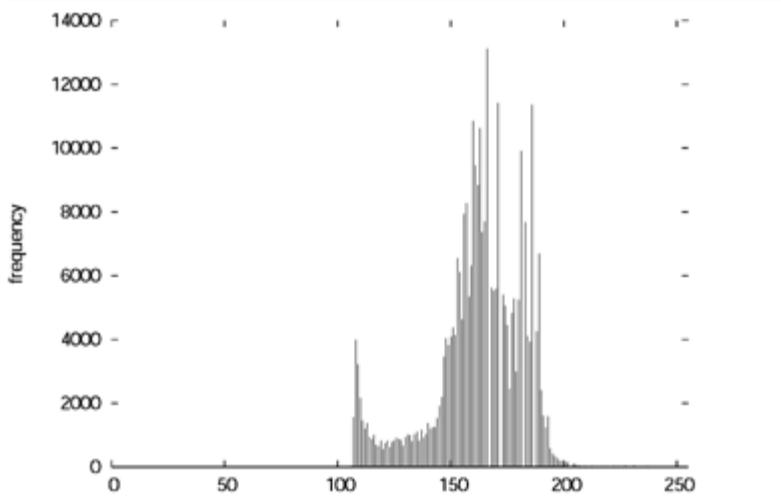
Histogram Eşitleme



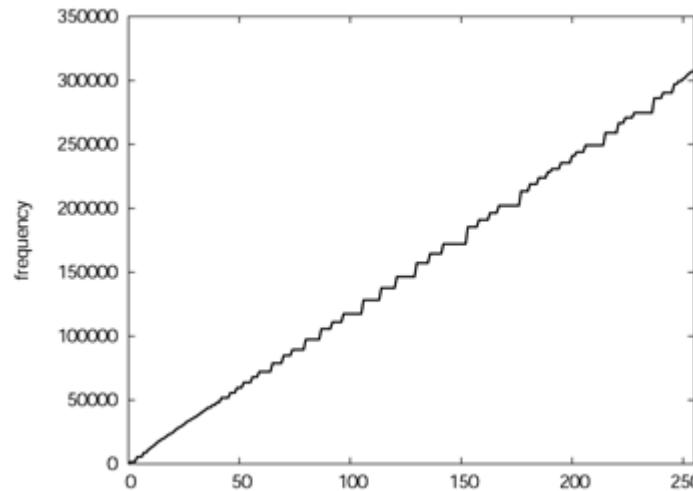
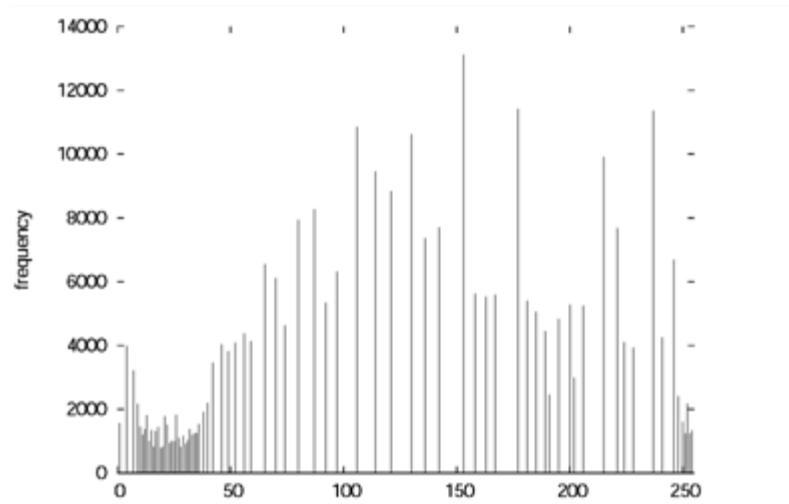
- İmgenin olasılık dağılım fonksiyonu doğrusallaştırılmaktadır.



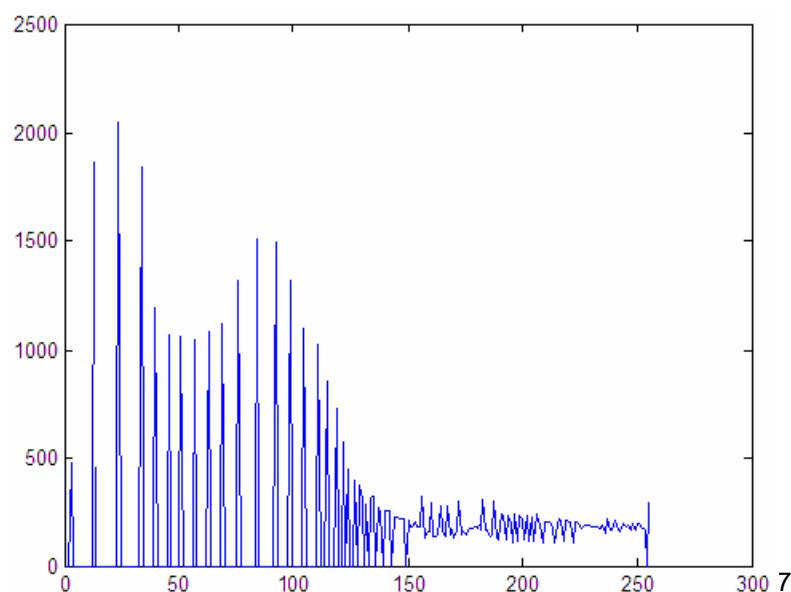
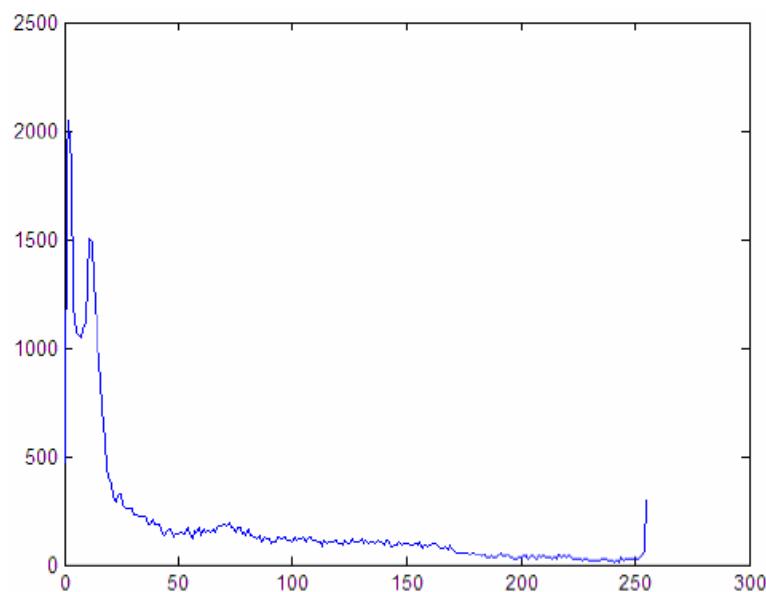
Histogram Eşitleme



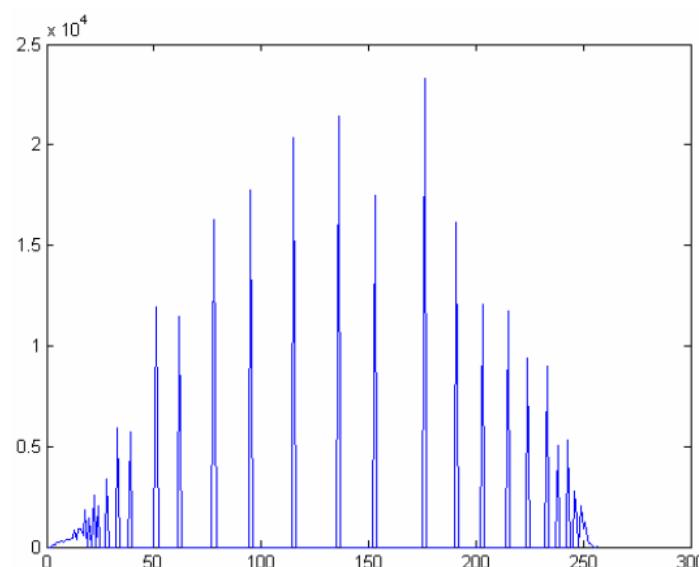
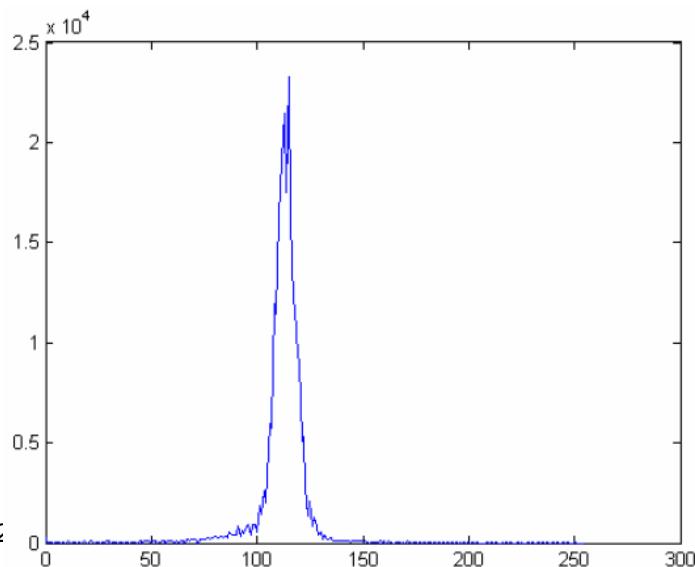
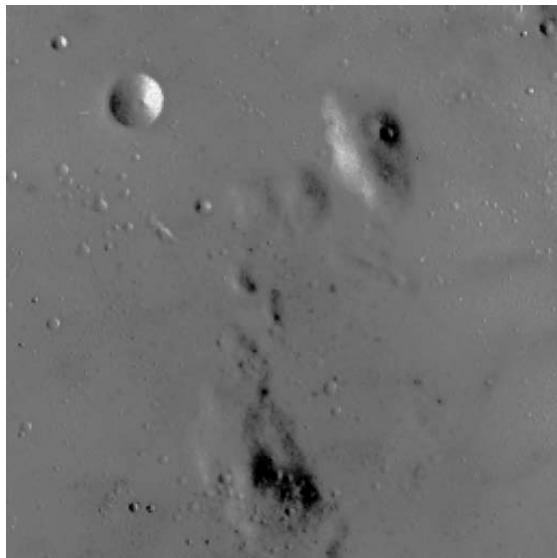
Histogram Eşitleme



Histogram Eşitleme



Histogram Eşitleme



Nicemleme



- Let $t_n \in \{0, 1, \dots, 255\}$ denote a sequence of **thresholds** ($n = 0, \dots, P - 1$).
- Consider the P “half-open, discrete intervals” $R_n = [t_n, t_{n+1})$ ($t_0 = 0, t_P = 256$).
- Let $r_n \in R_n$ be the **reproduction level** of the interval R_n .
- Define the quantizing point function or the **P-level quantizer** $Q(l)$ in terms of the R_n, r_n (or equivalently in terms of t_n, r_n) as follows:

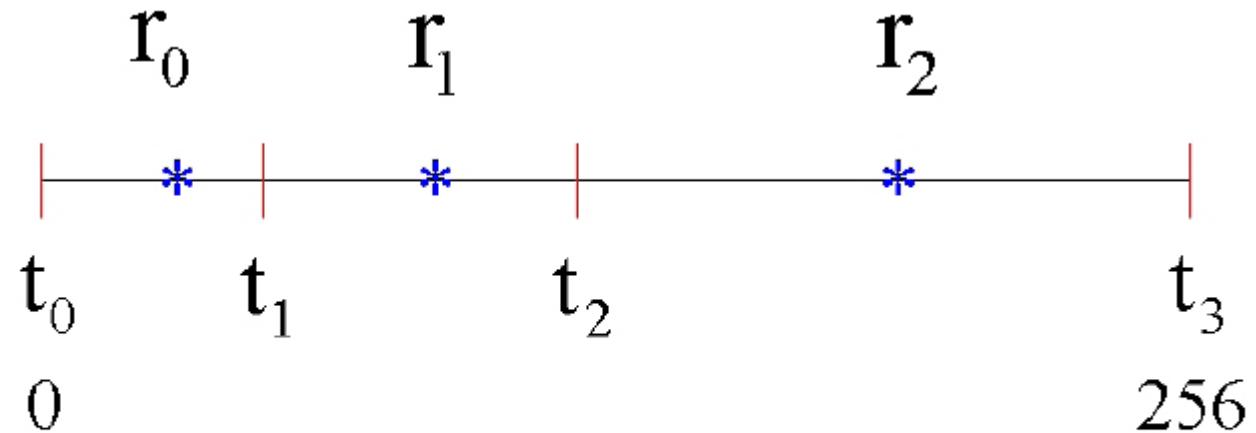
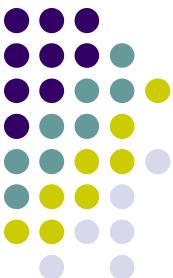
$$Q(l) = \{r_k | l \in R_k, k = 0, \dots, P - 1\}$$

i.e., $l \in R_k \Leftrightarrow Q(l) = r_k$.

- Quantizing an image A in matlab:

```
>> Q = zeros(256, 1); x = (0 : 255)';  
>> for i = 1 : P  
    Q = Q + r(i) * ((x >= t(i)) & (x < t(i + 1)));  
end;      % t(P + 1) = 256  
>> B = Q(A + 1);
```

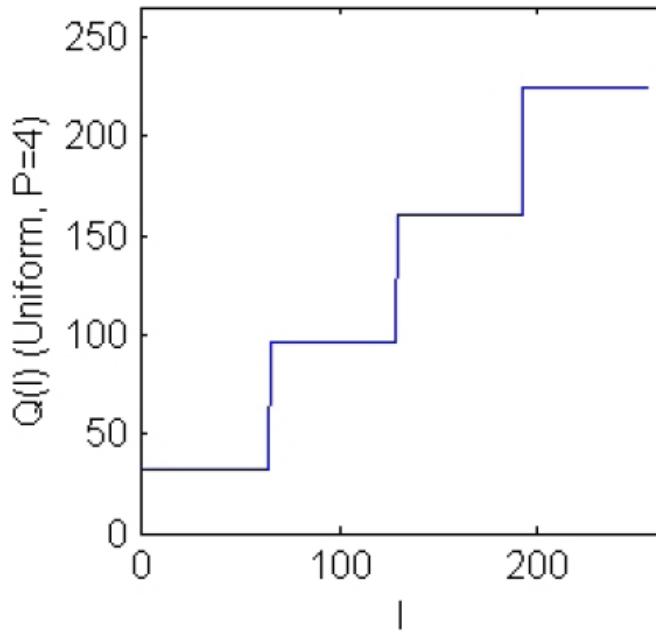
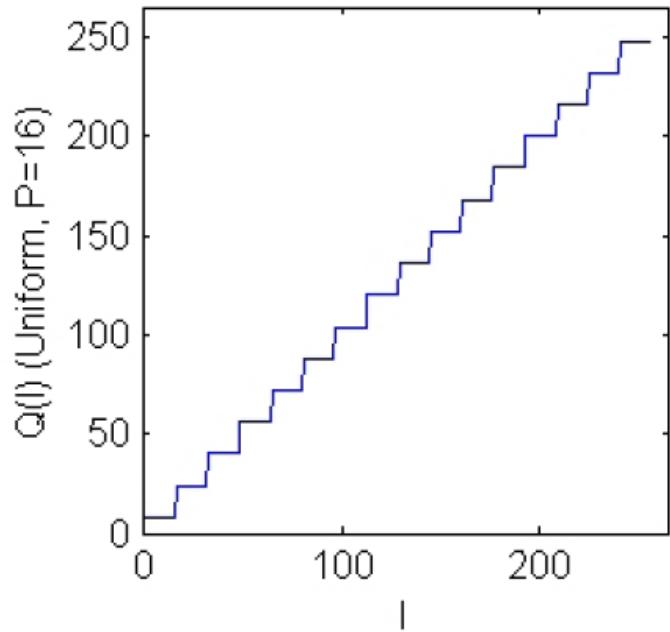
Nicemleme



- $R_n = [t_n, t_{n+1}]$.



- In uniform quantization $P = 256/\Delta$, $t_{n+1} - t_n = \Delta$, $\forall n$ and $r_n = \frac{t_n+t_{n+1}}{2}$.
- Δ is the **stepsize** of the uniform quantizer ($r_n = n\Delta + \Delta/2$).



Easy uniform quantization: `>> B = delta * floor(A/delta) + delta/2;`

Nicemleme



A



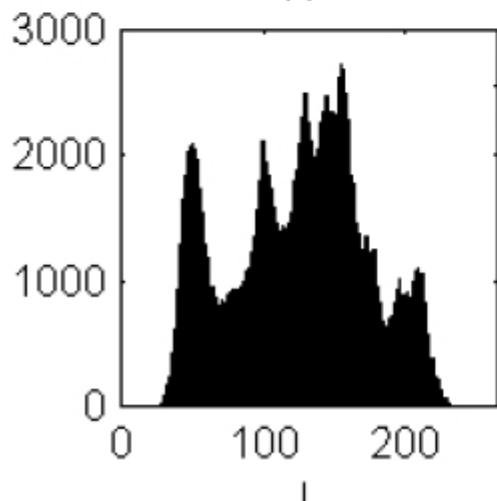
B1, P=64 ($\Delta=4$)



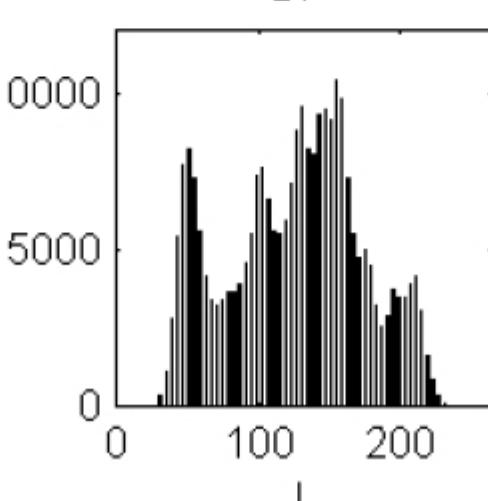
B2, P=32 ($\Delta=8$)



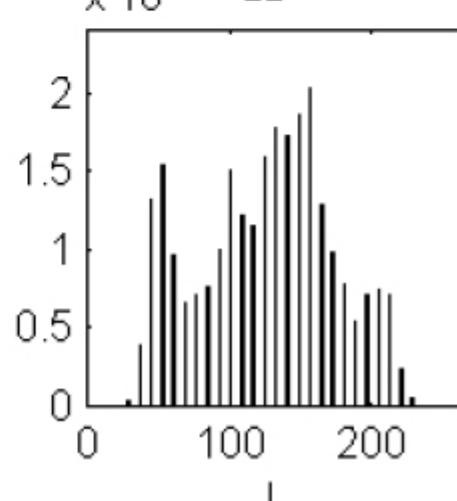
$h_A(l)$



$h_{B1}(l)$



$h_{B2}(l) \times 10^4$



Nicemleme



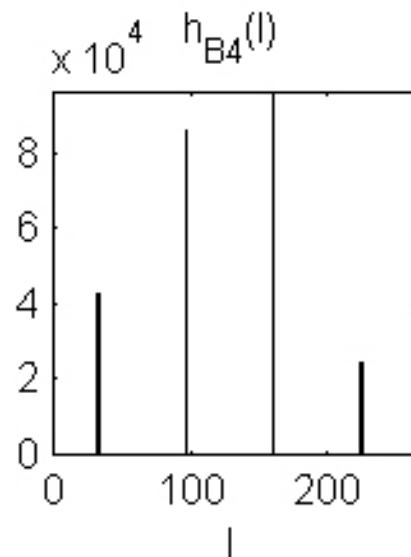
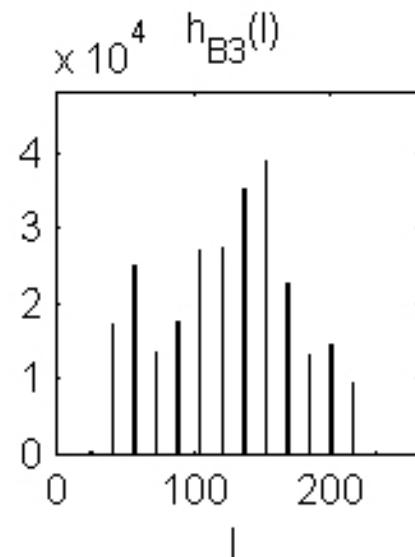
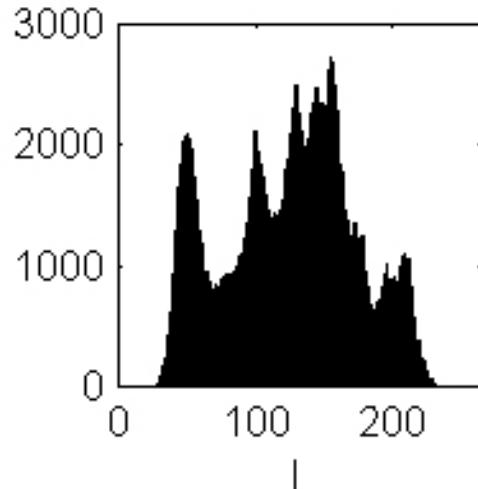
A



B3, P=16 ($\Delta=16$)



B4, P=4 ($\Delta=64$)





- The quantization error matrix is defined as $\mathbf{E} = \mathbf{A} - Q(\mathbf{A})$.
- The sample mean squared quantization error (MSQE) is:

$$\begin{aligned}\text{MSQE} &= \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (E(i, j))^2}{NM} \\ &= \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (A(i, j) - Q(A(i, j)))^2}{NM} \\ &= \sum_{l=0}^{255} (l - Q(l))^2 p_A(l)\end{aligned}$$

Example

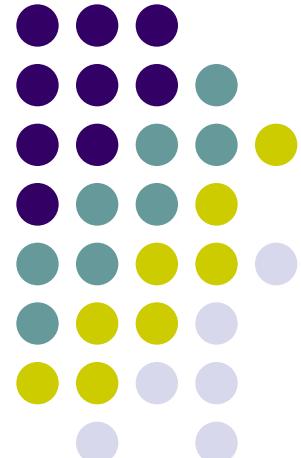
For the earlier example:

Δ	Quantized Image	MSQE
4	B1	1.50
8	B2	5.49
16	B3	22.18
64	B4	334.77

İMGE İŞLEME

Ders-4

Piksel Komşuluk İşlemleri



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İ işaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/elohab/kemalg/imge_web/odev.htm

İmgenin Ortalama ve Değişintisi



- Bir imgenin ortalaması (mean):

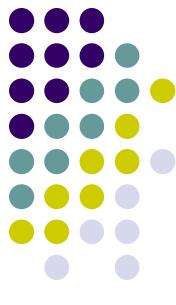
$$\mu = E(X) = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

- Bir imgenin değişintisi (variance):

$$\begin{aligned}\sigma^2 &= Var(X) = E((X - \mu)^2) \\ &= \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2\end{aligned}$$

- MATLAB'da 2-boyutlu matrisin ortalamasını almak için **mean2** işlevi kullanımaktadır.
- Değişinti hesabı için **std2** standart sapma bulma işlevi kullanılmaktadır. Daha sonra standart sapmanın karesi alınarak değişinti bulunabilmektedir.

İmgenin Ortalama ve Değişintisi



- MATLAB'da 8 bitlik bir İ imgesi açıp, imgenin her bir pikseline erişerek etrafından nxn boyutlu bir blok alın ve bloğun standart sapmasını hesaplayın.
- I ile aynı boyutta oluşturacağınız I2 imgesinin ilgili pikseline bulunan standart sapma değerini yazınız.
- Bu işlemi imgedeki bütün pikseller için yapınız.

Piksel Komşuluk İşlemleri



- Her bir piksel için yeni bir değer hesaplanmaktadır.
- İlgili pikselin yeni değeri, komşu piksellerin değerleri de dikkate alınarak bulunur.
- Kullanılacak piksellerin ağırlıkları, yapılacak işleme bağlı olarak değişmektedir.
- Kenar bulma, gürültü giderme, imge keskinleştirme, yumusatma gibi işlemlerde kullanılmaktadır.
- Hesapsal yükü, nokta işlemlerine göre oldukça fazla olabilmektedir.

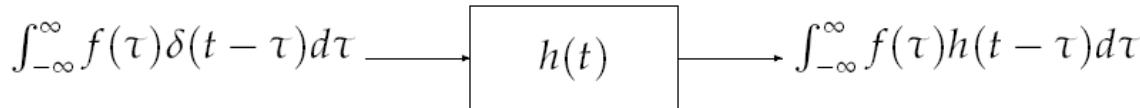
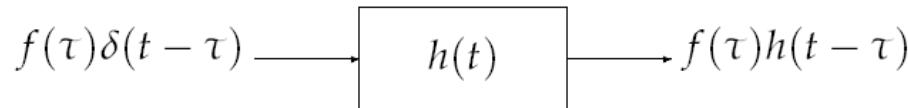
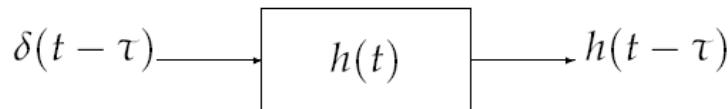
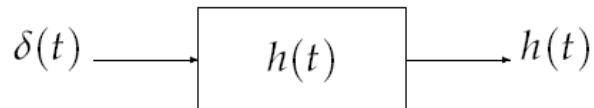
Evrişim (Convolution)



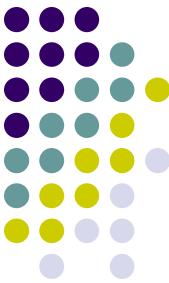
- İki fonksiyonun etkileşimi olarak ifade edilebilir.

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

- İmge işlemede sıkça kullanılmaktadır.
- Sistemin, giriş işaretine etkisini vermektedir.



Evrişim (Convolution)



- Evrişimin ayrık zamanlı 2-boyutlu ifadesi:

$$g(x, y) = k * f$$

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

k , evrişim çekirdeği (convolution kernel)

f , giriş imgesi

g , çıkış imgesi

(x, y) , ilgili piksel konumu

$(2m+1, 2n+1)$,
yükseklik ve düşey uzunluğu u

- Evrişim çekirdeği (kernel) genelde , evrişim maskesi (convolution mask), evrişim penceresi (convolution window) olarak da adlandırılabilir.

Evrişim (Convolution)

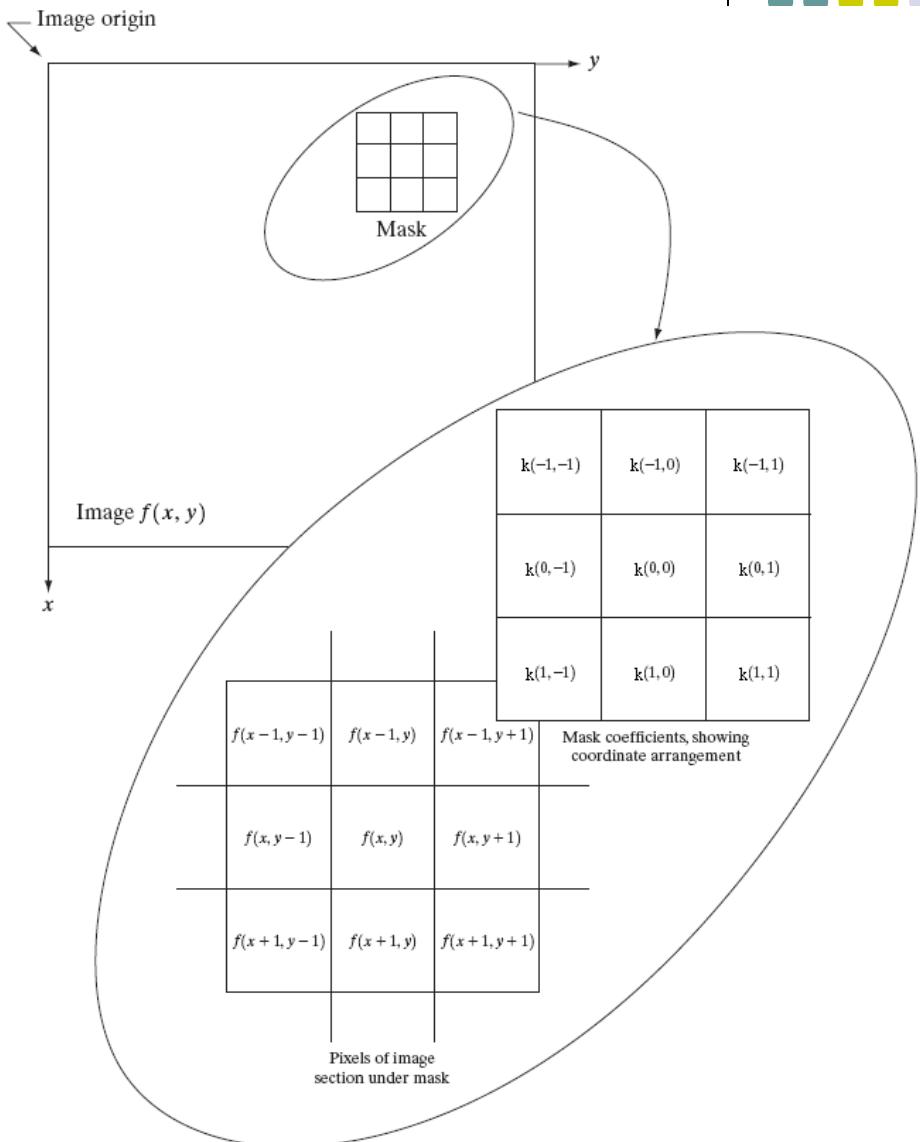


$$g(x, y) = k * f$$

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

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

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

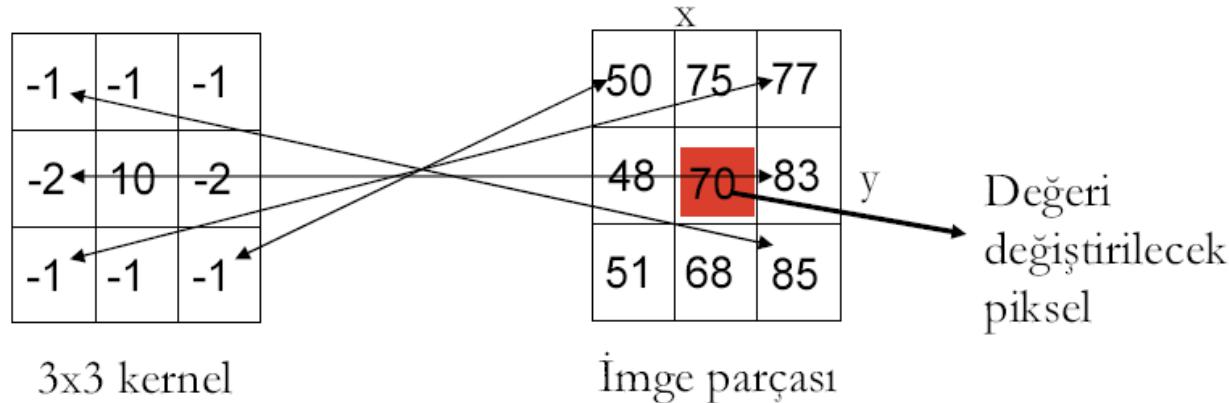


Evrişim (Convolution)



$$g(x, y) = k * f$$

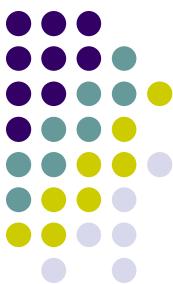
$$= \sum_{i=-m}^m \sum_{j=-n}^n k(i, j) f(x-i, y-j)$$



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

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

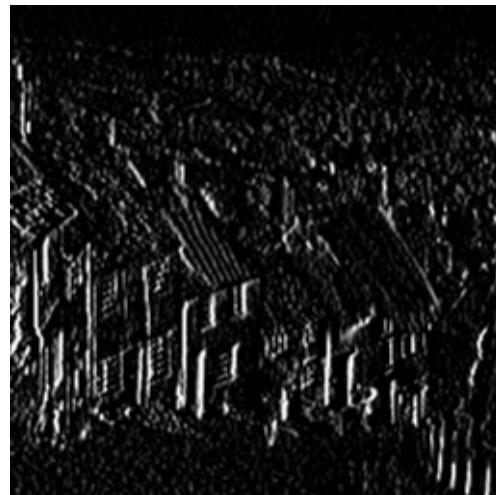
Evrişim (Convolution)



Giriş imgesi

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

Evrişim
çekirdeği



Çıkış imgesi

- MATLAB'da 2-boyutlu evrişim **conv2** işlevi ile yapılabilmektedir.
- Bunun yanında imge süzgeçlerken genellikle **imfilter** işlevi kullanılmaktadır.

Evrişim (Convolution)



Evrişim işleminde kenar bölgelerindeki taşıma durumunda olası işlemler:

- Kenar bölgelerini işlememe,
- Kenar bölgelerini kesme,
- Kenar bölgelerinde evrişim çekirdeğini kırpma,
- Kenar bölgelerini aynen kopyalama (imge boyutları büyür),
- Kenar bölgelerini aynalayarak kopyalama (imge boyutları büyür)...

Hesapsal yük:

- (m, n) boyutlu bir evrişim çekirdeği kullanıldığında bir piksel için çıkış değerinin hesaplanmasında gerekli işlem sayısı:

$$(m \times n)[\text{çarpma}] + (m \times n - 1)[\text{toplama}]$$

Evrişim (Convolution)



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

Delta fonksiyonu



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

Kaydır ve çıkart



Evrişim (Convolution)



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

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



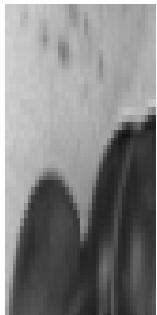
Uzamsal Frekans Kavramı



- İmgede pikseller arasındaki yumuşak geçişler *uzamsal düşük frekanslara* karşılık gelir.



- Sert geçişler (kenarlar, nesne sınırları...) *uzamsal yüksek frekanslara* karşılık gelir.



Evrişim (Convolution)-Yumuşatma



- En temel evrişim çekirdeğidir.
- İmgedeki gürültü etkilerini azaltır.
- Kenarları yumusatır.

$$1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

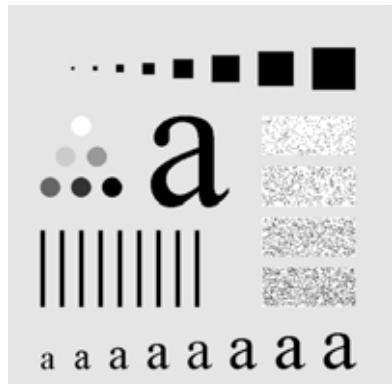
$$1/25 \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



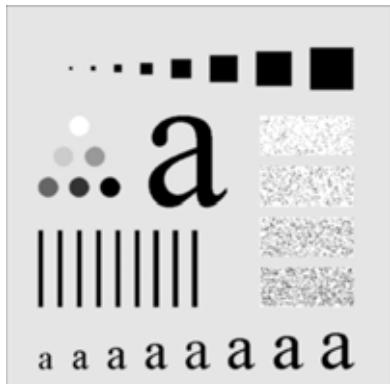
Evrişim (Convolution)-Yumuşatma



- Çekirdek boyutunun yumuşatmaya etkisi:



Orjinal imgé



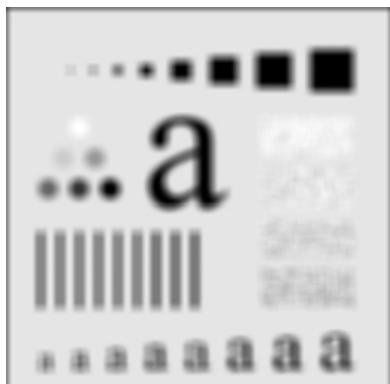
3x3



5x5



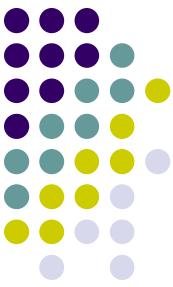
9x9



15x15



35x35



Evrişim (Convolution)-Yumuşatma

- Ağırlıklı ortalama alma işlemi de yapılabilmektedir.

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

1/15 × $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

- Kenar bölgelerindeki yumuşamayı azaltmak için kontrollü ortalama alma yapılabilir.

$$g(x, y) = \begin{cases} \frac{1}{ws \times ws} \sum_i \sum_j f(x-i, y-j) & , \quad \left| f(x, y) - \frac{1}{ws \times ws} \sum_i \sum_j f(x-i, y-j) \right| < T \\ f(x, y) & , \text{ già} \end{cases}$$

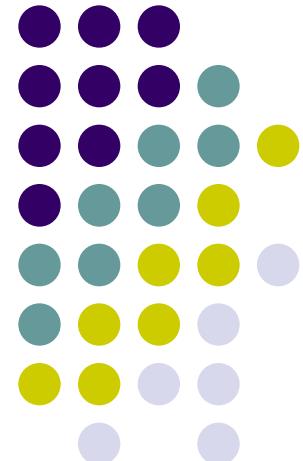
↓

MATLAB'da uygulayınız

İMGE İŞLEME

Ders-5

Piksel Komsuluk İşlemleri-2



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İ işaret ve Görüntü İşleme Laboratuvarı

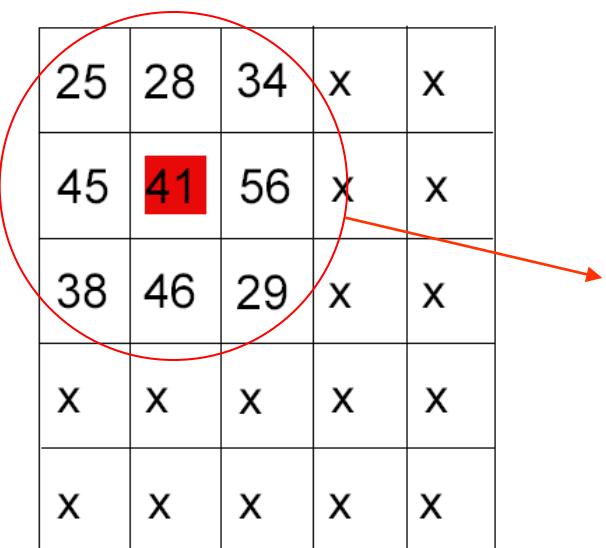
(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/lohab/kemalg/imge_web/odev.htm



Ortanca (Median) Süzgeç

- Süzgeçleme işlemi, pencere içerisindeki piksellerin sıralanması temelinde yapmaktadır.
- Doğrusal olmayan bir süzgeçlemedir.
- Dürtü ve tuz-biber gürültülerinin giderilmesinde etkin kullanım sağlamaktadır.
- İmgenin kenar bölgelerini bozmaktadır.



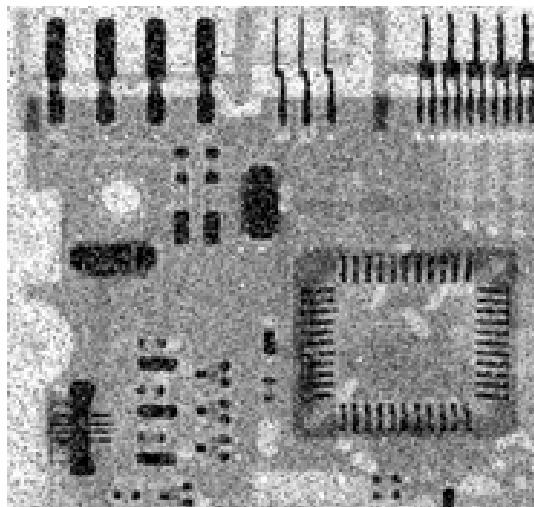
25, 28, 29, 34, 38, 41, 45, 46, 56

Yeni piksel değeri

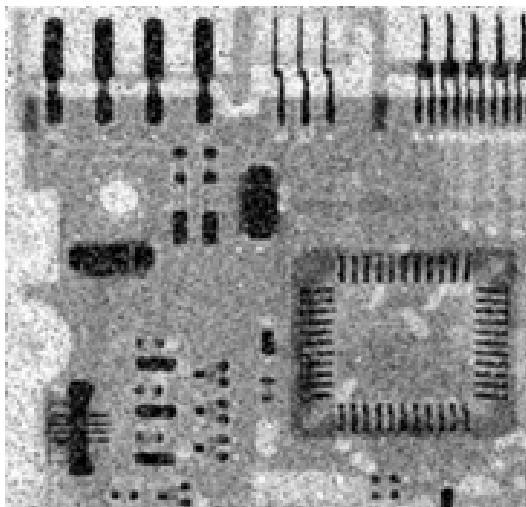


Ortanca (Median) Süzgeç

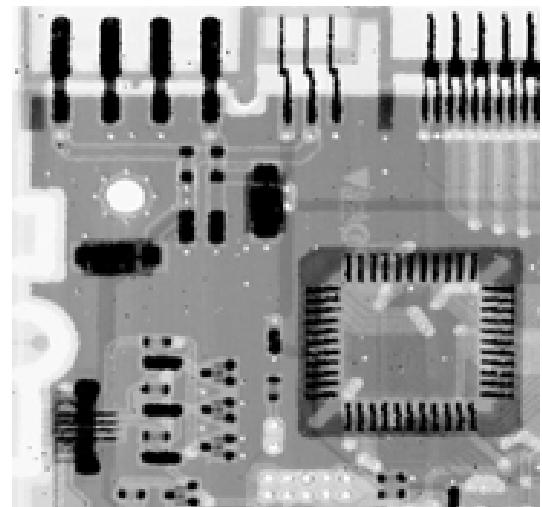
- Tuz ve biber gürültüsünün (salt and pepper noise) ortanca süzgeç ile giderilmesi



Gürültü
eklenmiş imge



3x3 ortalama
süzgeç ile
gürültü
giderme



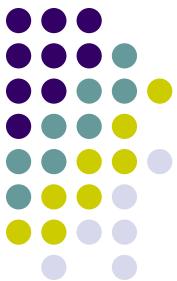
3x3 ortanca
süzgeç ile
gürültü
giderme



Evrişim (Convolution)-Keskinleştirme

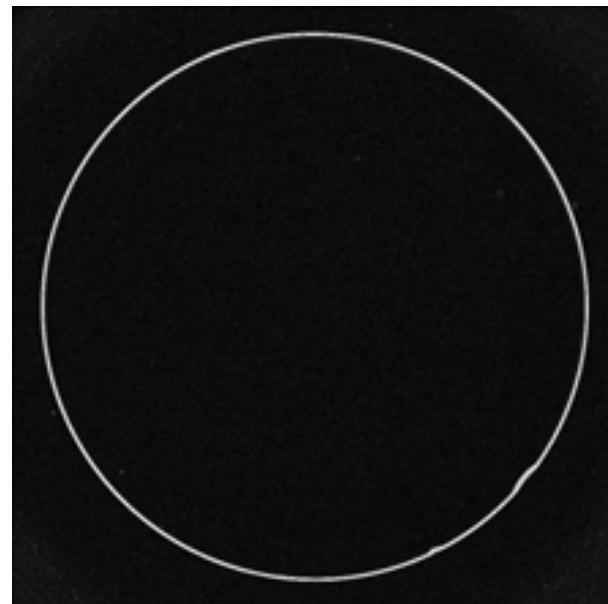
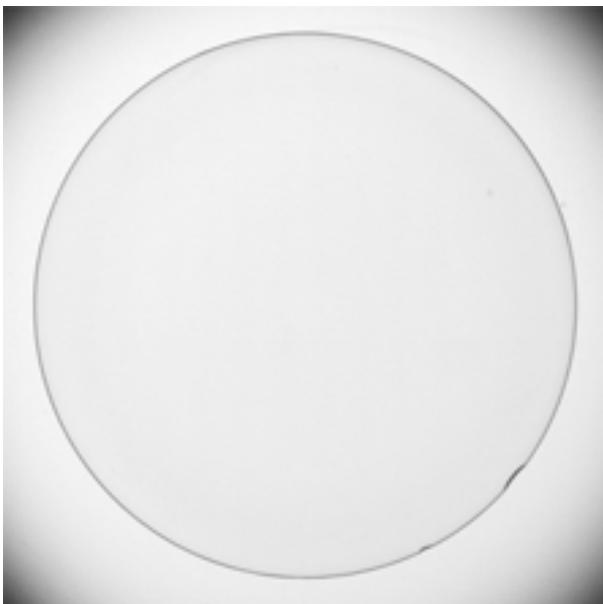
- Kenar: İmgedeki keskin ışıklılık değişimleridir.
- Keskinleştirme işlemindeki temel hedef detayları daha görünür hale getirmek ve bulanık bölgelerden detay çıkartmaya çalışmaktadır.
- Keskinleştirme işlemi, çıktı kalitesini artırma, tıbbi görüntüleme, endüstriyel denetim, kendi kendine dolaşan robot gibi uygulamalarda kullanılmaktadır.





Evrişim (Convolution)-Kenar Bulma

- Kenar bulma imge içerisindeki anlamlı kenarların bulunması olarak ifade edilmektedir.
- Bölümlemede nesne sınırlarının bulunması, tanımada örüntü çıkartma, hareket analizinde bölgeleri takip etme gibi uygulamalarda kenar bulma kullanılmaktadır.





Evrişim (Convolution)-Keskinleştirme

- Keskinleştirme işlemi, sayısal türevleme kullanılarak farklı yollarla yapılabilmektedir.
- Temelde, türev alma işleminin yanıtı, imge operatörün uygulandığı noktadaki süreksizlik ile orantılıdır.
- Tek boyutlu bir fonksiyon için 1. dereceden türev:

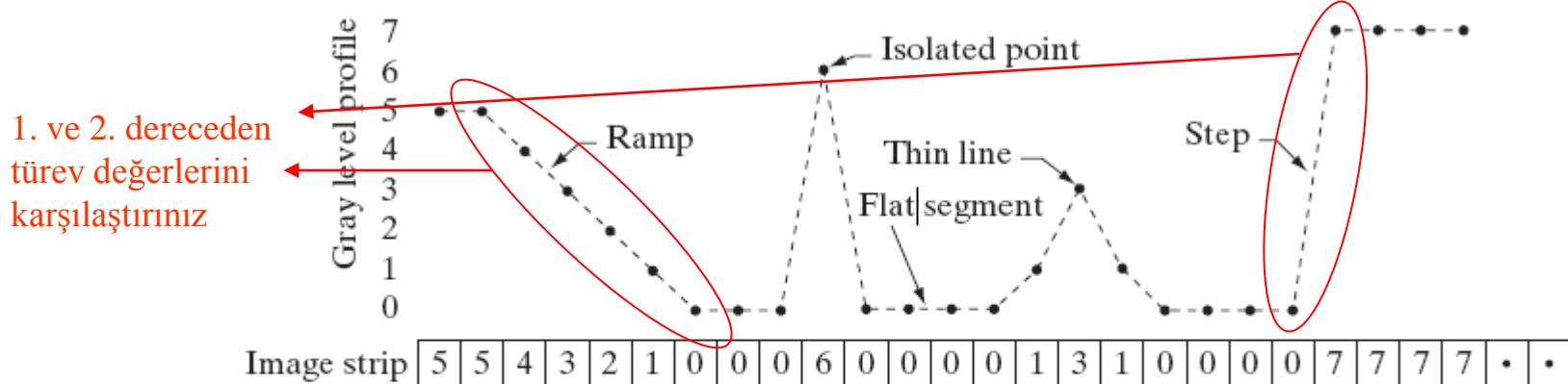
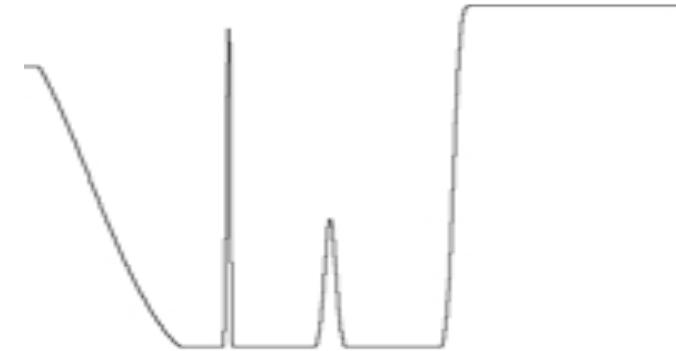
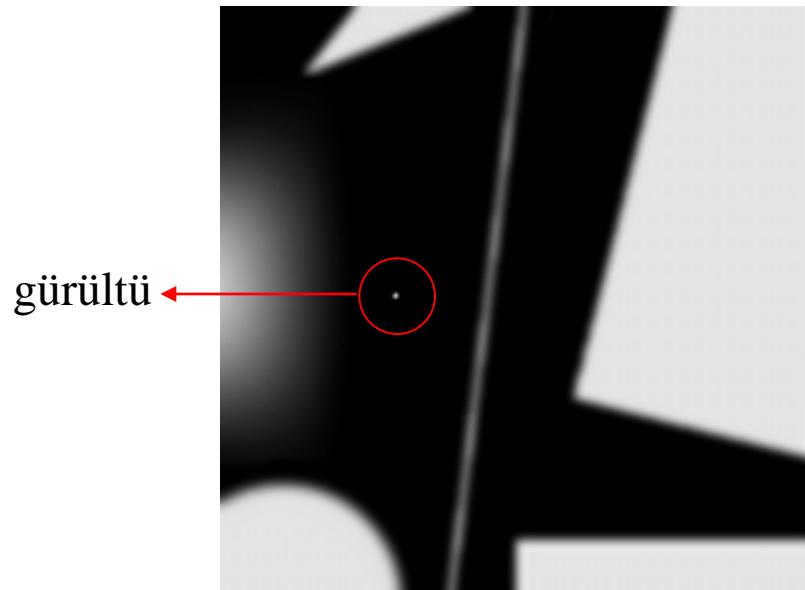
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- 2. dereceden türev:

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



Evrişim (Convolution)-Keskinleştirme



First Derivative	-1	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	
Second Derivative	-1	0	0	0	0	1	0	6	-12	6	0	0	0	1	1	-4	1	1	0	0	0	7	-7	0	0



Evrişim (Convolution)-Keskinleştirme

- 1. dereceden türev kalın kenarlar üretmektedir.
- 2. dereceden türev, detay bölgelerinde daha fazla tepki vermektedir (örn; dikey ince çizgi ve gürültü bölgeleri).
- 2. dereceden türev, kenar bölgelerinde ve nokta değişimlerinde daha fazla tepki vermesinden dolayı, keskinleştirmede daha fazla tercih edilmektedir.

2. Dereceden Türev Kullanımı - Laplacian Filtresi:

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

ayırık şekilde:

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

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



Evrişim (Convolution)-Keskinleştirme

- Tek bir ifade ile 2-B Laplacian:

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



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

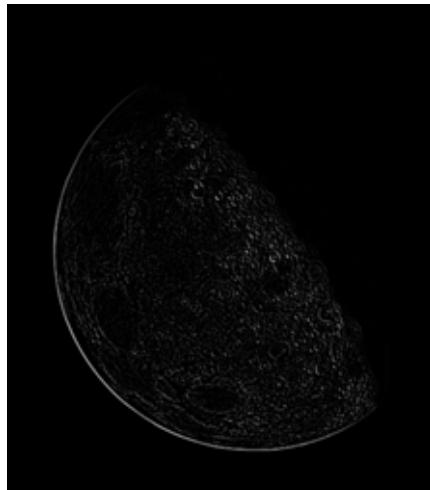
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Evrişim (Convolution)-Keskinleştirme

- 2-B Laplacian çekirdeği kullanılarak süzgeçlenen imge ile orjinal imge kullanılarak keskinleştirilmiş imge aşağıdaki şekilde elde edilir:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{değinemerkez değer negatif ise} \\ f(x, y) + \nabla^2 f(x, y) & \text{değinemerkez değer pozitif ise} \end{cases}$$





Evrişim (Convolution)-Kenar Bulma

1. Dereceden Türev Kullanımı – Eğim (Gradyan-The Gradient):

- 2-B sütun vektörü olarak gradyan:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Bu vektörün genliği:

$$\nabla f = \text{mag}(\nabla f) = \left[G_x^2 + G_y^2 \right]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

- Gradyeni bulurken kare ve karekök alma işlemlerinin hesapsal yükünü azaltmak için pratikte mutlak toplam kullanılmaktadır:

$$\nabla f \approx |G_x| + |G_y|$$



Evrişim (Convolution)-Kenar Bulma

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

$$G_x = z_8 - z_5$$

$$G_y = z_6 - z_5$$

$$\nabla f \approx |G_x| + |G_y|$$

$$\approx |z_8 - z_5| + |z_6 - z_5|$$

- Robert cross gradient operator:

$$G_x = z_9 - z_5$$

$$G_y = z_8 - z_6$$

$$\nabla f \approx |G_x| + |G_y|$$

$$\approx |z_9 - z_5| + |z_8 - z_6|$$

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

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

- Boyutu çift sayılarından (2×2) oluşan maske yerine 3×3 maske oluşturursak:

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

Sobel
Operatörleri

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

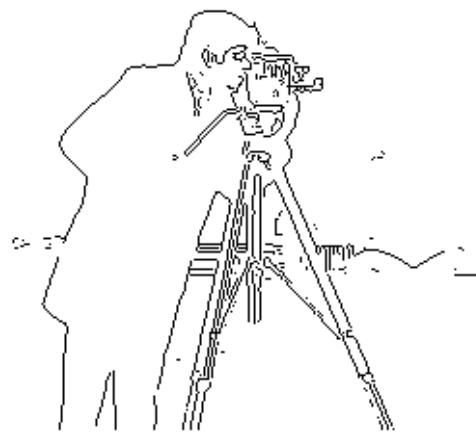


Evrişim (Convolution)-Kenar Bulma

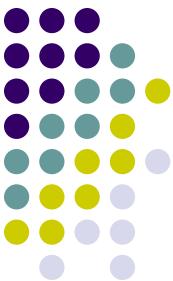
- Prewitt Operatörü:

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

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



- Not: Çekirdekler kullanılarak elde edilen imgenin eşiklenmesi ile ikili kenar imgesi oluşmaktadır.



Evrişim (Convolution)-Kenar Bulma

Kirsch Operatörü:

- Örütü tanımada şablon eşlestirmede kullanılmaktadır.
- Kenar yönlerine çok duyarlıdır.

$$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}$$

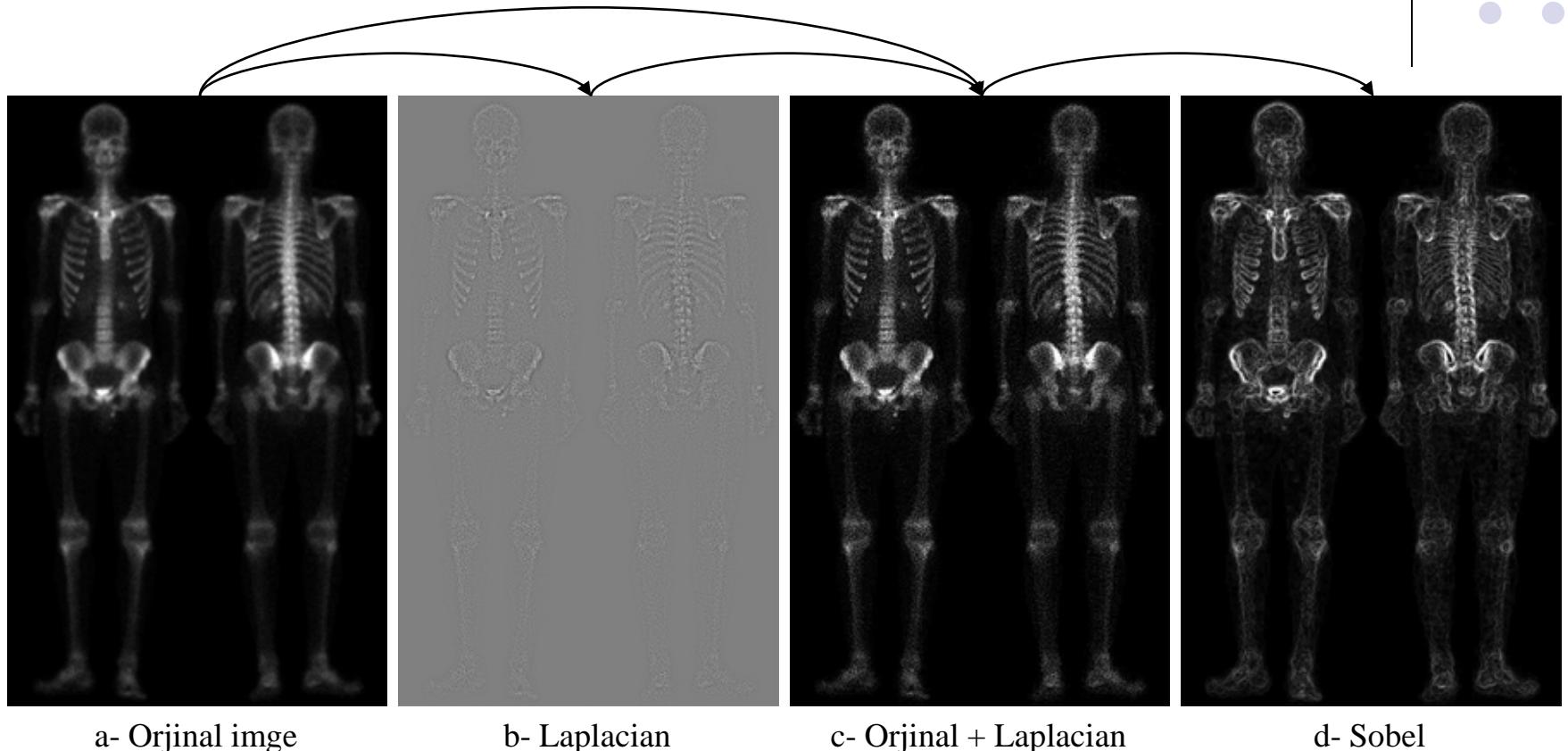
$$h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix}$$

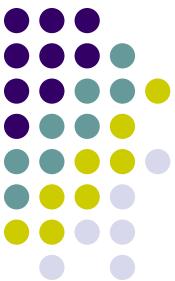
$$h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}$$



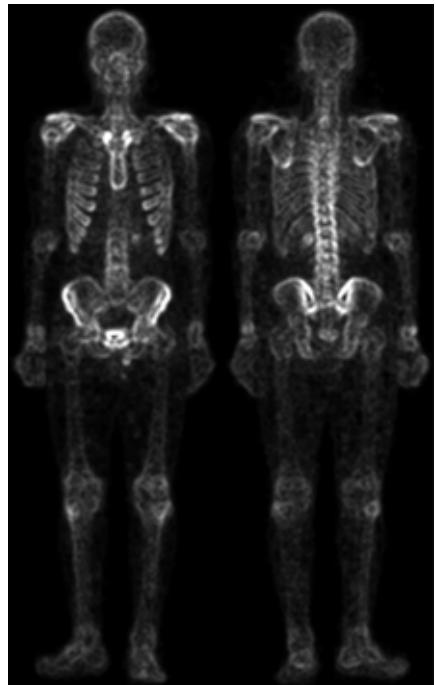


Uygulama

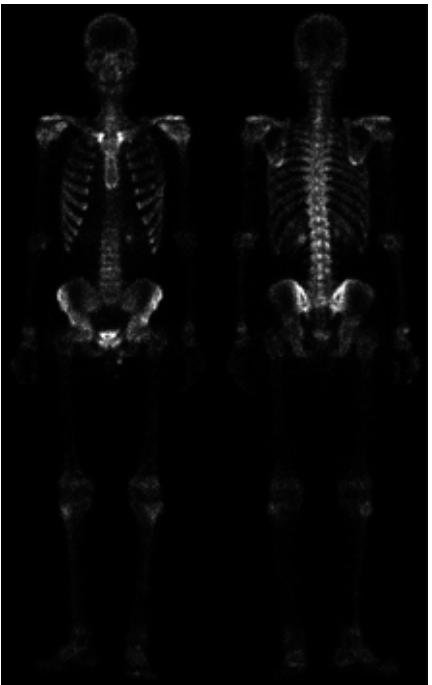




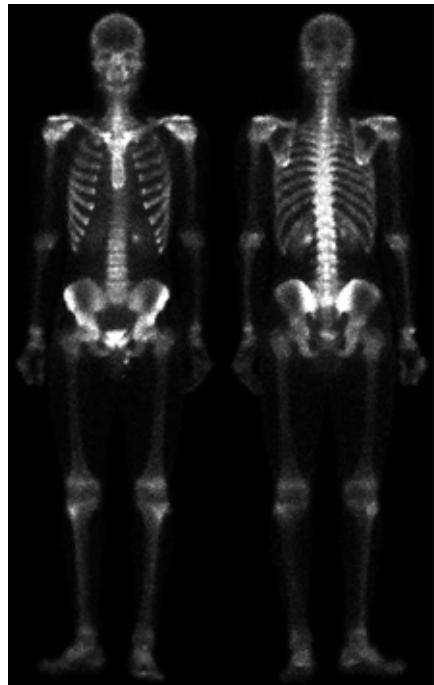
Uygulama-devam



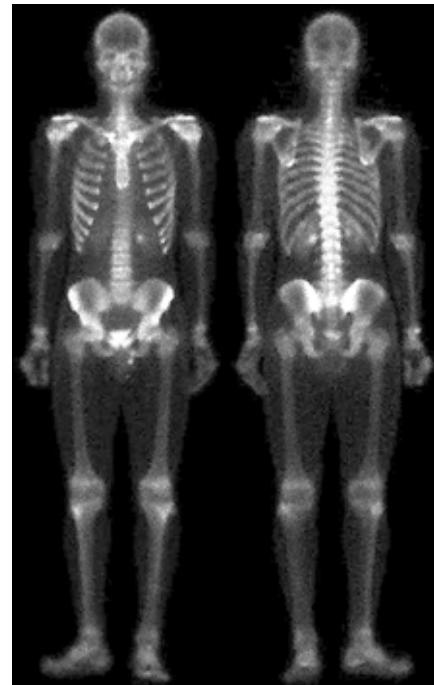
e- 5x5 ortalama



f-
(c x e)

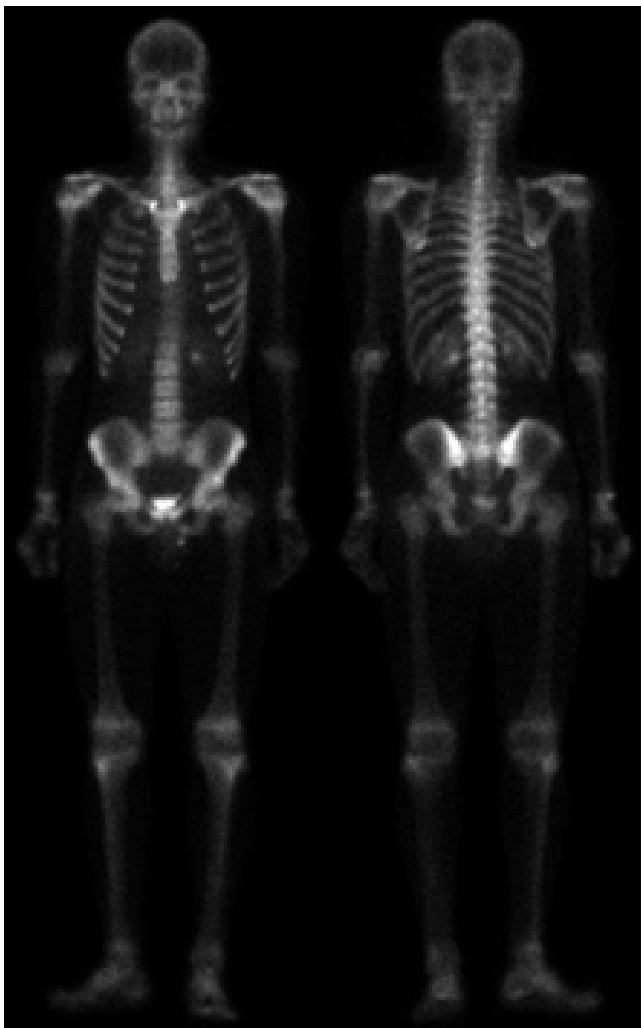
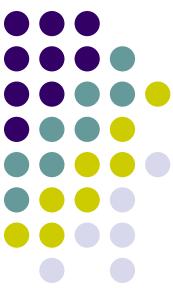


g-
(a+f)



h- İmge
pekiştirme

Uygulama-devam





İlenti (Correlation)

- İki işaret ya da imge arasındaki ilişkinin bulunması,
- Bir imgenin içerisinde imge parçası arama gibi işlemlerde kullanılmaktadır.

$$r(x, y) = \frac{\sum_{i=-m}^m \sum_{j=-n}^n (h(i, j) - \bar{h})(f(x+i, y+j) - \bar{f})}{\sqrt{\left(\sum_{i=-m}^m \sum_{j=-n}^n (h(i, j) - \bar{h})^2 \right) \left(\sum_{i=-m}^m \sum_{j=-n}^n (f(x+i, y+j) - \bar{f})^2 \right)}}$$

h : aracak blok

f : aramanın yapılacak imge

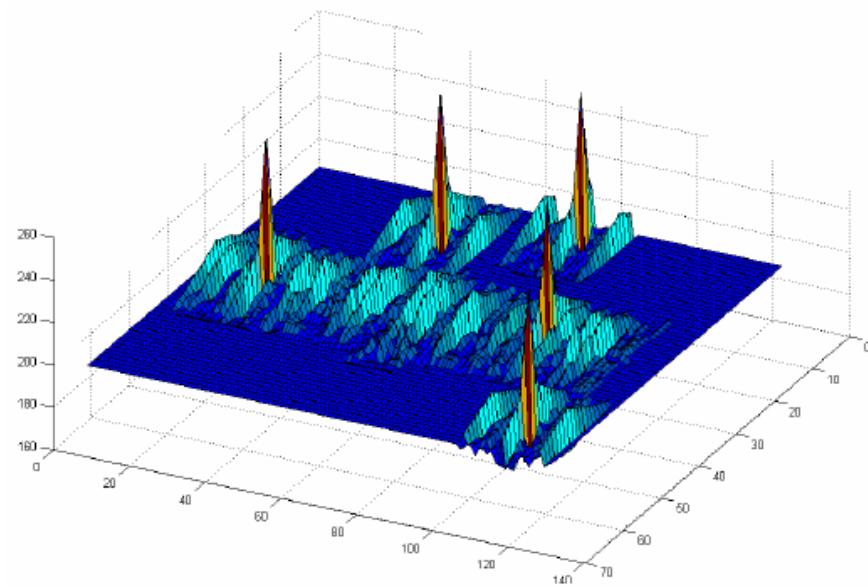
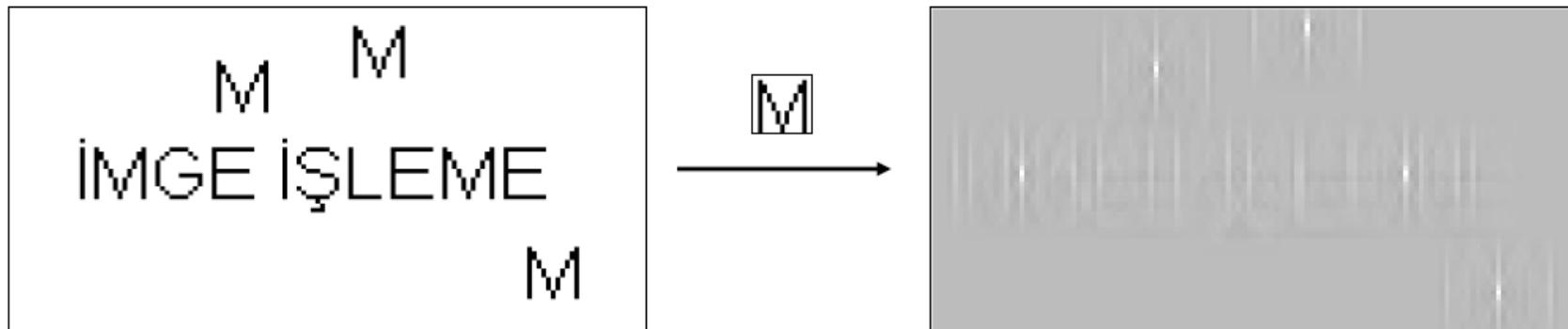
\bar{h} : blok ortalaması

\bar{f} : imge ortalaması

(x, y) : ilgili piksel konumu



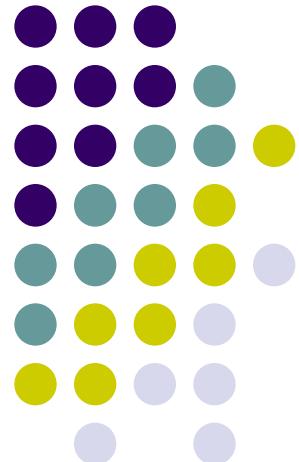
İlenti (Correlation)



İMGE İŞLEME

Ders-6

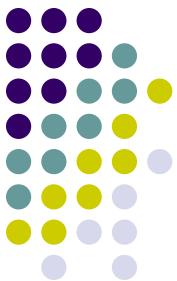
Renkli İmge İşleme – Renk Uzayları



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İ işaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/lohab/kemalg/imge_web/odev.htm

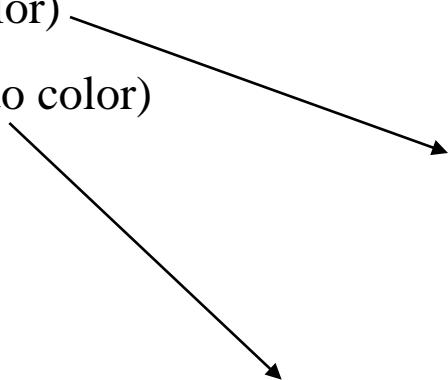


Renk

- Nesne tanımlama ve sahneden çıkartmada önemli bir tanımlayıcıdır.
- İnsan gözü ıshıklılık tonlarını ayırt etmede zayıfken, renk ve ıshık tonlarının bir arada olması durumunda ayırt edicilik çok artmaktadır.
- Renkli imgé işleme temelde iki grupta incelenebilir:

- Tam renk (full color)

- Sahte renk (pseudo color)



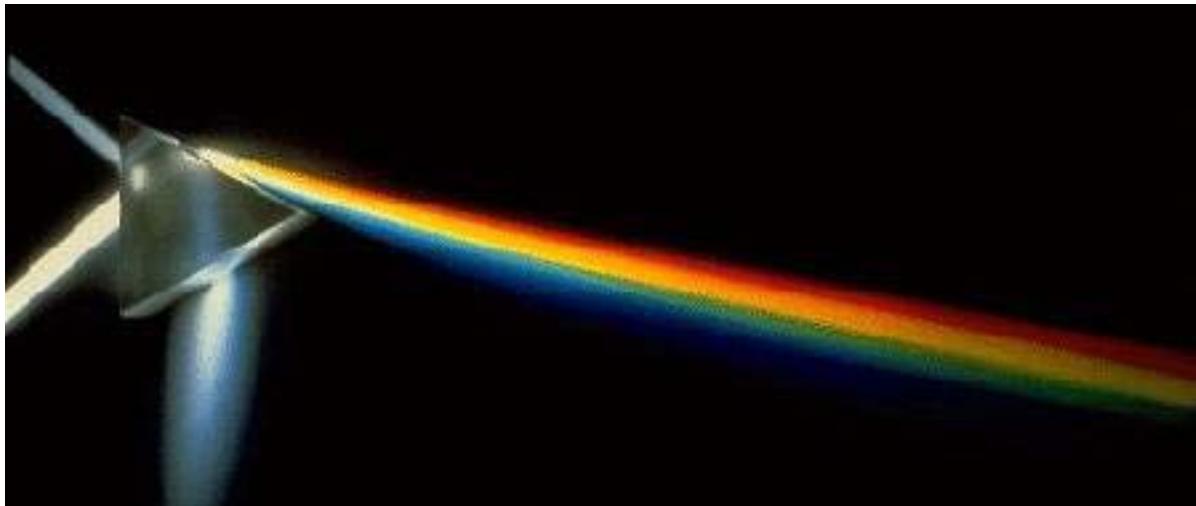
Renkli bir algılayıcı vasıtayıla alınan imgeler.

Gri tonlu imgelerin renklendirilmesiyle oluşan imgeler.

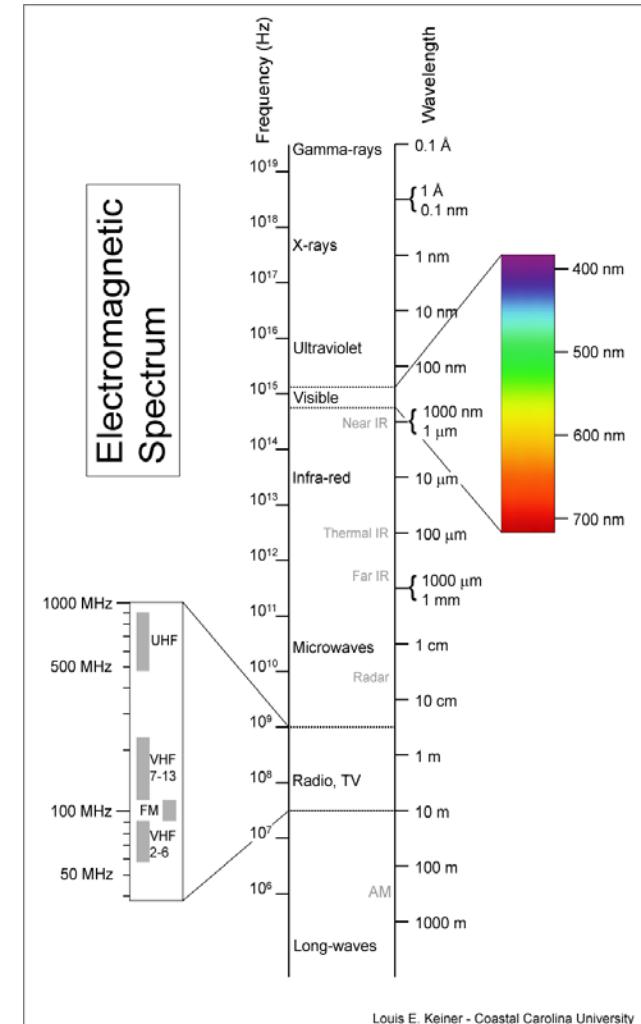


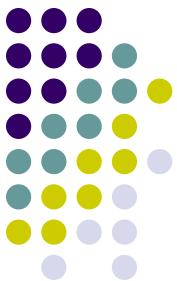
Renk - Elektromanyetik Spektrum

- Optik Prizma (Isaac Newton-1666)



- Eğer ışık renksiz ise ışığın yalnızca şiddeti (intensity) ya da miktarı vardır.

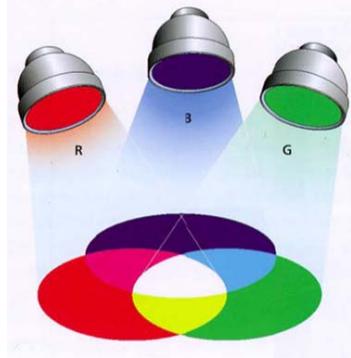




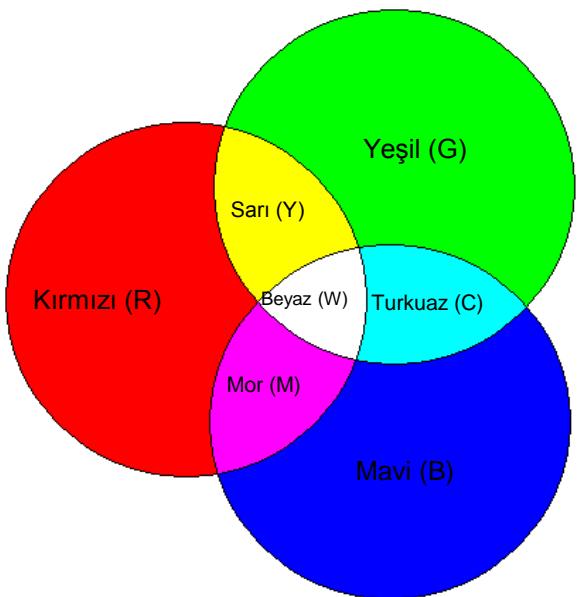
Renk - Algılama

- Renkli ışık kaynağını tanımlamada üç ayrı nicelik kullanılır:
 - Işınırılık (Radiance): ışık kaynağından çıkan enerji miktarı (Watt).
 - Işıklılık (Işık): ışık kaynağından alınan enerji miktarı (lümen).
 - Parlaklık (Brightness): Pratik olarak ölçülemeyen, rengin ışık miktarının öznel tanımlayıcısı.
- İnsan gözündeki koni (cone) algılayıcılarının yaklaşık olarak %65'i kırmızı, %33'ü yeşil ve %2'si de mavi renge duyarlıdır.
- İnsan gözünde renkler de bu üç ana rengin birleşimi şeklinde alınmaktadır.
- Bu üç renk kullanılarak görünürlük bülgedeki bütün renkler elde edilebilmektedir.
- Ana renklerin (kırmızı (R), yeşil (G), mavi (B)) birbirlerine eklenmesi ile ara renkler ya da ikincil renkler (turkuaz (C), mor (M), sarı (Y)) elde edilir.

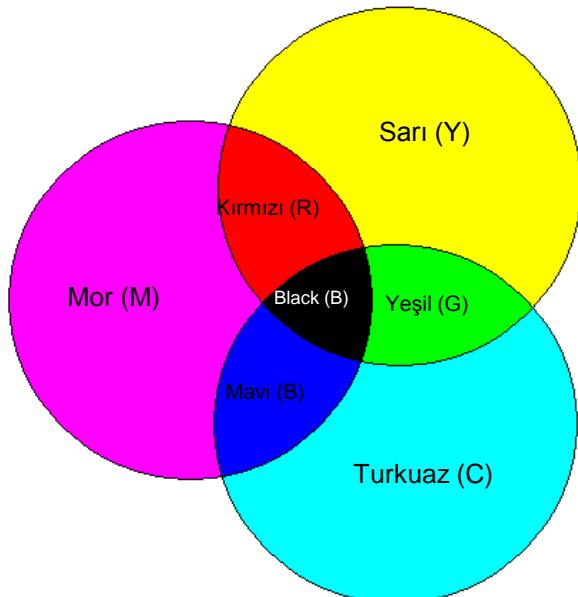
Renk – Ana ve Ara Renkler



Ana renkler (RGB)



Ara renkler (CMY)





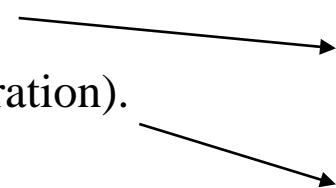
Renk – Ayırt Edicilik

- Bir rengi diğerinden ayırt etmede kullanılan özellikler:

- Parlaklık (brightness),

- Renk özü (hue),

- Doygunuk (saturation).



Renkteki baskın dalga boyunu verir.

Rengin saflığı ya da renk özündeki beyaz ışık katkısı. Saf spektrum renkleri tamamen doygundur. Örn;

pembe=kırmızı+beyaz (az doygun)

- Renk özü ve doygunluk, renksellik olarak ele alınır.
- Parlaklık ile renksellik kullanılarak bir renk ifade edilebilir.



Renk – Üç Uyartı

- Herhangi bir rengi oluşturmak için kullanılan kırmızı, yeşil ve mavi miktarlarına üç uyartı değeri denir ve X , Y , Z ile gösterilir.
- Bu gösterimde bir renk:

$$x = \frac{X}{X + Y + Z}$$

$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}$$

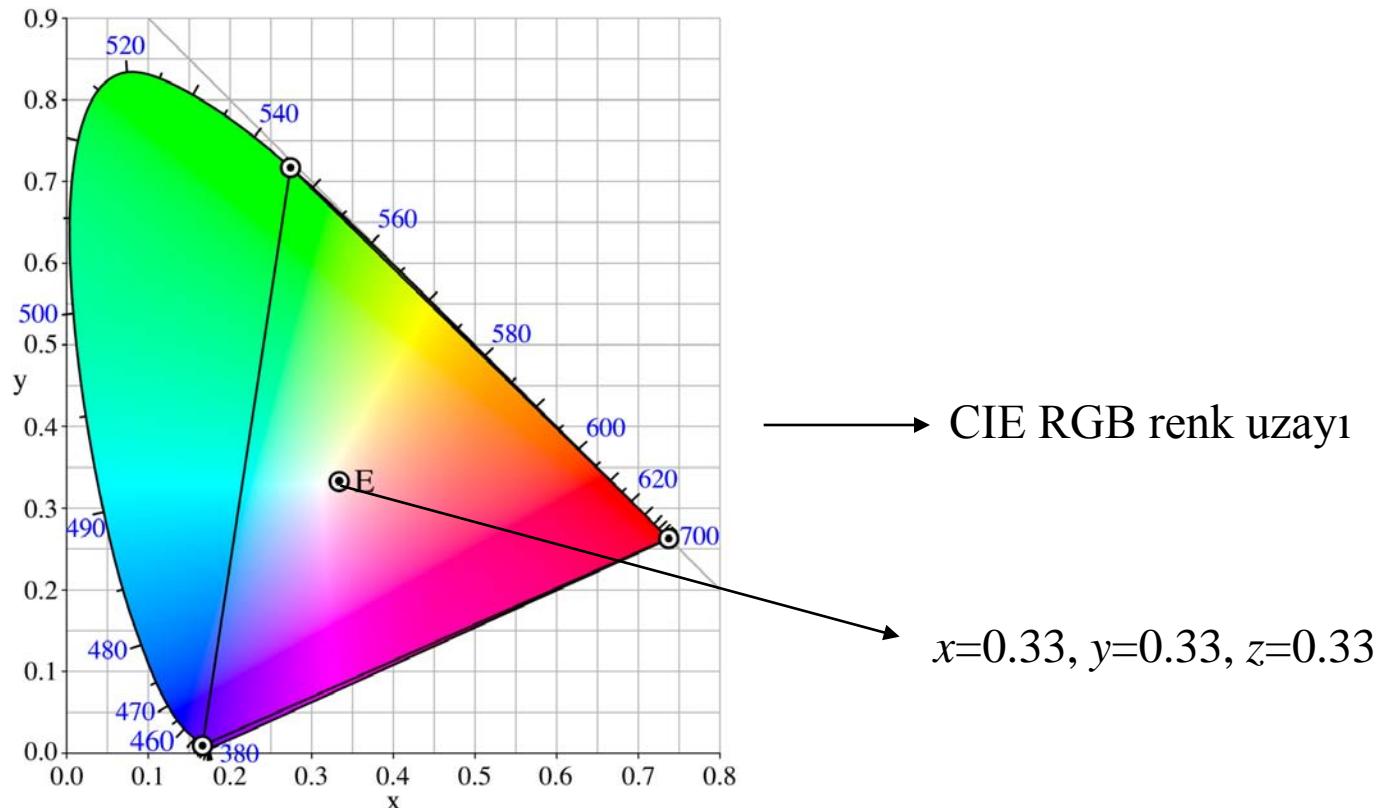
$$x + y + z = 1$$

katsayıları ile ifade edilebilir.



Renk – CIE* Renksellik Çizgesi

- Bu gösterimde herhangi bir renk x (kırmızı) ve y (yeşil) ile ifade edilir ve $z = 1 - (x + y)$ eşitliğinden z (mavi) çekilerek bileşenlerin tamamı bulunmuş olur.





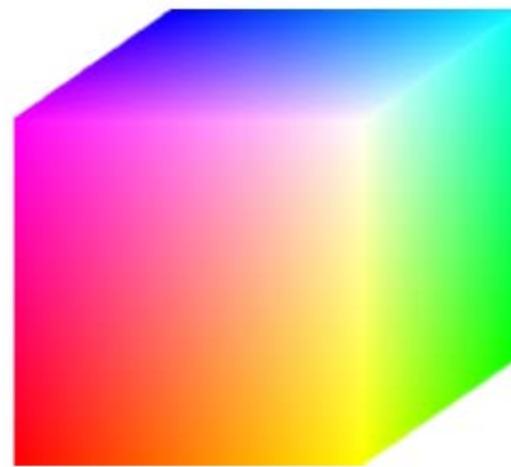
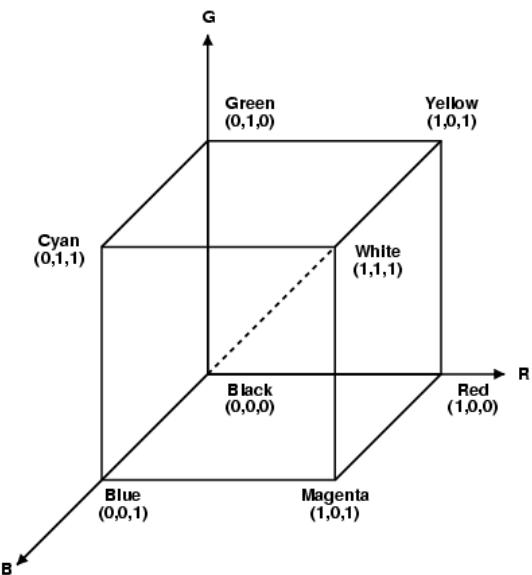
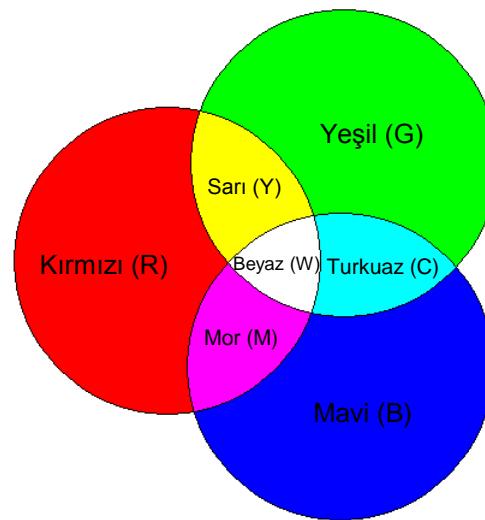
Renk Modelleri

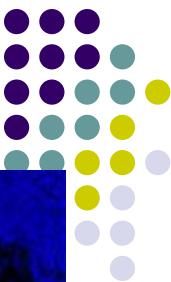
- RGB (Red, Green, Blue):
 - Renkli monitör, renkli video kamera.
- CMY (Cyan, Magenta, Yellow), CMYK (Cyan, Magenta, Yellow, black)
 - Renkli yazıcı.
- HSI (Hue, Saturation, Intensity)
 - İnsan renk tanımlama ve ayırt etmede.
- YIQ
 - Renkli NTSC yayını.
- YCbCr
 - Renkli sayısal TV yayını.



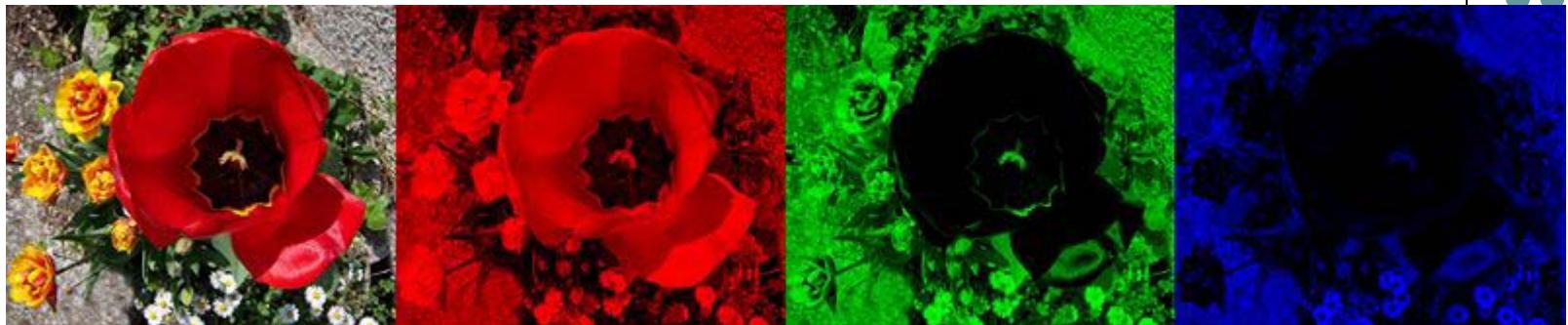
Renk Modelleri - RGB

- RGB





Renk Modelleri - RGB



Red



Green



Blue

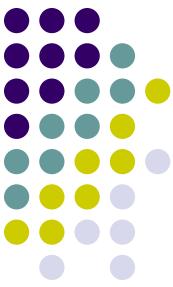


Renk Modelleri - CMY

- RGB renk uzayı ile CMY renk uzayı arasındaki geçişler:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} , \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

- CMYK renk modelinin ortaya çıkışının nedeni ise, C, M ve Y ile saf siyahın elde edilememesidir. Renkli bir yazıcının saf siyah basabilmesi bu sayede mümkün kılınmıştır.



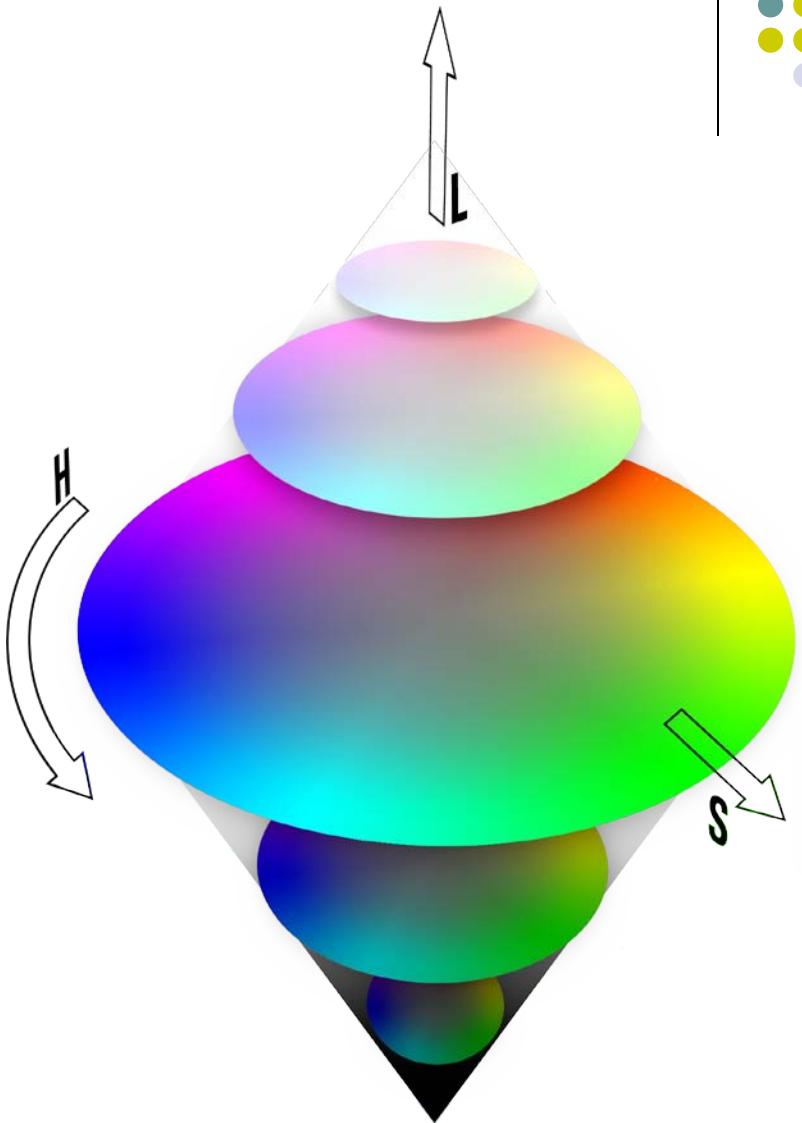
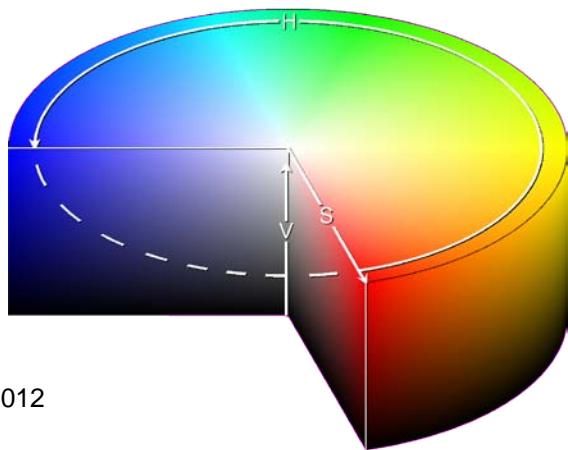
Renk Modelleri - HSI

- RGB, CMY gibi renk uzayları insan gözünün renk alma yapısındadır.
- İnsan beyninde renkler tanınırken ya da birbiri ile karşılaştırılırken bu modellerin kullanımı zordur.
- Bu nedenle bu tür çalışmalarda renk özü (hue-H), doygunluk (saturation-S) ve şiddet (intensity-I) tanımlamaları kullanılır.
- Bu bileşenlerden oluşan modele de **HSI (hue, saturation, intensity)** renk modeli adı verilir.
- Sonuçta söyleyebiliriz ki, RGB renk modeli renk oluşturma için idealdır (örn; monitör) fakat betimlemeye kötüdür. HSI renk modeli ise renge bağlı tanımlamada çok iyidir.



Renk Modelleri - HSI

- Renk Özü (Hue-H): Baskın renk dalgalaboyunu gösterir ve açısal olarak $[0^\circ, 360^\circ]$ aralığında ifade edilir.
- Doygunluk (Saturation-S): Saf rengin beyaz ışık ile hangi oranda karıştığını gösterir. Yarıçap'a karşılık gelir ve $[0,1]$ aralığında değer alır.
- Şiddet (Intensity-I): Işık miktarını gösterir.





Renk Modelleri - HSI

RGB → HSI geçiş:

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)]$$

$$I = \frac{1}{3} (R + G + B)$$



Renk Modelleri - HSI

HSI → RGB geçiş:

$$(0^\circ \leq H < 120^\circ) \quad B = I(1 - S)$$

$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad G = 3I - (R + B)$$

$$(120^\circ \leq H < 240^\circ) \quad H = H - 120^\circ$$

$$R = I(1 - S)$$

$$G = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad B = 3I - (R + G)$$

$$(240^\circ \leq H \leq 360^\circ) \quad H = H - 240^\circ$$

$$G = I(1 - S)$$

$$B = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad R = 3I - (G + B)$$



Renk Modelleri - YUV

- PAL, NTSC, SECAM kompozit renkli video standartlarında kullanılır.
- Y, ıshıklılık (luma); U ve V renklilik (chrominance) bileşenleridir.
- YUV bileşenleri RGB'den türetilir.
- Y, ortalama parlaklıği veren ve R, G, B bileşenlerinin ağırlıklı ortalaması ile elde edilen ıshıklılık bileşeni; U, mavi bileşeninden Y'nin; V, kırmızı bileşeninden Y'nin çıkartılması ile elde edilen fark bileşenleridir.

RGB → YUV geçiş:

$$\begin{aligned}Y &= 0.299 * R + 0.587 * G + 0.114 * B \\U &= 0.436 * (B - Y) / (1 - 0.114) \\V &= 0.615 * (R - Y) / (1 - 0.299)\end{aligned}\quad \begin{bmatrix}Y \\U \\V\end{bmatrix} = \begin{bmatrix}0.299 & 0.587 & 0.114 \\-0.14713 & -0.28886 & 0.436 \\0.615 & -0.51499 & -0.10001\end{bmatrix} \begin{bmatrix}R \\G \\B\end{bmatrix}$$
$$R, G, B \in [0, 1] \quad Y \in [0, 1], \quad U \in [-0.436, 0.436], \quad V \in [-0.615, 0.615]$$

YUV → RGB geçiş:

$$\begin{bmatrix}R \\G \\B\end{bmatrix} = \begin{bmatrix}1 & 0 & 1.13983 \\1 & -0.39465 & -0.58060 \\1 & 2.03211 & 0\end{bmatrix} \begin{bmatrix}Y \\U \\V\end{bmatrix}$$

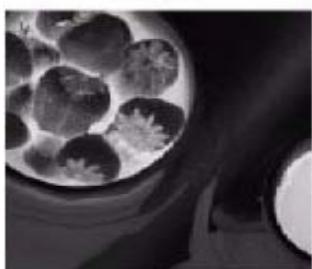
¹⁶ Mart 2012 İmge sayısal ise bu model YCbCr renk modeli olarak isimlendirilir.



Renk Modelleri - Karşılaştırma



Full color



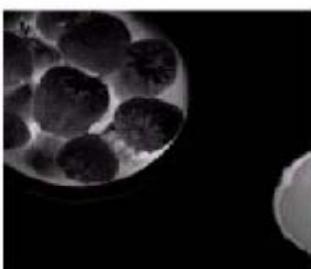
Cyan



Magenta



Yellow



Black



Red



Green



Blue



Hue



Saturation



Intensity



Renk Modelleri – MATLAB Komutlar

MATLAB'da renk uzayı dönüşümleri için:

- RGB → ...
 - YCbCr **rgb2ycbcr**
 - YIQ **rgb2ntsc**
 - HSI **rgb2hsv**
- ... → RGB
 - YCbCr **ycbcr2rgb**
 - YIQ **ntsc2rgb**
 - HSI **hsv2rgb**



RGB Renk Modelinden Gri Tonlu İmge Oluşturma

$$Y = \frac{R + G + B}{3} \quad , \quad \text{renk özü ve doygunluk atılsa}$$

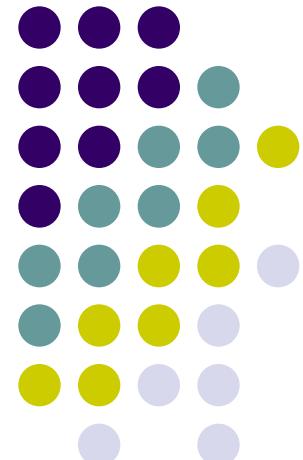
$$Y = 0.299R + 0.587G + 0.114B \quad , \quad \text{NTSC standard göre}$$



iMGE İŞLEME

Ders-7

Morfolojik İmge İşleme



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İ işaret ve Görüntü İşleme Laboratuvarı

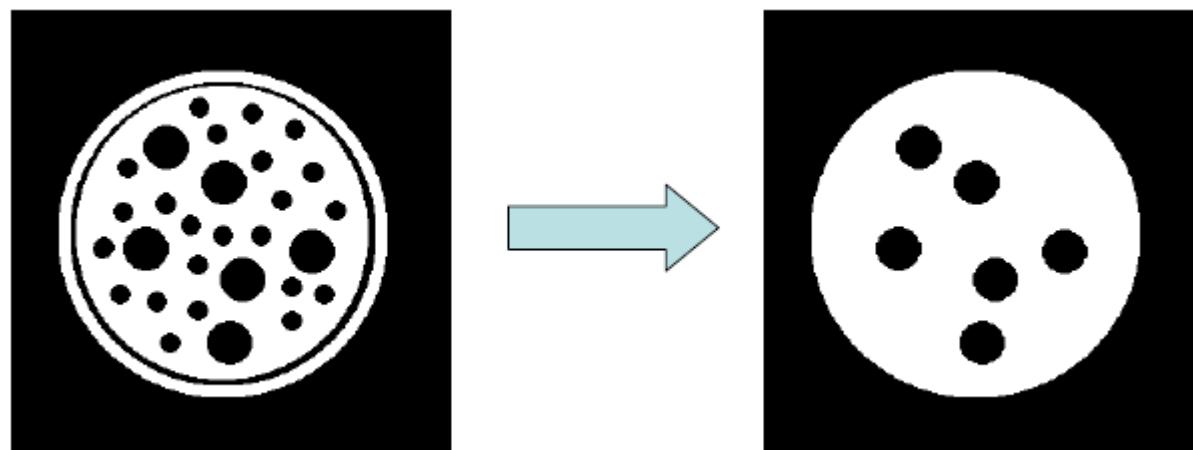
(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/lohab/kemalg/imge_web/odev.htm

Morfoloji

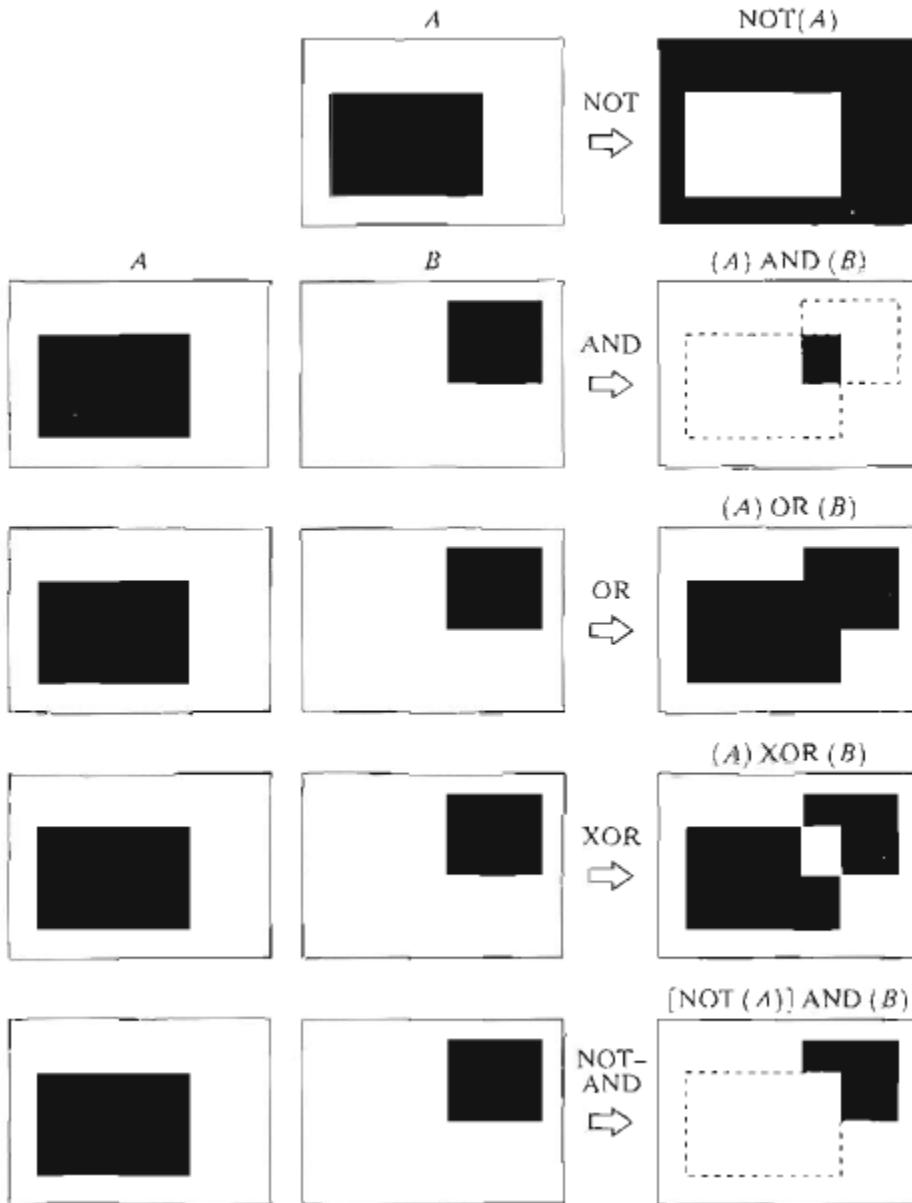


- Biyolojinin canlıların şekil ve yapıları ile ilgilenen dalına morfoloji (biçim bilim) adı verilmektedir.
- Matematiksel morfoloji ise temel küme işlemlerine dayanan, imgedeki sınırlar (borders), iskelet (skeleton) gibi yapıların tanımlanması ve çıkartılması, gürültü giderimi, bölütleme gibi uygulamalar için gerekli bir araçtır.
- İmge işlemede genellikle, morfolojik süzgeçleme, inceltme (thinning), budama (pruning) gibi ön/son işlem olarak da sıkça kullanılırlar.
- Gri tonlu imgeler üzerinde de yapılabileceği gibi, genellikle ikili imgeler üzerinde yapılan işlemlerdir.





İkili İmgelerde Mantıksal İşlemler



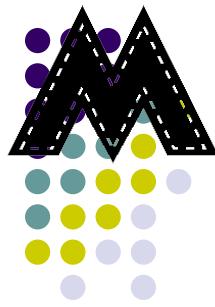
Morfolojik İmge İşleme – Yayma ve Aşındırma



- Aşındırma (Erosion): Matematiksel morfolojinin temel operasyonlarından biridir.
- Ele alınan bölgenin sınır bölgelerinin aşındırılmasında kullanılmaktadır.



- Yayma (Dilation): Diğer bir temel morfolojik işlemidir.
- Ele alınan bölgenin sınırlarının genişletilmesinde kullanılmaktadır.



Morfolojik İmge İşleme – Yayma

- Morfolojik operatörlerin iki girişi vardır:
 1. Yayılacak imge,
 2. Yayma işleminin şeklini belirleyen yapı elemanı (structure element).
- Z^2 uzayında verilen A ve B kümeleri için yagma işlemi aşağıdaki gibi tanımlanmaktadır:

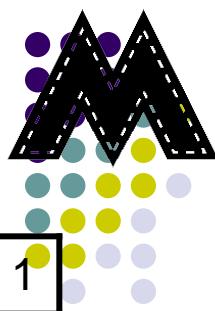
$$A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \emptyset \right\}$$

\hat{B} : B 'nin bire tümleyeni

Yagma operatörü

A : işlenecek imge

B : yapı elemanı



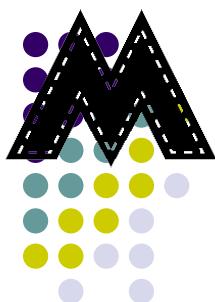
Morfolojik İmge İşleme – Yayma

- Yapı elemanı, yayma işleminin nasıl yapılacağını belirlemektedir.

1	1	1
1	1	1
1	1	1

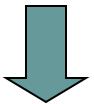


Morfolojik İmge İşleme – Yayma



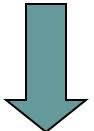
Input image

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



Structuring Element

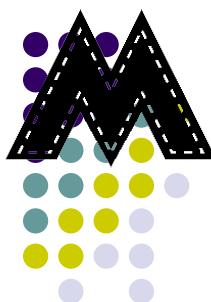
1	1	1
---	---	---



Output Image

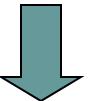
	1								
--	---	--	--	--	--	--	--	--	--

Morfolojik İmge İşleme – Yayma



Input image

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



Structuring Element

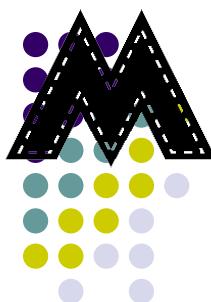
1	1	1
---	---	---



Output Image

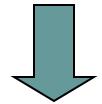
	1	0							
--	---	---	--	--	--	--	--	--	--

Morfolojik İmge İşleme – Yayma



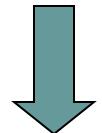
Input image

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



Structuring Element

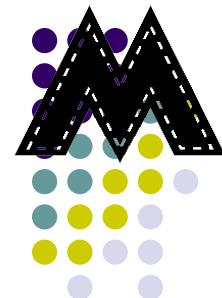
1	1	1
---	---	---



Output Image

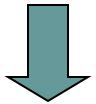
	1	0	1						
--	---	---	---	--	--	--	--	--	--

Morfolojik İmge İşleme – Yayma



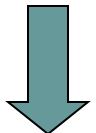
Input image

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



Structuring Element

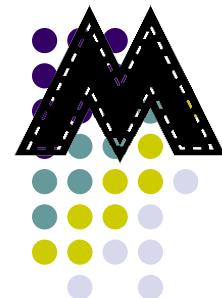
1	1	1
---	---	---



Output Image

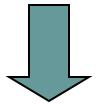
	1	0	1	1					
--	---	---	---	---	--	--	--	--	--

Morfolojik İmge İşleme – Yayma



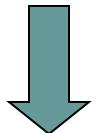
Input image

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



Structuring Element

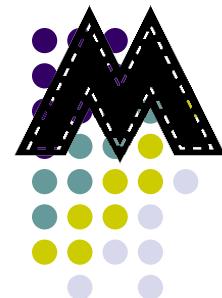
1	1	1
---	---	---



Output Image

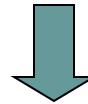
	1	0	1	1	1				
--	---	---	---	---	---	--	--	--	--

Morfolojik İmge İşleme – Yayma



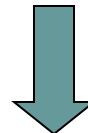
Input image

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



Structuring Element

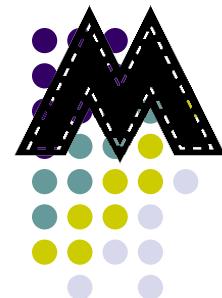
1	1	1
---	---	---



Output Image

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

Morfolojik İmge İşleme – Yayma



Input image

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



Structuring Element

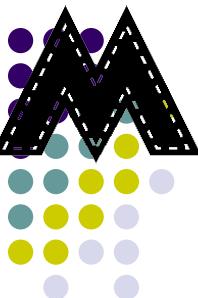
1	1	1
---	---	---



Output Image

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

Morfolojik İmge İşleme – Yayma



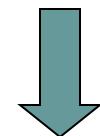
Input image

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



Structuring Element

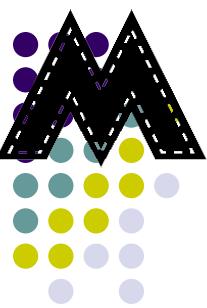
1	1	1
---	---	---



Output Image

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

Morfolojik İmge İşleme – Yayıma

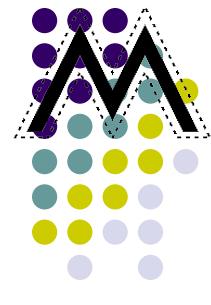


0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1
0	0	1	1	1	1	1	0	0	0	0	1	1	1
0	0	1	1	1	1	1	0	0	0	0	1	1	1
0	0	0	1	1	0	0	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	1	1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



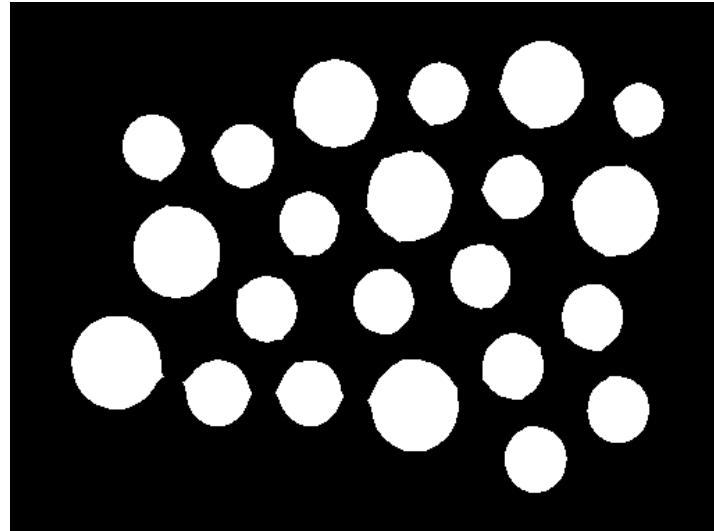
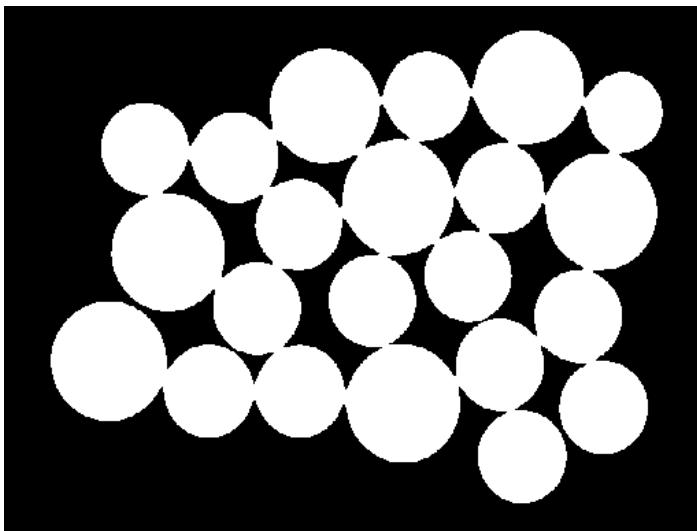
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0	0	1	1	1
0	1	1	1	1	1	1	1	1	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1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

Morfolojik İmge İşleme – Aşındırma

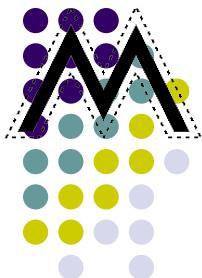


- Z^2 uzayında verilen A ve B kümeleri için aşındırma işlemi

$$A \ominus B = \left\{ z \mid (B)_z \subseteq A \right\}$$

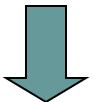


Morfolojik İmge İşleme – Aşındırma



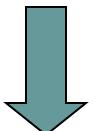
Input image

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



Structuring Element

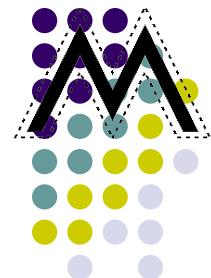
1	1	1
---	---	---



Output Image

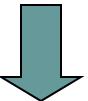
	0								
--	---	--	--	--	--	--	--	--	--

Morfolojik İmge İşleme – Aşındırma



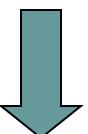
Input image

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



Structuring Element

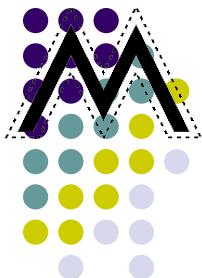
1	1	1
---	---	---



Output Image

	0	0							
--	---	---	--	--	--	--	--	--	--

Morfolojik İmge İşleme – Aşındırma



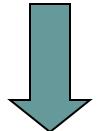
Input image

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



Structuring Element

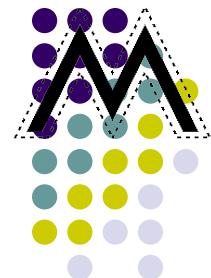
1	1	1
---	---	---



Output Image

	0	0	0						
--	---	---	---	--	--	--	--	--	--

Morfolojik İmge İşleme – Aşındırma



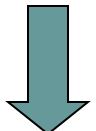
Input image

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



Structuring Element

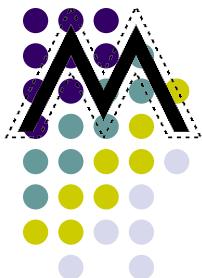
1	1	1
---	---	---



Output Image

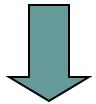
	0	0	0	0					
--	---	---	---	---	--	--	--	--	--

Morfolojik İmge İşleme – Aşındırma



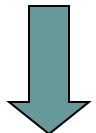
Input image

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



Structuring Element

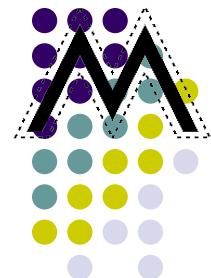
1	1	1
---	---	---



Output Image

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

Morfolojik İmge İşleme – Aşındırma



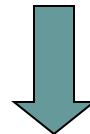
Input image

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



Structuring Element

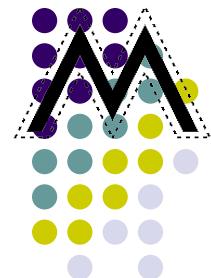
1	1	1
---	---	---



Output Image

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

Morfolojik İmge İşleme – Aşındırma



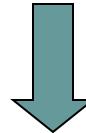
Input image

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



Structuring Element

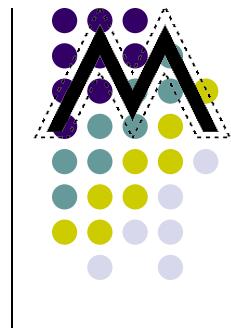
1	1	1
---	---	---



Output Image

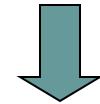
	0	0	0	0	1	0	0		
--	---	---	---	---	---	---	---	--	--

Morfolojik İmge İşleme – Aşındırma



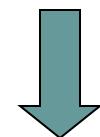
Input image

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



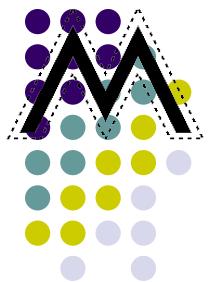
Structuring Element

1	1	1
---	---	---



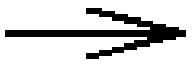
Output Image

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

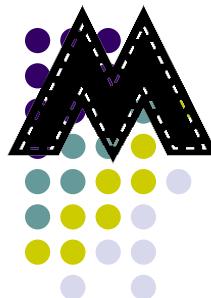


Morfolojik İmge İşleme – Aşındırma

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

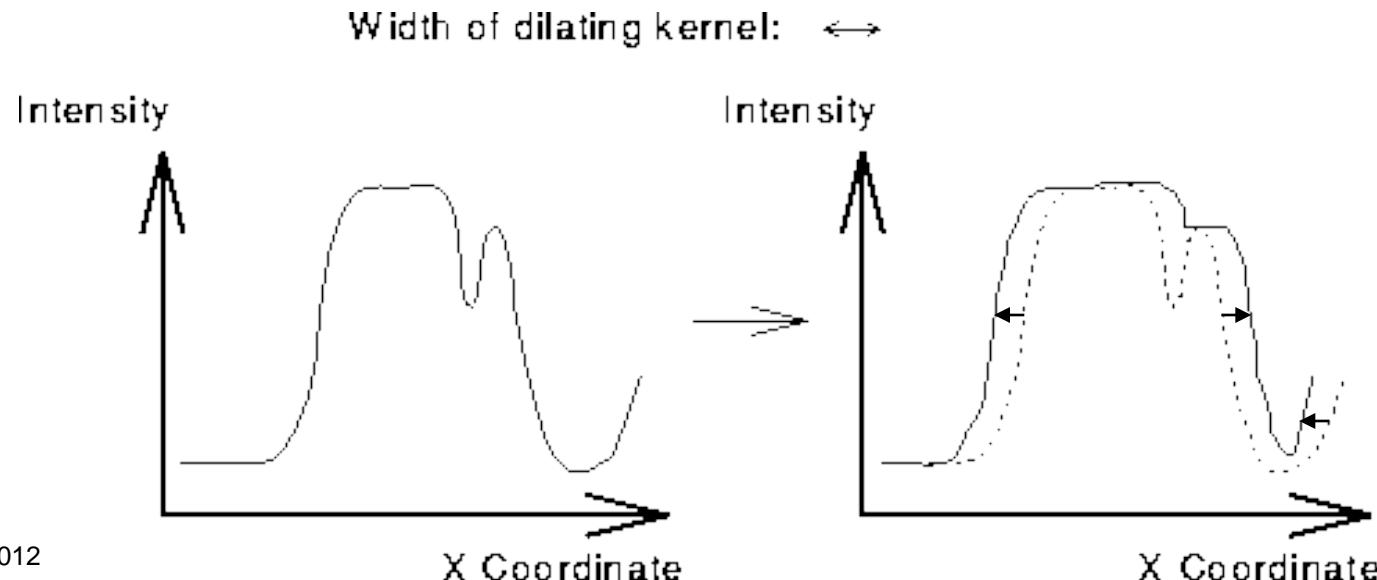


0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0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

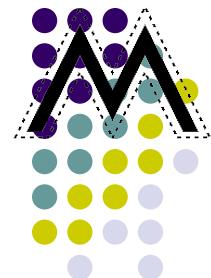


Morfolojik İmge İşleme – Gri Tonlu Yayma

- Morfolojik operasyonlar ikili imgelere uygulanabileceği gibi, gri tonlu imgeler için de tanımlıdır.
- Gri tonlu imgede yayma işlemi, imgedeki koyu tonlu bölgeleri açmaktadır.
- İmgede koyu tonlu bölgelerle çevrili parlak bölgeler genişlerken, parlak bölgelerle çevrili koyu tonlu bölgeler zayıflamakta, hatta yapı elemanı ve koyu tonlu bölgenin boyutuna bağlı olarak kaybolabilmektedir.

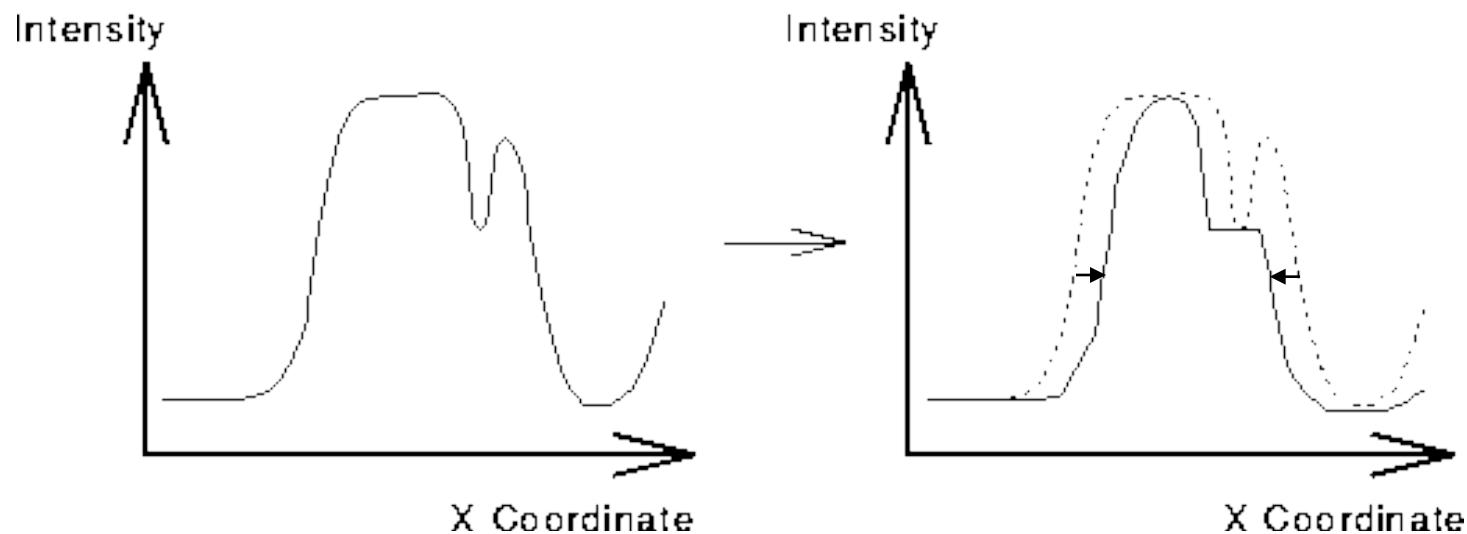


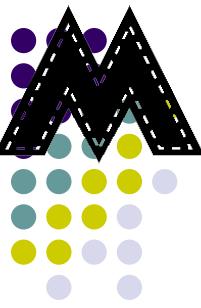
Morfolojik İmge İşleme – Gri Tonlu Aşındırma



- Gri tonlu imgede aşındırma işlemi, imgenin parlaklığını artttırmaktadır.
- İmgede koyu tonlu bölgelerle çevrili parlak bölgeler daralırken, parlak bölgelerle çevrili koyu tonlu bölgeler genişlemektedir.

Width of eroding kernel:





Morfolojik İmge İşleme – Gri Tonlu Morfoloji

Let $F, K \subseteq E^{N-1}$ and $f : F \rightarrow E$ and $k : K \rightarrow E$ where E is the Euclidean Space.

Dilation $f \oplus k : F \oplus K \rightarrow E$ can be computed as:

$$(f \oplus k)(x) = \max_{\substack{z \in K \\ x-z \in F}} \{f(x-z) + k(z)\}$$

Erosion $f \ominus k : F \ominus K \rightarrow E$ can be computed as:

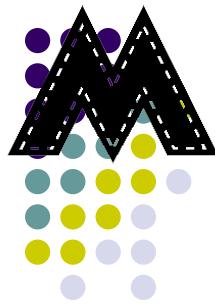
$$(f \ominus k)(x) = \min_{z \in K} \{f(x+z) - k(z)\}$$

Opening $f \circ k : F \circ K \rightarrow E$ can be computed as:

$$(f \ominus k) \oplus k$$

Closing $f \bullet k : F \bullet K \rightarrow E$ can be computed as:

$$(f \oplus k) \ominus k$$



Morfolojik İmge İşleme – Gri Tonlu Morfoloji

- İşlemler basitleştirilirse:

Given a *zero-height flat structuring element* k , i.e.
 $k(x) = 0$ for all x , dilation can be computed as

$$(f \oplus k)(x) = \max_{x \in F} \{f(x)\}$$

→ max filter

likewise, erosion can be computed as

$$(f \ominus k)(x) = \min_{x \in F} \{f(x)\}$$

→ min filter



Morfolojik İmge İşleme – Açıma ve Kapama

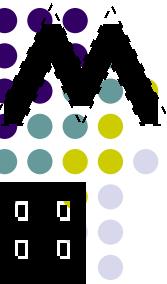
- Açıma ve kapama, yayma ve aşındırma işlemlerinin iki değerli imgeye ardışılı uygulanmasıyla yapılan işlemlerdir.
- Açıma (Opening):

$$A \circ B = (A \Theta B) \oplus B$$

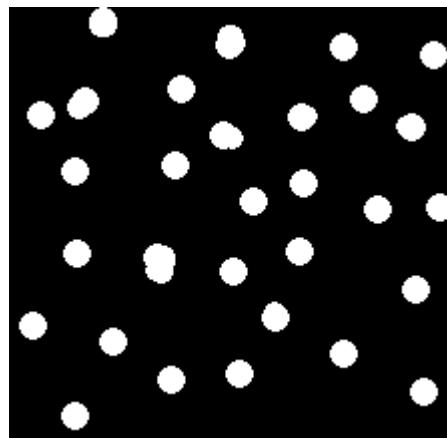
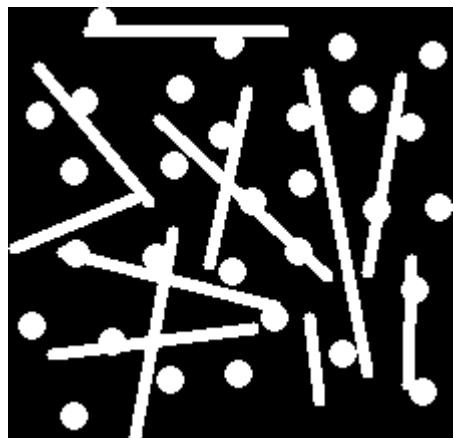
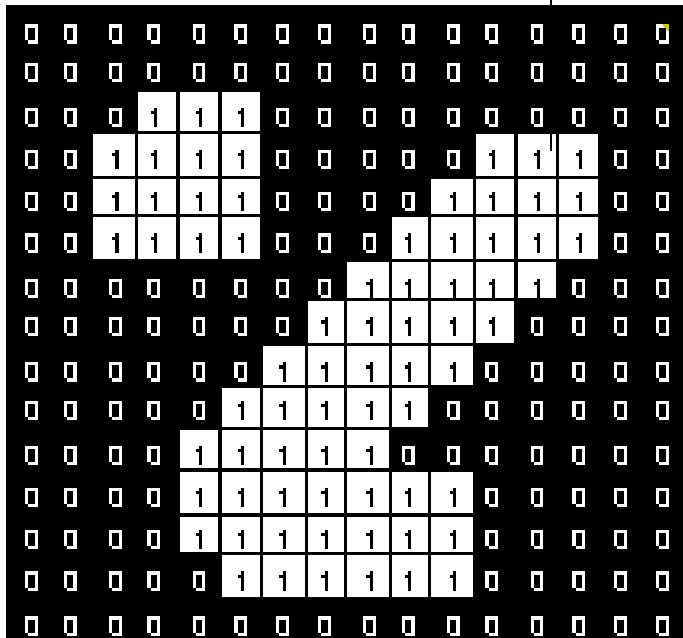
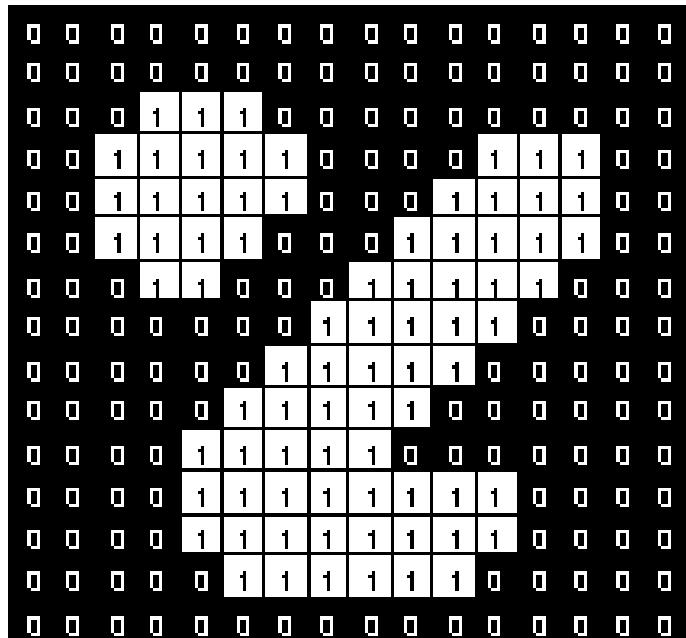
- Kapama (Closing):

$$A \bullet B = (A \oplus B) \Theta B$$

MATLAB'da açma işlemi için **imopen** fonksiyonu, kapama işlemi için **imclose** fonksiyonu kullanılmaktadır.



Morfolojik İmge İşleme – Açıma



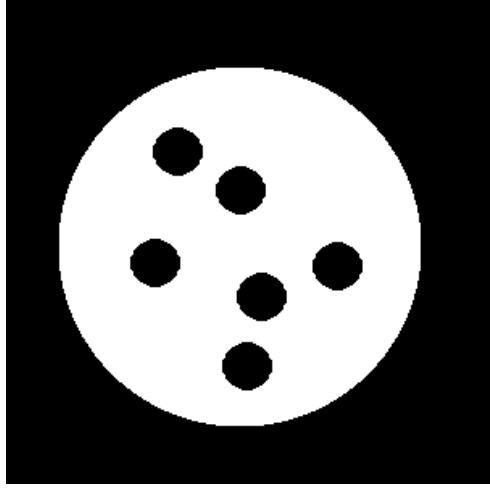
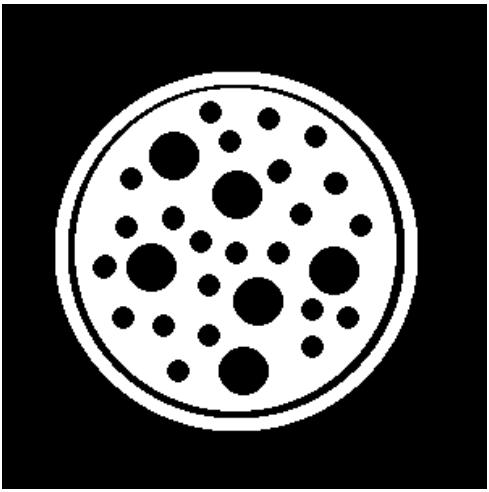


Morfolojik İmge İşleme – Kapama

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



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



Morfolojik İmge İşleme – Sınır Çıkarma

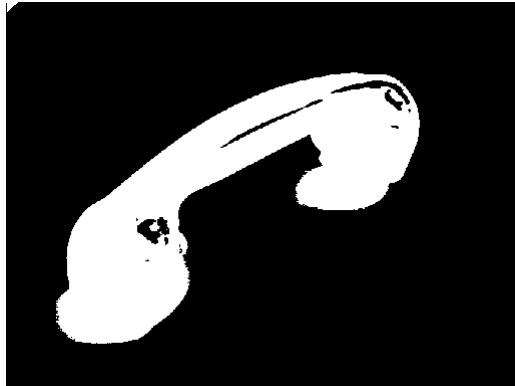


$$\beta(A) = A - (A \Theta B)$$

A kümesinin
sınırları



Morfolojik İmge İşleme – Uygulamalar



Morfolojik İmge İşleme – Uygulamalar



$$A \xrightarrow{A \ominus B}$$

1	1	1
1	1	1
1	1	1



$$(A \ominus B) \oplus B = A \circ B$$

$$(A \circ B) \oplus B$$

$$[(A \circ B) \oplus B] \ominus B = (A \circ B) \cdot B$$



a b
d c
e f

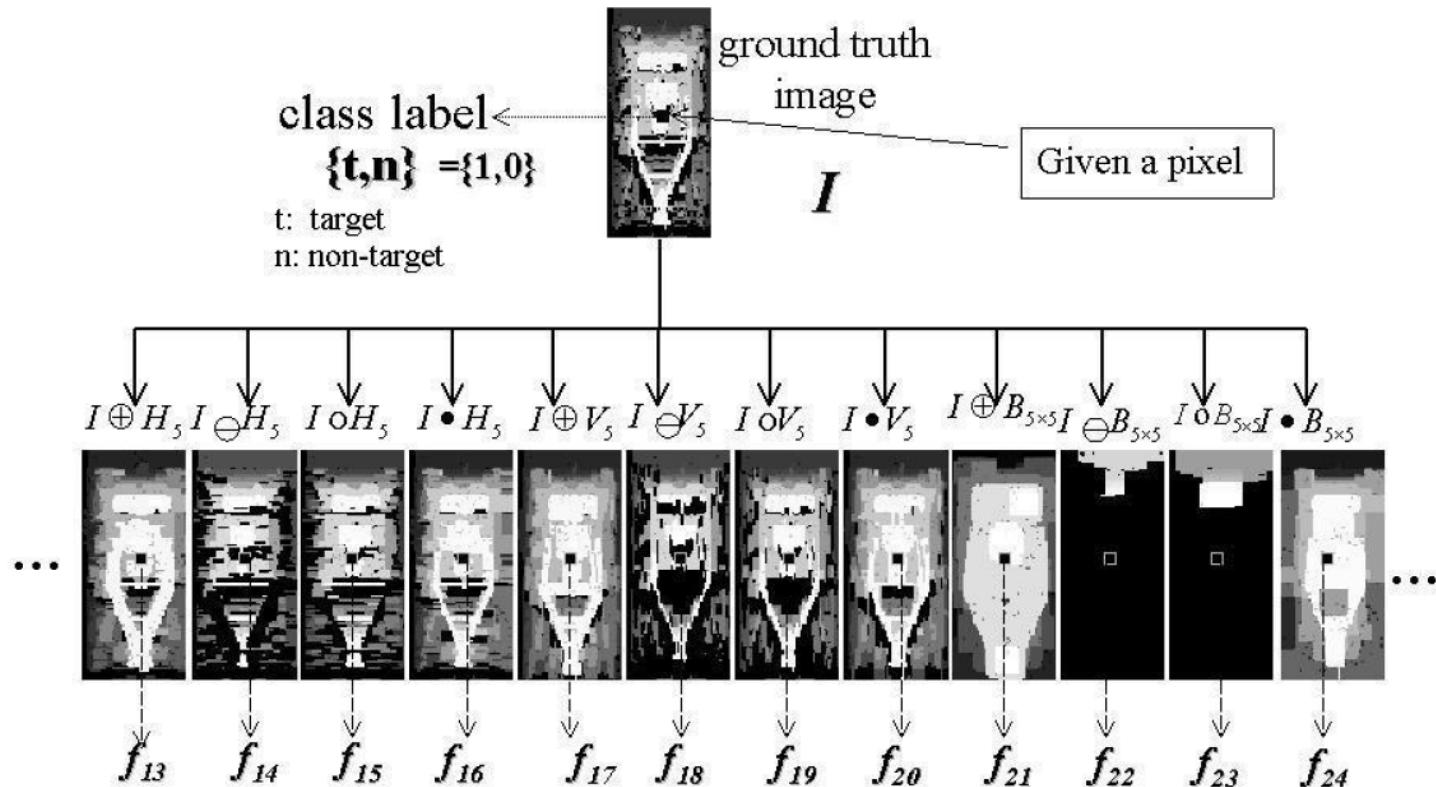
FIGURE 9.11

- (a) Noisy image.
- (c) Eroded image.
- (d) Opening of A.
- (d) Dilation of the opening.
- (e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

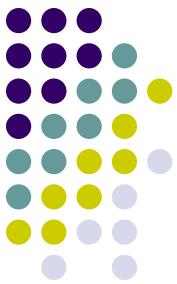


Morfolojik İmge İşleme – Uygulamalar (Gri ton)

Hedef Algılama:



Morfolojik İmge İşleme – Uygulamalar (Gri ton)

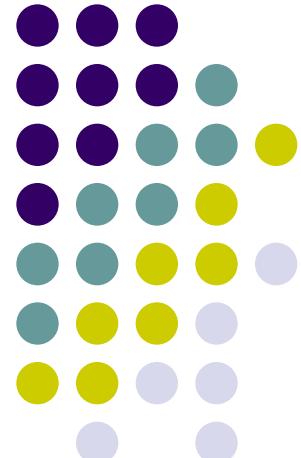


Distance hesabı
(bwdist) eklenecek

iMGE İŞLEME

Ders-8

Frekans Uzayında İmge İşleme

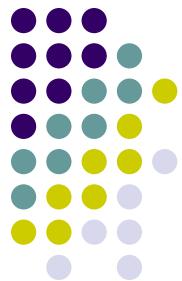


KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İ işaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/lohab/kemalg/imge_web/odev.htm

Frekans Uzayı



- İmge uzayında yapılabilecek işlemlerin yanında, frekans düzlemindeki bilgi de imge işlemede sıkça kullanılmaktadır.
- Daha önce imge süzgeçleme için evrişimden bahsedilmişti. İmge uzayında yapılan bu işlem her bir piksel için tekrarlanmakla birlikte, çekirdek elemanına bağlı olarak hesapsal yükü oldukça fazla olabilmektedir.
- Frekans uzayına geçildiğinde evrişim işlemi çarpma işlemine dönüşeceğinden, bu uzayda yapılacak süzgeçleme işlemlerinde frekans uzayına geçiş ve geri dönüş işlemleri için hesapsal yükten bahsedilebilir.
- Ayrıca frekans uzayında imgedeki piksellerin dağılımına ilişkin bilgileri gözlemlemek de mümkündür.
- Frekans uzayına geçiş için genellikle Fourier dönüşümü kullanılmaktadır.

Frekans Uzayı – Fourier Dönüşümü

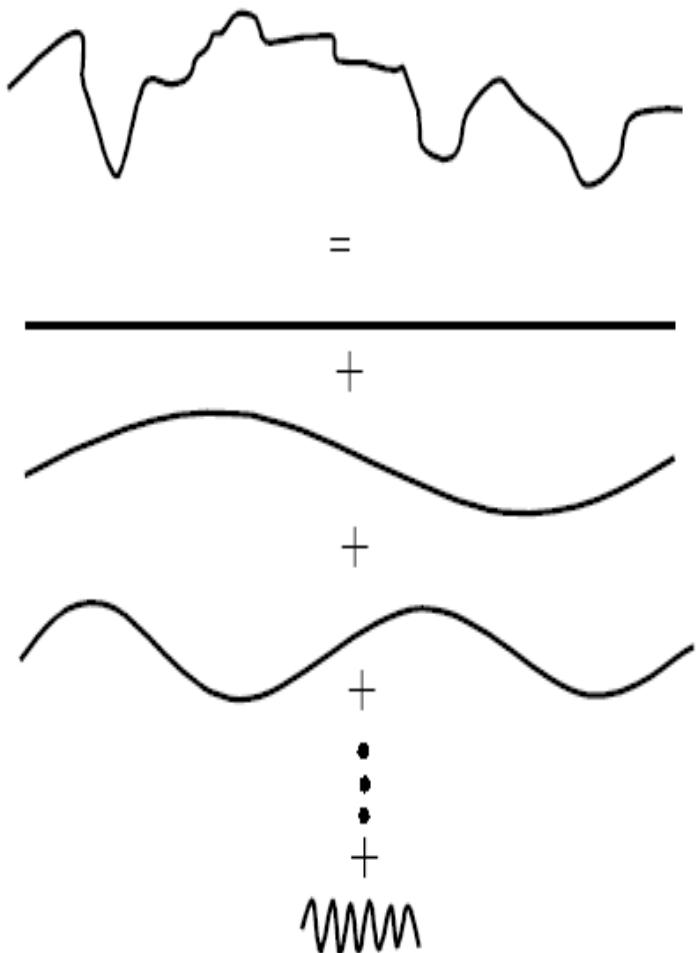


- 1-B Fourier Dönüşümü:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$j = -1$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$





Frekans Uzayı – Fourier Dönüşümü

- 2-B Fourier Dönüşümü:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$



Frekans Uzayı – Ayrık Fourier Dönüşümü

- Ayrık uzayda 1-B Fourier Dönüşümü (AFD):

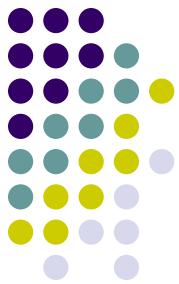
$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}$$

$$u = 0, 1, 2, \dots, M-1$$

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M}$$

$$x = 0, 1, 2, \dots, M-1$$

Frekans Uzayı – Ayrık Fourier Dönüşümü



$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}$$
$$u = 0, 1, 2, \dots, M-1$$

- $F(u)$ yu bulmak için:
 - $u=0$ için x 'in bütün değerlerinde yukarıdaki toplamı hesapla,
 - $u=1$ için x 'in bütün değerlerinde yukarıdaki toplamı hesapla,
 -
 -
 -
 - $u=M$ için x 'in bütün değerlerinde yukarıdaki toplamı hesapla.



Frekans Uzayı – Ayrık Fourier Dönüşümü

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}$$

$$u = 0, 1, 2, \dots, M-1$$

- $F(u)$ yu bulmak için:

- Bu işlem için M^2 çarpma ve toplama gerekli.

- Euler teoremine göre: $e^{j\theta} = \cos\theta + j\sin\theta$

$$e^{-j\theta} = \cos\theta - j\sin\theta$$

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) [\cos(2\pi ux/M) - j\sin(2\pi ux/M)]$$

$$u = 0, 1, 2, \dots, M-1$$



Frekans Uzayı – Ayrık Fourier Dönüşümü

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) [\cos(2\pi ux/M) - j \sin(2\pi ux/M)]$$
$$u = 0, 1, 2, \dots, M-1$$

- Özetle $f(x)$, farklı frekanslardaki Sin ve Cos bileşenleri ile çarpılıyor.

$$\left. \begin{array}{l} F(0) \\ F(1) \\ \vdots \\ \vdots \\ F(M) \end{array} \right\}$$

Dönüşümün
frekans
bileşenleri

$$F(u) = |F(u)| e^{-j\varphi(u)}$$

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad \text{Genlik}$$

$$\varphi(u) = \tan^{-1} \left[\frac{I(u)}{R(u)} \right] \quad \text{Faz açısı}$$



Frekans Uzayı – Ayrık Fourier Dönüşümü

- Güç spektrumu:

$$\begin{aligned} P(u) &= |F(u)|^2 \\ &= R^2(u) + I^2(u) \end{aligned}$$

Spektral yoğunluğu verir

- 2-B AFD:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

$$u = 0, 1, 2, \dots, M-1$$

$$v = 0, 1, 2, \dots, N-1$$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

$$x = 0, 1, 2, \dots, M-1$$

$$y = 0, 1, 2, \dots, N-1$$

Frekans Uzayı – Ayrık Fourier Dönüşümü



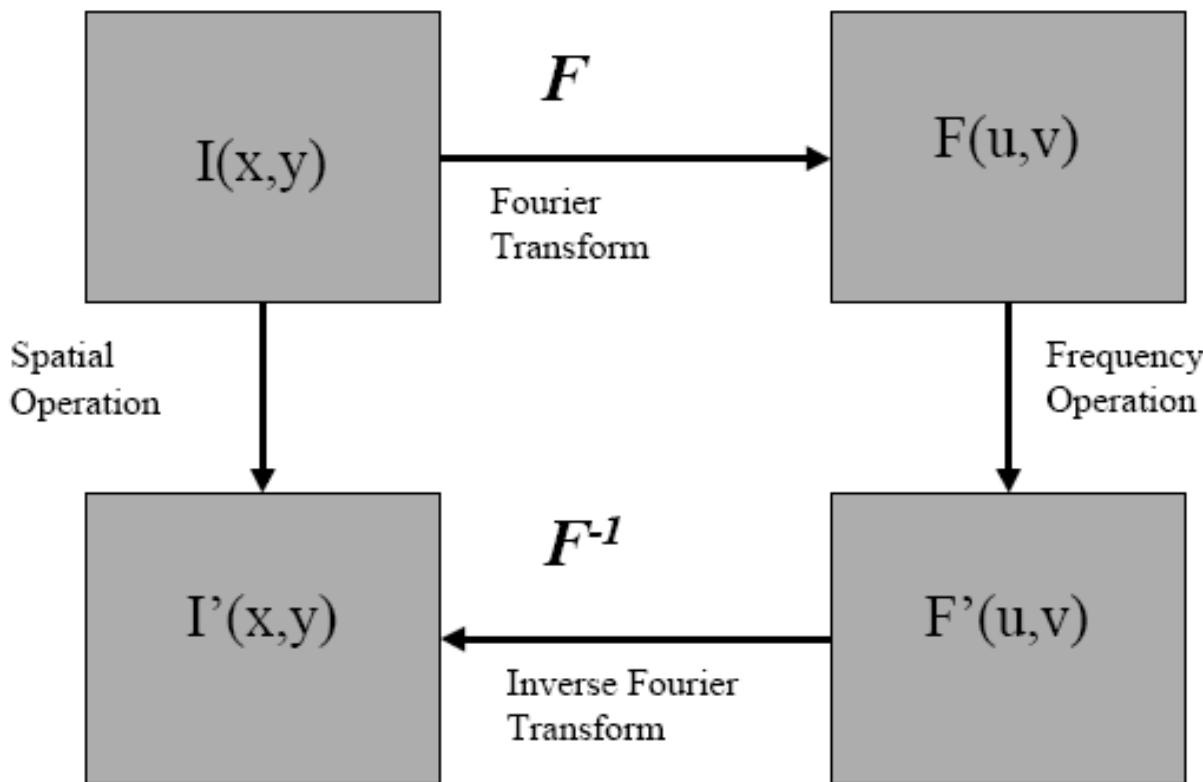
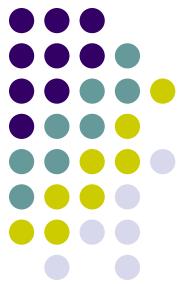
$$F(u, v) = |F(u, v)| e^{-j\varphi(u, v)}$$

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

$$\varphi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$$

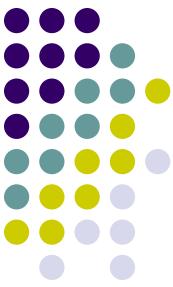
$$\begin{aligned} P(u, v) &= |F(u, v)|^2 \\ &= R^2(u, v) + I^2(u, v) \end{aligned}$$

Frekans Uzayı – İmge İşleme Aşamaları

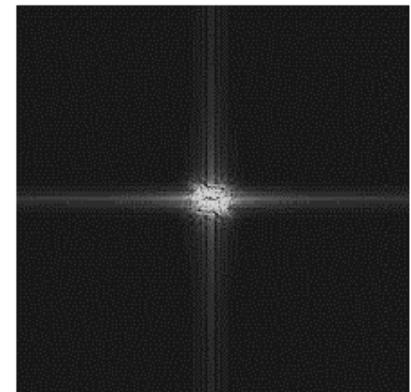
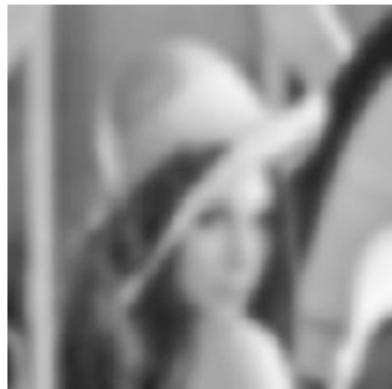
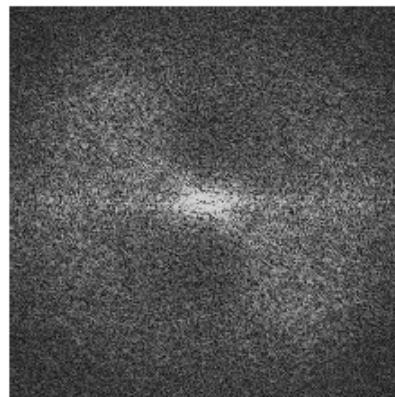
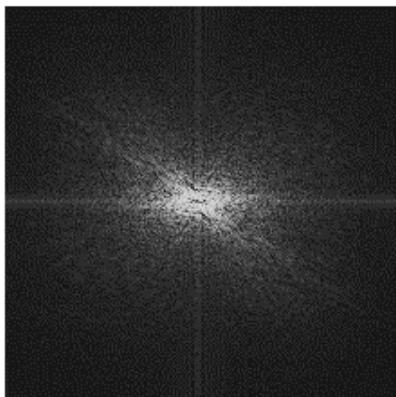


$$F'(u,v) = F(u,v)H(u,v)$$

H : Süzgeç



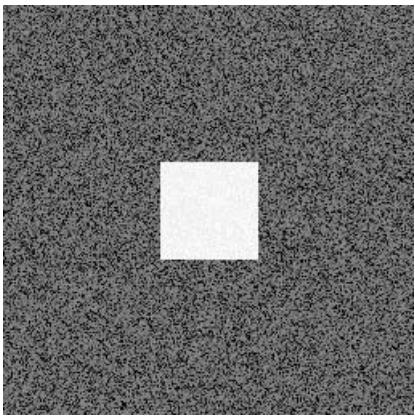
Frekans Uzayı – Örnek Genlik İmgeleri



Frekans Uzayı – AFD

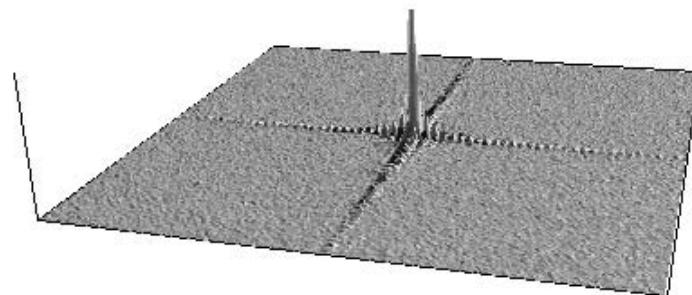
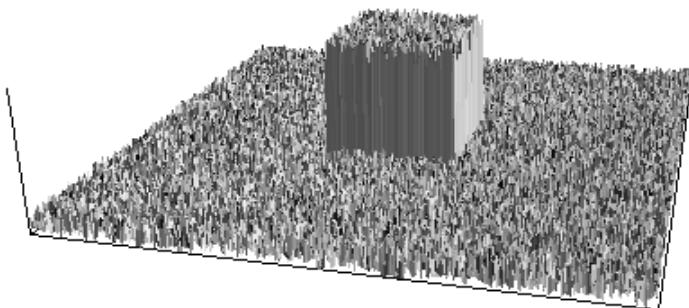
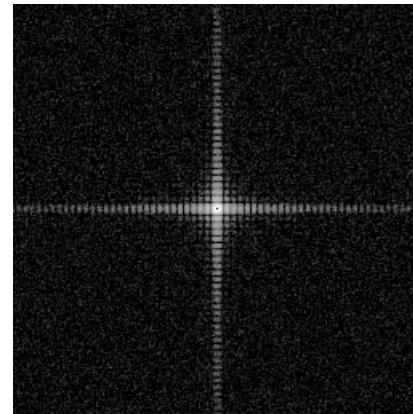


imge

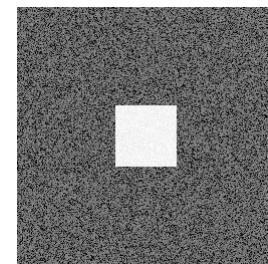


AFD

genlik

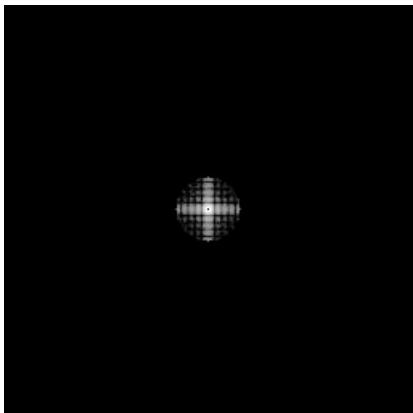


Frekans Uzayı – Temel AGS Örneği



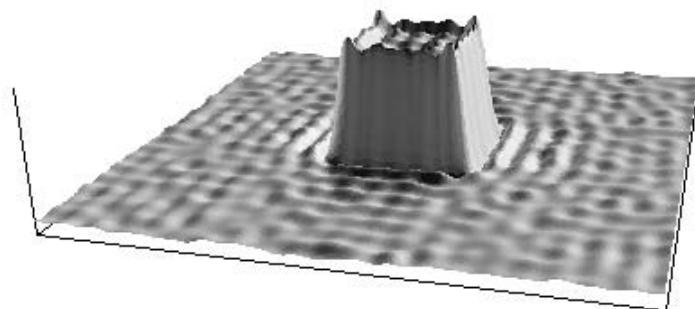
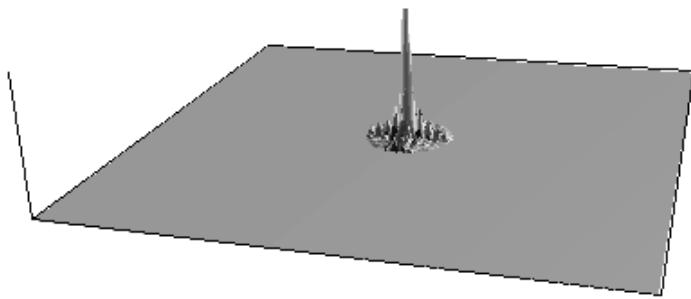
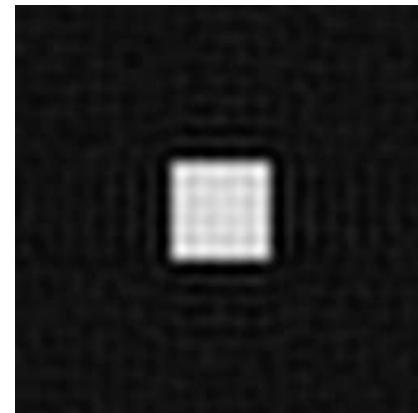
Alçak geçiren süzgeç

genlik

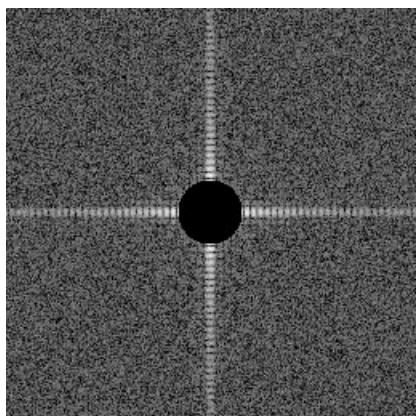


Ters AFD
→

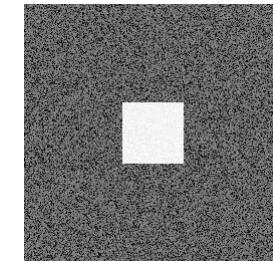
imge



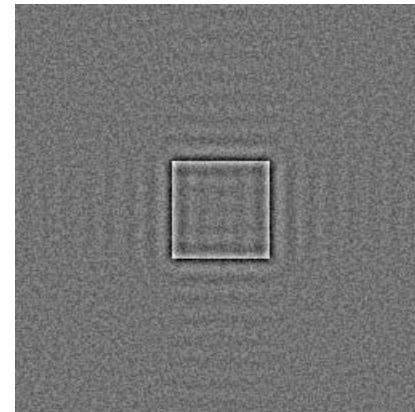
Frekans Uzayı – Temel YGS Örneği



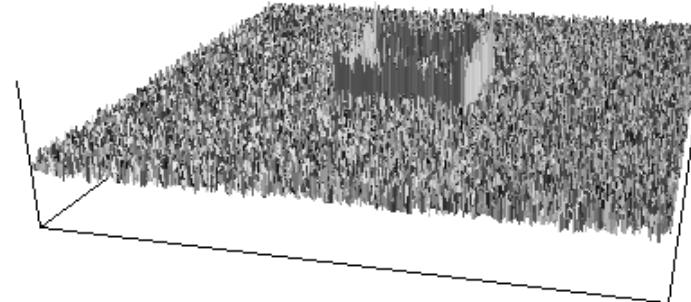
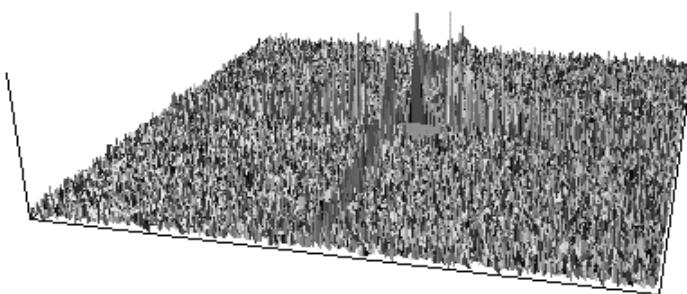
Yüksel geçiren süzgeç



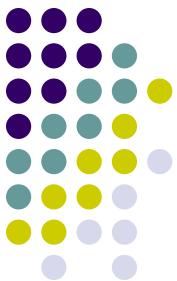
imge



Ters AFD
→

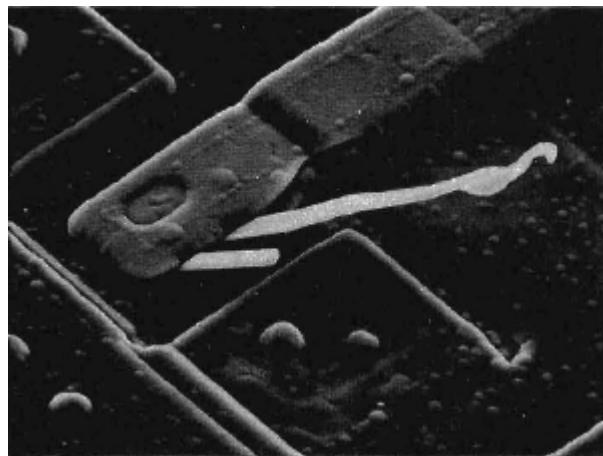
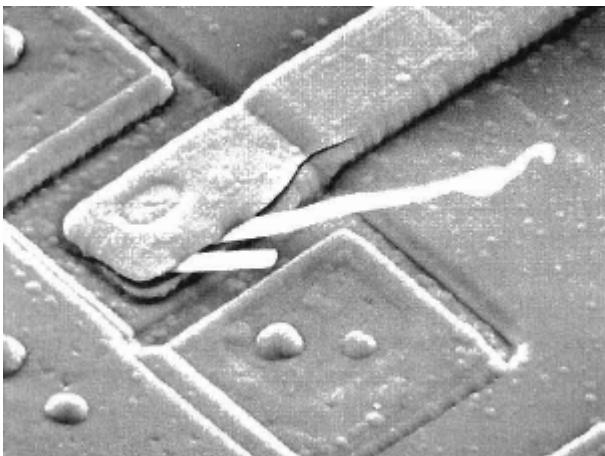


Frekans Uzayı – Çentik Süzgeci



- Bazı temel süzgeç yapıları:
 - Çentik süzgeci (notch filter):

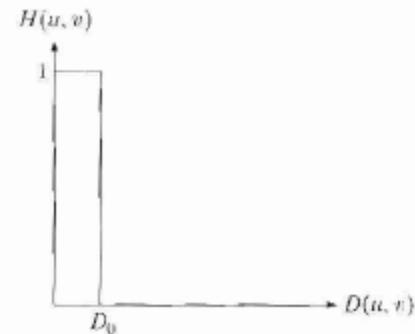
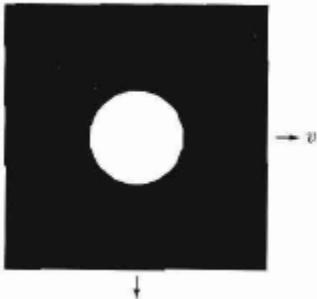
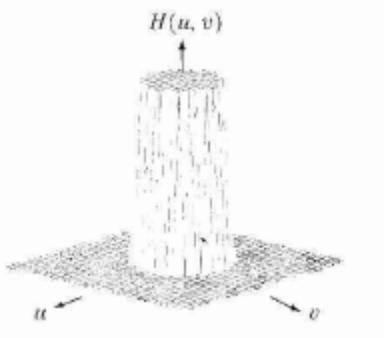
$$H(u, v) = \begin{cases} 0 & , \text{ eğer } (u, v) = (M/2, N/2) \\ 1 & , \text{ diğer} \end{cases}$$



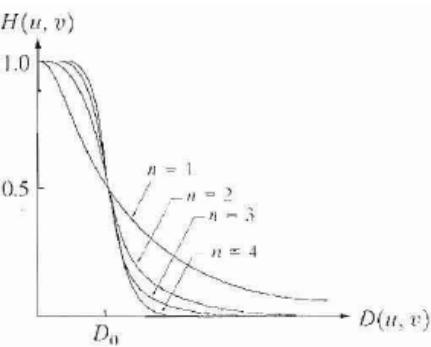
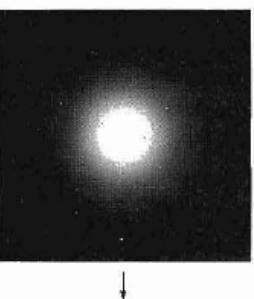
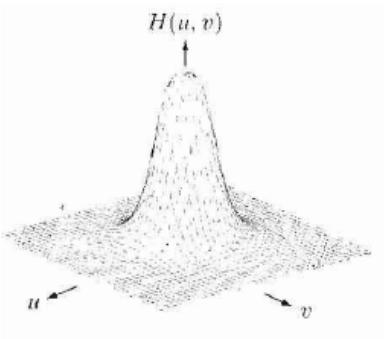
Frekans Uzayı – AGS



- Ideal AGS:



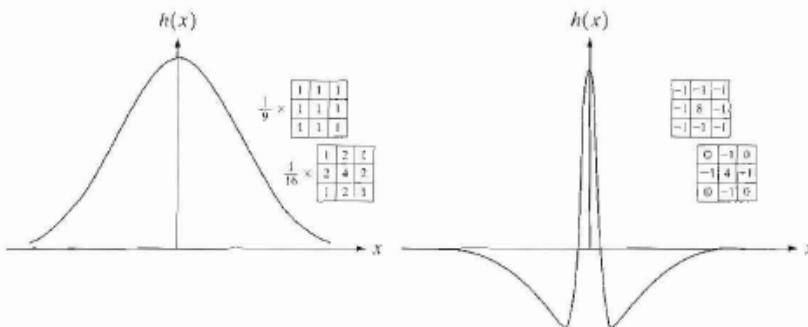
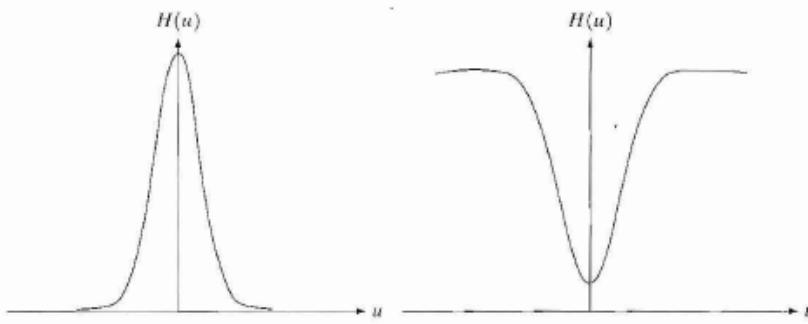
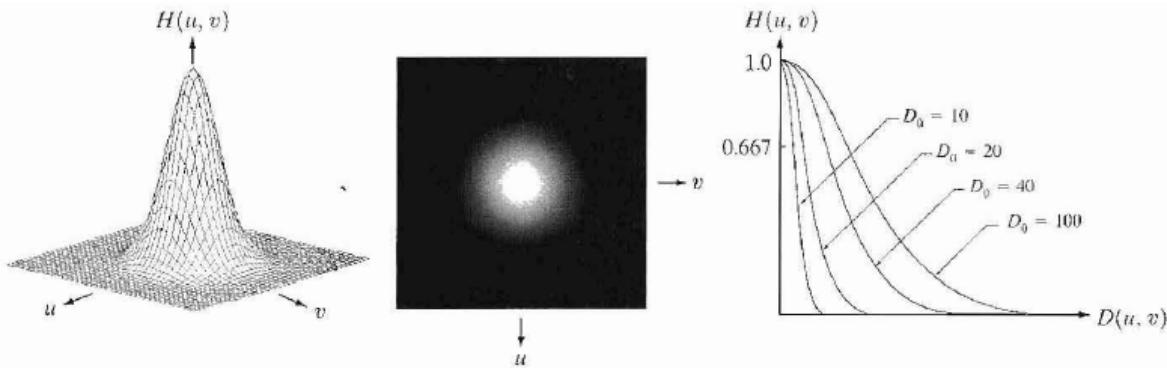
- Butterworth AGS:





Frekans Uzayı – AGS

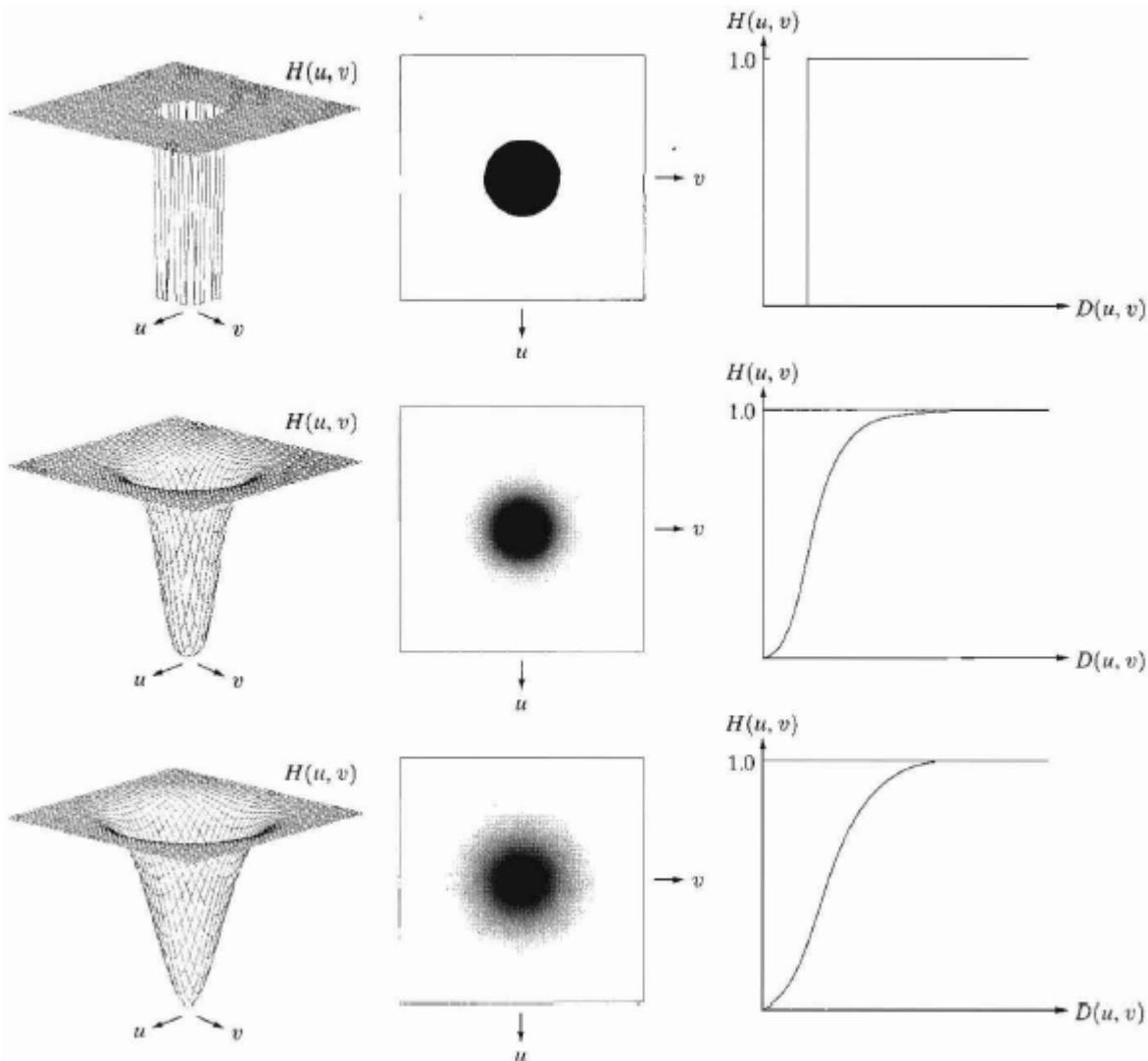
- Gauss Süzgeci:



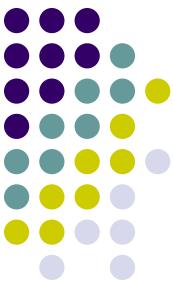


Frekans Uzayı – YGS

- YGS:

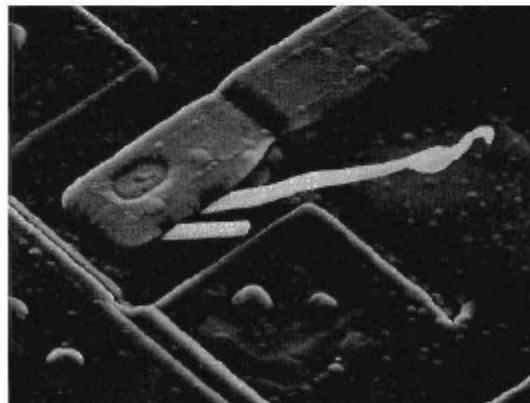
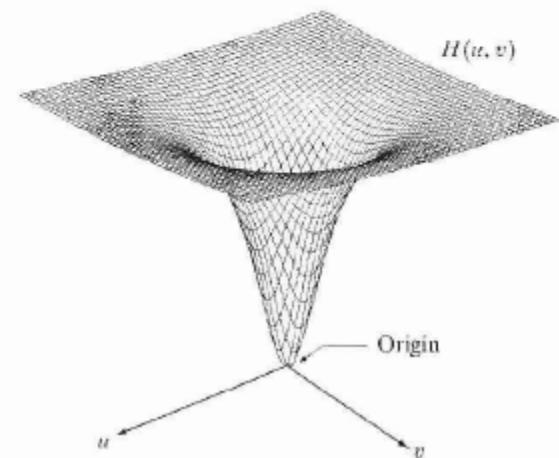
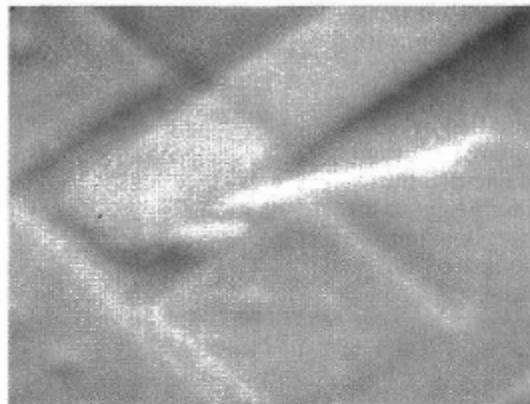
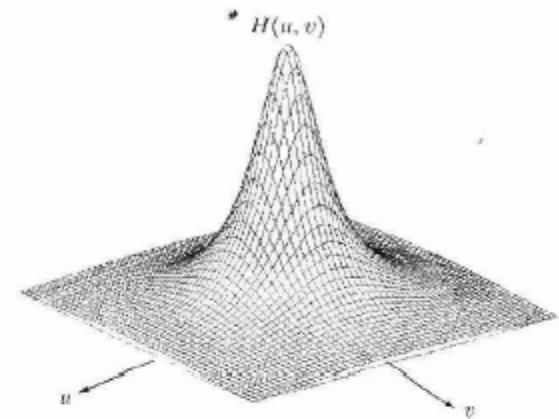


$$H_{YG}(u, v) = 1 - H_{AG}(u, v)$$

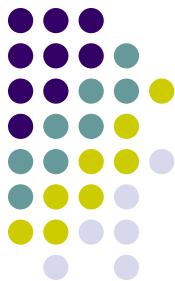


Frekans Uzayı – AGS ve YGS

- AGS ve YGS:



Frekans Uzayı – Matlab'da Frekans Uzayı İşlemleri



- MATLAB'da frekans uzayı süzgeçleme için:
 1. İmgenin AFD'sini al (**fft2** fonksiyonu).
 2. Karmaşık imgeye kaydırma işlemi uygula (**fftshift** fonksiyonu).
 3. Kaydırılmış $F(u,v)$ 'yi $H(u,v)$ ile noktasal çarp.
 4. Karmaşık imgeye tekrar kaydırma işlemi uygula (**fftshift**).
 5. Karmaşık imgenin ters AFD'sini al (**ifft2** fonksiyonu).
 6. Elde edilen sonucun gerçek kısmını süzgeçlenmiş imge olarak kullan (**real** fonksiyonu).

NOT: Farklı tipte süzgeç yapıları için **fspecial** fonksiyonunu kullanabilirsiniz.



Frekans Uzayı – İlinti

- Frekans Uzayında İlinti (correlation):

$$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v) H(u, v)$$

*: karmaşık eşleniği göstermektedir.

- Frekans Uzayında Öz ilinti (autocorrelation):

$$f(x, y) \circ h(x, y) \Leftrightarrow |F(u, v)|^2$$

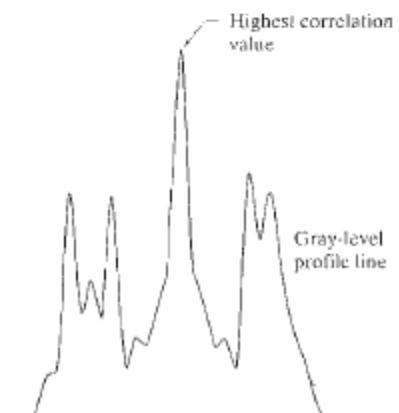
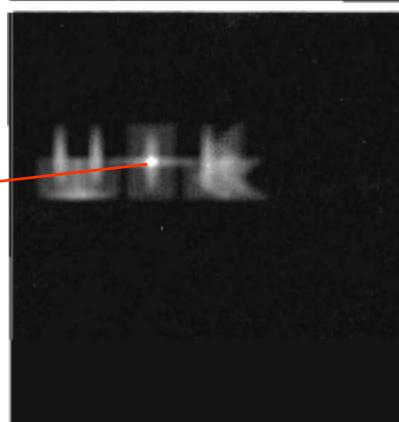


Frekans Uzayı – İlinti



İlintideki en
büyük genlik
seviyesi (aranan
nesne konumu)

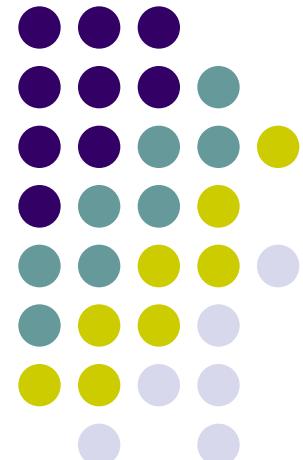
16 Mart 2012



İMGE İŞLEME

Ders-9

İmge Sıkıştırma



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İşaret ve Görüntü İşleme Laboratuvarı

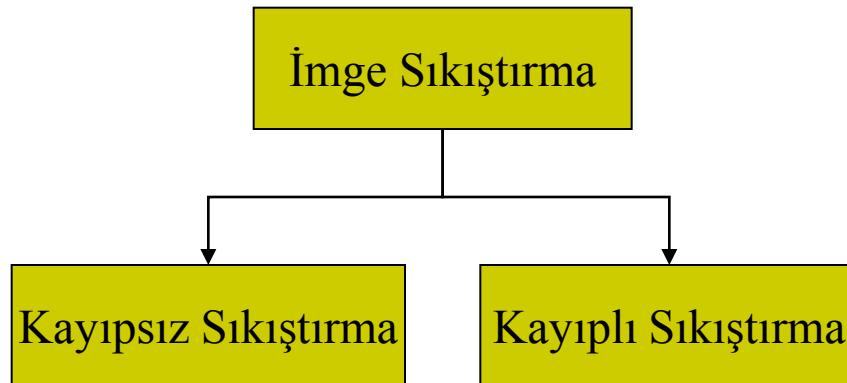
(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/lohab/kemalg/imge_web/odev.htm

İmge Sıkıştırma



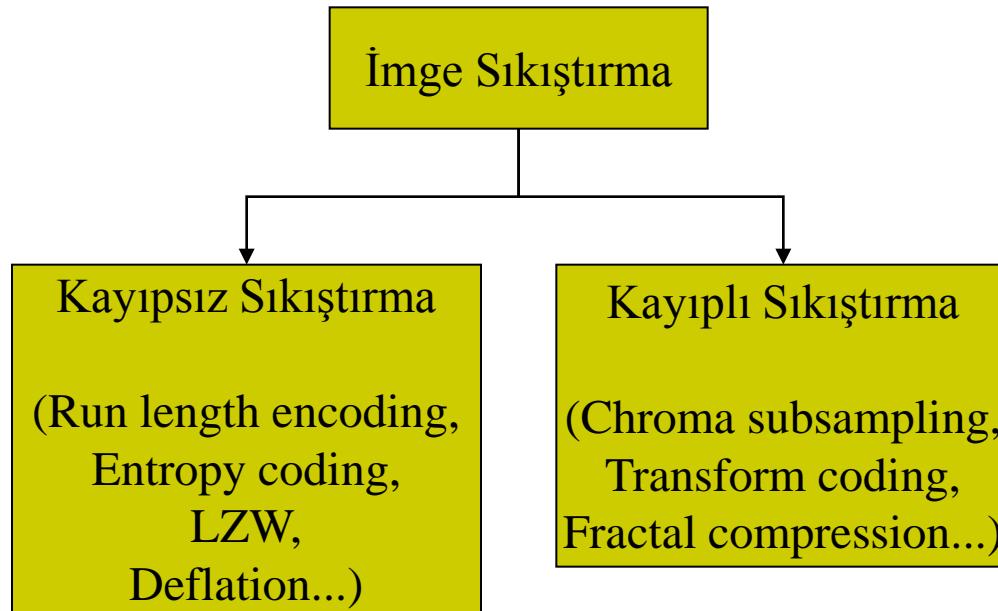
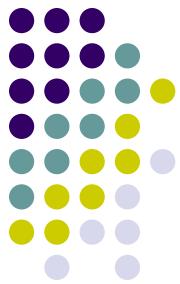
- Veri sıkıştırmanın sayısal imgeler üzerindeki uygulamasıdır.
- Amaç; imgeyi en verimli şekilde saklamak ya da iletmek.



Sıkıştırılan imge;

- Bellekte daha az yer kaplar,
- Aynı bellekte daha fazla veri saklanmasını sağlar,
- İletimi kolaylaştırır (Bant genişliğini azaltır),

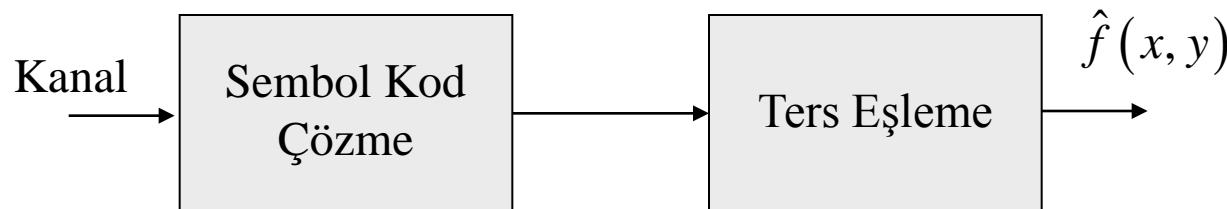
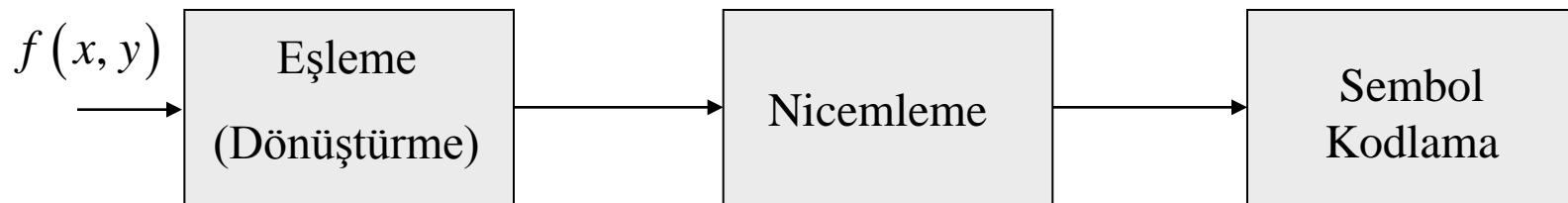
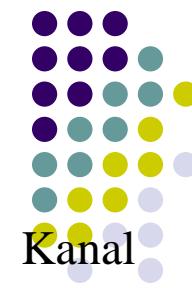
İmge Sıkıştırma



Genel Yapı:



İmge Sıkıştırma

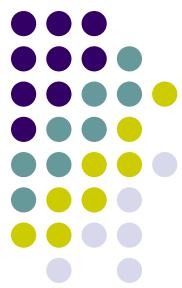


Eşleme:



- Pikseller arası artıklığı sıkıştırma için uygun biçimde dönüştürme işlemi yapar.
- İmgenin yeniden elde edilebilmesi için tersi alınabilen dönüşümler kullanılır.

İmge Sıkıştırma



Nicemleme:



- Eşleyicinin çıkışındaki verinin verimliliğini, önceden belirlenmiş belirli kurallar çerçevesinde düşürür.
- Geri dönüşümü yoktur. Bu nedenle kayıpsız sıkıştırmada ihmäl edilmelidir.

Sembol Kodlama:



- Nicemleyici çıkışındaki veriyi sabit veya değişken uzunluklu kodlar.

İmge Sıkıştırma - JPEG



JPEG (Joint Photographic Experts Group):

- 1992 yılında uluslararası standart olarak belirlendi ve 1994 yılında bu standard ISO 10918-1 olarak onaylandı.
- Kodlayıcı, kod çözücü ve dosya yapısını standartlaştmaktadır.
- Genellikle kayıplı sıkıştırma tekniği olarak bilinir fakat kayıpsız sıkıştırma yapan temel JPEG de vardır.
- Kayıpsız sıkıştırma yöntemlerinin veri sıkıştırma oranları düşüktür (Huffmann, LZW, Aritmetik kodlama...). JPEG'de ise yüksek sıkıştırma oranları elde edilebilir.
- Dönüşüm kodlama kullanılır.

İmge Sıkıştırma - JPEG



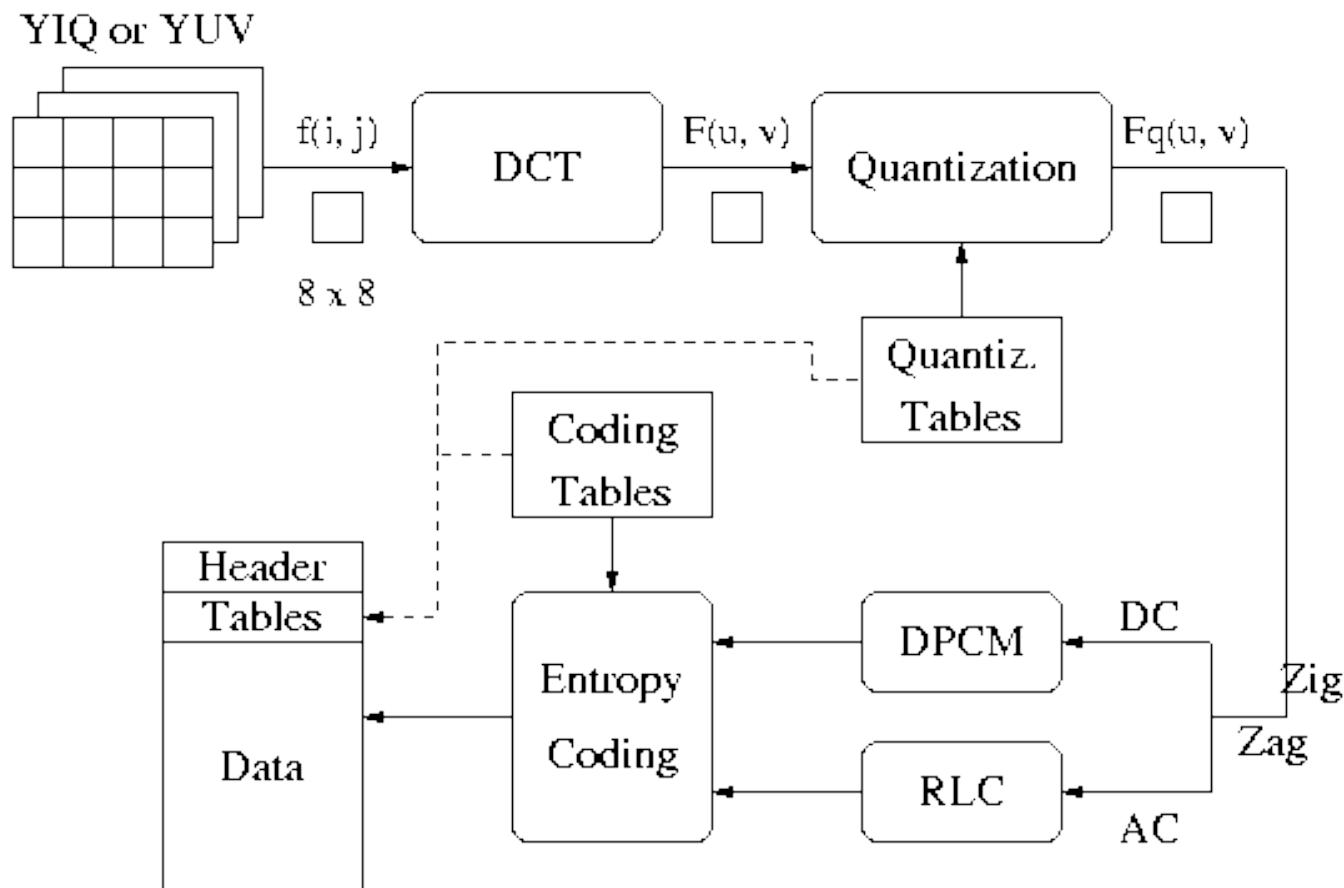
JPEG Kodlama Aşamaları:

- RGB → YCbCr dönüşümü ile ıshıklılık ve renk bileşenleri ayrılır.
- Işıklılık bileşeni korunur ve renk bileşenlerinin sayısı yarıya indirilir (insan gözü ışığa, renkten daha fazla duyarlıdır).
- İmge 8x8 lik bloklara ayrılır ve her bir bloğa Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform-DCT) uygulanarak frekans dönüşümü yapılır.
- Frekans bileşenleri nicemlenerek veri boyutunda azaltma sağlanır. Nicemleme temel olarak düşük frekans bileşenlerini yüksek oranda, yüksek frekans bileşenlerini düşük oranda koruyan yapıdadır.
- Her bir 8x8 blok için elde edilen nicemlenmiş frekans verisi kayıpsız olarak sıkıştırılır.

İmge Sıkıştırma - JPEG



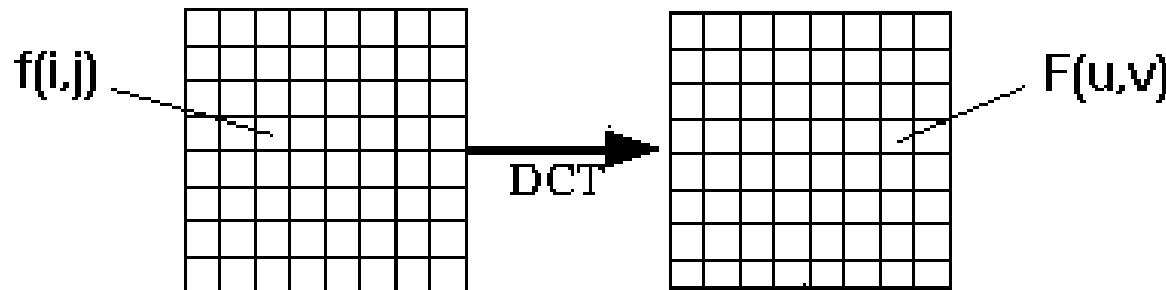
Kodlayıcı Yapısı:



İmge Sıkıştırma - DCT



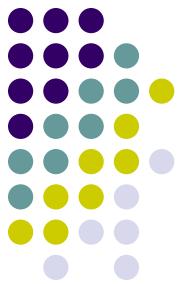
- JPEG'de blokların 8x8 olmasının nedeni, tümdevre teknolojisi ile kullanılmasının daha kolay olmasındandır (2^3). Tek bir işlem süresinde alınan bir blok kolaylıkla işlenebilir.
- JPEG' de ayrık kosinüs dönüşümü (Discrete Cosine Transform-DCT) kullanılmaktadır.
- Fourier dönüşümünde sinüs ve kosinüs bileşenleri kullanılırken, ayrık kosinüs dönüşümünde yalnızca kosinüs bileşenleri kullanılmaktadır. Yani, işaret kosinüs bileşenleri ile ifade edilmekte ve karmaşık uzaya geçilmemektedir.



$$F(u, v) = \frac{\Lambda(u)\Lambda(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos \frac{(2i+1) \cdot u\pi}{16} \cdot \cos \frac{(2j+1) \cdot v\pi}{16} \cdot f(i, j)$$

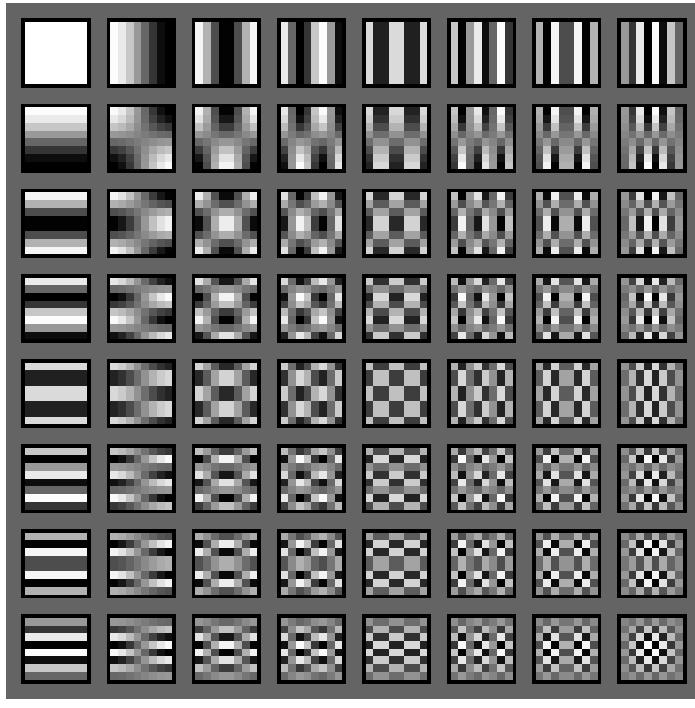
$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

İmge Sıkıştırma - DCT



8x8 DCT temel fonksiyonlar:

DC



HF

Ters DCT (IDCT):

$$\hat{f}(i, j) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 \Lambda(u)\Lambda(v) \cos \frac{(2i+1) \cdot u\pi}{16} \cdot \cos \frac{(2j+1) \cdot v\pi}{16} \cdot F(u, v)$$

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$



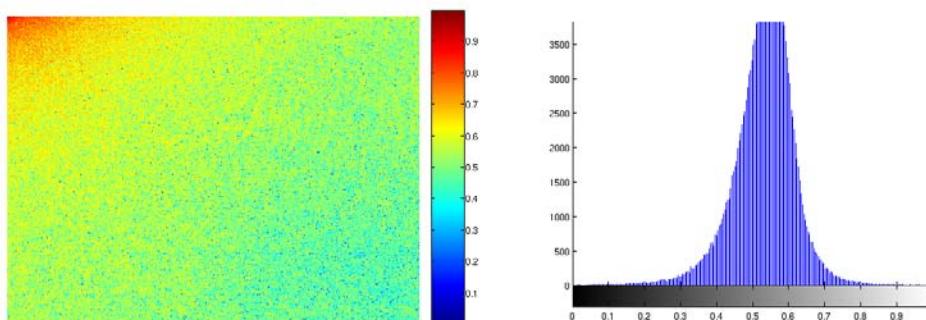
İmge Sıkıştırma - DCT

Avantajı:

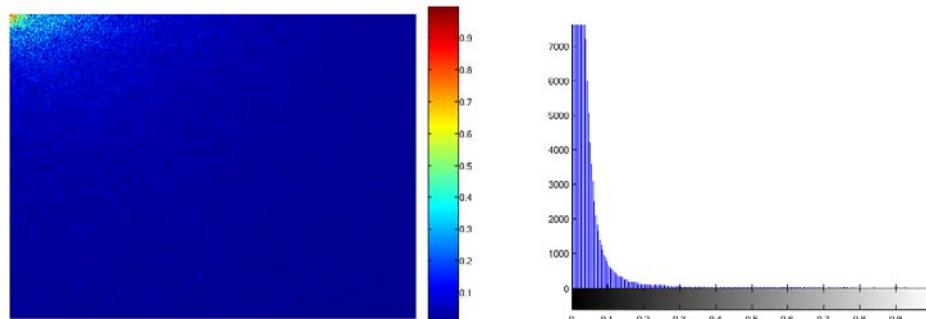
- Enerji yoğunluğu



DFT



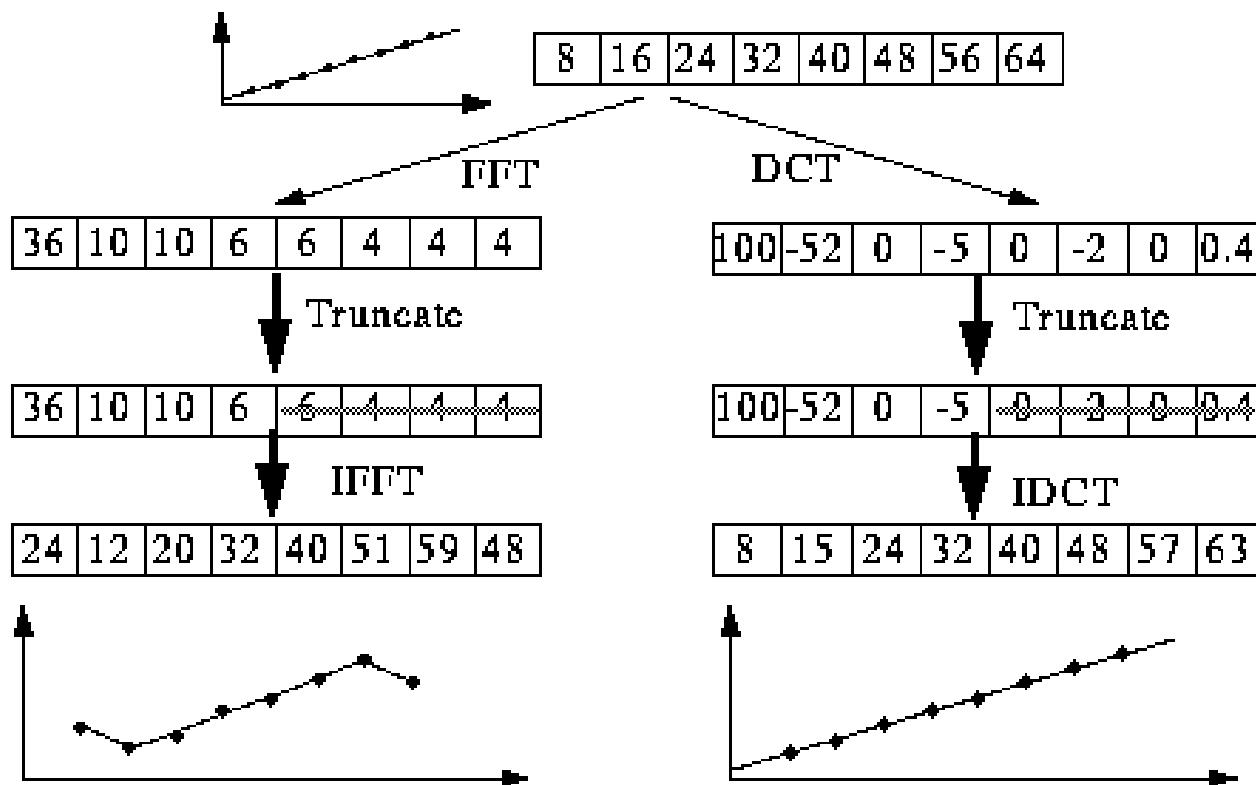
DCT



İmge Sıkıştırma - DCT



Avantajı (Örnek):



İmge Sıkıştırma - DCT



8x8 bir imge bloğunun DCT'sini alma:

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

$\rightarrow (-128) \rightarrow$

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-73	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-60	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

y

DC bilgisi

		u							
		-415	-30	-61	27	56	-20	-2	0
		4	-22	-61	10	13	-7	-9	5
		-47	7	77	-25	-29	10	5	-6
		-49	12	34	-15	-10	6	2	2
		12	-7	-13	-4	-2	2	-3	3
		-8	3	2	-6	-2	1	4	2
		-1	0	0	-2	-1	-3	4	-1
		0	0	-1	-4	-1	0	1	2

v

İmge Sıkıştırma - DCT



DCT katsayılarının nicemlenmesi:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

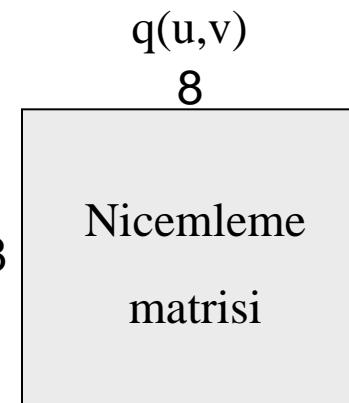
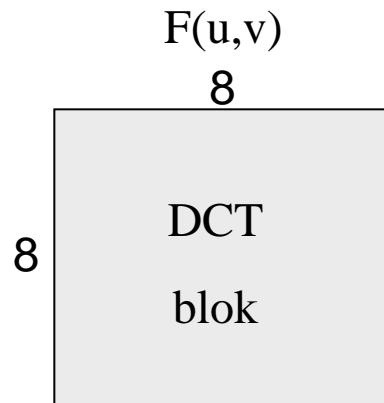
$$B_{j,k} = \text{round} \left(\frac{A_{j,k}}{Q_{j,k}} \right) \text{ for } j = 0, 1, 2, \dots, N_1 - 1; k = 0, 1, 2, \dots, N_2 - 1$$

$$\text{round} \left(\frac{-415}{16} \right) = \text{round} (-25.9375) = -26$$

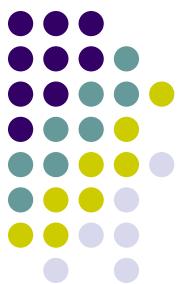
Nicemleme
tablosu

Nicemleme
matrisi

DCT
katsayıları



İmge Sıkıştırma - DCT



DCT katsayılarının nicemlenmesi:

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

Düşük sıkıştırma oranı,
yüksek kalite

1	2	4	8	16	32	64	128
2	4	4	8	16	32	64	128
4	4	8	16	32	64	128	128
8	8	16	32	64	128	128	256
16	16	32	64	128	128	256	256
32	32	64	128	128	256	256	256
64	64	128	128	256	256	256	256
128	128	128	256	256	256	256	256

Yüksek sıkıştırma oranı,
düşük kalite

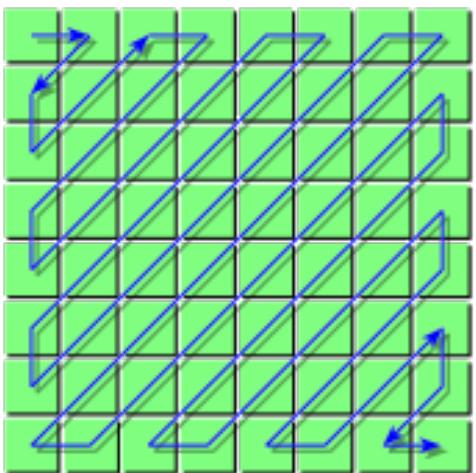


İmge Sıkıştırma – Entropi Kodlama

Zig-zag tarama:

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

-26								
-3	0							
-3	-2	-6						
2	-4	1	-4					
1	1	5	1	2				
-1	1	-1	2	0	0			
0	0	0	-1	-1	-1	-1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0





İmge Sıkıştırma – Entropi Kodlama

Bloğu sonlandırma: -26

-26

-26

-3 0

-3 0

-3 -2 -6

-3 -2 -6

2 -4 1 -4



2 -4 1 -4

1 1 5 1 2

1 1 5 1 2

-1 1 -1 2 0 0

-1 1 -1 2 0 0

0 0 0 -1 -1 0 0

0 0 0 -1 -1 0 EOB

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0

0 0 0 0 0

0 0 0 0

0 0 0

0 0

İmge Sıkıştırma – Entropi Kodlama



- Blokların DC katsayıları büyük ve değişkendir fakat yanındaki bloğun değerine yakındır (ilişki yüksektir).
- Bloklar için DC değerlerin farkları, Farksal Darbe Kod Modülasyonu (Differential Phase Code Modulation-DPCM) ile kodlanır.
- Zig zag tarama sonrası oluşan veride çok sayıda sıfır bulunabilmektedir (Bu sıfırlar yüksek frekans bileşenlerini içermekteydi).
- Bu verileri daha verimli kodlamak için Dizi Uzunluğu Kodlaması (Run Length Encoding-RLE) kullanılmaktadır.

Örn;

WWWWWWWWWWWWBWWWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWW
WWWWWWWWWWWWBWWWWWWWWWWWWWWWWWW

RLE sonrası:

12W1B12W3B24W1B14W

Seklini almaktadır.

İmge Sıkıştırma – Entropi Kodlama



- Tek bir blok için kodlama ve kod çözme:

201	195	188	193	169	157	196	15
193	188	187	201	195	193	213	15
184	192	180	195	182	151	199	15
176	172	179	179	152	148	198	18
196	195	169	171	159	185	218	17
214	213	205	170	173	185	206	15
207	205	207	184	180	167	173	16
198	203	205	186	196	149	159	16

DCT

1480	49	33	-15	-14	33	-38	2
10	-52	11	-12	16	17	-13	-1
19	32	-22	-10	22	-20	9	
16	10	17	27	-31	12	6	
-30	-6	13	-12	8	4	-3	
-25	16	6	-24	9	3	3	
-2	17	4	-6	0	-4	-9	
1	-2	6	0	7	-5	-8	

Q

185	3	1	1	1	-3	2	-1	0
1	1	-1	0	-1	0	0	0	1
0	0	1	0	-1	0	0	0	0
1	1	0	-1	0	0	0	-1	0
0	0	1	0	0	0	-1	0	
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(185 3 1 0 1 1 1 -1 0 1 0 1 1 0 -1
2 -1 0 0 0 0 0 0 1 -1 -1 0 -1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 -1 -1 EOB)

run-level-
coding

Mean of block: 185
(0,3) (0,1) (1,1) (0,1) (0,1)
(0,-1) (1,1) (1,1) (0,1) (1,-3)
(0,2) (0,-1) (6,1) (0,-1) (0,-1)
(1,-1) (14,1) (9,-1) (0,-1)
(EOB)

transmission

Mean of block: 185
(0,3) (0,1) (1,1) (0,1) (0,1)
(0,-1) (1,1) (1,1) (0,1) (1,-3)
(0,2) (0,-1) (6,1) (0,-1) (0,-1)
(1,-1) (14,1) (9,-1) (0,-1)
(EOB)

run-level-
decoding

(185 3 1 0 1 1 1 -1 0 1 0 1 1 0 -1
2 -1 0 0 0 0 0 0 1 -1 -1 0 -1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 -1 -1 EOB)

inverse
zig-zag-

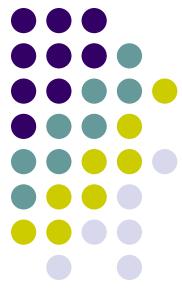
Original 8x8 block

196	193	187	192	179	176	196	189
198	188	182	198	196	192	208	200
185	189	191	197	174	159	184	189
167	181	182	177	154	153	187	189
201	199	178	165	163	185	206	179
220	217	193	176	165	179	197	170
194	198	195	193	169	156	180	179
210	196	192	209	185	149	157	160

scaling
and inverse
DCT

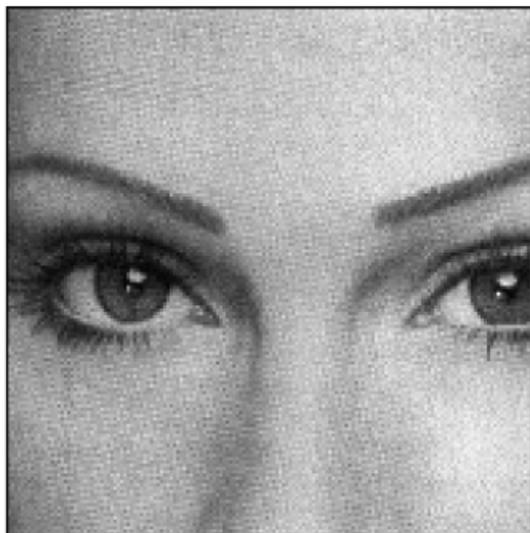
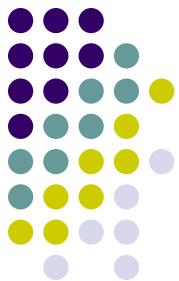
185	3	1	1	1	-3	2	-1	0
1	1	-1	0	-1	0	0	0	1
0	0	1	0	-1	0	0	0	0
1	1	0	-1	0	0	0	-1	0
0	0	1	0	0	0	-1	0	
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

İmge Sıkıştırma – Entropi Kodlama



- RLE ile kodlanmış dizi en son Entropi Kodlama'dan geçirilerek JPEG dosyası oluşturulur.
- Entropi kodlamada:
 - Aritmetik Kodlama
 - Huffman Kodlamakullanılır.

İmge Sıkıştırma - JPEG



orijinal imge



10:1 sıkıştırma oranı



45:1 sıkıştırma oranı

İmge Sıkıştırma - JPEG



%100 kalite, 83 kbayt



%25 kalite, 9.5 kbayt



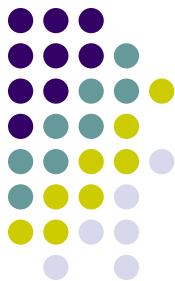
16 Mart 2012

%10 kalite, 4.8 kbayt



%1 kalite 1.5 kbayt

İmge Sıkıştırma – Kalite Ölçütü



Kalite Ölçütü:

- Sıkıştırılmış imgenin orjinaline ne kadar benzer olduğunu bulmak için genelde kullanılan metrik RMS (root mean square) dir.
- Hata:

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

- Toplam hata:

$$e = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]$$

İşaret bağımlı

- Toplam ortalama karesel hatanın karekökü (RMS): ▲

$$e_{RMS} = \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right)^{1/2}$$



İmge Sıkıştırma

SNR (Signal to Noise Ratio):

$$SNR_{RMS} = \left(\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2} \right)^{1/2}$$

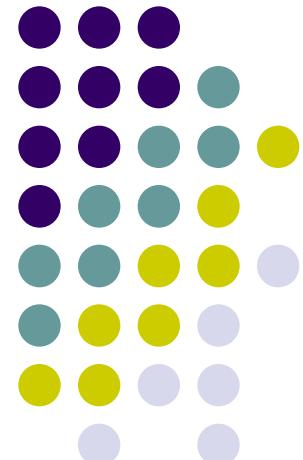
PSNR (Peak Signal to Noise Ratio):

$$PSNR = 20 \log_{10} \left(\frac{MAX_I}{e_{RMS}} \right)$$

iMGE İŞLEME

Ders-10

İmge Sıkıştırma-2



KULIS Kocaeli University Laboratory of Image and Signal Processing
Kocaeli Üniversitesi İşaret ve Görüntü İşleme Laboratuvarı

(Yrd. Doç. Dr. M. Kemal GÜLLÜ)

Dersin web sayfası: http://mf.kou.edu.tr/lohab/kemalg/imge_web/odev.htm

İmge Sıkıştırma-JPEG2000



JPEG2000

- JPEG'e alternatif olarak geliştirilen yeni imge sıkıştırma standardıdır.
- Aynı anda kayıplı ve kayıpsız sıkıştırmaya olanak sağlar.
- 64Mp'den büyük imgeleri sıkıştırabilmektedir.
- Düşük bit hızlarında, JPEG'e göre çok daha iyi sonuçlar vermektedir.
- İkili imgelerdeki sıkıştırma kalitesi arttırlımaktadır.
- Bölgesel (ROI) kodlama sağlamaktadır.
- İletim hattında bozulmalardan daha az etkilenmektedir.
- İmge içerisinde bilgi saklama mümkündür.
- Ayrık dalgacık dönüşümünü (discrete wavelet transform) kullanmaktadır.

İmge Sıkıştırma - 2B DWT



Ayrık Dalgacık Dönüşümü

- İşaretin farklı ölçeklerde ve farklı çözünürlüklerde ayrıştırılmasına olanak sağlar.
- 2-Boyutlu ayrık dalgacık dönüşümü (Discrete Wavelet Transform-DWT), imgeyi farklı çözünürlük seviyelerinde dört bileşenin toplamı olarak açar.
- Dalgacık dönüşüm işlemi satır ve sütunlar için ayrı ayrı 1-boyutlu olarak ayrılabilir.
- 1-B alçak geçiren süzgeç ve yüksek geçiren süzgeç çiftinden oluşmaktadır.

Table 2 Daubechies 9/7 Analysis Filter Coefficients

k	Lowpass Filter (h_k)	Highpass Filter (g_k)
0	0.6029490182363579	1.115087052456994
± 1	0.2668641184428723	-0.5912717631142470
± 2	-0.07822326652898785	-0.05754352622849957
± 3	-0.01686411844287495	0.09127176311424948
± 4	0.02674875741080976	

Table 3 Le Gall 5/3 Analysis Filter Coefficients

k	Lowpass Filter (h_k)	Highpass Filter (g_k)
0	6/8	1
± 1	2/8	$\frac{1}{2}$
± 2	-1/8	

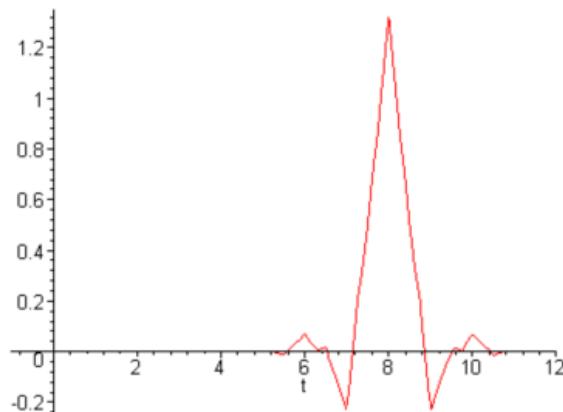
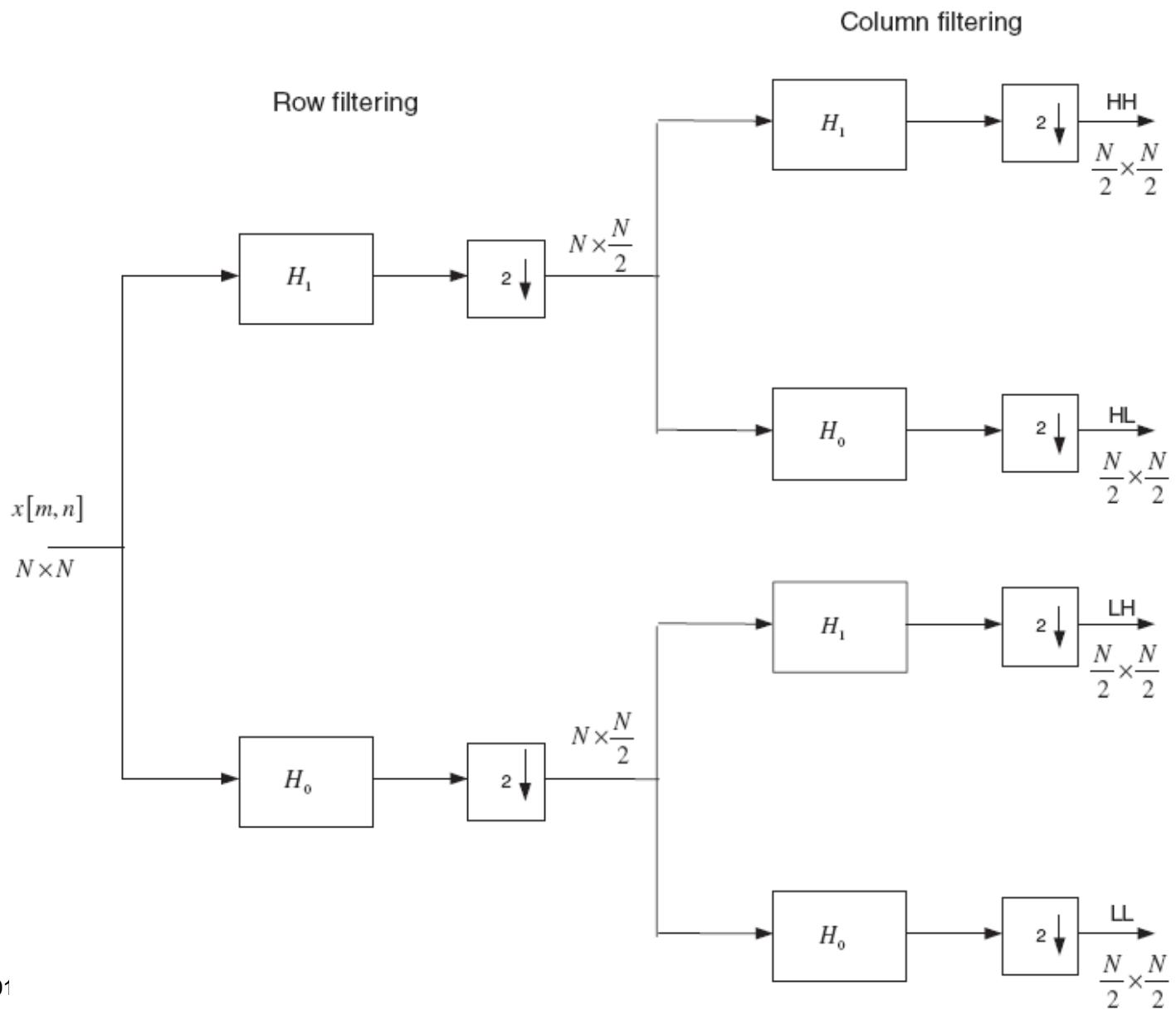
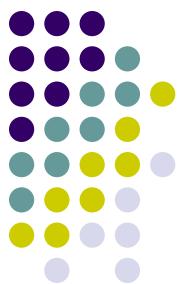
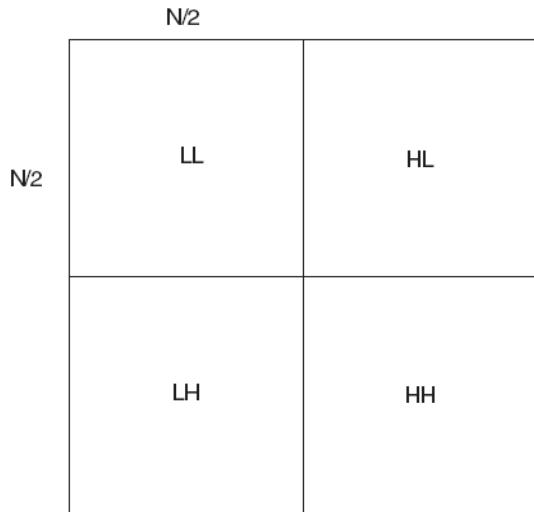


Figure 8 Wavelet corresponding to Daubechies 9-tap/7-tap filter

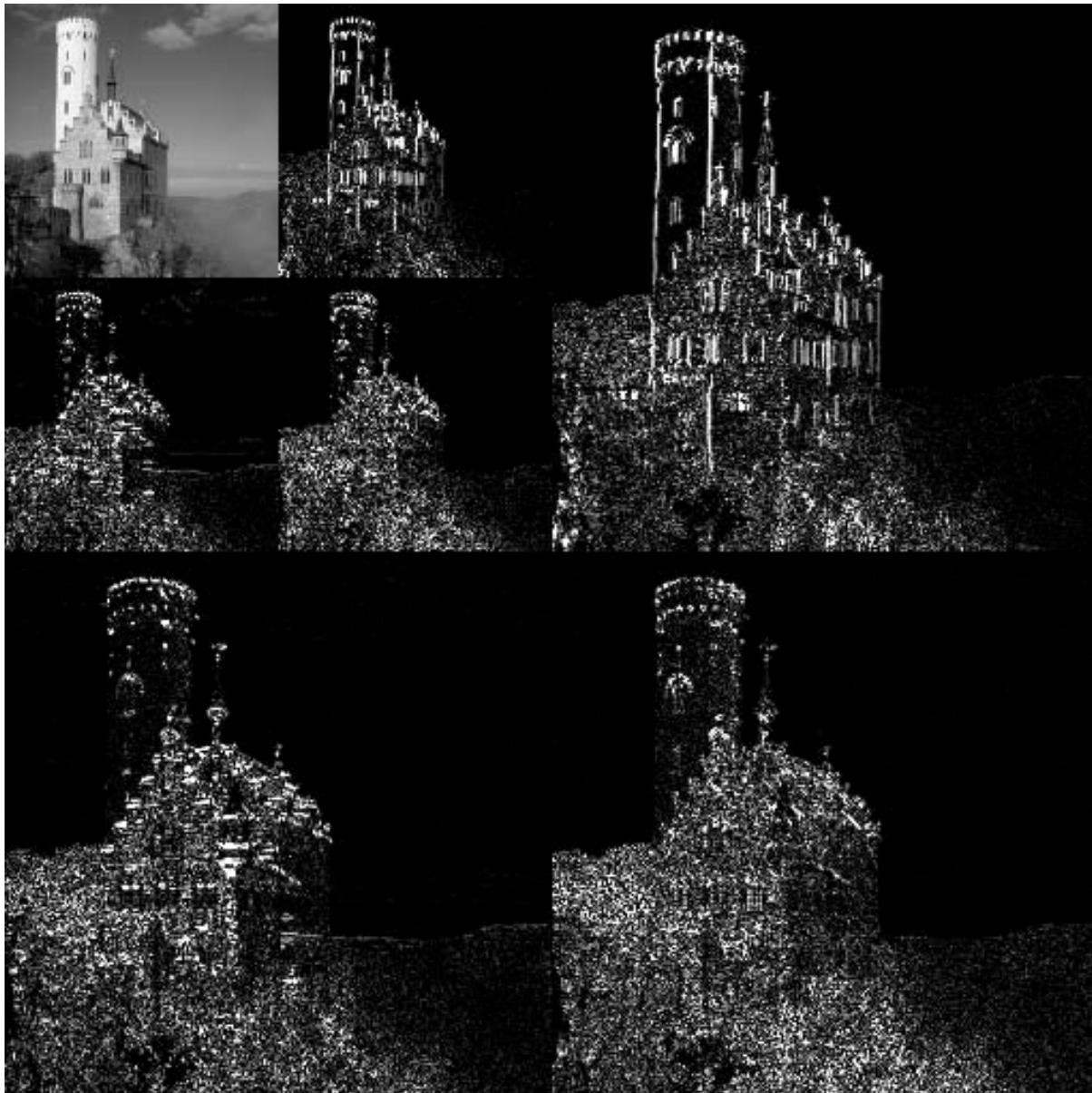
İmge Sıkıştırma – 2B DWT



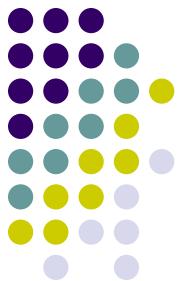
İmge Sıkıştırma - 2B DWT



İmge Sıkıştırma - 2B DWT



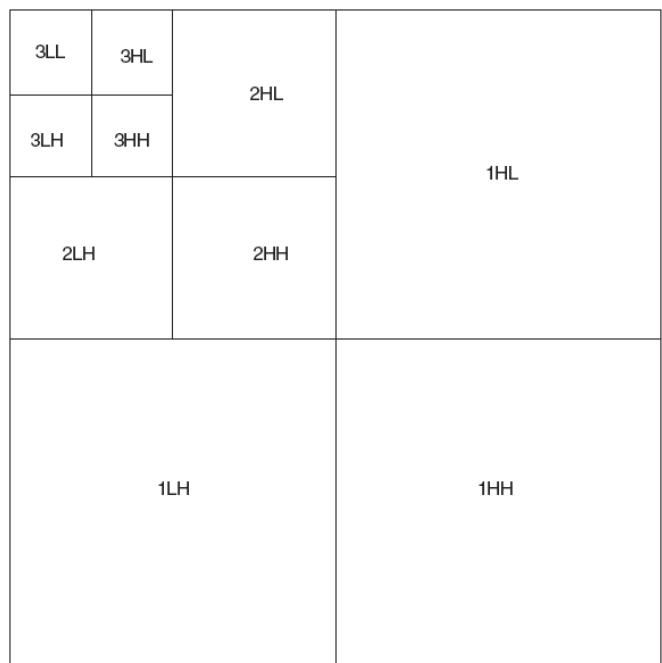
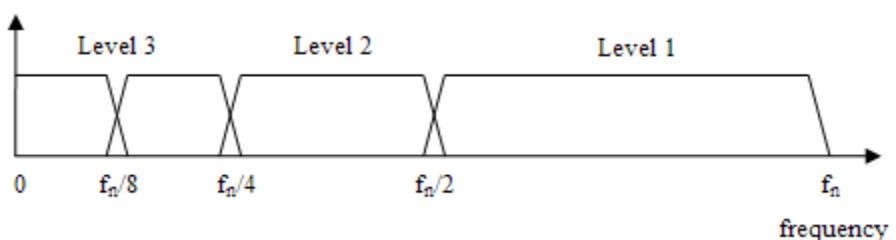
İmge Sıkıştırma - 2B DWT



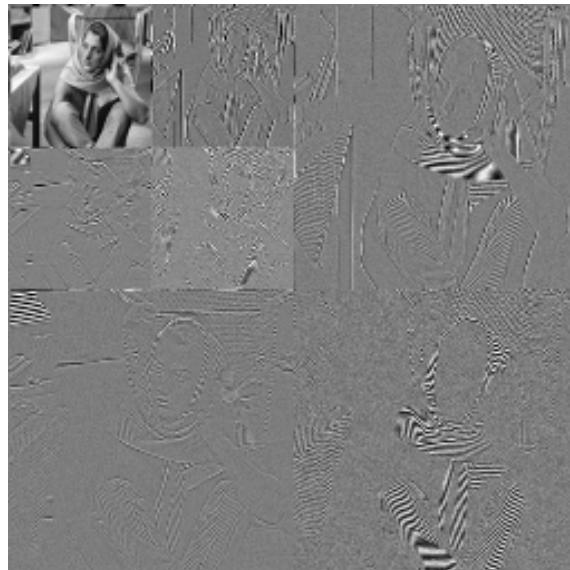
32 örnekli bir işaret için

Level Frequencies Samples

3	0 to $f_n / 8$	4
3	$f_n / 8$ to $f_n / 4$	4
2	$f_n / 4$ to $f_n / 2$	8
1	$f_n / 2$ to f_n	16



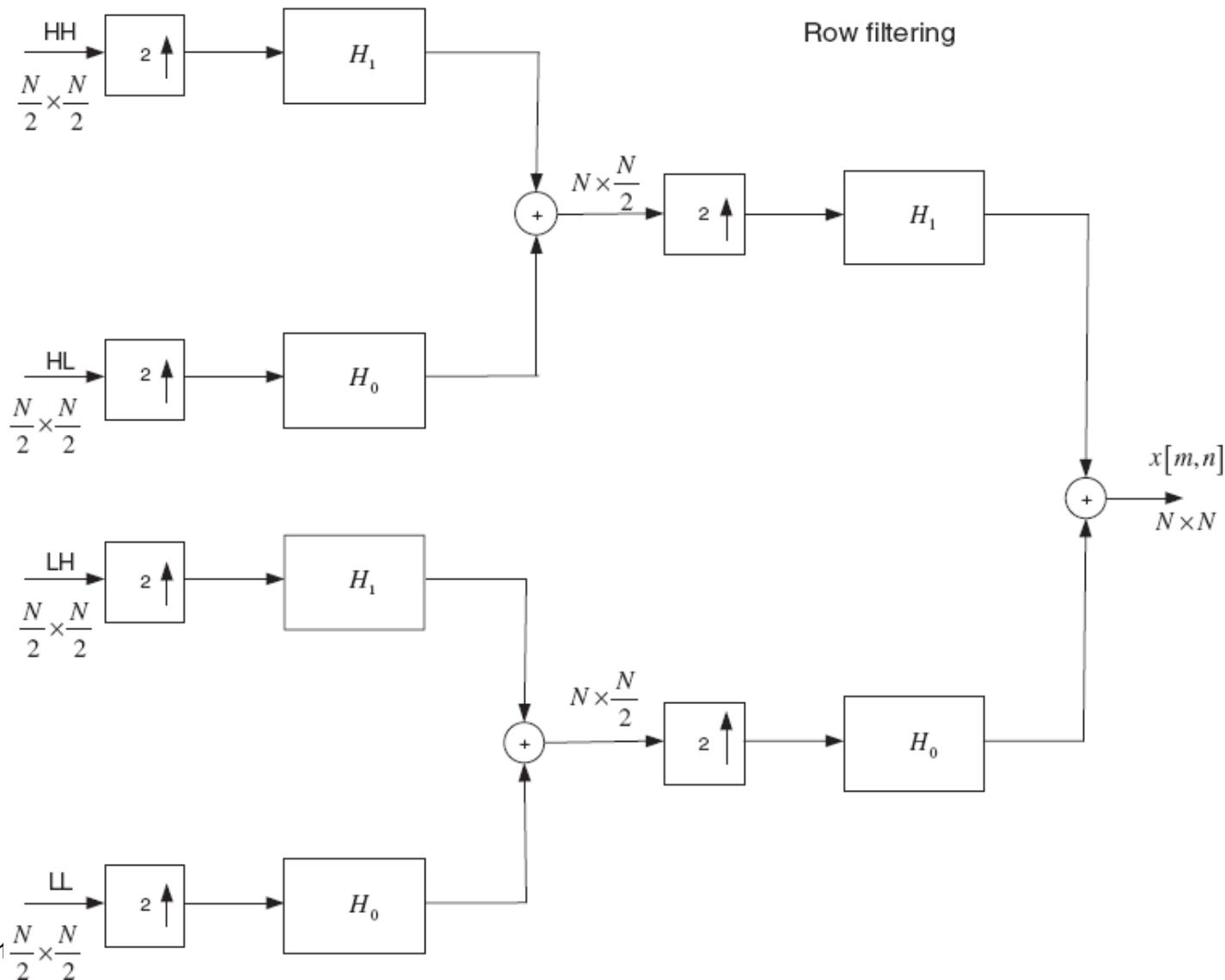
İmge Sıkıştırma - 2B DWT



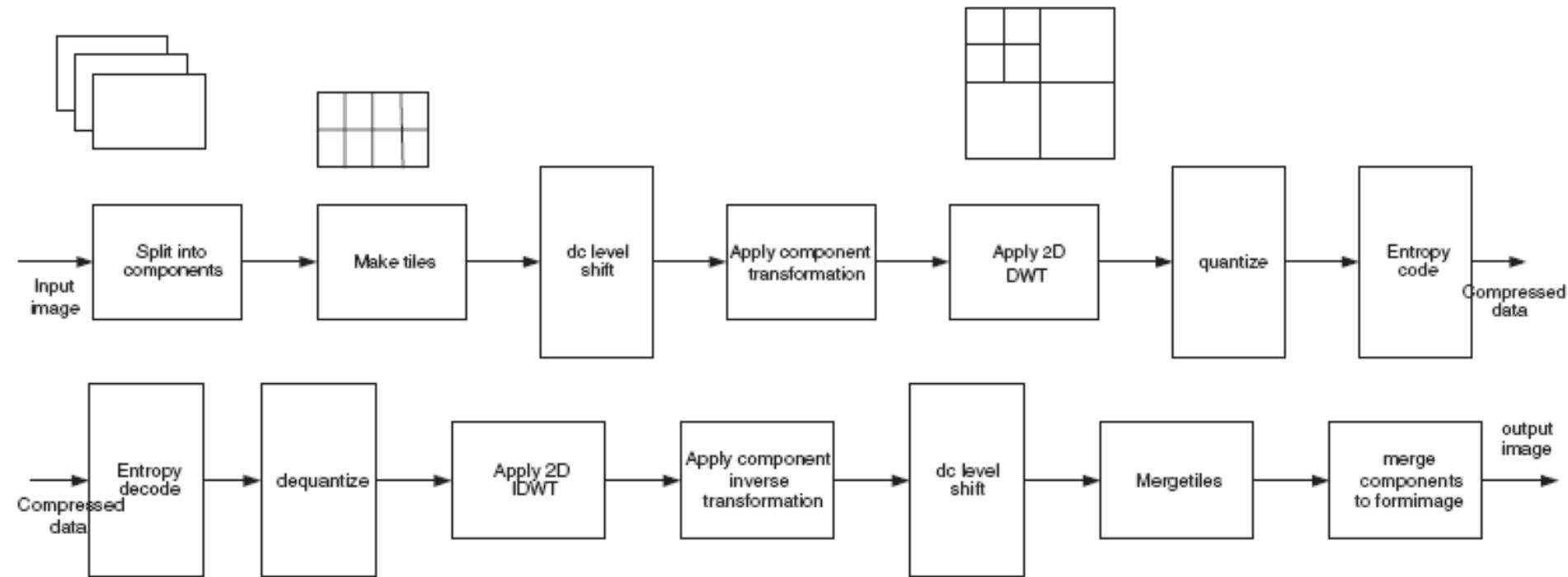
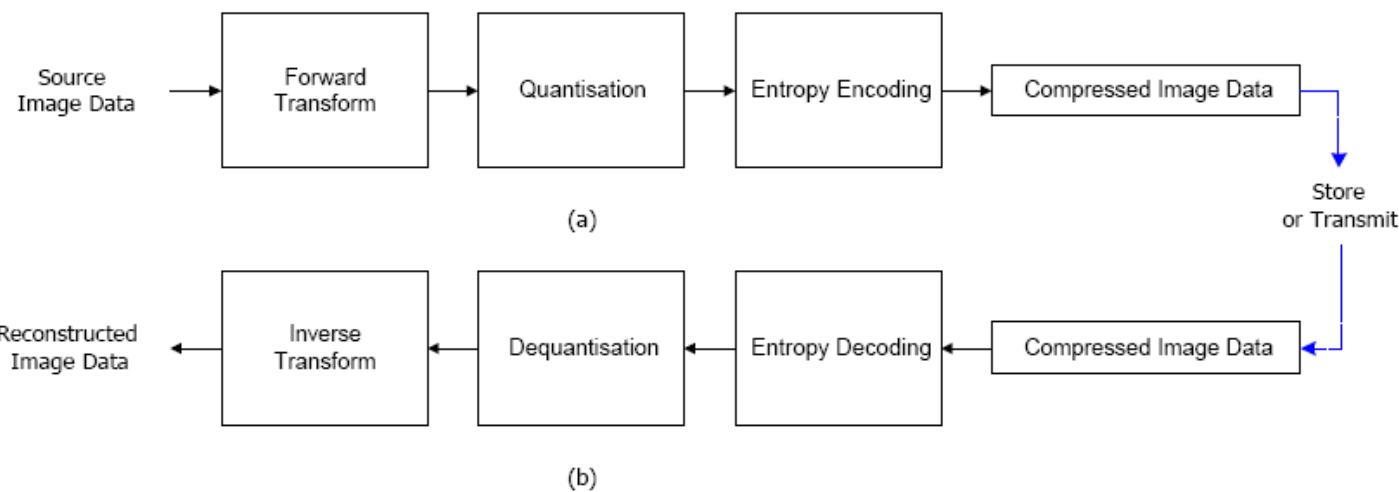
İmge Sıkıştırma - 2B DWT



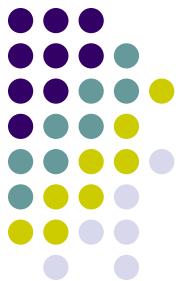
Column filtering



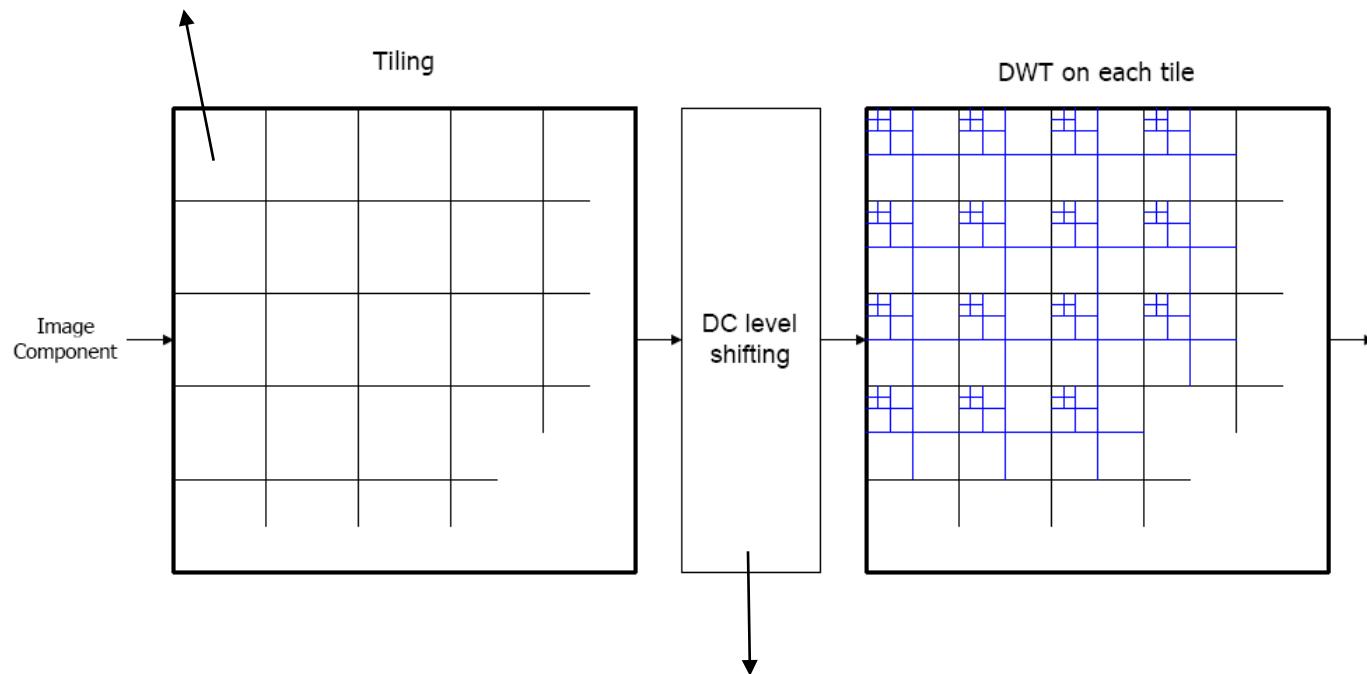
İmge Sıkıştırma-JPEG2000



İmge Sıkıştırma-JPEG2000

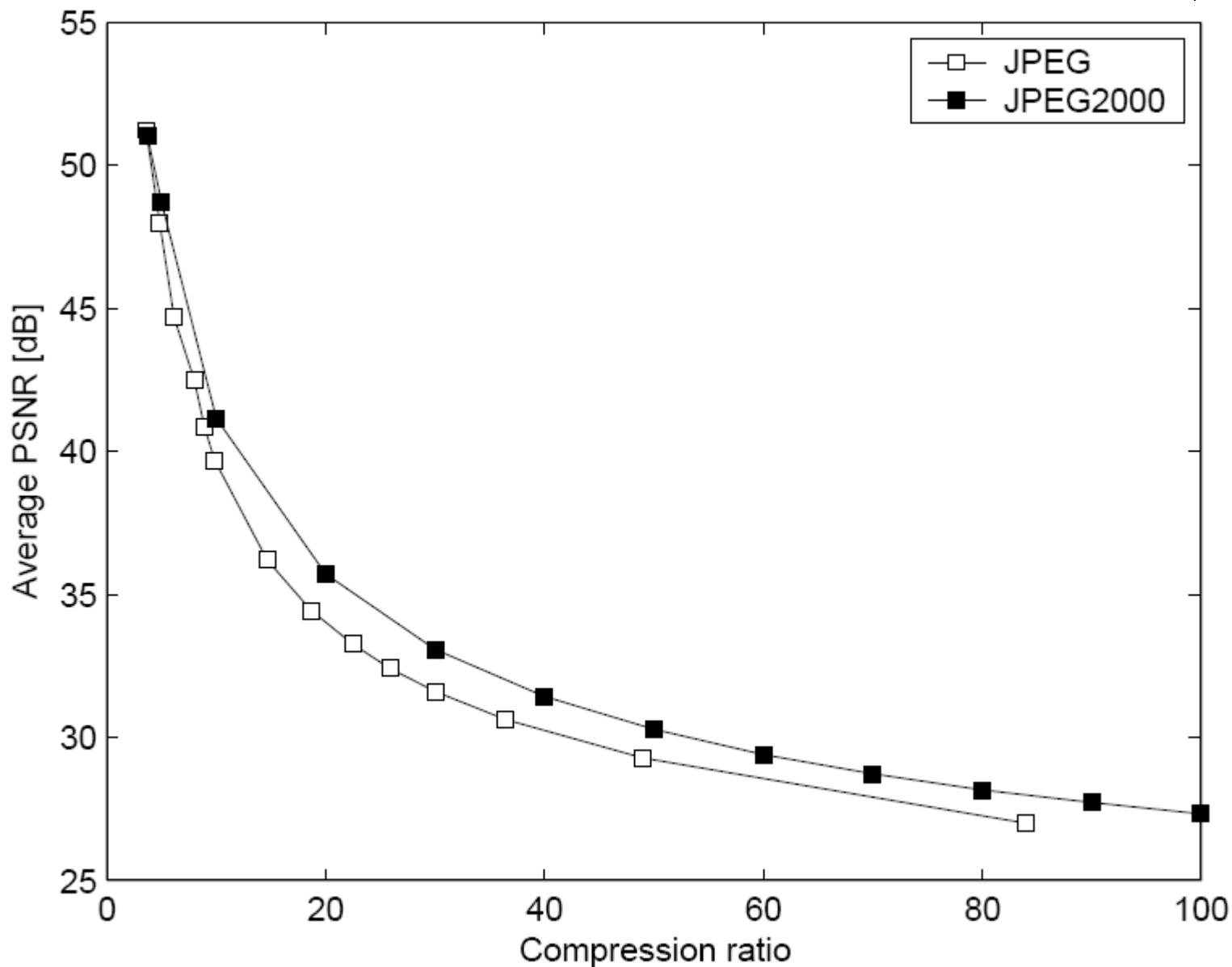


Her bir blok ayrı kodlanmaktadır.

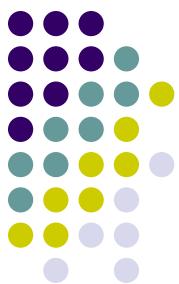


Her bloğun ortalaması aynı oranda kaydırılır.

İmge Sıkıştırma-JPEG2000



İmge Sıkıştırma-JPEG2000



JPEG



JPEG 2000



JPEG



JPEG 2000

İmge Sıkıştırma-JPEG2000



JPEG 5kB



JPEG2000 5kB

Dalgacık Uzayı



Matlab: wavemenu

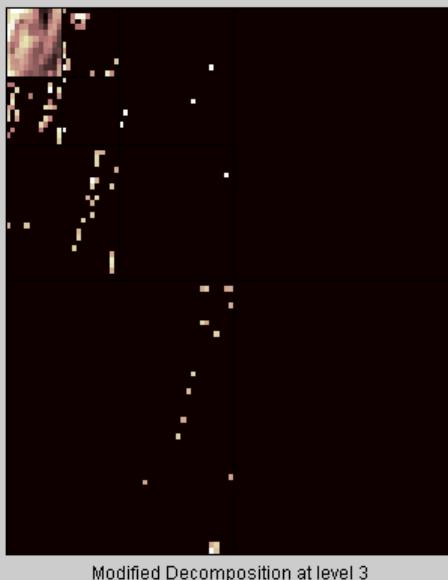
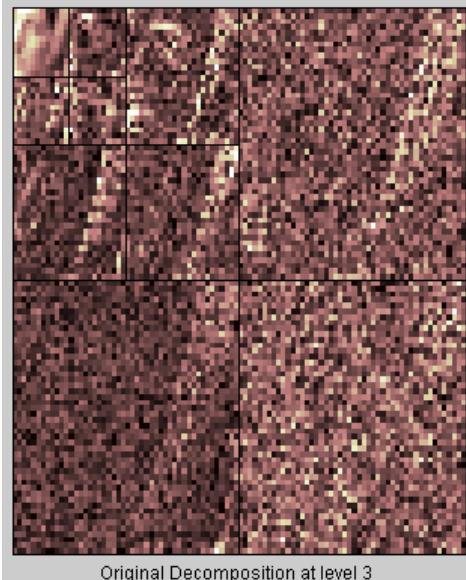
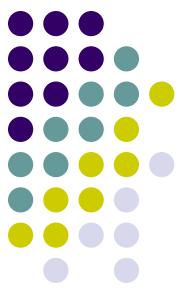


Dalgacık uzayı gürültü giderme

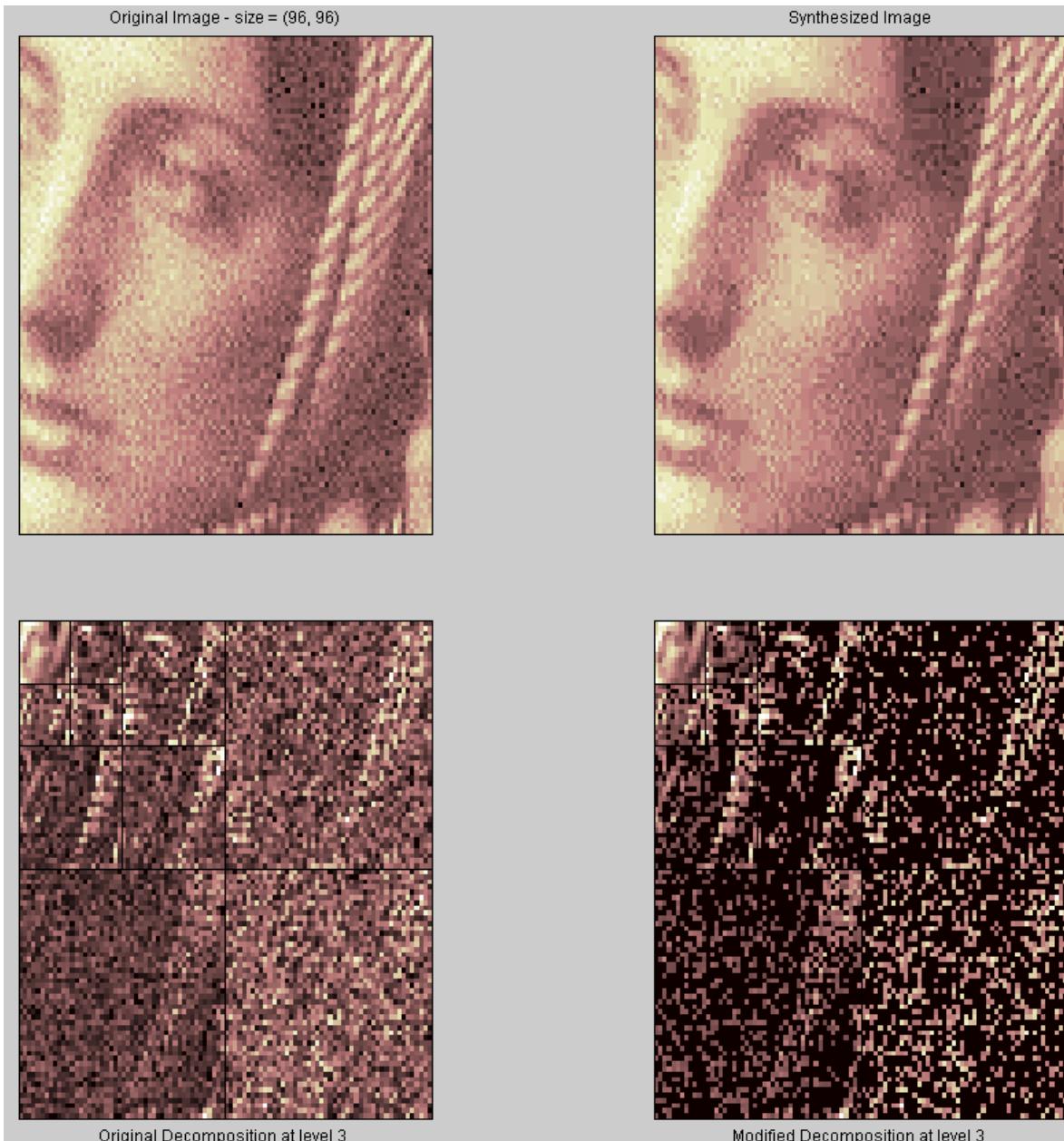
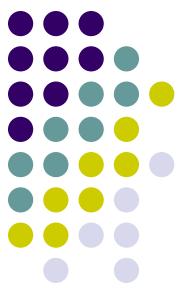


1-B bir sinüs işaretini oluşturup AWGN ekleyelim ve gürültülü işaretin dalgacık uzayında süzgeçleyelim.

Dalgacık uzayı gürültü giderme



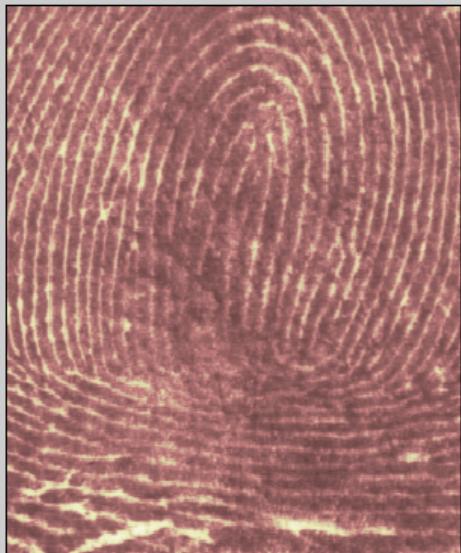
Dalgacık uzayı gürültü giderme



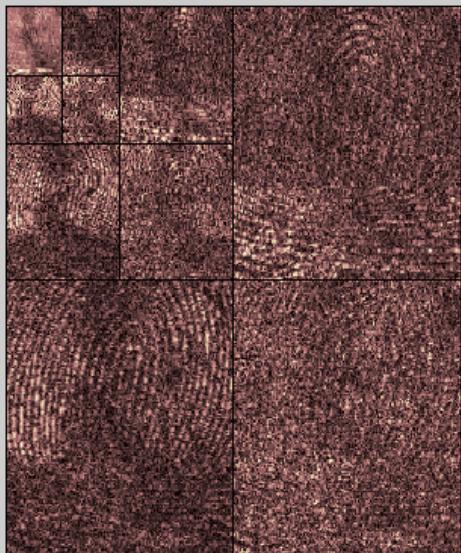
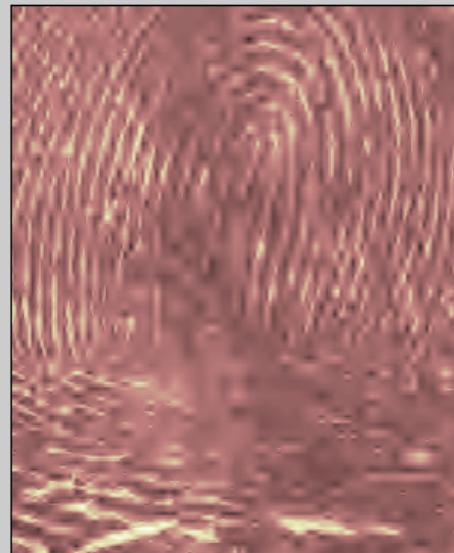
Dalgacık uzayı gürültü giderme



Original Image - size = (296, 296)



Synthesized Image

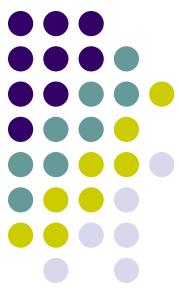


Original Decomposition at level 3

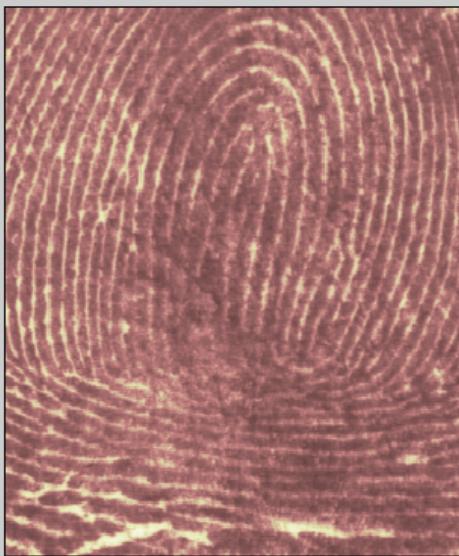


Modified Decomposition at level 3

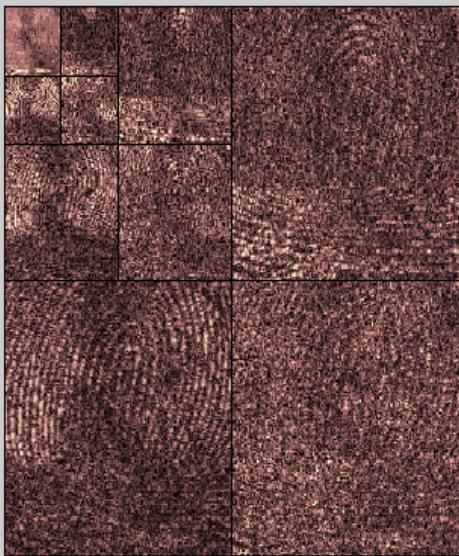
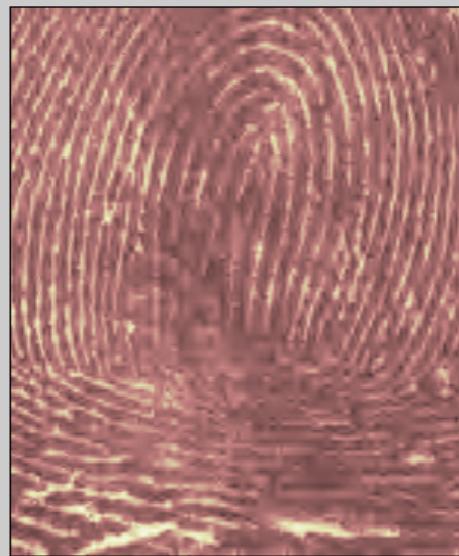
Dalgacık uzayı gürültü giderme



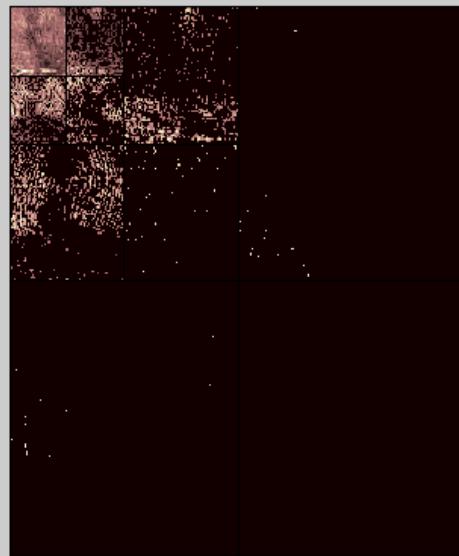
Original Image - size = (296, 296)



Synthesized Image

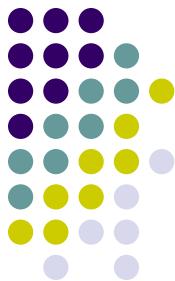


Original Decomposition at level 3



Modified Decomposition at level 3

İmge Sıkıştırma – Kalite Ölçütü



Kalite Ölçütü:

- Sıkıştırılmış imgenin orjinaline ne kadar benzer olduğunu bulmak için genelde kullanılan metrik RMS (root mean square) dir.
- Hata:

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

- Toplam hata:

$$e = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]$$

- Toplam ortalama karesel hatanın karekökü (RMS):

$$e_{RMS} = \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right)^{1/2}$$



İmge Sıkıştırma

SNR (Signal to Noise Ratio):

$$SNR_{RMS} = \left(\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2} \right)^{1/2}$$

PSNR (Peak Signal to Noise Ratio):

$$PSNR = 20 \log_{10} \left(\frac{MAX_I}{e_{RMS}} \right)$$