

SAYISAL GÖRÜNTÜ İŞLEME

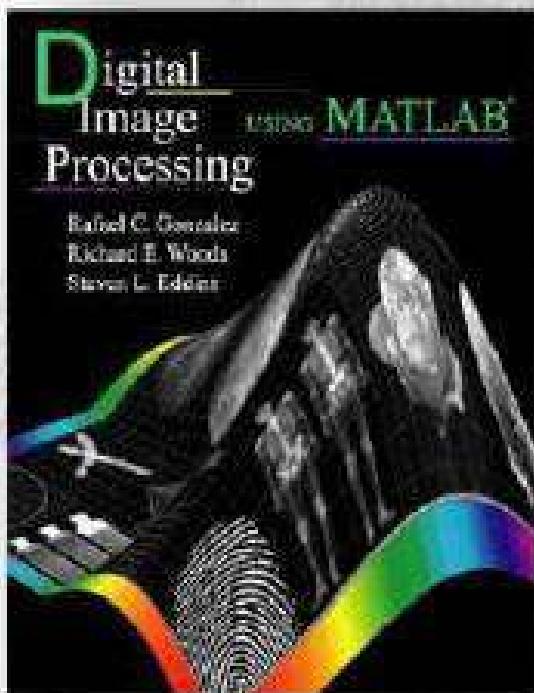
Hafta1: Sayısal Görüntü İşlemeye Giriş

Ders İçeriği

- Görüntü işleme nedir ?
- Görüntü işleme tekniklerinin kullanma alanları
- Matlab görüntü işleme araç kutusu

Digital Image Processing

Gonzalez • Woods • Eddins Using MATLAB®



The first textbook dealing with MATLAB software implementation of image processing concepts.

Support Materials

- [M-Files](#)
- [Projects](#)
- [Tutorials](#)
- [Teaching Materials](#)
- [Publications](#)
- [Book Updates](#)
- [Links](#)
- [Image Databases](#)
- [Search](#)
- [Contact Us](#)

About the Book

- [Description](#)
- [Sample Material](#)
- [About the Authors](#)
- [How to Order](#)

Visit the Site for Gonzalez & Woods
Digital Image Processing, 2nd edition

[Home](#)

Sayısal Görüntü İşleme Nedir?

- **İmgenin Tanımı**

- İmge 2-D bir fonksiyondur : $f(x,y)$.
 - x ve y spatial (uzaysal) koordinatlardır
 - f fonksiyonunun (x,y) noktasındaki genliği yoğunluk(intensity) ya da imgenin (x,y) 'deki gri seviyesi olarak bilinir.

- **Sayısal Görüntü (İmge) Tanımı**

- Eğer x ve y sonlu ve kesikli değerler ise, $f(x,y)$ sayısal imge olarak tanımlanır
- $f(x,y)$ negatif değerler almaz.

- **Sayısal Görüntü İşleme**

- Sayısal imgenin bilgisayar yardımıyla işlenmesidir

- **Pixel'in tanımı**

- Sayısal bir imgenin belli bir yer ve değere sahip elemanlarıdır

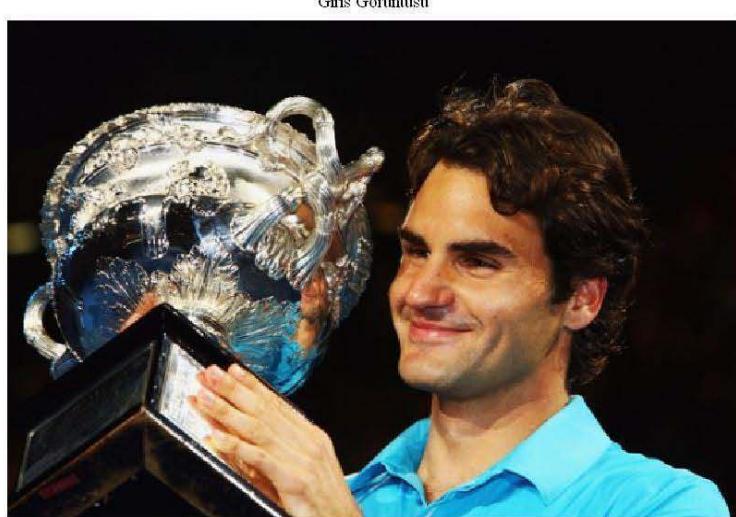
SAYISAL GÖRÜNTÜ NEDİR?

- Renkli Görüntü
- Üç fonksiyonla gösterilebilir.
 - $R(x,y)$: kırmızı bileşen
 - $G(x,y)$:yeşil bileşen
 - $B(x,y)$:mavi bileşen

Sayısal Görüntü İşleme Nedir?

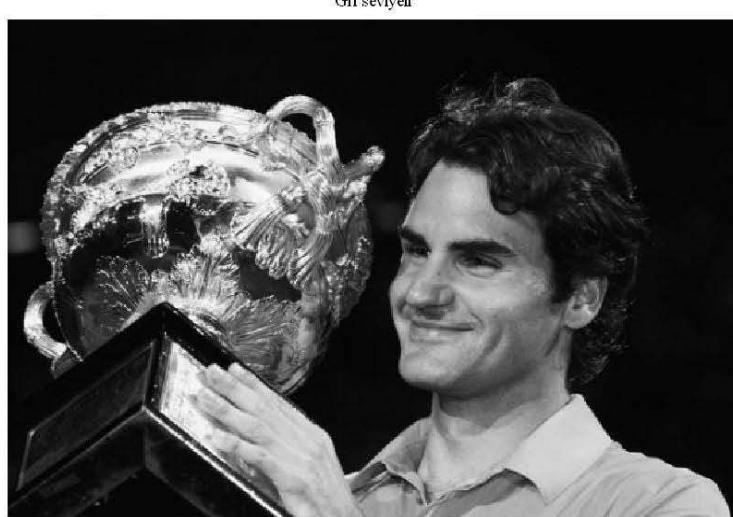
- Sayısal görüntü işleme, sayısal görüntülerin bilgisayarla işlenmesidir.
- Sayısal görüntü, her birinin özel bir konumu ve değeri olan sonlu sayıdaki elemanlardan oluşmuştur.
 - Bu elemanlar, resim elemanları, görüntü elemanları, pel, imgecik veya piksel olarak adlandırılır. Piksel terimi en yaygın kullanılıdır.

Gri Seviyeli (Grayscale) / Renkli (RGB) Görüntü



Renkli görüntü

$$I(m, n, 3)$$



Gri Seviyeli görüntü

$$I(m, n)$$



“Lena” görüntüsü

Gri Seviyeli (Grayscale) / Renkli (RGB) Görüntü



Sayısal Görüntü İşleme - Başlangıcı

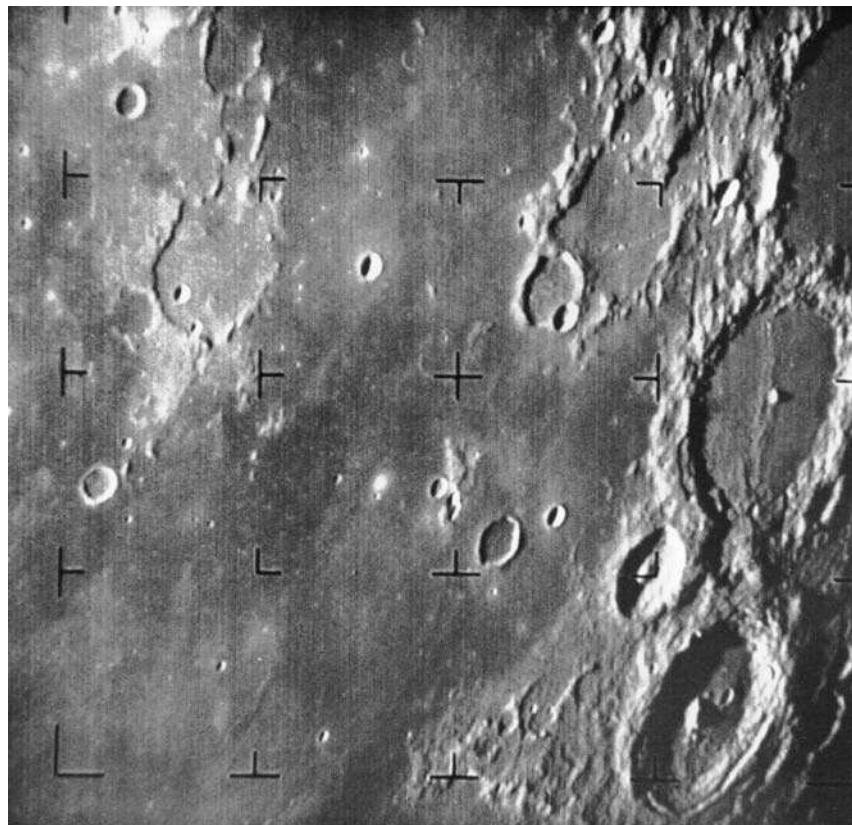
- **Sayısal İmgelerin ilk kullanımı**

- Sayısal imgeler ilk olarak denizaltı kabloları aracılığı ile Londra – New York arasında gönderilmiştir

- **Bilgisayarların Sayısal İşaret İşlemede kullanımı**

- Bilgisayarlar sayısal görüntü işlemede ilk olarak, Jet Propulsion Laboratuvarlarında (Pasadena, California) televizyon kameralarından kaynaklanan çeşitli bozulmaları düzeltmek için kullanılmıştır.

Sayısal Görüntü İşleme - Başlangıcı



- Ay'ın çekilmiş olan ilk resmi
- Amerikan uzay gemisi (Ranger 7, 1964) tarafından çekilmiştir

Sayısal Görüntü İşlemenin Uygulama Alanları

- **Tıbbi Uygulamalar**
- **Uzay Uygulamaları**
- **Dünya Kaynaklarının uzaktan izlenemsi**
- **Astronomi**
- **Makine Algısı**

Sayısal Görüntü İşlemenin Uygulama Alanları

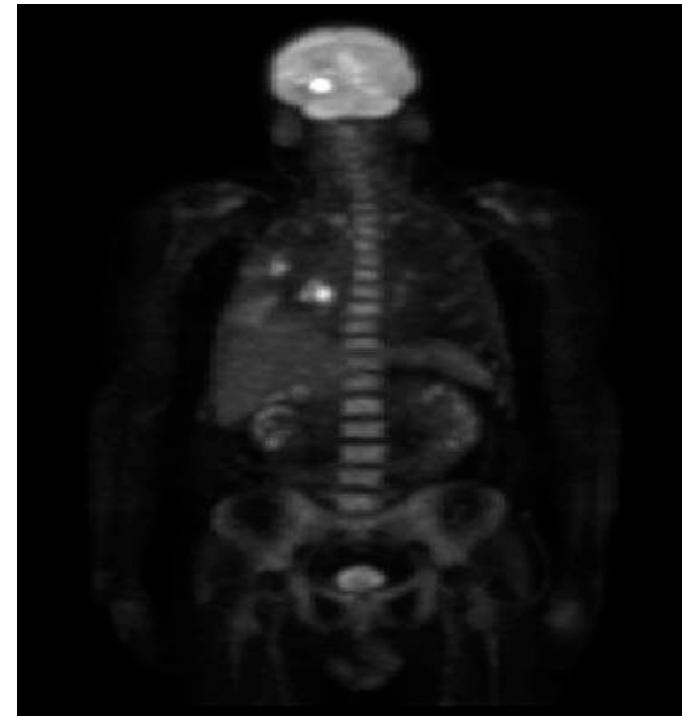
- **Tıbbi Uygulamalar**

- Biyomedikal görüntülerin işlenmesi
- Nükleer manyetik rezonans(NMR) ve ultrasonik taramada elde edilen tıbbi görüntülerin işlenmesi.
- Bu görüntüler hastalığı görüntülemek ve teşhis etmek, tümör, damar tikanıklığı, kemiklerdeki kırıklar gibi hastalıkların anlaşılmasıında kullanılır.

Sayısal Görüntü İşlemenin Uygulama Alanları - Örnekler



Kemik Taraması



PET imgesi

Sayısal Görüntü İşlemenin Uygulama Alanları - Örnekler

- X-Ray Görüntüleme



(a) Göğüs X-Ray Görüntüsü



(b) Aort Anjiogram

Sayısal Görüntü İşlemenin Uygulama Alanları - Örnekler



Aortic angiogram



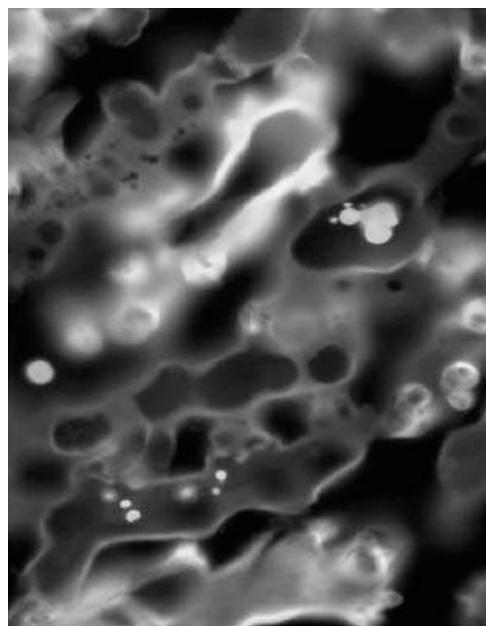
Head CT.

Sayısal Görüntü İşlemenin Uygulama Alanları

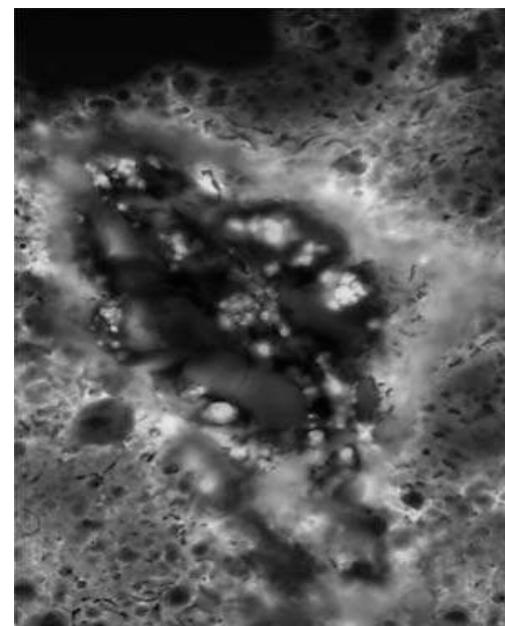
- Coğrafi Bilimler
- Jeodezi, hava ve uydu görüntülerinden hava tahmini
- Yeryüzündeki doğal kaynakların izlenmesi,
- Coğrafi haritaların çıkarılması,
- Zirai ürünlerin tahmini,
- Ormanların gelişiminin izlenmesi,
- Hava durumu,
- Sel ve yangın kontrolü ve birçok çevresel uygulamalar

Sayısal Görüntü İşlemenin Uygulama Alanları - Örnekler

- Ultraviole Bandında Görüntüleme



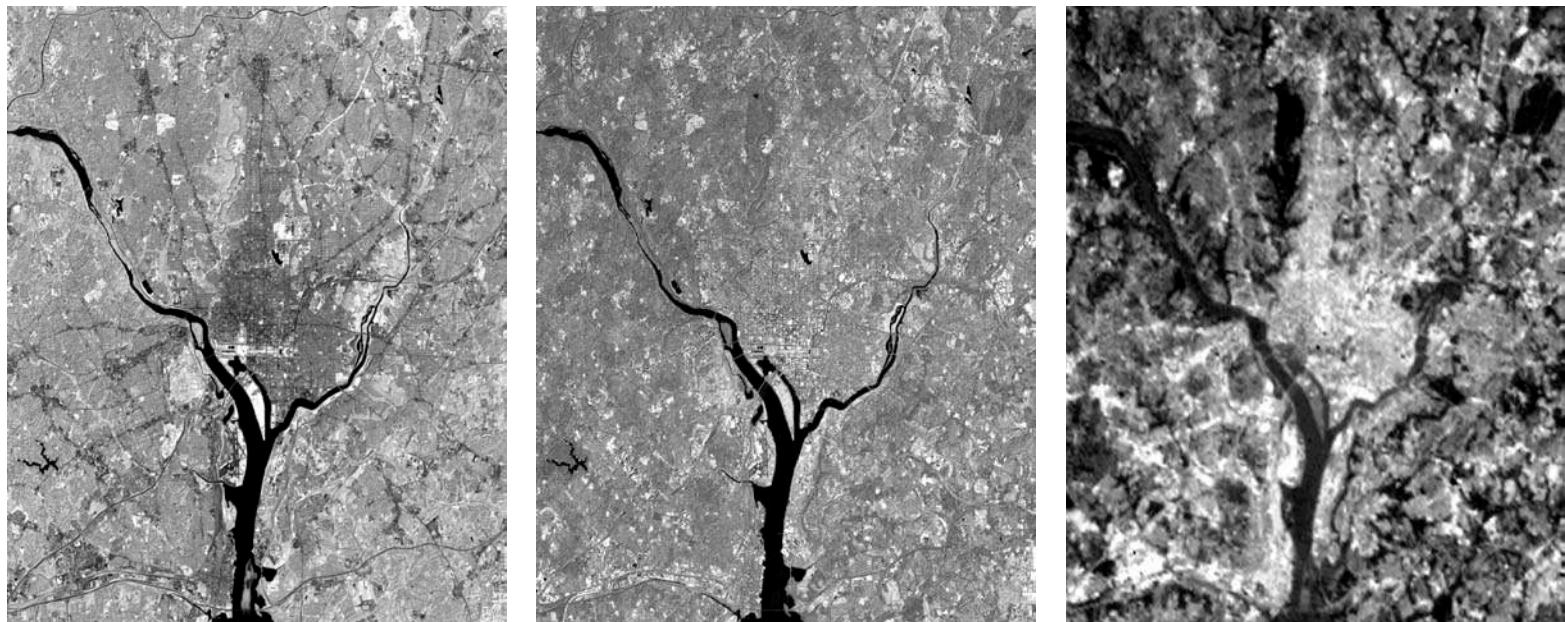
Normal mısırın Floresan mikroskop imagesi



Bozulmuş mısır imagesi

Sayısal Görüntü İşlemenin Uygulama Alanları - Örnekler

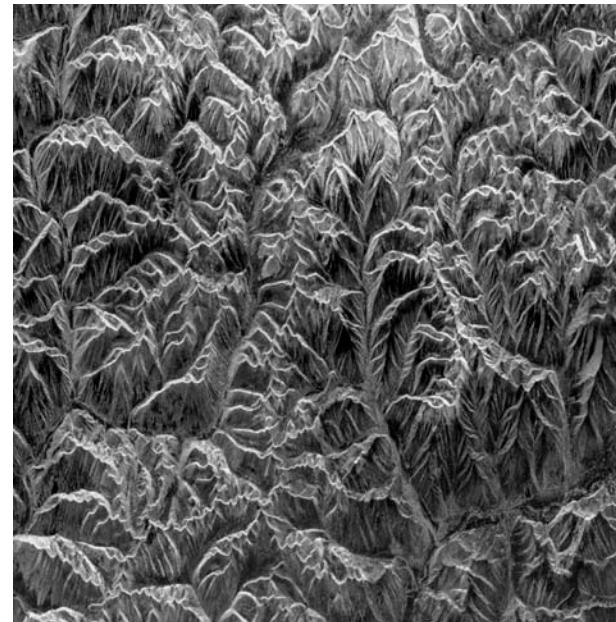
- Görülebilir ve kızılötesi bantta görüntüleme



Washington, D.C. Bölgesinin uzaydan çekilmiş resimleri

Sayısal Görüntü İşlemenin Uygulama Alanları - Örnekler

- Mikrodalga bandında görüntüleme



Güneydoğu Tibet dağlarının uzaydan çekilmiş radar imgesi

Sayısal Görüntü İşlemenin Uygulama Alanları - Örnekler

- Farklı görüntüleme teknikleri



Ultrasound görüntüleme örnekleri

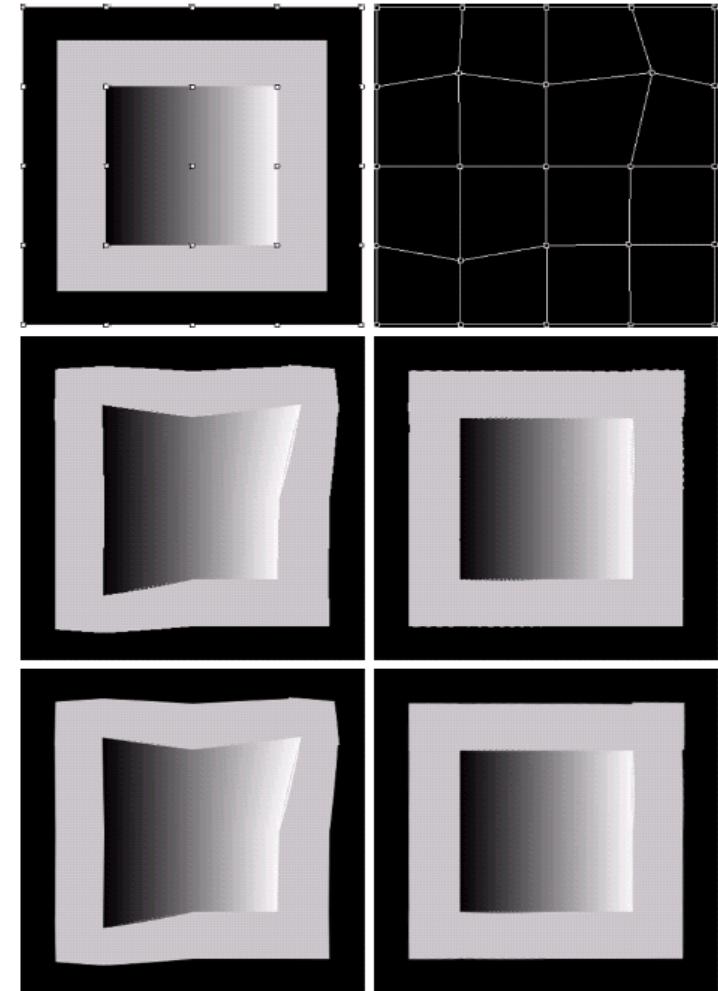
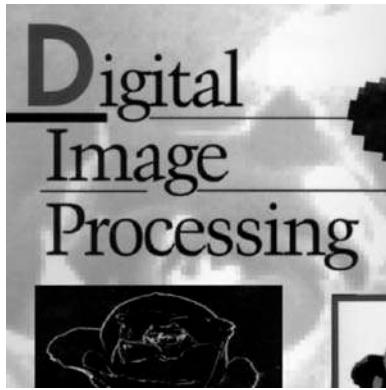
Sayısal Görüntü İşlemede Temel Adımlar

- Görüntü İyileştirme



Sayısal Görüntü İşlemede Temel Adımlar

- Gürültü Temizleme**
- İmge İyileştirmesi**
- Geometrik düzeltme**
- Radyometrik düzeltme**



Sayısal Görüntü İşlemede Temel Adımlar

- **İmge Sıkıştırma**

- Kayıpsız Sıkıştırma
- Kayıplı Sıkıştırma

Sayısal Görüntü İşlemede Temel Adımlar

- İmge Bölütleme

images\xrp4.tif



Wang's method, optimal k=9



MATLAB İLE GÖRÜNTÜ İŞLEME



- Görüntü işlemede kullanılan standart resimlerden en ünlüsü: lena imgesi
- <http://sipi.usc.edu/database/database.php?volume=misc>
 - sayfasından indirilebilir

LENA image: 512x512 pixels, 768kb

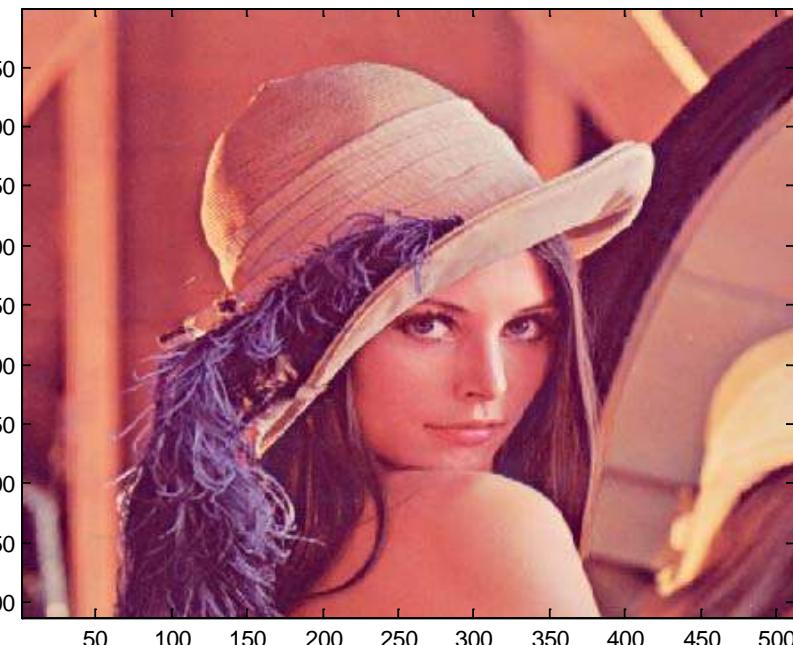
Color (24 bits/pixel)

Dosyaadı: 4.2.04.tiff

MATLAB İLE GÖRÜNTÜ İŞLEME

- Resimlerin MATLAB formatına dönüştürülmesi
- `imread`: Read image from graphics file
 - `A = imread(filename, fmt)`
 - reads a grayscale or color image from the file specified by the string `filename`.
 - `filename`: Dosyaadı
 - `fmt`: specifies the format of the file by its standard file extension.
 - standart dosya uzantısı
 - `[X, map] = imread(...)`
 - reads the indexed image in `filename` into `X` and its associated colormap into `map`.

MATLAB İLE GÖRÜNTÜ İŞLEME

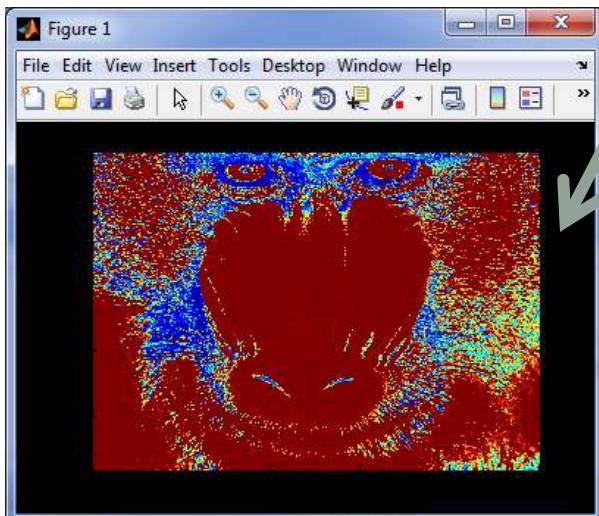
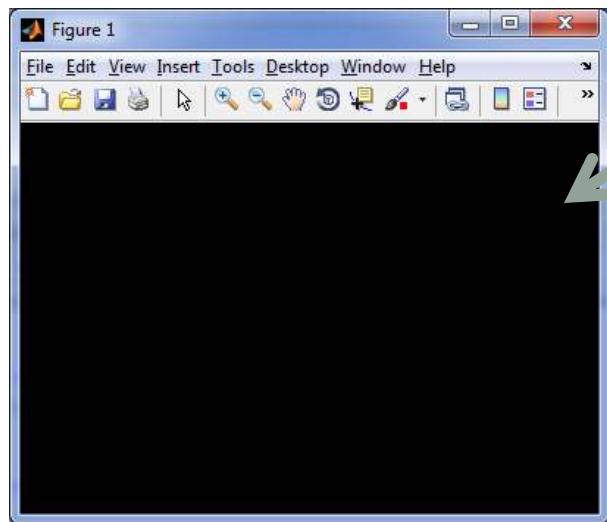


- Lena görüntüsünü indirdikten sonra:
 - `lena=imread('4.2.04','tiff');`
 - lena: MATLAB formatına dönüştü
 - Görüntülemek için
 - `image(lena)`

MATLAB İLE GÖRÜNTÜ İŞLEME

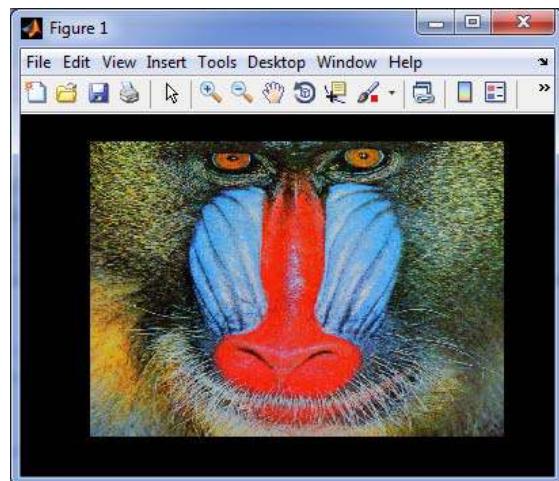
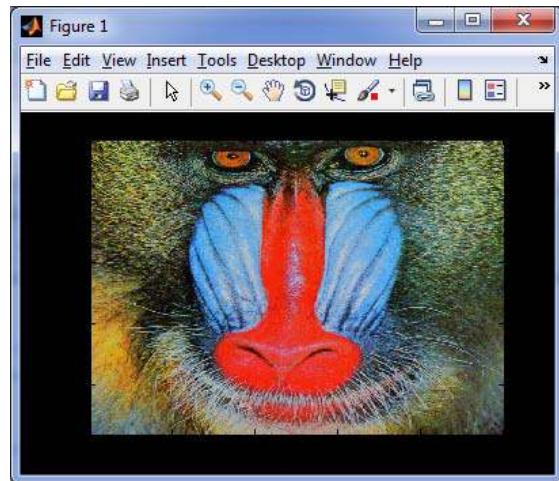
- MATLAB hazır örnek
 - load mandrill
 - X, caption ve map değişkenleri yüklenir
 - figure('color','k')
 - yeni figür aç arka plan siyah olsun
 - image(X)
 - Resmi görüntüle (standart map ile)
 - colormap(map)
 - Kendi map'i ile görüntüle
 - axis off % Remove axis ticks and numbers
 - axis image % Set aspect ratio to obtain square pixels

MATLAB İLE GÖRÜNTÜ İŞLEME

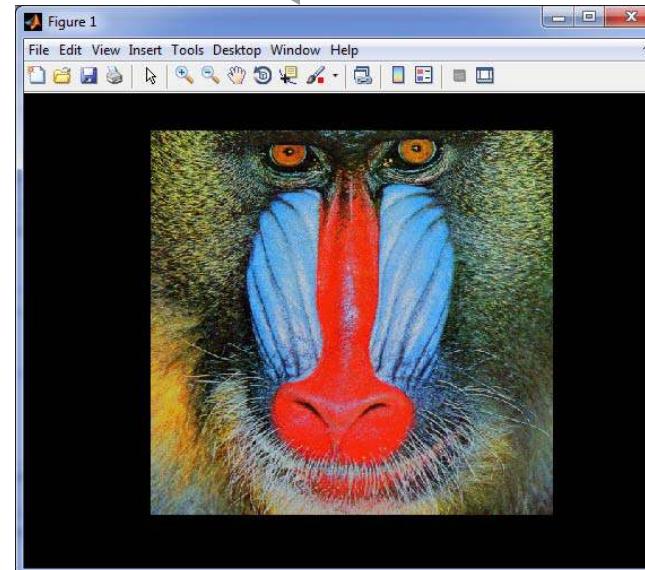


- `figure('color','k')`
 - yeni figür aç arka plan siyah olsun
- `image(X)`
 - Resmi görüntüle
 - standart map (jet) ile görüntülenir
 - colormap(jet) denilerek bu görüntü tekrar elde edilebilir

MATLAB İLE GÖRÜNTÜ İŞLEME



- `colormap(map)`
 - Kendi map'i ile görüntüle
- `axis off`
 - % Remove axis ticks and numbers
- `axis image`
 - % Set aspect ratio to obtain square pixels

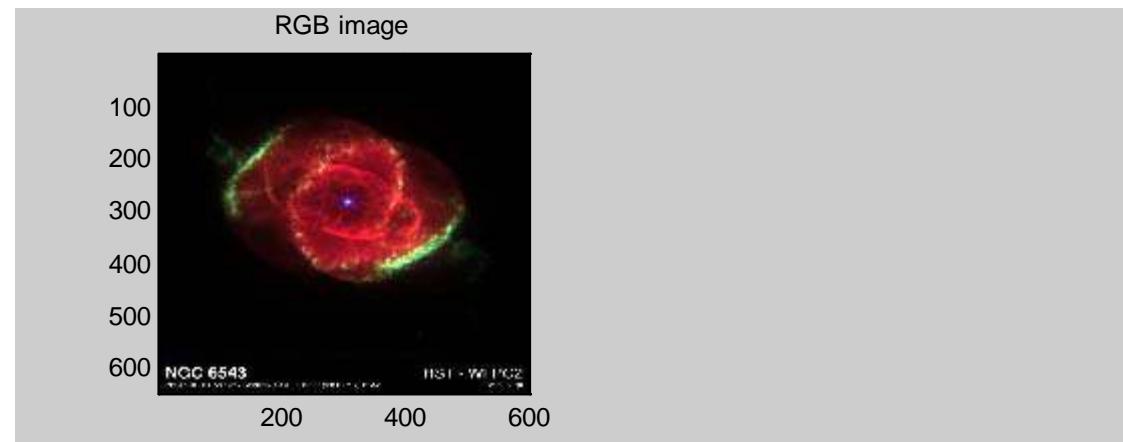


MATLAB İLE GÖRÜNTÜ İŞLEME

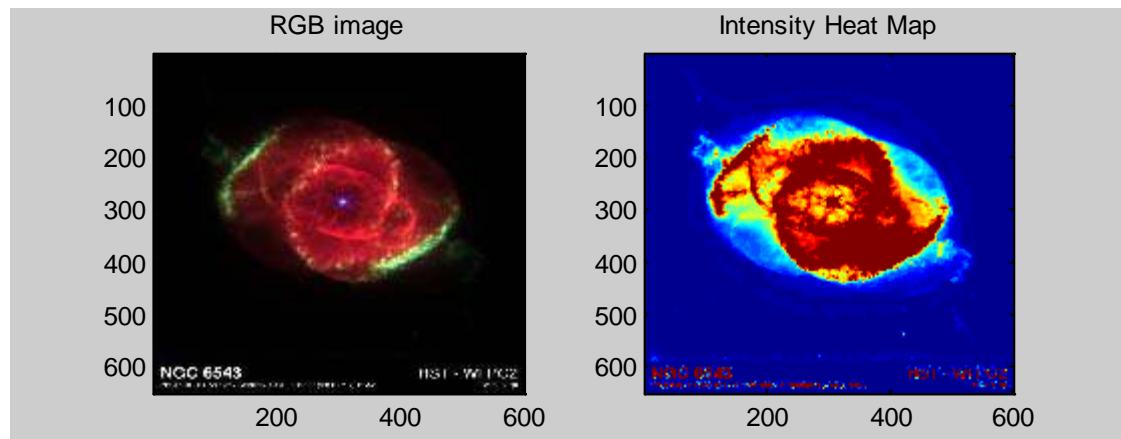
- Başka bir örnek
 - figure
 - ax(1) = subplot(1,2,1);
 - rgb = imread('ngc6543a.jpg');
 - image(rgb); title('RGB image')
 - ax(2) = subplot(1,2,2);
 - im = mean(rgb,3);
 - image(im); title('Intensity Heat Map')
 - colormap(hot(256))
 - linkaxes(ax,'xy')
 - axis(ax,'image')

MATLAB İLE GÖRÜNTÜ İŞLEME

```
figure  
ax(1) = subplot(1,2,1);  
rgb = imread('ngc6543a.jpg');  
image(rgb); title('RGB image')
```

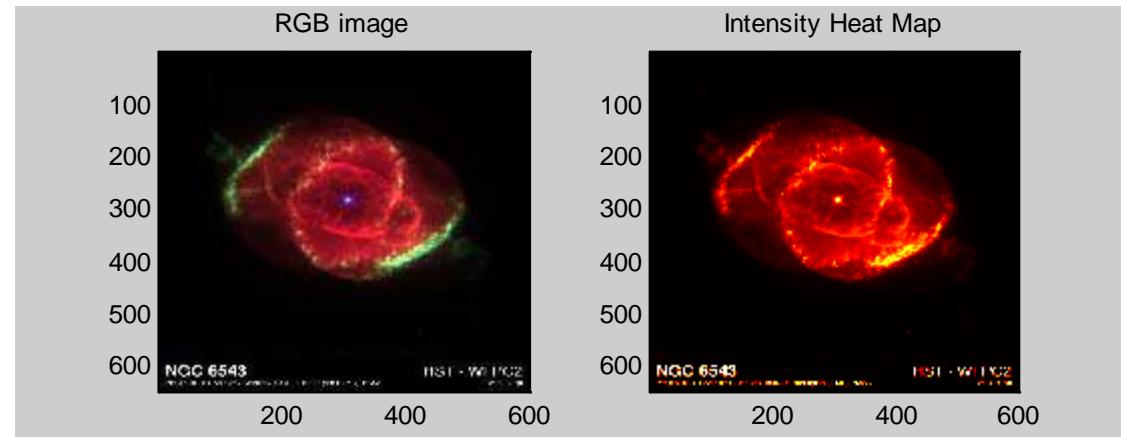


```
ax(2) = subplot(1,2,2);  
im = mean(rgb,3);  
image(im);  
title('Intensity Heat Map')
```

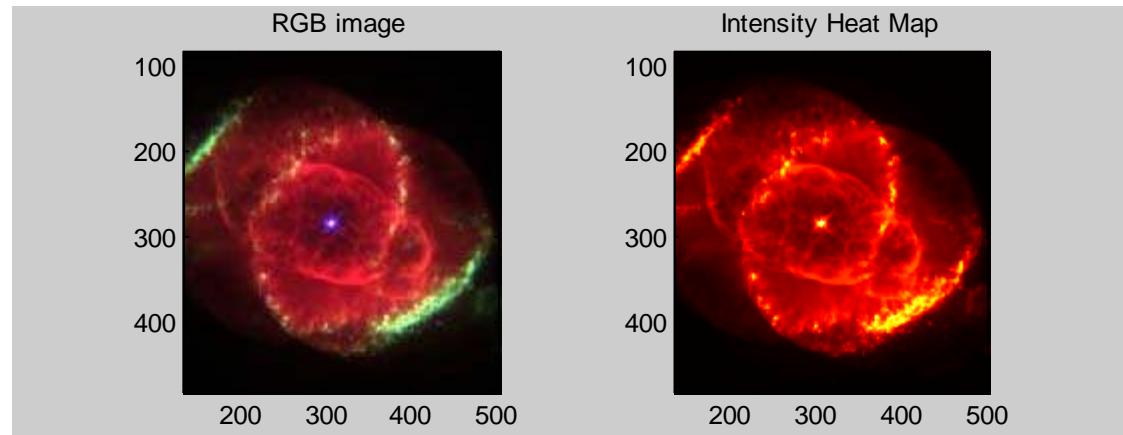


MATLAB İLE GÖRÜNTÜ İŞLEME

```
figure  
ax(1) = subplot(1,2,1);  
rgb = imread('ngc6543a.jpg');  
image(rgb); title('RGB image')  
ax(2) = subplot(1,2,2);  
im = mean(rgb,3);  
image(im);  
title('Intensity Heat Map')  
colormap(hot(256))
```



```
linkaxes(ax,'xy')  
% iki imgenin axislerini birbirine bağlıyor.  
% birinde zoom yapılınca  
% diğer resmin aynı bölgesini de büyütüyor  
axis(ax,'image')
```



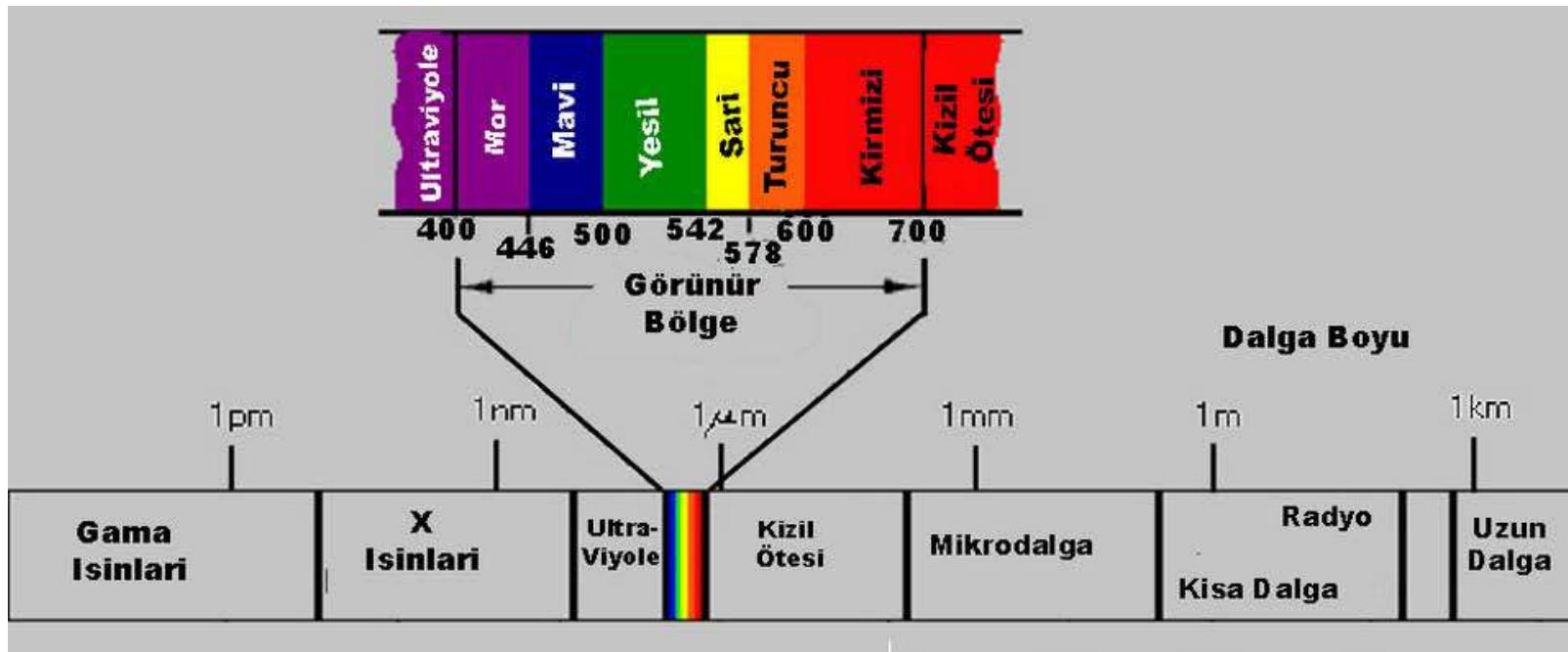
GÖRÜNTÜ İŞLEMEDE

TEMEL KAVRAMLAR

Görüntü İşleme - İnsan

- Görüntü işleme açısından ele alındığında insan algılama sistemi; görüntü yakalama, gruplama ve analiz konusunda bilinen en karmaşık sistemdir.
- İnsan görme sistemi gözlerimizle başlar.
- Işığın çok kanallı ve pankromatik dalga boyları her biri birer algılama sistemi olan gözlerimiz yardımcı ile algılanır.
- Görülebilen spektrum;
 - insan gözünün görebileceği elektro manyetik dalga boyu aralığını tanımlar.
 - Buna karşın bir arının görebildiği spektral aralık ultraviyole bölgede başlar ve yeşil dalga boylarında sona erer.

Görüntü İşleme



- insan gözünün görebileceği alana ait dalga boyları $0.4\mu\text{m}$ - $0.7\mu\text{m}$ arasındadır
- Buna karşın bir arının görebildiği spektral aralık ultraviyole bölgede başlar ve yeşil dalga boylarında sona erer

Görme Sistemi

- Gözlerimizle görülebilen alandaki elektro manyetik dalgaları algılayabiliriz ve beynimiz yardımımı ile yorumlanabilir görüntü haline dönüştürebiliriz.
- Gözün ana bileşenleri; Kornea, göz bebeği, mercek, retina ve optik sinirlerdir.
- Kornea gözün dış kısmında olup geçirgen, kubbe formunda olup, ışığa odaklama fonksiyonuna sahiptir.
- Göz bebeği kendisini tutan kaslar yardımımı ile ışık göze ulaştığında gözün açılıp kapanmasına yarar.
- Göz bebeği göz merceği örter. Kaslar yardımımı ile mercek göze giren ışığın şiddetine göre kalınlaşır veya incelir.

Görme Sistemi

- Gözlerin farklı kontrastlara adapte olabilme yeteneği ***parlaklık adaptasyonu*** (brightness adaption) olarak adlandırılır,
- İki parlaklık düzeyleri arasında ayırım yapabilme yeteneğine ise ***kontrast duyarlılığı*** adı verilir. Bu da gözün etrafını çevreleyen parlaklık düzeylerine bağlıdır. Güneşli bir günde farları yanmış bir aracın farlarını görmek güçtür, fakat gece değildir.

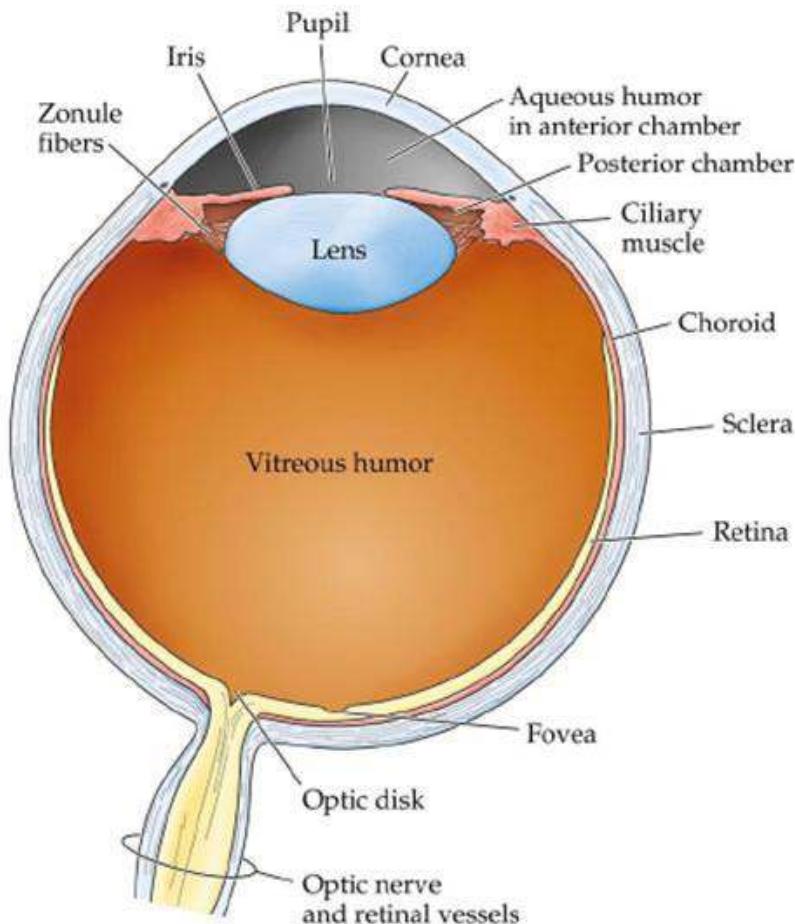
Görme Sistemi

- Özet olarak; sayısal görüntü işleme için görme sistemlerimizin altında yatan temel mekanizmaların bilinmesi oldukça önemlidir.
- Kısaca göz bir fotoğraf makinesi gibi düşünülebilir ve beynin görme bölümleri de karmaşık bir sayısal görüntü işleme sistemi olarak düşünülebilir.
- Görüntü işleme yaşam ver oldukça söz konusu olmuştur.
- İnsanlar ve hayvanlar gözleri ile analog temele dayanan görüntü işleme yapmaktadır.
- Bu olay beyin yardımımı ile (akıllı sistem) online, paralel ve çok spektrumlu (multispektral) olmaktadır

Görüntü İşleme

- Resimlerin bilgisayar ortamında değerlendirilebilmeleri için veri formatlarının bilgisayar ortamına uygun hale getirilmeleri gerekmektedir.
- Bu dönüşümü sayısallaştırma (digitizing) adı verilir.
- Bir resmin fotografik sunumunu daha doğrusu sayısal forma dönüştürülmesi çeşitli şekillerde olanaklıdır.
- Buna farklı teknikler kullanılarak resmin sayısallaştırıldığı tarayıcılar örnek olarak verilebilir.
- Ya da Analog/Sayısal dönüşümün kullanılarak resmin sayısal hale dönüştürüldüğü sistemler (FrameGrabber), uzaktan algılamada uçak ya da uydulara yerleştirilen çok kanallı tarayıcılar yine örnek olarak verilebilir

GÖRÜNTÜ ALGILAMA

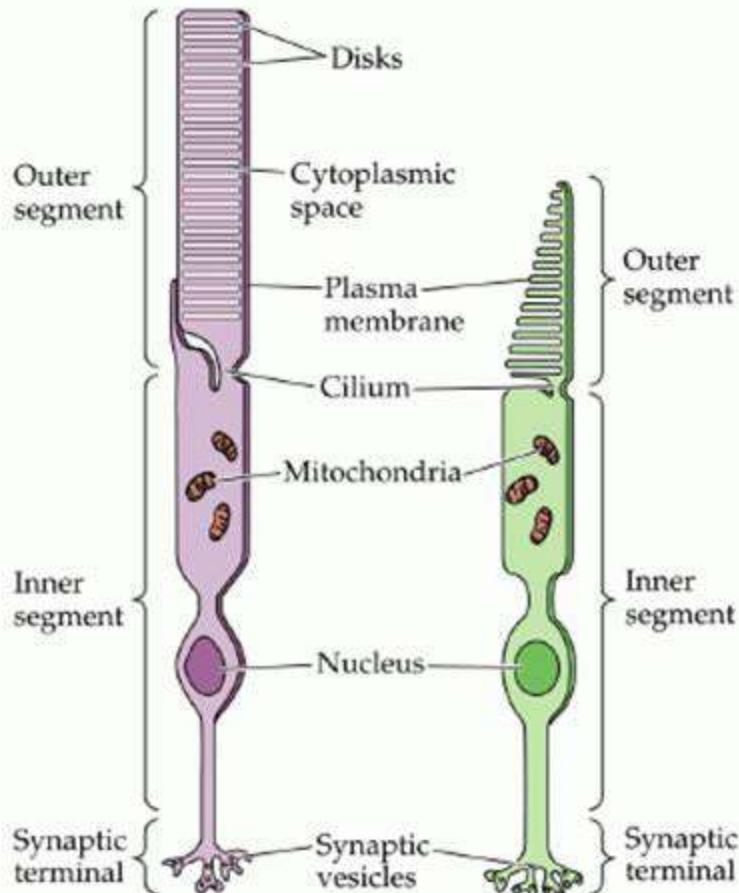


- Üç temel zar ile kaplıdır.
 - 1.Dış Zar (kornea ve Sklera)
 - 2.Koroid
 - 3.Retina

GÖRÜNTÜ ALGILAMA

- Dış Zar
 - İki kısımdan oluşur. Kornea ve Sklera
 - Kornea sert ve saydam bir yapıya sahiptir.
 - Sklera saydam olmayan bir yapıdadır ve optik bölümünü kuşatır.
- Koroid
 - Koroid sklera boyunca uzanan bir zardır.
 - Bu zarda gözü besleyen kan damarları yoğun şekilde bulunur.
- Retina
 - Gözün en önemli tabakası retinadır.
 - Nesnenin görüntüsü bu tabakaya düşer.
 - Nesneler retinanın yüzeyindeki algılayıcılar ile algılanır.
 - Cone(Koni) ve Rod(Çubuk)

GÖRÜNTÜ ALGILAMA



- Cone ve Rod

- Cone

- 6-7 milyon/göz
 - Renklere karşı çok hassas
 - Detaylar bu algılayıcılar ile algılanır
 - Cone algısı *photopic* ya da *parlak ışık görmesi* olarak adlandırılır

- Rod

- 75 - 150 milyon/göz
 - Işığa karşı çok hassas
 - Görüntü ile ilgili genel özellikler bu algılayıcılar ile algılanır
 - *Scotopik* ya da *düşük ışık görmesi* olarak adlandırılır.
 - En çok 498 nm (green-blue) dalga boylarına hassastır. 640 nm (red)'den yüksek dalgalara karşı hassas değildir.

Gözde İmge Oluşumu

- Gözde görüntü oluşturulurken görüntü uzaklığı lens ve retina ile ayarlanır. Görüntü oluştururken uygun odaklanma uzaklığı ayarlanması için lensin şekli değişir.
- Lens esnektir ve ışığı kırma oranı kalınlığı ile kontrol edilir. Kalınlık ise kasların gerilmesi ile gerçekleşir.
 - Göz ile optik lens arasındaki temel fark gözdeki lenslerin daha esnek olduğunu.
- Uzaktaki cismi görmek için yassılaşır ve kırma minimumdur.
- Yakındaki cismi görmek için ise kırma maksimumdur.

GÖRÜNTÜ ALGILAMA

Source emits photons



Light

Photons travel in a straight line



And then some reach the eye/camera.



When they hit an object they:

- bounce off in a new direction
- or are absorbed

Gözde İmge Oluşumu

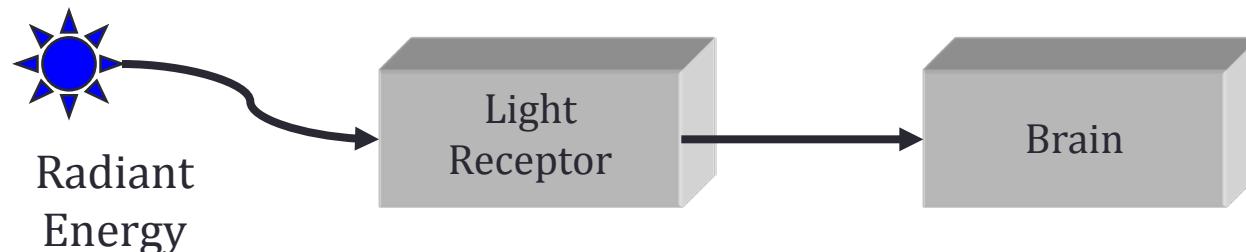
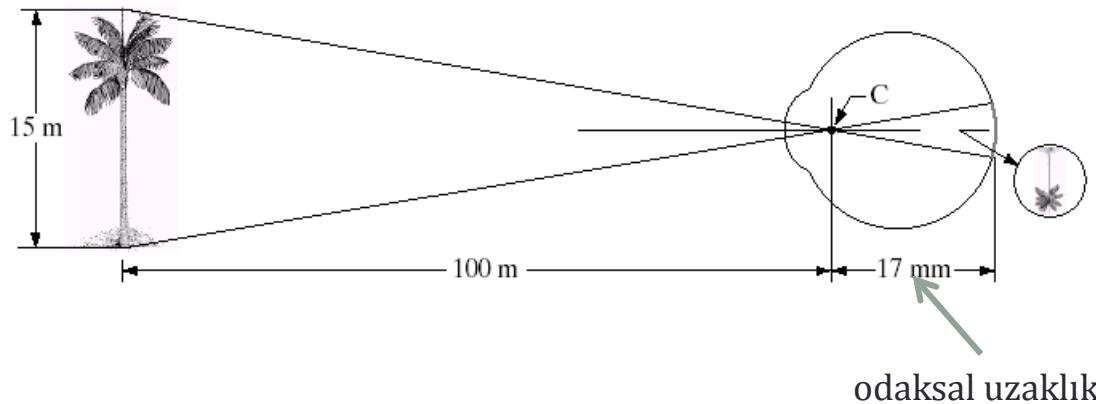


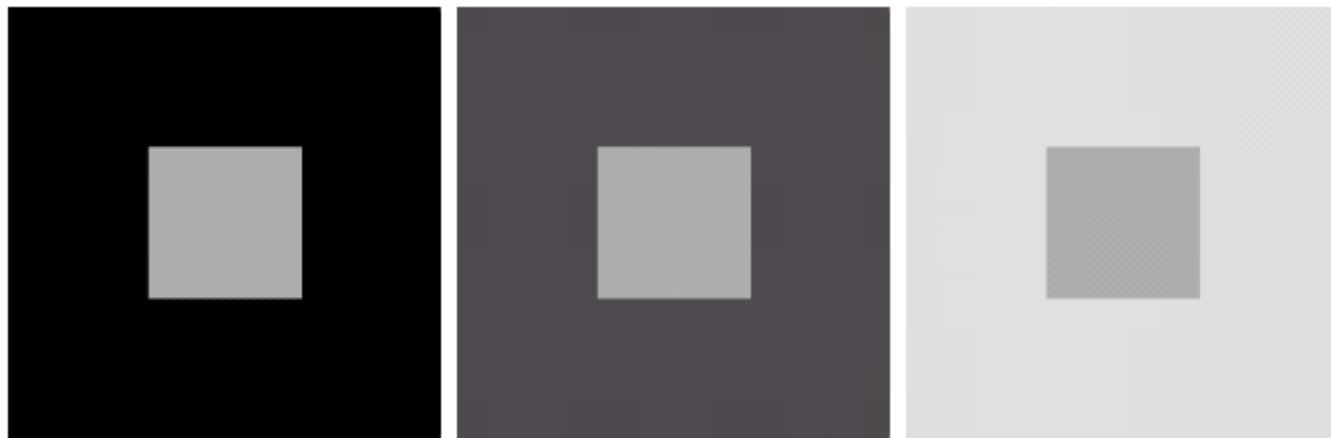
FIGURE 2.3
Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.



Retinada oluşan imgenin boy hesaplamasında:

$$\frac{\text{İmgenin retinadaki boyu}}{\text{Odaksal uzaklık}} = \frac{\text{İmgenin boyu}}{\text{İmgeye uzaklık}}$$
$$\frac{h}{17} = \frac{15}{100} \rightarrow h = 2.55\text{mm}$$

Göz Yanılgıları



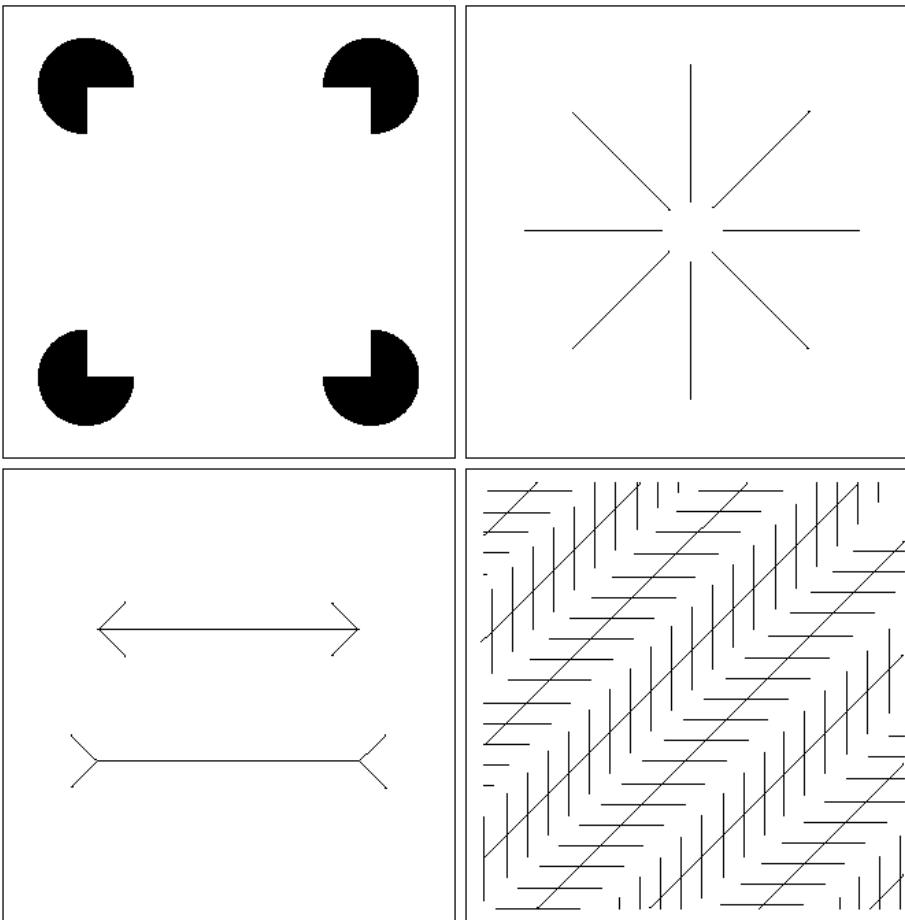
a b c

FIGURE 2.8 Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

Göz Yanılgıları

a
b
c
d

FIGURE 2.9 Some well-known optical illusions.



- a) Ortada kare olmamasına rağmen çok net bir şekilde kare görüntüsü alınıyor
- b) Aynı olay çember görüntüsü için oluşuyor. Çok az sayıda çizgi ile çember illüzyonu yaratılabildiğine dikkat edin
- c) Çizgi uzunluklar eşit olmasına rağmen aşağıdakiler daha uzun görünüyor.
- d) Tüm çizgiler 45 derecelik açıyla paralel olmasına rağmen üzerlerindeki çizgiler bunu görmemizi engelliyor

Henüz tam olarak anlaşılamamış olan optik illüzyonlar, insan göz sisteminin karakteristik özelliklerindendir.

IŞIK VE ELEKTROMAGNETİK SPEKTRUM

- Elektromanyetik dalgalar değişen dalga boylardaki sinüzoidal dalgaların ışık hızında yayılımı olarak veya dalga şekli biçiminde ışık hızında ilerleyen kütlesiz parçacıkların akışı olarak düşünülebilir.
- Her kütlesiz parçacık belli bir miktarda enerji içerir.
 - Her bir enerji yiğinına foton denir.
- Işığın çok kanallı ve pankromatik dalga boyları her biri birer algılama sistemi olan gözlerimiz yardımı ile algılanır.
- Görülebilen spektrum;
 - insan gözünün görebileceği elektro manyetik dalga boyu aralığını tanımlar.
 - Buna karşın bir arının görebildiği spektral aralık ultraviyole bölgede başlar ve yeşil dalga boylarında sona erer.

IŞIK VE ELEKTROMAGNETİK SPEKTRUM

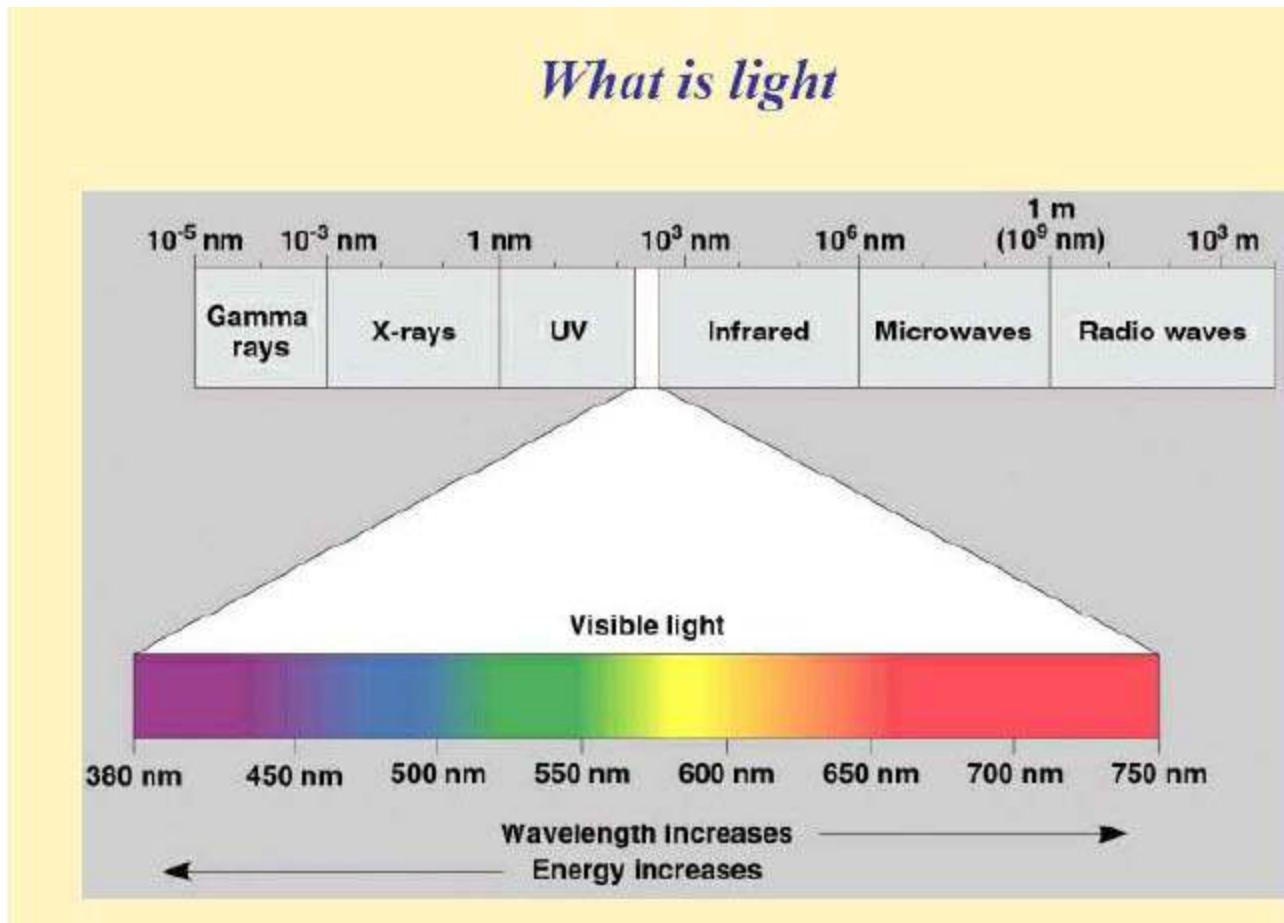
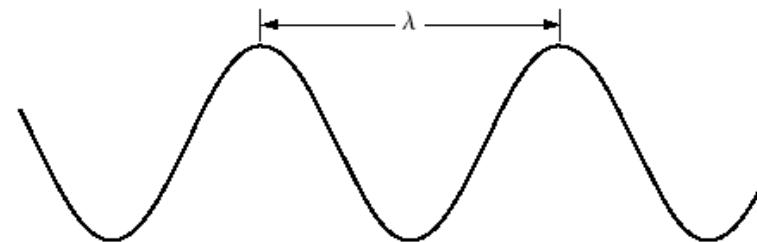


FIGURE 2.11
Graphical representation of one wavelength.



$$\lambda = c/v$$

λ = Dalga Boyu
 c = Işık Hızı (2.998×10^8 m/sn)
 V = frekans (1/sn)(Hz)

$$c = \frac{1}{\sqrt{\mu_0 \epsilon_0}}$$
$$\mu_0 = 4\pi \times 10^{-7} N/A^2$$
 boşluğun manyetik geçirgenliği
$$\epsilon_0 = 8,854187817 \times 10^{-12} C^2/N \cdot m^2$$
 boşluğun elektrik geçirgenliği

Elektromanyetik Spektrumun parçalarının enerjisi aşağıdaki şekilde hesaplanır:

$$E = hv$$

h = Plank Sabiti
 E = Enerji
 V = frekans

İşık ve Renk

- İnsan gözünün bir nesnede algıladığı renkler nesneden *yansıyan* ışık tarafından belirlenir
- İnsan gözünün bir nesnede algıladığı renkler nesneden *yansıyan* ışık tarafından belirlenir
- Tüm görünür dalga boyalarında ışıkların dengeli şekilde birleştiği ve üzerine gelen tüm ışığı yansıtan bir nesne gözlemciye beyaz olarak görünecektir.
- Ancak gelen ışığı belli dalga boyları arasında yansıtıyorsa nesne renkli görünecektir.
 - Örneğin, yeşil bir nesne 500 to 570 nm aralığındaki dalga boyununa sahip ışığı yansıtırken, diğer dalga boylarındaki enerjileri yutar

Işık ve Renk

- Renk bilgisine sahip olmayan ışık “Monochromatic” ışık olarak isimlendirilir.
 - Bu tür ışığın tek niteliği yoğunluk (intensity) veya miktarıdır (amount).
 - Monochromatic ışığın yoğunluğu siyahdan beyaza doğru farklı grilik seviyelerinde olduğundan Gri-seviyeli terimi (Gray-Scale) kullanılır.
- Chromatic ışıklar ise monochromatic ışığın tersine renk bilgisini de barındırırlar.
 - 3 temel ölçüt bu tarz ışık kaynağını tanımlamak için kullanılır.
 - **Radiance**
 - **Luminance**
 - **Brightness**

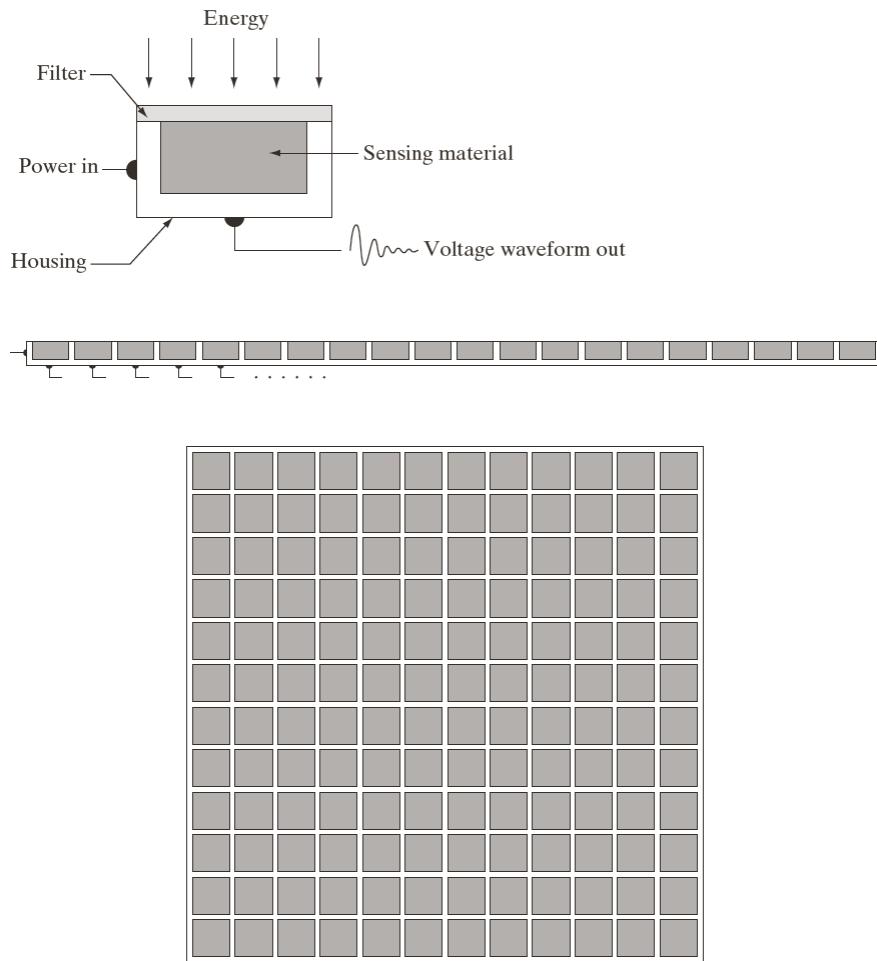
Işık

- **Radiance** : Işık kaynağından yayılan enerjinin toplam miktarıdır. Genellikle watt (W) ile gösterilir.
- **Luminance**: Lümen birimi ile ifade edilir. Gözlemcinin, ışık kaynağından gelen enerjinin algıladığı miktarı belirtir.
- **Brightness**: Bu değer ise gözlemciden gözlemciye değişebilen ve ölçümlenmesi zor olan bir özelliktir.

SAYISAL GÖRÜNTÜ İŞLEME

Hafta2: Görüntü Yakalama(Image Acquisition)

Sensörler



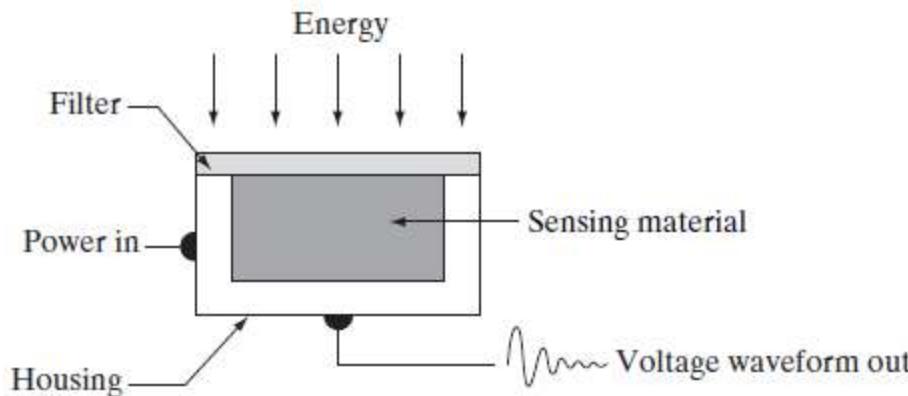
a
b
c

FIGURE 2.12
(a) Single imaging
sensor.
(b) Line sensor.
(c) Array sensor.

- Gelen enerjiyi sayısal imgeye dönüştürmek için kullanılan 3 temel sensör düzenlemesi:
- Düşünce basittir:

- Gelen enerji elektrik güç girişi ve incelenenek belli enerji tipine duyarlı sensör materyali tarafından voltaja dönüştürülür
- Dönen voltaj, sensör(ler)in gelen enerjiye cevap olarak verdikleri dalga şeklidir ve sayısal sonuç bu dalga(lar)nın sayısallaştırılmasıyla elde edilir.

Tek Sensör İle Görüntü Elde Etme



Tek bir sensörün parçaları

- Bu türde ait en çok bilinen sensör *fotodiyottur*.
 - *Fotodiyot* silikondan yapılmıştır ve çıkış voltajı dalga şeklinde ışık ile doğru orantılıdır
- Sensör öncesinde bir filtre kullanmak seçiciliği artıracaktır
 - Örneğin, yeşil geçen bir filtre renk spektrumunun yeşil bandını öne çıkaracaktır
 - Bunun sonucunda sensör çıkışı yeşil ışık için spektrumun görülebilen diğer tüm parçalarından daha güçlü olacaktır.

Tek Sensör İle Görüntü Elde Etme

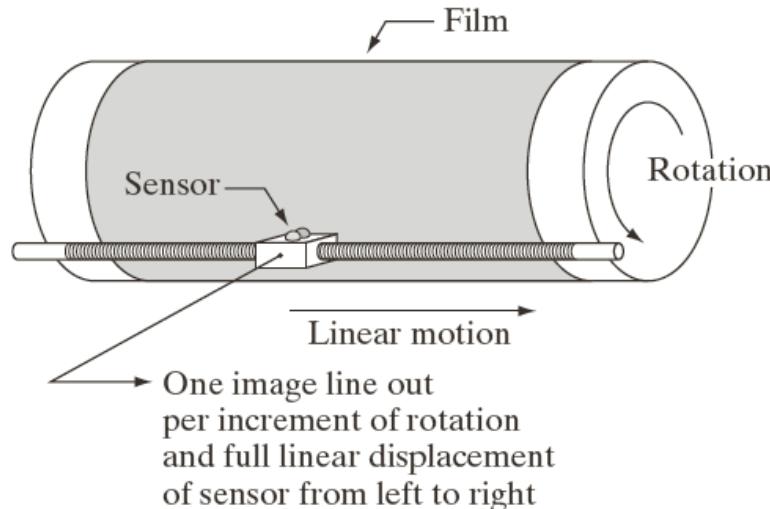
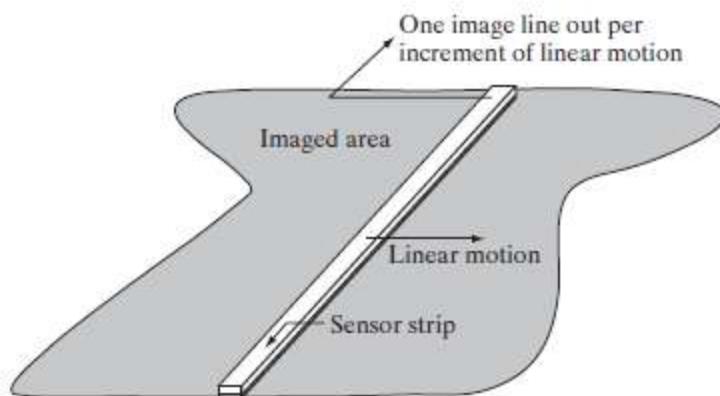


FIGURE 2.13. Combining a single sensor with motion to generate a 2-D image

- Yandaki figür yüksek hassasiyetli tarama için kullanılan bir düzeneği göstermektedir
- 2 boyutlu bir imgé oluşturmak için sensör ve görüntülenecek alanın x ve y boyutları arasında bağıl farklılıklar olması gerekmektedir
 - Yukarıdaki düzenekte:
 - Negatif film mekanik rotasyonu bir boyutta farklılığı (displacement) sağlayan bir sütun üzerine yerleştirilir
 - Tek sensör dikey düzlemede hareket sağlayan Kurşun vidası üzerine takılmıştır
 - Bu düzenek, mekanik hareket yüksek hassasiyetle kontrol edilebildiği için yüksek çözünürlüklü imgeler elde etmenin ucuz (ama yavaş) bir yoludur

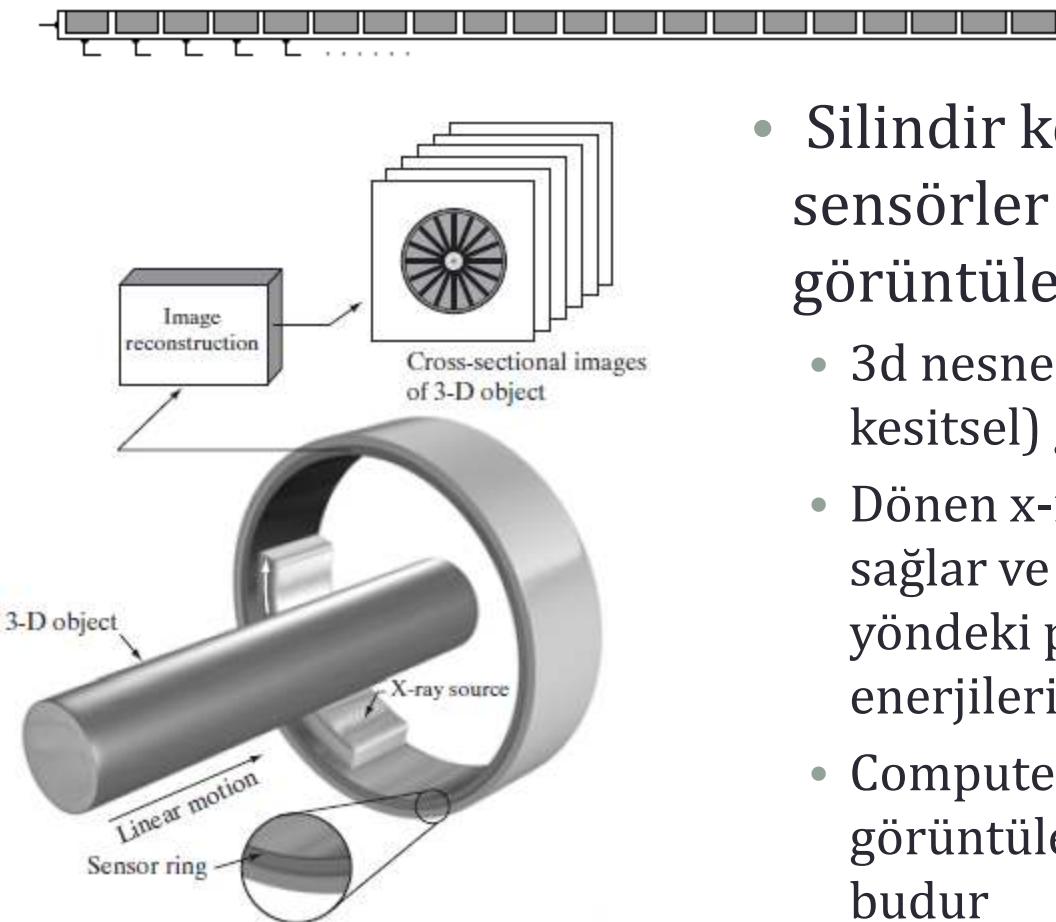
Lineer Sensör Şeritleri İle Görüntü Elde Etme



**Image acquisition using
a linear sensor strip.**

- Sensör striplerden oluşan oluşan başka bir geometri tek sensörlerden çok daha yaygın olarak kullanılmaktadır.
- Stripler bir yönde görüntüleme sağlar
- Strip'e dik yönde hareketlilik ise diğer yönde görüntüleme sağlar
- 4000 ya da daha fazla sıralı sensörler vardır
- Sıralı sensörler havadan görüntüleme uygulamalarında rutin olarak kullanılmaktadır.
 - Görüntüleme sistemi, görüntülenecek coğrafik alan üzerinde sabit bir yükseklikte ve hızla uçan bir uçağa takılır

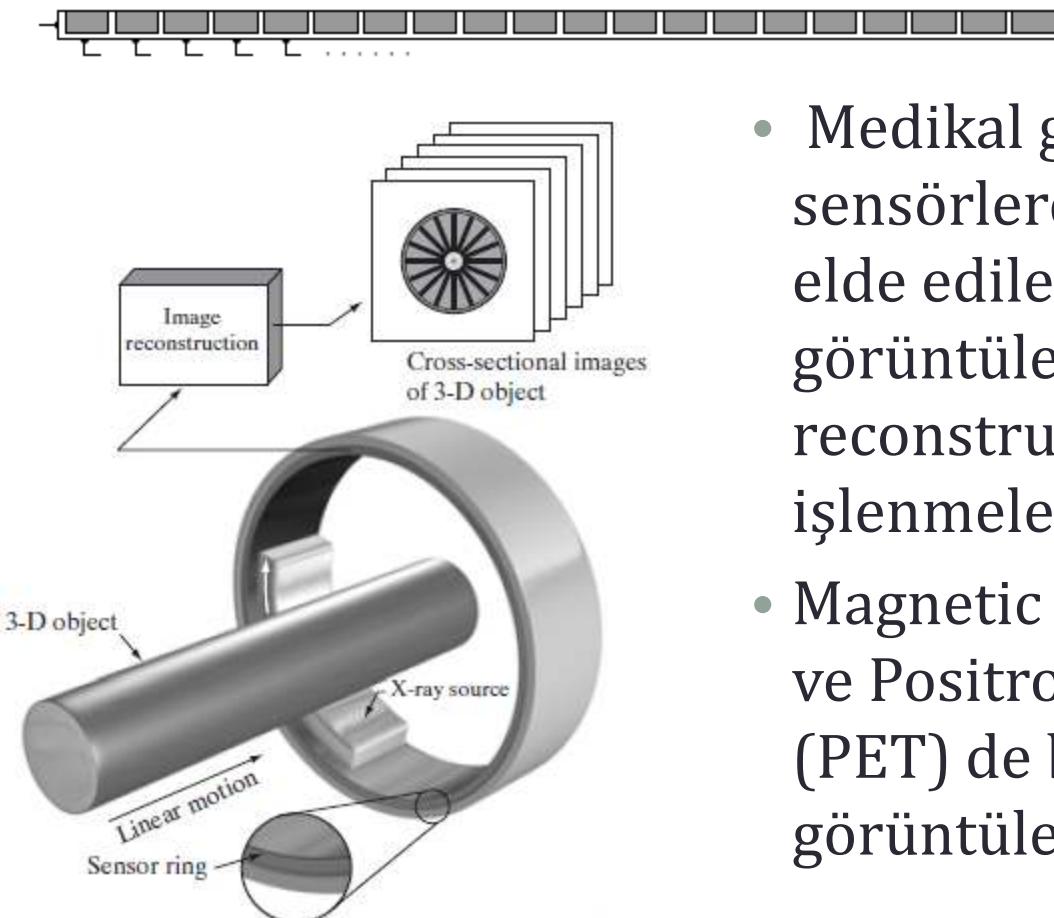
Dairesel Sensör Şeritler İle Görüntü Elde Etme



**Image acquisition using
a circular sensor strip.**

- Silindir konfigürasyon içine takılı sensörler medikal ve endüstriyel görüntülemede kullanılır.
 - 3d nesnelerin cross-sectional (slice - kesitsel) görüntülerini almaya yarar
 - Dönen x-ray kaynağı illumunitaion sağlar ve sensörlerin kaynağı ters yöndeeki parçaları nesneden geçen x-ray enerjilerini toplar
- Computerized axial tomography (CAT) görüntüleme tekniğinin temel prensibi budur

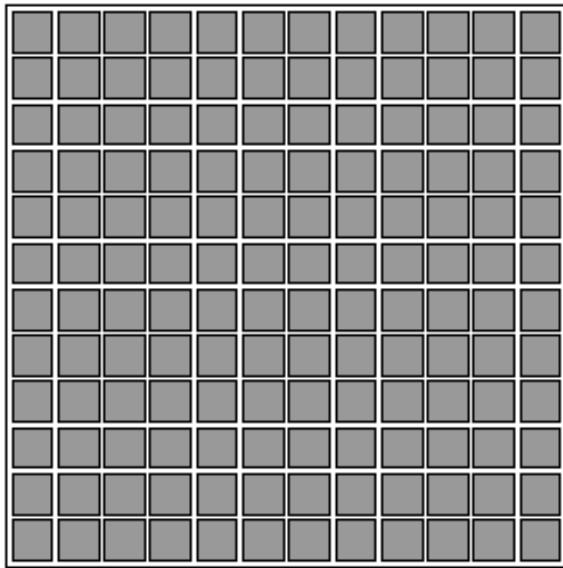
Dairesel Sensör Şeritler İle Görüntü Elde Etme



**Image acquisition using
a circular sensor strip.**

- Medikal görüntüleme tekniklerinde sensörlerden alınan çıktıların amacı elde edilen verilerin anlamlı kesitsel görüntülere çevirmek olan reconstruction algoritmaları ile işlenmeleri gereklidir
- Magnetic Resonance Imaging (MRI) ve Positron Emission Tomography (PET) de bu prensipler ile görüntüleme yapar

Sensör Dizileri ile Görüntü Elde Etme



- Sensör dizileri tek sensörlerin 2D şekilde dizilmesi ile elde edilir
- Dijital fotoğraf makinelерinde en çok bu düzenleme kullanılır.
- Bu kameralar için en çok CCD dizileri kullanılır.
 - 4000×4000 ya da daha fazla eleman ile üretilebilir
- Her sensörün çıkışı yüzeyine gelen ışık enerjisi ile orantılıdır
- Sensör düzenlemesi 2D olduğu için tam görüntü sensörlerin hareket etmesine gereksinim duyulmaksızın elde edilebilir.

Sensör Dizileri ile Görüntü Elde Etme

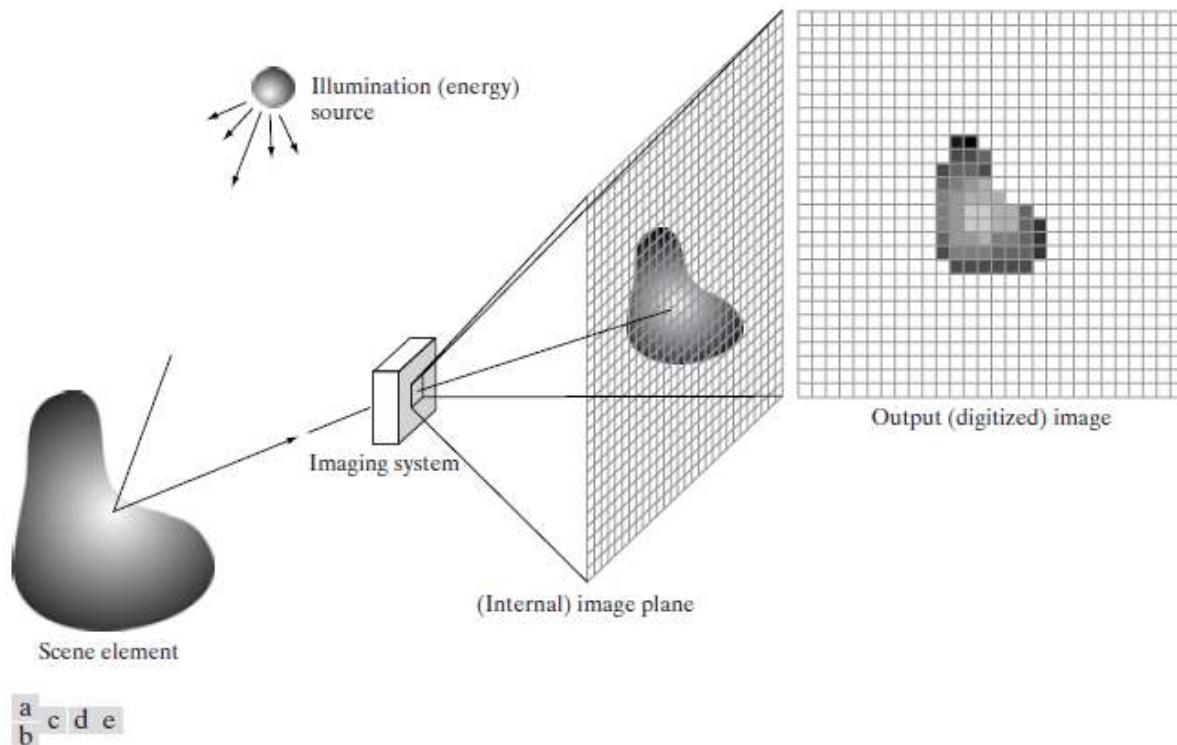


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

- Sensör dizilerinin kullanma düzenekleri yanda gösterilmektedir
- Figürde illuminasyon kaynağı enerjisinin (a) bir nesneden yansımaları gösteriliyor (b)
 - Dikkat: Enerji nesnenin içinden de iletilebilir.

- Görüntüleme sistemi tarafından gerçekleştirilen ilk fonksiyon gelen enerjiyi toplamak ve bir görüntü düzlemine odaklamaktır (c).

Sensör Dizileri ile Görüntü Elde Etme

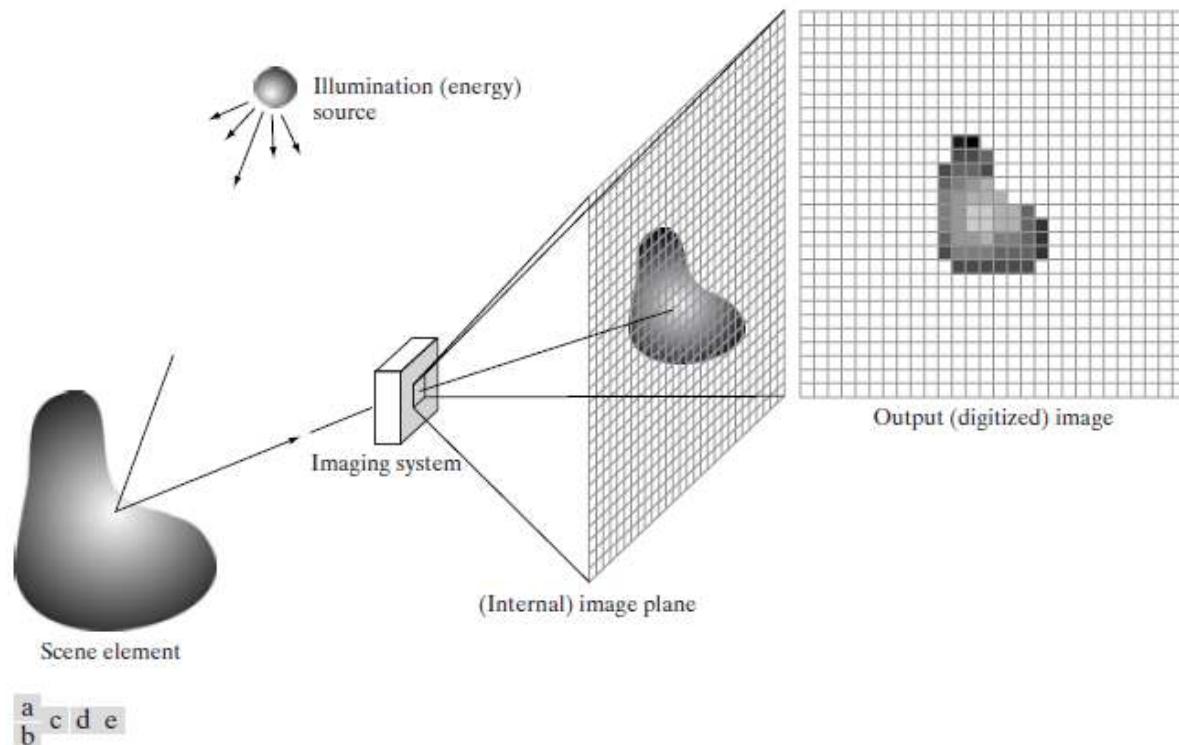


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy (“illumination”) source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

- Görüntüleme sistemi görüntüyü odak düzleme yansıtır (d)
- Odak düzlemi ile çakışan sensör dizisi, her sensör tarafından alınan ışığın toplamına orantılı olacak şekilde bir çıkış üretir.
- Kullanılan analog ve dijital devreler yardımıyla bu çıkışlar görüntü sinyaline dönüştürülür

- Görüntüleme sisteminin başka bir bölümü tarafından bu görüntüler sayısal forma dönüştürülür.
- Sonuç olarak dijital bir görüntü elde edilir (e).

Basit Bir Görüntü Oluşumu Modeli

- İmge 2 boyutlu bir fonksiyondur: $f(x,y)$
 - F' nin x ve y uzaysal koordinatlarındaki genliği ya da değeri,fiziksel anlamı görüntünün kaynağı tarafından belirlenen *pozitif* bir değerdir
 - Bir işlem sonrasında elde ettiğimizi söylediğimiz imgenin değerleri fiziksel bir kaynak tarafından yayılan enerji ile orantılıdır.
 - Sonuç olarak $f(x,y)$ 0'dan farklı sonlu bir değer olmalıdır.

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

- $F(x,y)$ 2 komponent ile karakterize edilebilir:
 - Görüntülenen sahnedeki aydınlatma kaynağı miktarı
 - *Illumination – Aydınlatma Bileşeni* olarak bilinir ve $i(x,y)$ ile gösterilir
 - Sahnedeki nesneler tarafından yansıtılan ışık miktarı
 - *Reflectance – Yansıma Bileşeni* olarak bilinir ve $r(x,y)$ ile gösterilir

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

$$0 < i(x, y) < \infty \text{ ve } 0 < r(x, y) < 1$$

Tamamen absorbe
durumu

Tamamen yansıtma
durumu

Tipik Örnek Değerler

- $i(x,y)$
 - Güneş:
 - Havanın açık olduğu bir günde $90,000 \text{ lm/m}^2$ aydınlichkeit sağlarken
 - Bulutlu bir günde bu değer $10,000 \text{ lm/m}^2$ 'ye düşebilir
 - Ay:
 - Açık bir havada dolunay $0,1 \text{ lm/m}^2$ aydınlichkeit sağlar
 - Standart bir ofis yaklaşık 1000 lm/m^2
- $r(x,y)$
 - Siyah kadife: 0,01
 - Paslanmaz çelik : 0,65
 - Düz beyaz duvar: 0,80
 - gümüş kaplama metal: 0,90
 - Kar: 0,93

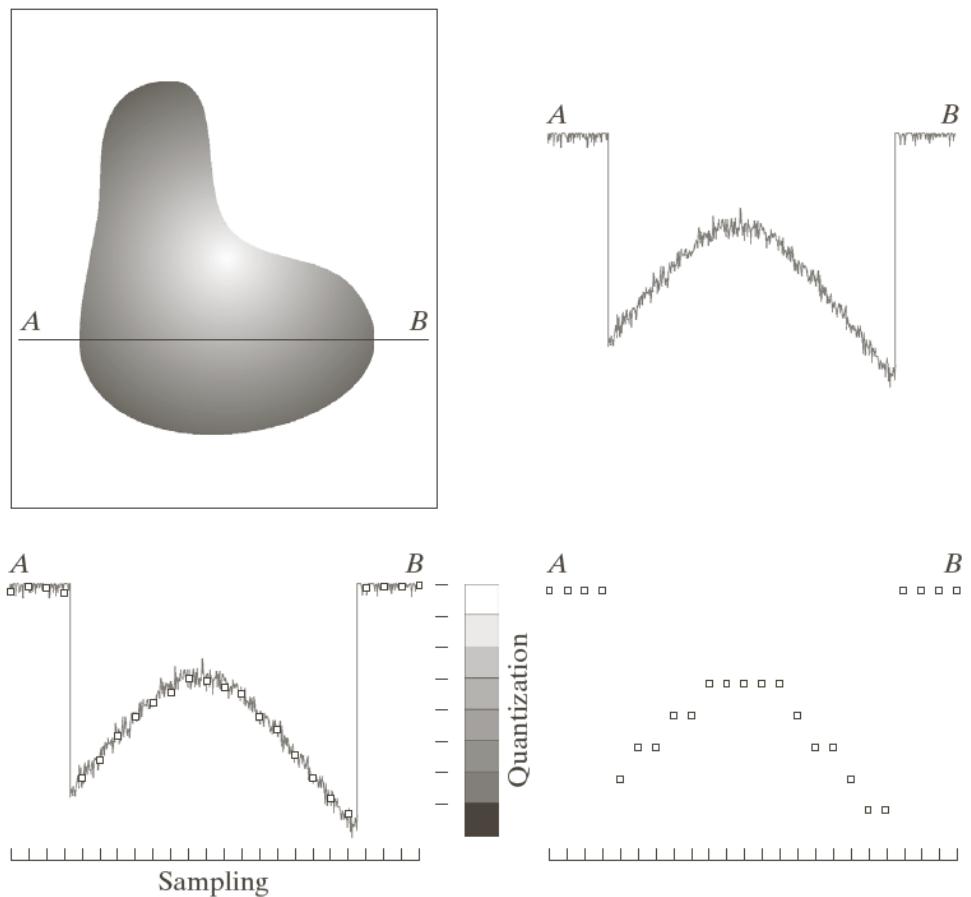
Monokrom İmge

- Monokrom imgenin bir noktadaki yoğunluğu:
 - $I=f(x_0, y_0)$
 - $L_{min} < I < L_{max}$
 - $[L_{min}, L_{max}]$ aralığı gray scale olarak bilinir
 - Genel uygulama bu aralığın $[0, L-1]$ 'e çekilmesidir
 - 0 : Siyah
 - L-1: Beyaz
 - Aradaki değerler grinin tonları

Örnekleme & Kuantalama

- Görüntü elde etmenin pek çok yolu olduğunu gördük.
- Asıl amaç elde edilen verilerden görüntü elde etmektir.
- Çoğu sensörün çıkışı, genliği ve uzaysal davranışını görüntülenen olguya bağlı değişen sürekli voltaj dalgasıdır.
- Sayısal görüntü elde etmek için bu sürekli veriyi sayısal forma çevirmek gereklidir.
- İki işlem gereklidir:
 - Örnekleme (Sampling)
 - Kuantalama (Quantization)

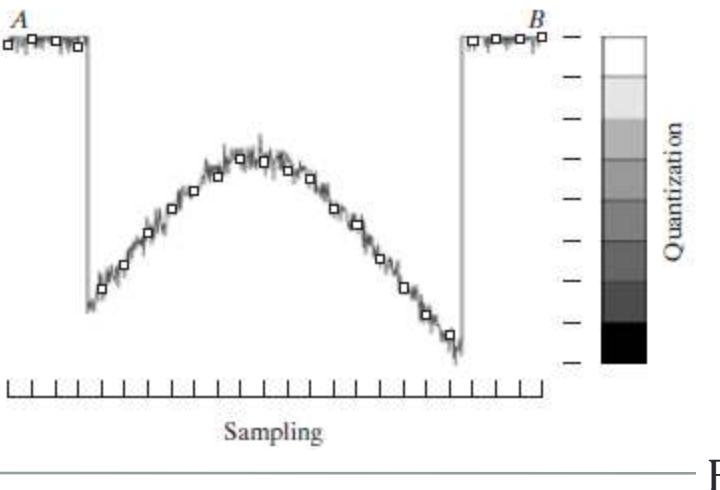
Örnekleme & Kuantalama



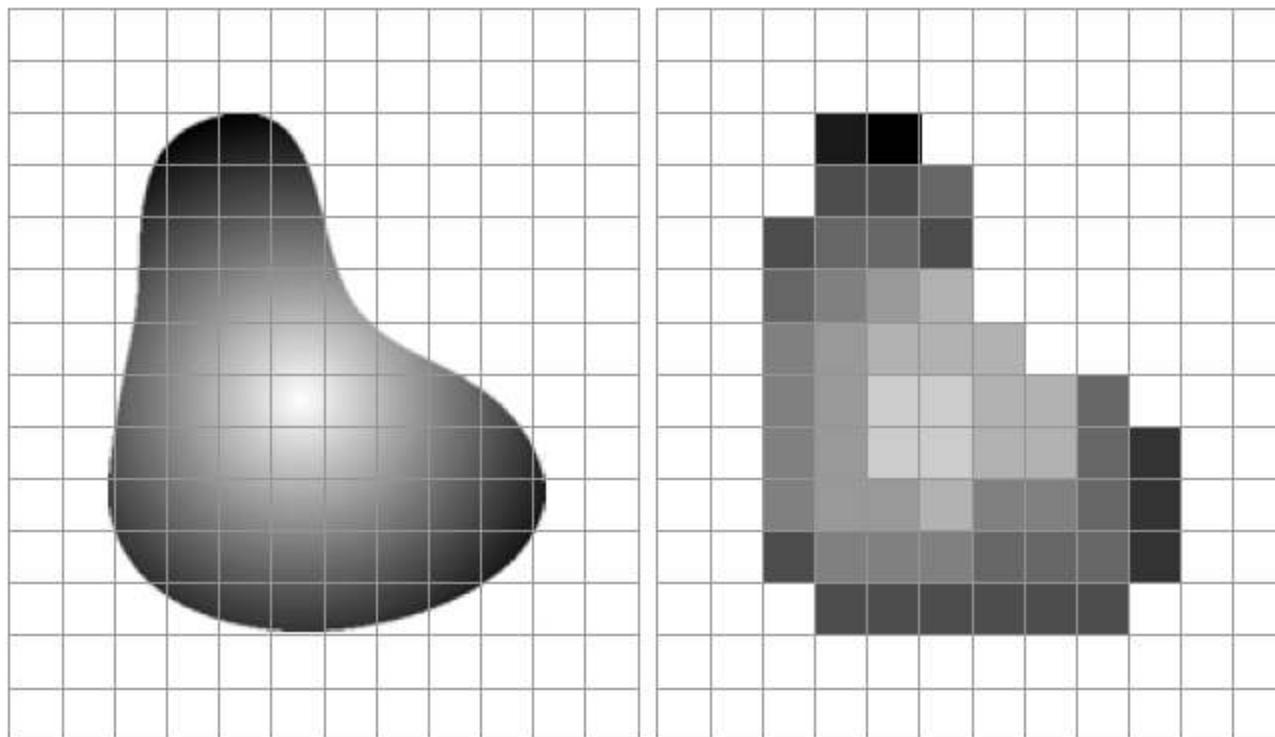
a
b
c
d

FIGURE 2.16
Generating a digital image.
(a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization.
(c) Sampling and quantization.
(d) Digital scan line.

Örnekleme & Kuantalama

- Elde edilen imge $f(x,y)$ hem x,y koordinatlarında hem de genlikte sürekli olabilir
 - Dijital imge için ikisini de ayrik yapmak gereklidir
 - Örnekleme: Koordinat değerlerini sayısallaştırma
 - Kuantalama: $f(x,y)$ genlik değerlerini sayısallaştırma
- 
- Örnekleme: AB hattı boyunca **eşit aralıklı** örnekler alınmasıdır
 - Örneklenmiş noktalardaki genlik değerleri sürekli olabilir.
 - Sayısal imge için bu değerler de ayrik (discrete) değerlere dönüştürülmelidir. (Kuantalama)
 - Sağ taraftaki şekil genlik değerlerinin 8 parçaya ayrılmasını gösteriyor

Dijital Görüntü



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Dijital İmgenin Gösterilişi

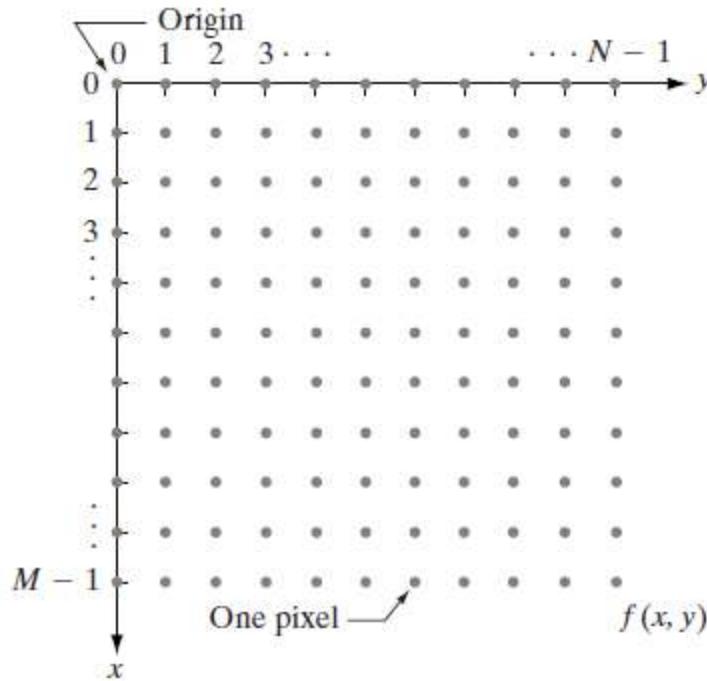


FIGURE 2.18
Coordinate convention used in this book to represent digital images.

- Elde edilen dijital imgenin M satır ve N sütunu olduğunu varsayalım
 - Örneklemeye sayesinde x ve y değerleri ayrık değerler.
 - Uygulama kolaylığı açısından bu değerler tamsayı ile ifade edilir

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

↓

al Görüntü Görüntü Elemanı (pixel)

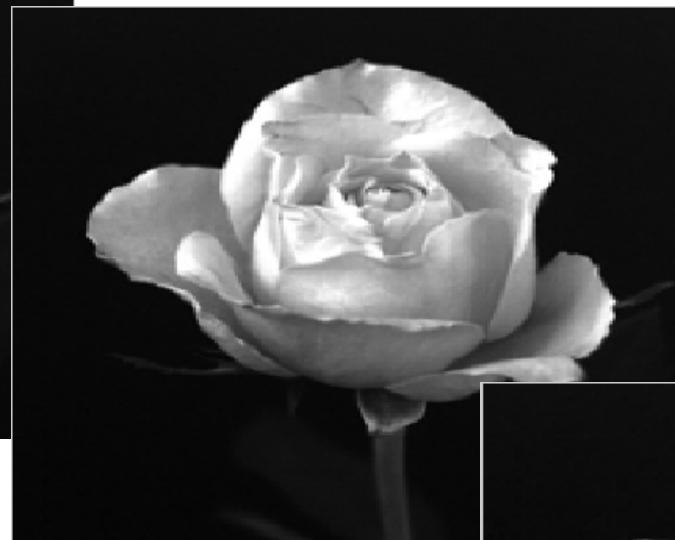
Uzaysal Çözünürlük

- Sayısal imgede örnekleme yaparak imgenin çözünürlüğü değiştirilebilir.
- Çözünürlük imgenin pixel sayısı / birim fiziksel boyut ifadesi ile bulunur
- Birimi ppi (pixels per inch) ya da dpi (dots per inch)
- Bir nesnenin bir bölümü ne kadar çok pixel ile ifade edilirse uzamsal çözünürlük o kadar yüksek olur
 - Ancak saklamak için gereken yerin de daha fazla olacağını unutmayalım.

Alt Örnekleme



512x512 imge



Alt örneklenmiş imge:
256x256



Alt örneklenmiş imge:
128x128

Alt Örnekleme

Orjinal Resim



```
gul=imread('gul.jpg');
imshow(gul)
title('Orjinal Resim')
```

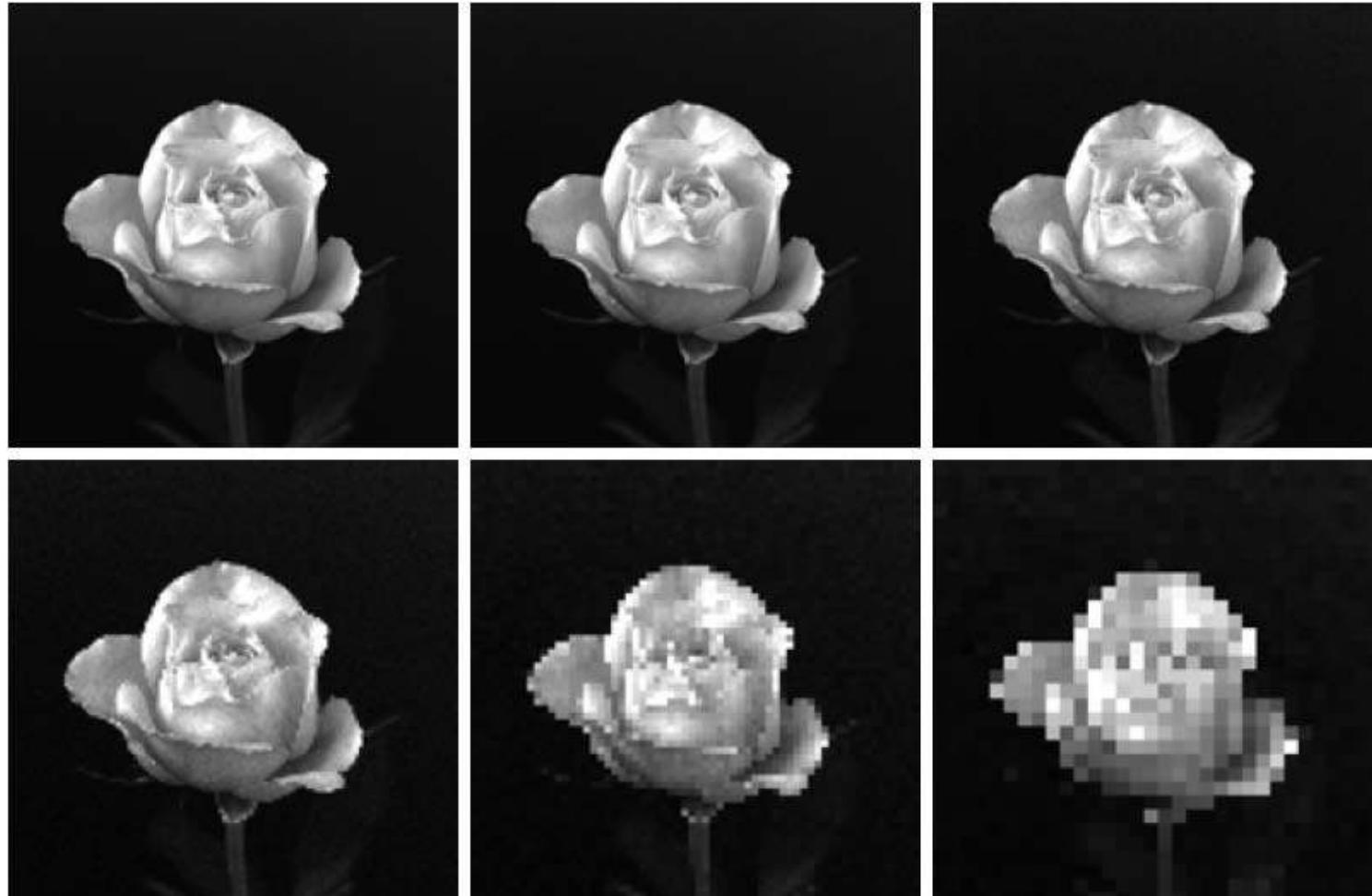
1/2 Oranında Alt Örneklenmiş Resim



Alt örneklem: Her iki örnekten birini al.

```
gul256=gul(1:2:end,1:2:end)
imshow(gul256)
title('1/2 Oranında Alt Örneklenmiş Resim')
```

Alt Örnekleme Geri Elde Etme



a b c
d e f

FIGURE 2.20 (a) 1024×1024 , 8-bit image. (b) 512×512 image resampled into 1024×1024 pixels by row and column duplication. (c) through (f) 256×256 , 128×128 , 64×64 , and 32×32 images resampled into 1024×1024 pixels.

Alt Örnekleme Geri Elde Etme

Orjinal Resim



1/8 Oranında Alt Örneklenmiş Resimden Reconstruction



- Orijinal Resim : 512x512 pixel
- Ölçeklenmiş Resim: 64x64 pixel
- Geri elde etme işlemi : Pixel Replication

Farklı Kalitede gösterim

- Her pixeli kaç bit ile gösterdiğimiz resmin kalitesini etkiler

8 Bits



4 Bit



Her griseviyesi 8 bit ile gösteriliyor:
 $2^8 = 256$ seviye var

Her griseviyesi 4 bit ile gösteriliyor:
 $2^4 = 16$ seviye var

8 Bit



7 Bit



6 Bit



5 Bit



4 Bit



3 Bit

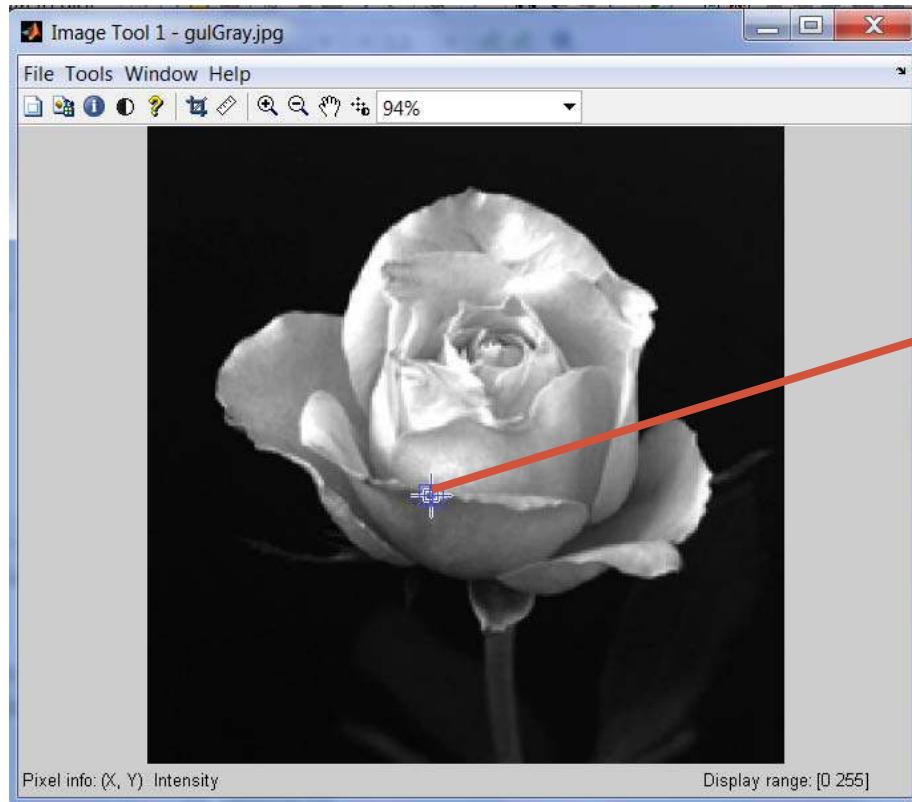


Zooming and Shrinking Digital Images

- Zoom için 2 adım gereklidir:
 - Yeni pixel yerleri yaratma
 - Bu yeni yerlere değerler atama
- 500x500'lük bir resmimiz olduğunu varsayıalım
- Bu resmi 1.5 katına genişletmek istiyoruz (750x750 pixel)
- Yeni pixele değer atarken yeni pixelin en yakın komşularına bakarız ve yeni pixele bu değeri atarız
 - Bu olay *nearest neighbor interpolation* olarak bilinir

Pixel Kopyalama (*Pixel replication*)

- Eğer resmi iki katına zoomlayacaksak:
 - Her kolon ve satırı tekrarlayabiliriz

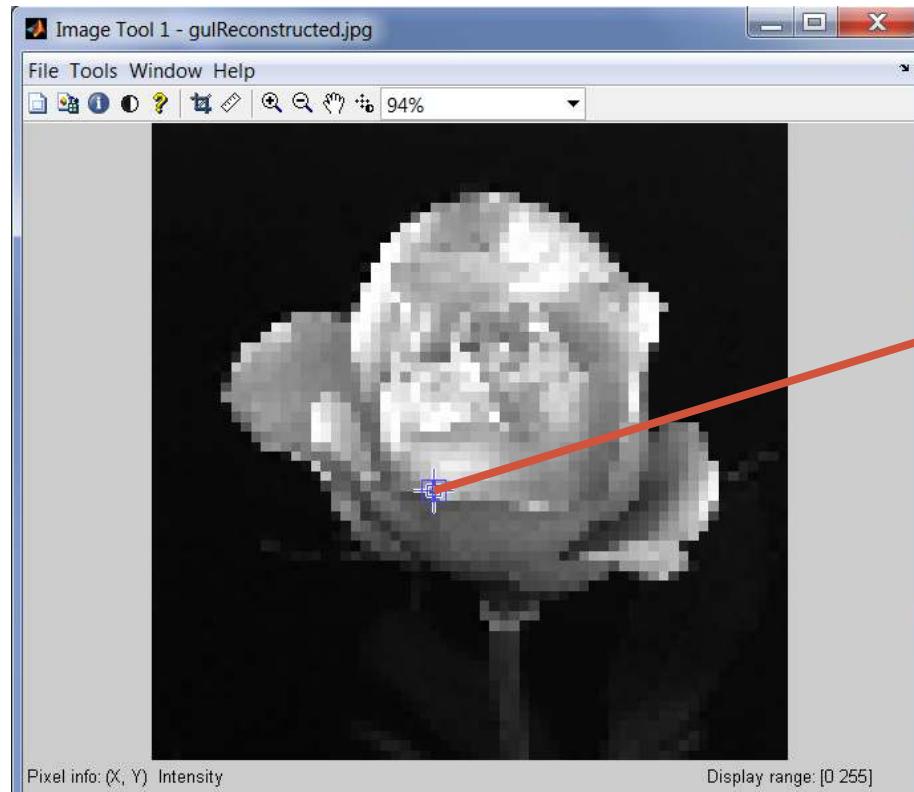


Pixel Region (Image Tool 1)													
File Edit Window Help													
Pixel info: (224, 299) 74													
1	136	147	150	154	156	157	158	161	163	161	163	163	163
4	131	135	139	146	151	156	161	166	169	171	173	173	173
5	102	105	113	125	139	152	165	176	183	198	200	200	200
L	93	109	120	121	145	145	170	173	185	191	200	200	200
D	81	76	85	85	106	107	133	140	154	159	168	168	168
I	74	78	83	81	94	97	118	128	142	155	165	165	165
S	79	82	83	81	84	86	97	106	115	119	127	127	127

Orijinal resim pixel değerleri

Pixel Kopyalama (*Pixel replication*)

- *Nearest neighbor interpolation* olarak da biliniyor
- Aşağıdaki resimde her pixel 8 kere tekrarlandı



Pixel Region (Image Tool 1)															
File Edit Window Help															
?															
9	149	149	159	159	159	159	159	159	159	159	159	159	159	159	159
9	149	149	159	159	159	159	159	159	159	159	159	159	159	159	159
9	149	149	159	159	159	159	159	159	159	159	159	159	159	159	159
9	149	149	159	159	159	159	159	159	159	159	159	159	159	159	159
5	75	75	109	109	109	109	109	109	109	109	109	109	109	109	191
5	75	75	109	109	109	109	109	109	109	109	109	109	109	109	191
5	75	75	109	109	109	109	109	109	109	109	109	109	109	109	191

1/8 Oranında Alt Örnekleme
Resimden pixel kopyalama yoluyla
geri elde edilmiş resmin pixel
değerleri

Pixel Kopyalama (*Pixel replication*)

- *Nearest neighbor interpolation* yöntemi hızlı olsa da satranç tahtası etkisi pek istenen bir yöntem değildir.
- Bunun yerine biraz daha sofisitike bir yöntem olan *bilinear interpolation* kullanılabilir.
 - Bu yöntemde sadece 1 değil 4 komşuya bakılır.
- Shrinking – Resm küçültme
 - Alt örneklemde olduğu gibi bazı pixelleri silme ile yapılır

x =

9	4	4	2	5
6	2	9	6	2
3	9	3	3	2
4	4	8	8	3
2	1	6	8	7

Pixel kopyalayarak 2 katına zoomlama

9	9	4	4	4	4	2	2	5	5
9	9	4	4	4	4	2	2	5	5
6	6	2	2	9	9	6	6	2	2
6	6	2	2	9	9	6	6	2	2
3	3	9	9	3	3	3	3	2	2
3	3	9	9	3	3	3	3	2	2
4	4	4	4	8	8	8	8	3	3
4	4	4	4	8	8	8	8	3	3
2	2	1	1	6	6	8	8	7	7
2	2	1	1	6	6	8	8	7	7

x =

9	6	3	6	4	5	10	9	4	10
7	8	10	9	1	7	10	4	8	8
3	8	9	2	4	5	1	5	10	10
2	7	7	10	3	5	8	9	7	4
3	7	7	10	5	1	5	4	1	3
7	3	9	4	8	10	4	5	5	7
1	6	3	7	2	1	9	1	7	4
5	7	7	6	9	5	9	7	6	5
10	9	1	5	10	4	10	8	2	10
10	1	6	4	9	6	2	4	3	9

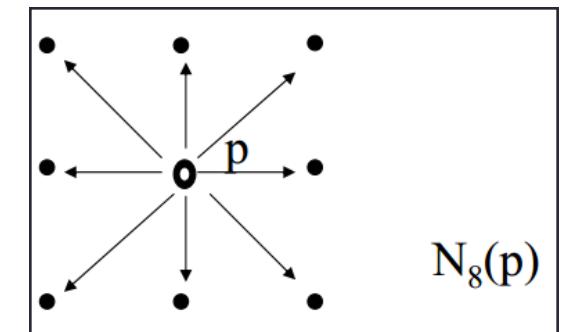
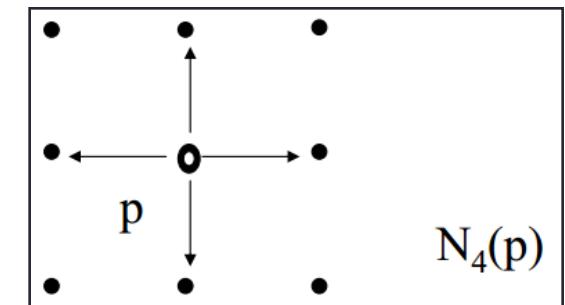
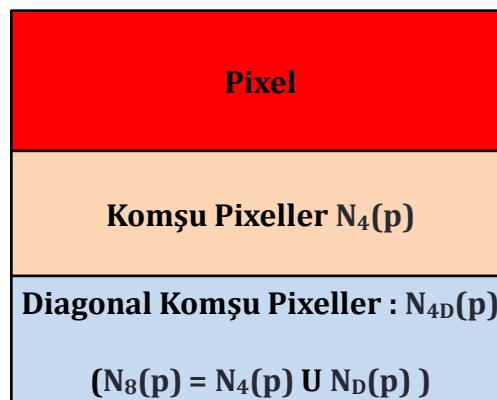
9	3	4	10	4
3	9	4	1	10
3	7	5	5	1
1	3	2	9	7
10	1	10	10	2

Pixeller Arasındaki Temel İlişkiler

- Komşuluk:

- (x,y) noktasındaki pixel p , 4 yatay ve dikey komşuya sahiptir
 - $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$
- Bu pixeller p 'nin 4 komşusu olarak bilinir ve $N_4(p)$ ile gösterilir
 - Her komşu bir birim uzaklıktadır.
- Köşegenlerdeki komşular $N_8(p)$ komşulardır

$(x-1,y-1)$	(x,y)	$(x+1,y-1)$
$(x,y-1)$	(x,y)	$(x,y+1)$
$(x+1,y-1)$	$(x+1,y)$	$(x+1,y+1)$



Adjacency, Connectivity, Regions, and Boundaries

- Adjacency (Bitişiklik)
- Connectivity (Bağlılık)
- Regions (Bölgeler)
- Boundaries (Sınırlar)
- *Pixellerarası bağlılık* bölge ve sınır gibi temel sayısal imge kavramlarını kolaylaştırır
- İki pixelin *bağlı* olup olmadığını anlamak için
 - İki pixel komşu olmalıdır
 - Grilik seviyeleri belli bir kriteri sağlamalıdır.
 - Örneğin binary bir imgede iki pixel komşu olabilir ancak ikisinin de değeri birbirine eşitse (1-1 veya 0-0) bağlı oldukları söylenir.

Adjacency (Bitişiklik)

- V 'nin bitişikliği test etmek için tanımlanmış değerler setini simgelediğini varsayalım
 - Örneğin, Binary imge için $V=\{1\}$ olabilir
 - Gri-seviyeli imge için 0-255 arası herhangi bir değeri içerebilir. V bu 256 değerin herhangi bir alt kümesi olabilir
- 3 çeşit bitişiklik tanımı vardır:
 - *4-adjacency*: V kümesine ait değerler sahip iki pixel, p ve q , eğer $q \in N_4(p)$ kümesi içindeyse 4-bitişik kabul edilir
 - *8-adjacency*: V kümesine ait değerler sahip iki pixel, p ve q , eğer $q \in N_8(p)$ kümesi içindeyse 8-bitişik kabul edilir
 - *m-adjacency* (mixed adjacency): V kümesine ait değerler sahip iki pixel, p ve q , aşağıdaki şartları taşırsa m -bitişik kabul edilir
 - $q \in N_4(p)$ içinde ya da
 - $q \in N_D(p)$ içinde ve $[N_4(p) \cap N_4(q) = \emptyset]$

Adjacency (Bitişiklik)

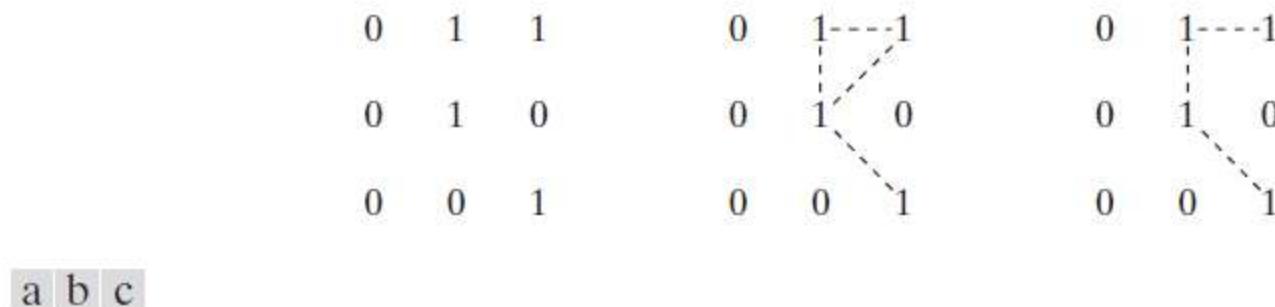
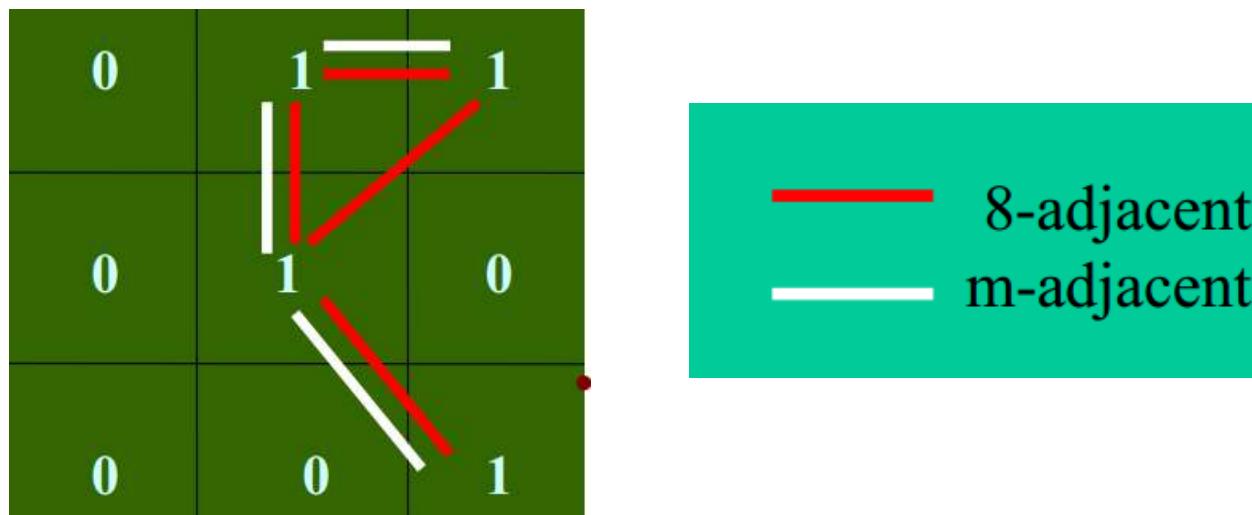


FIGURE 2.26 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.



Sayısal Yol (Path)

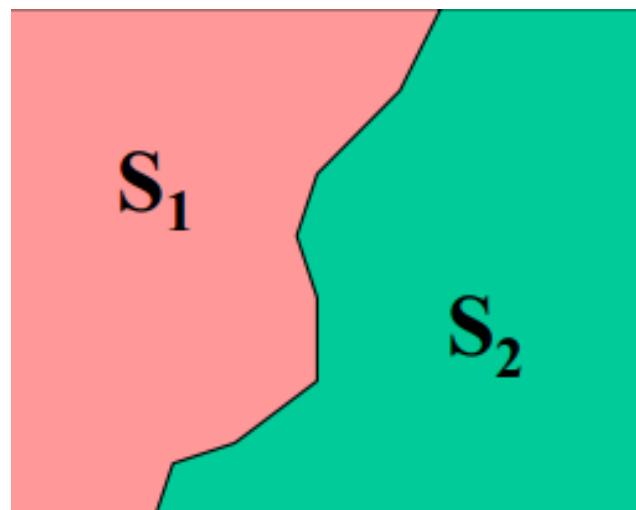
- Koordinatları (x,y) olan pixel p'den koordinatları (s,t) olan pixel q'ya yol koordinatları aşağıda verilen birbirinden ayrık pixeller dizisidir:
 - $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ ve aşağıdakiler sağlanır:
 - $(x_0, y_0) = (x, y)$ $(x_n, y_n) = (s, t)$
 - $(x_0, y_0) (x_{i-1}, y_{i-1})$ $0 \leq i \leq n$ bitişik pixellerdir
 - N: yolun uzunluğudur
 - $(x_0, y_0) = (x_n, y_n)$ ise Kapalı yol (Closed Path) adını alır
 - Bitişiklik kavramı 3 tane tanımdan herhangi biri olabilir

Adjacency (Bitişiklik)/Connectvity (Bağlılık)

- S bir imgenin alt kümesi olsun
- Pixel p ile pixel q arasında S 'nin bütün pixellerini içeren bir bir yol bulunabiliyorsa bunlar S içinde birbirine bağlıdır.
- S içinde her pixel için, S içinde bu pixele bağlı olan pixeller seti S 'nin bağlı elemanları (***connected components***) olarak bilinir
- Eğer S sadece bir tane connected componente sahipse ***connected set*** olarak adlandırılır

Adjacency (Bitişiklik)/Connectvity (Bağlılık)

- S1 ve S2 alt imgeleri, eğer eğer S1'deki bazı pixeller S2deki bazı pixellerle bitişik ise bu iki alt imge bitişiktir.



Region

- R bir imgenin alt kümesi olsun
- R connected set isimgenin bir bölgesi (region) olarak adlandırılır
- Bir bölgenin sınırı (boundary, border ya da contour) bu R içerisinde olan ve en az bir komşusu R dışında olan pixeller setidir.
- Eğer R tüm imgenin (rectangular set of pixels) sınır ilk ve son satır ve sütunlardan oluşur.

Uzaklık Ölçüleri (Distance Measures)

- p, q, ve z, with koordinatları sırasıyla (x, y), (s, t), ve (v, w), olan pixeller için uzaklık fonksiyonu , D aşağıdaki koşulları sağlar:
 - $D(p, q) \geq 0$ ($D(p, q)=0$ iff $p=q$)
 - $D(p, q)=D(q, p)$, ve
 - $D(p, z) \leq D(p, q)+D(q, z)$.
- p ve q arasında *Euclidean distance*
 - $D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$
- p ve q arasında D_4 uzaklık (*city-block distance*)
 - $D_4(p, q) = |x - s| + |y - t|$
- p ve q arasında D_8 uzaklık (*chessboard distance*)
 - $D_8(p, q) = \max(|x - s|, |y - t|)$

Uzaklık Ölçüleri (Distance Measures)

D₄ uzaklıkları

D₄ = 1 (x,y)'nin 4 komşularıdır

			2	
2	2	1	2	
2	1	0	1	2
2	1	2		
			2	

D₈ uzaklıkları (chessboard distance)

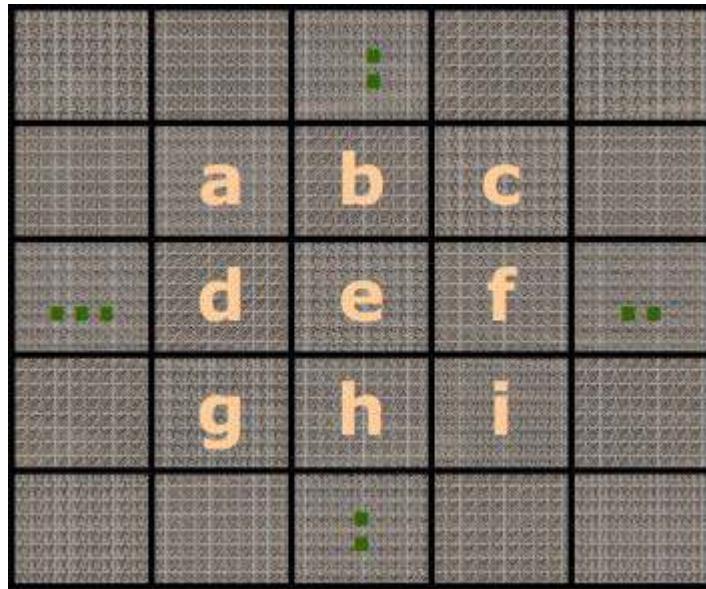
D₈ = 1 (x,y)'nin 8- komşularıdır

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Pixel Bazında İmge İşlemleri

- Arithmetic/Logic Operations
 - Toplama: $p + q$
 - Çıkarma: $p - q$
 - Çarpma: $p * q$
 - Bölme: p/q
 - AND: $p \text{ AND } q$
 - OR : $p \text{ OR } q$
 - Complement: $\text{NOT}(q)$
- Komşuluk Bazlı aritmetik İşlemler

Komşuluk Bazlı Aritmetik İşlemler



w₁	w₂	w₃
w₄	w₅	w₆
w₇	w₈	w₉

$$\begin{aligned} p &= (w_1\mathbf{a} + w_2\mathbf{b} + w_3\mathbf{c} + w_4\mathbf{d} + w_5\mathbf{e} + w_6\mathbf{f} + w_7\mathbf{g} + w_8\mathbf{h} + w_9\mathbf{i}) \\ &= \sum w_i f_i \end{aligned}$$

Arithmetric/Logic İşlemler

- Komşuluk göz önünde bulundurularak gerçekleştirilen işlemler:
 - Smoothing / averaging
 - Noise removal / filtering
 - Edge detection
 - Contrast enhancement
- Önemli noktalar:
 - w_i seçimi
 - Pencere genişliğinin, N, seçimi
 - Sınırarda hesaplamalar:
 - Sınırarda hesap yapma
 - İmgeyi 0'lar ekleyerek genişlet (sınır noktalarında zero padding)
 - İmgeyi periyodik olarak varsayıarak sınırlara ekleme yap

SAYISAL GÖRÜNTÜ İŞLEME

Hafta3: Uzaysal Domainde Görüntü iyileştirme

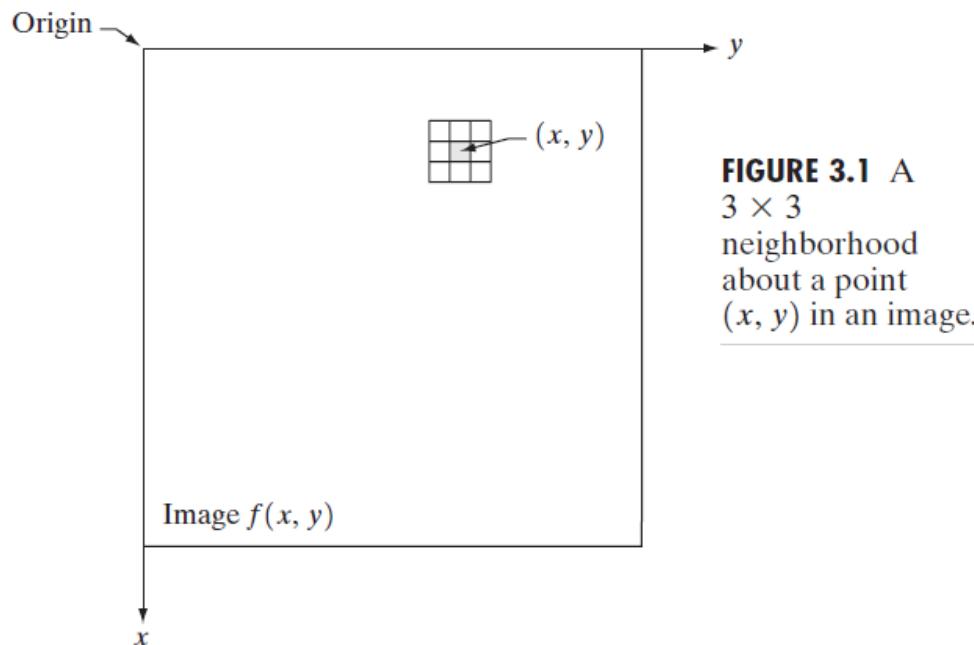
Görüntü İyileştirme

- Amaç
 - İmgeler, iyileştirilerek elde edilen sonucun özel bir uygulama için daha uygun olması için işlenir
- İki domainde iyileştirme metotları bulunur:
 - Uzaysal (Spatial) Domainde iyileştirme metotları
 - Pixellerin direk manipülasyonu
 - Frekans Domain Domainde iyileştirme metotları
 - İmgenini Fourier transformu üzerinde yapılan işlemler

Spatial Domainde Görüntü İyileştirme

- İşlem şu şekilde simgelenenecek:

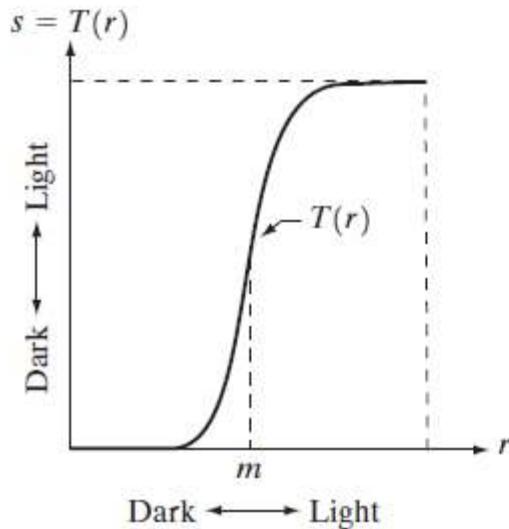
- $g(x,y) = T[f(x,y)]$
 - $f(x,y)$: Input image
 - $g(x,y)$: İşlenmiş (Output) imge
 - T : f üzerinde belli bir (x,y) 'nin belli komşuluğunda tanımlı bir fonksiyon



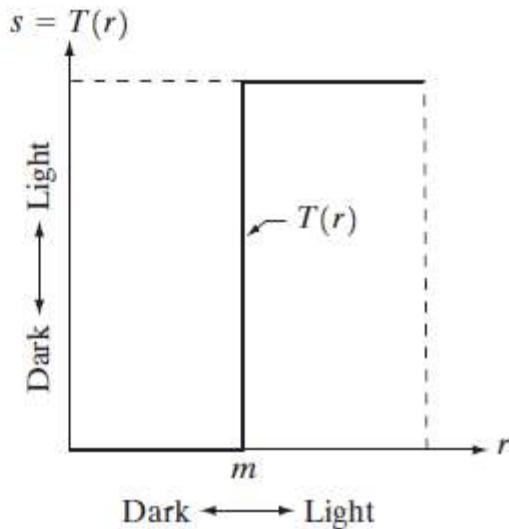
Komşuluk 1 olarak seçilirse işlem her pixel üzerinde yapılır.
En basit fonksiyon bu şekilde elde edilir. (**gray-level (intensity or mapping) transformation function**)

$$s = T(r)$$

Kontrast İyileştirme



Contrast streching :
m'den yüksek değerler
açılırken, diğerleri
koyulaştırılır.



thresholding function:
Binary İmge oluşur.

a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

Gri Seviyeli İmgeler İçin Temel Dönüşümler

- Image Negatives
- Log Transformations
- Power-Law Transformations
- Piecewise-Linear Transformation Functions

Image Negatives

- $[0, L-1]$ aralığındaki bir imgenin negatifi için şekilde gösterilen fonksiyon kullanılır:
 - $s = L - 1 - r$.

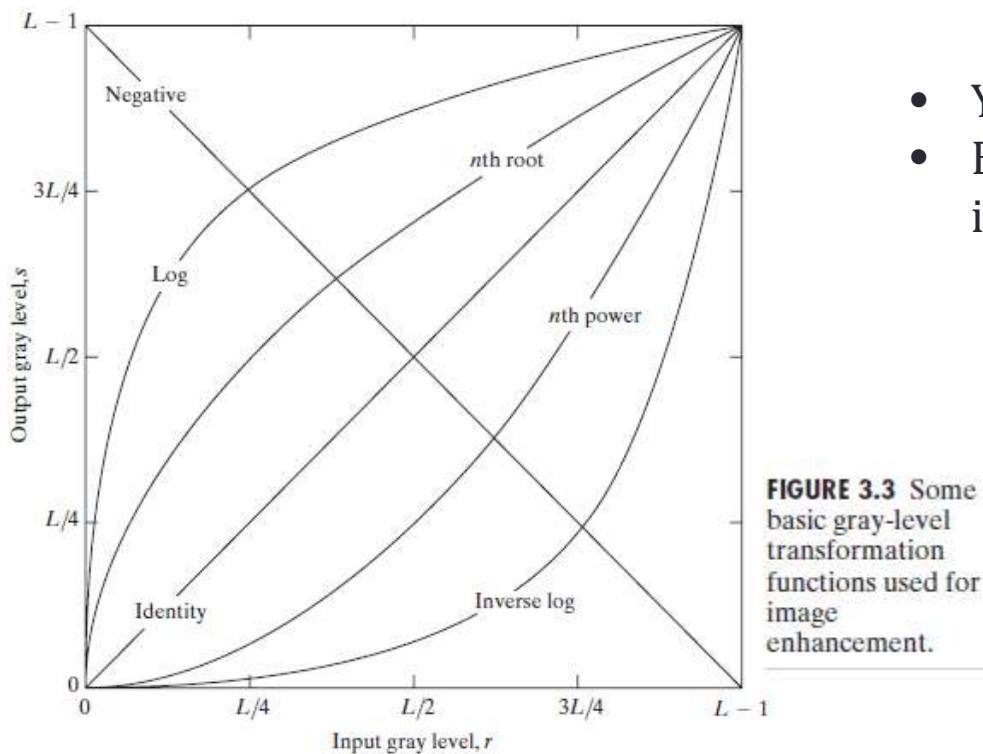


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

- Yoğunluk seviyeleri tersine çevrilir
- Beyaz ya da siyah ayrıntıların iyileştirilmesi gereğinde uygundur

Image Negatives

Original Image



Negative Image

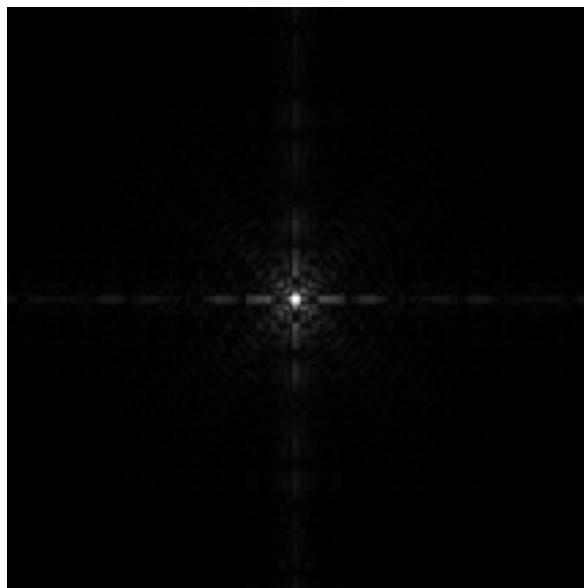


Log Transformations

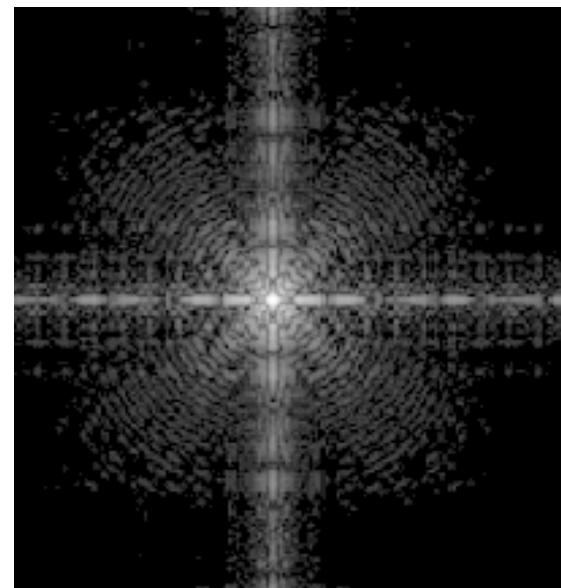
- Genel formu:
 - $s = c \log (1 + r)$
 - c : sabit, $r \geq 0$
- Bu dönüşüm az aralıklı düşük gri seviyeli değerleri (siyaha yakın değerler) yüksek aralıklı hale getirir.
 - Yüksek seviyeli (beyaza yakın) değerler için tersi geçerli
- Koyu renkli pixelleri genişletmek için kullanılabilir
- Açık renklileri genişletmek için ters log dönüşüm kullanılır

Log Transformations

Original Image



Log Transform Sonucu



Power-Law Transformations

- Genel formu:

- $s = cr^\gamma$ ya da input 0 olduğunda $c(r + \varepsilon)^\gamma$
- c ve γ sabit sayılar
- Gamma correction olarak da bilinir

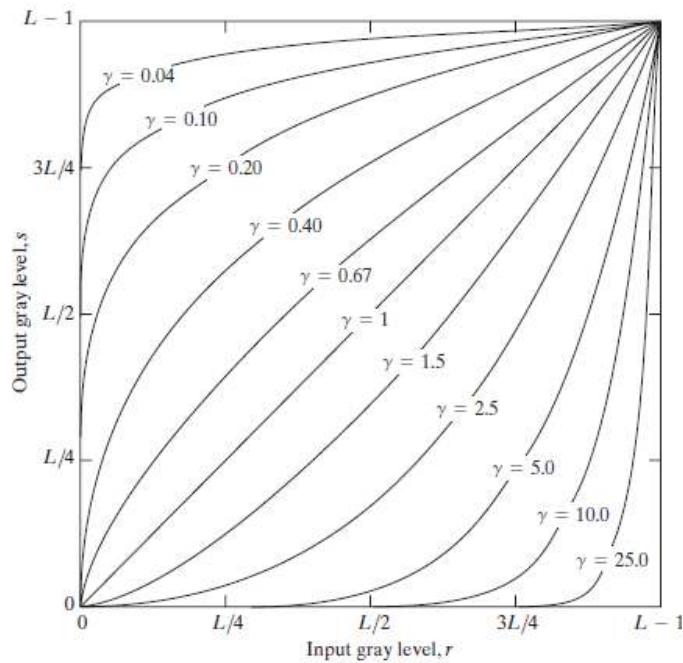
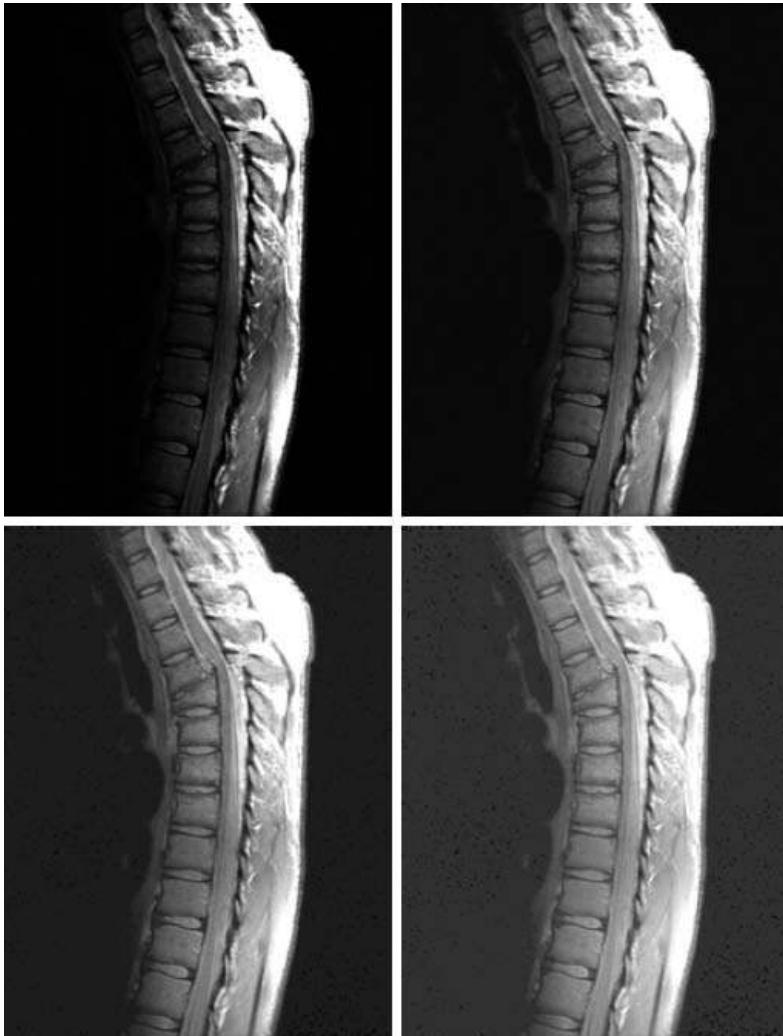


FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases).

Log fonksiyonuna benzer. Ancak burada çok daha farklı curve elde edilebilir.

Power-Law Transformations



a b
c d

FIGURE 3.8
(a) Magnetic resonance (MR) image of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3 , respectively.
(Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Power-Law Transformations

a
b
c
d

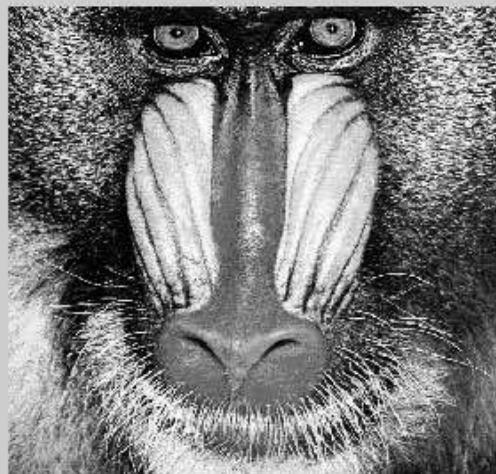
FIGURE 3.9

(a) Aerial image.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0 , respectively.
(Original image for this example courtesy of NASA.)

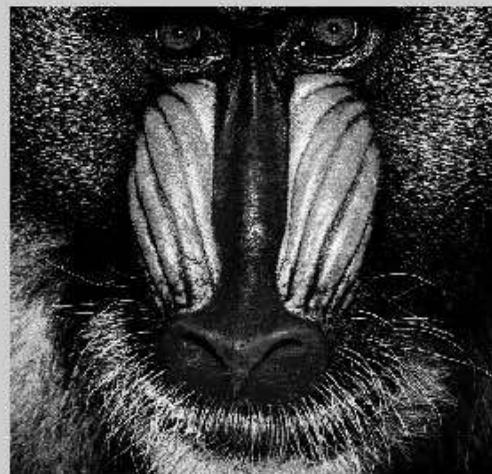


Power-Law Transformations

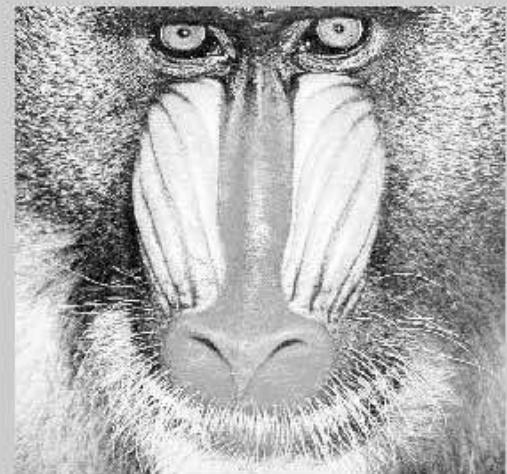
Original Image



Gamma = 3



Gamma = 0.33



Piecewise-Linear Transformation Functions

- Önceki metotları tamamlayıcı bir yaklaşım
- En önemli avantajı
 - Fonksiyon sabit değil
- Dezavataj
 - Çalışırken daha çok bilgi girilmesi gereklidir
- En basit fonksiyon
 - **Contrast stretching**



MATLAB İLE GÖRÜNTÜ İŞLEME

TEMEL İŞLEMLER

iMGE OKUMA



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



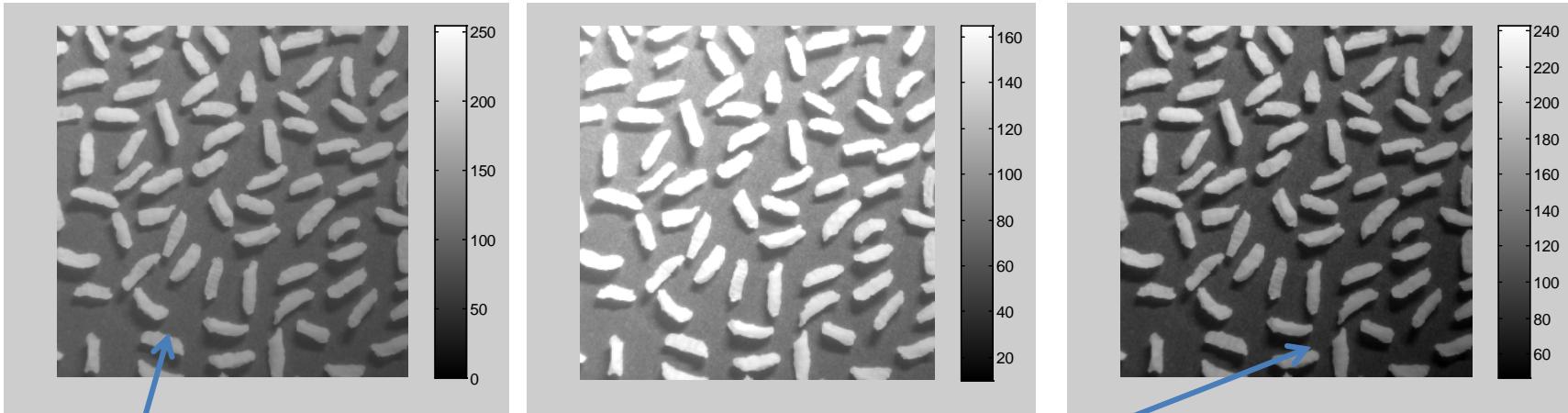
55, 249 = 126

```
a =imread('flowers.tif');  
imshow(a);  
pixval on;
```



185.07, 135.602 = 35, 34, 49

İMGE OKUMA



```
[rice,map]=imread('Fig0940(a)(rice_image_with_intensity_gradient).tif');  
imshow(rice),colormap(gray(256))  
imshow(rice,[10 165])  
imshow(rice,[])
```

imshow(rice) : Colormap otomatik olarak 0-255 arasında oluyor (colormap(gray(256)) ile aynı)
imshow(rice,[L1 L2]) : Colormap L1 – L2 arasında olacak şekilde çiziliyor. **L1=>Siyah L2=>Beyaz**
imshow(rice,[]) : Colormap L1 = minimum yoğunluk, L2= maksimum genlik olacak şekilde çiziliyor

İMGİE OKUMA



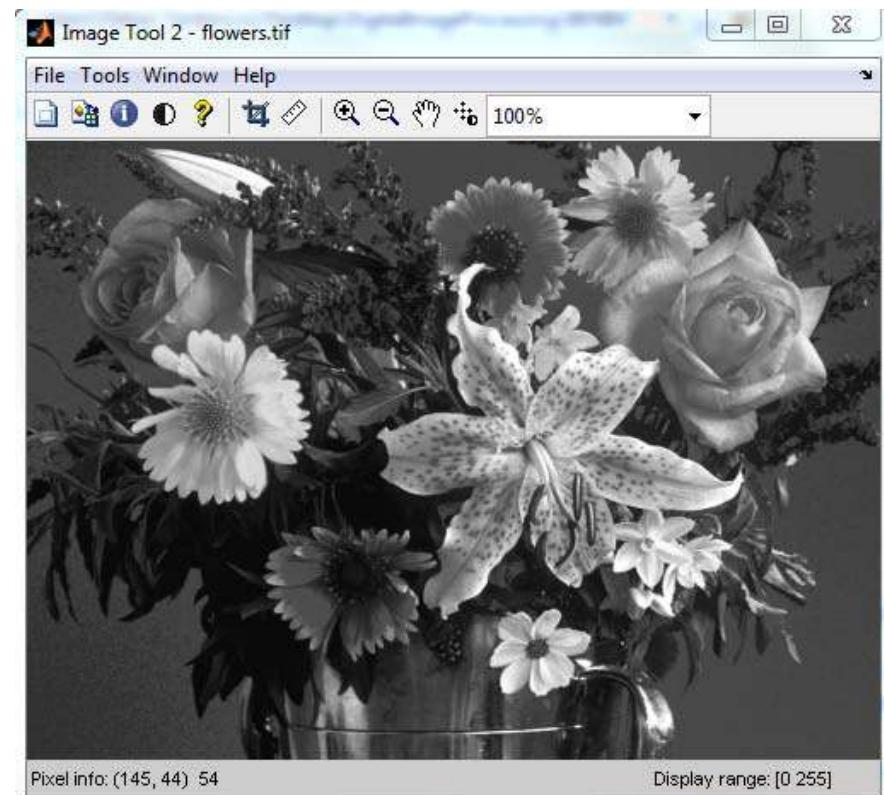
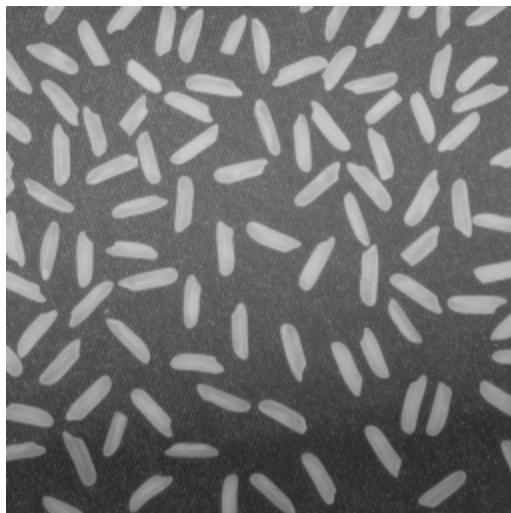
getimage : Açık olan figure'deki resmi alma

imshow rice.png

I = getimage;%I=imread('rice.png') ile aynı

imtool : İmgeyi imtool ile açma

imtool flowers.tif



İmge Görüntüleme



truesize : Görüntülenen ve ekrandaki görüntüsünü değiştirilen (figürün boyunu büyütme, küçültme yoluyla) imgenin asıl boyutu ile görüntülenmesini sağlar.

zoom : Görüntüyü yaklaştırip uzaklaştırma için kullanılır



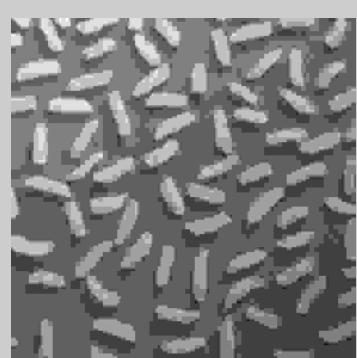
İmge Yazma ve Bilgilerine Ulaşma

```
info=imfinfo('rice.tif');
```

Dosya boyutu 355KB

```
[rice,map]=imread('rice.tif');
imwrite(rice,'rice.jpg','Quality',0)
r=imread('rice.jpg');
imshow(r)
imfinfo('rice.jpg')
```

Düşük kaliteli kayıt
yapılıyor. Dosya
boyutu 6KB



```
Filename: 'rice.jpg'
FileModDate: '21-Sub-2013 13:30:24'
FileSize: 5597
Format: 'jpg'
FormatVersion: ""
Width: 600
Height: 600
BitDepth: 8
ColorType: 'grayscale'
FormatSignature: ""
NumberOfSamples: 1
CodingMethod: 'Huffman'
CodingProcess: 'Sequential'
Comment: {}
```

Filename: [1x179 char]
FileModDate: '20-Ağu-2006 00:27:18'
FileSize: 360574
Format: 'tif'
FormatVersion: []
Width: 600
Height: 600
BitDepth: 8
ColorType: 'grayscale'
FormatSignature: [73 73 42 0]
ByteOrder: 'little-endian'
NewSubFileType: 0
BitsPerSample: 8
Compression: 'Uncompressed'
PhotometricInterpretation: 'BlackIsZero'
StripOffsets: [47x1 double]
SamplesPerPixel: 1
RowsPerStrip: 13
StripByteCounts: [47x1 double]
XResolution: 320
YResolution: 320
ResolutionUnit: 'Inch'
Colormap: []
PlanarConfiguration: 'Chunky'
TileWidth: []
TileLength: []
TileOffsets: []
TileByteCounts: []
Orientation: 1
FillOrder: 1
GrayResponseUnit: 0.0100
MaxSampleValue: 255
MinSampleValue: 0
Thresholding: 1
Offset: 360008

Sıkıştırma Oranı Hesaplama

```
imwrite(rice,'rice.jpg','Quality',8)
```

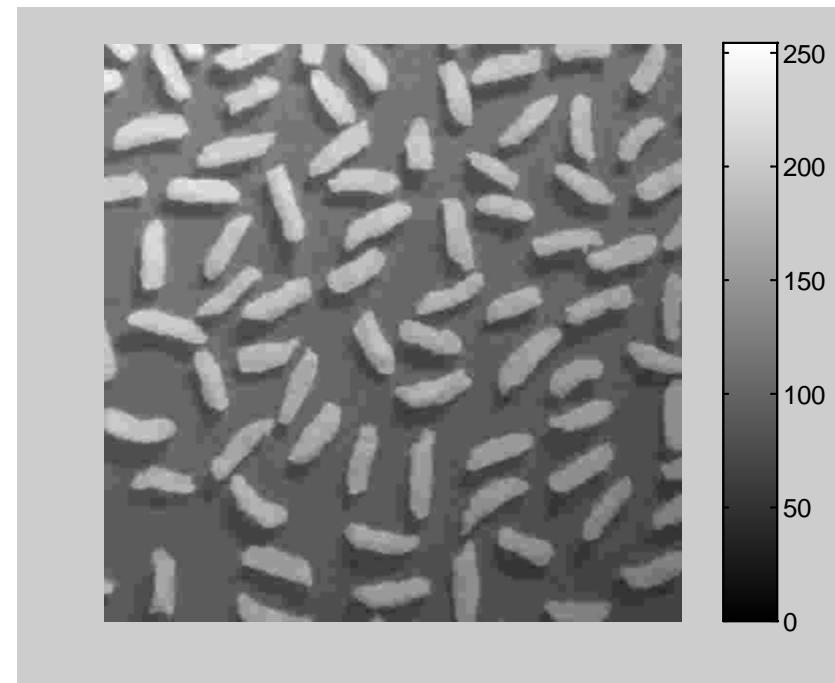
Dosya boyutu 9 KB

```
K=imfinfo('rice.jpg')
image_bytes = K.Width*K.Height*K.BitDepth/8;
compressed_bytes = K.FileSize;
compression_ratio = image_bytes/compressed_bytes;
display(compression_ratio);
```

```
Filename: 'rice.jpg'
FileModDate: '21-Jun-2013 13:43:29'
FileSize: 8319
Format: 'jpg'
FormatVersion: ""
Width: 600
Height: 600
BitDepth: 8
ColorType: 'grayscale'
FormatSignature: ""
NumberOfSamples: 1
CodingMethod: 'Huffman'
CodingProcess: 'Sequential'
Comment: {}
```

```
compression_ratio =
```

43.2744



İmge Tipleri



- Index
 - Data matrisi(uint8, uint16, double)
 - Colormap matrisi(m x 3 array of double [0 1])
- Intensity (black = 0, white = ∞)
- Binary (0, 1)
 $B = \text{logical}(\text{uint8}(\text{round}(A)));$
- RGB (m x n x 3 of truecolor)

İmge Tiplerinin Birbirine Çevrilmesi

- dither
- gray2ind
- grayslice
- im2bw
- ind2gray
- ind2rgb
- mat2gray
- rgb2gray
- rgb2ind

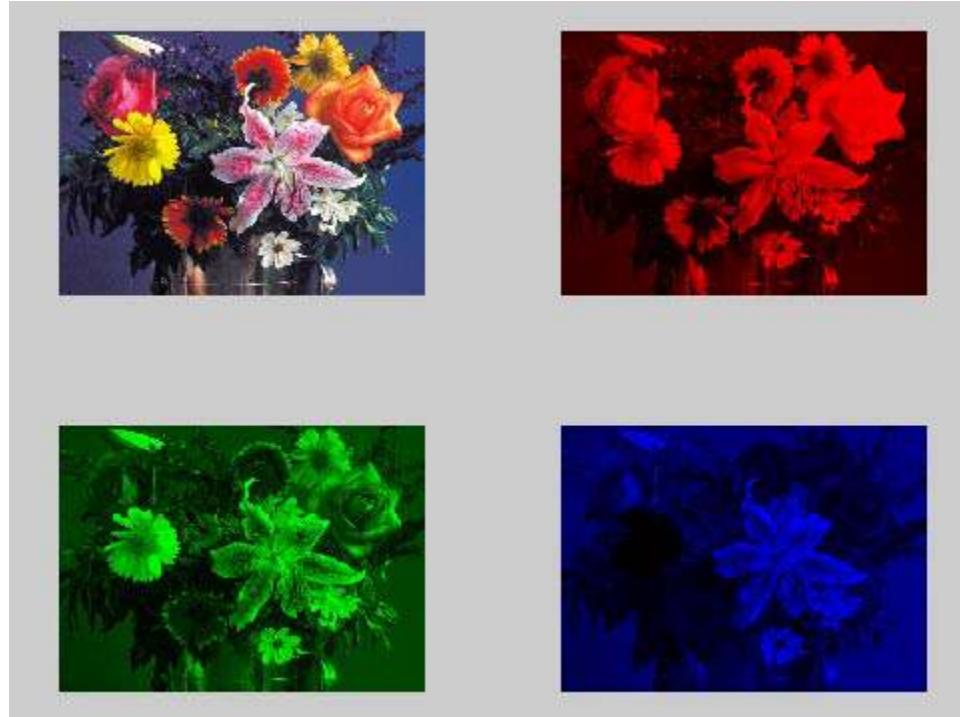
İmgelerin Gri ve Binary Hale Dönüşürtülmesi

```
a= imread('flowers.tif');
subplot(2,2,1);
imshow(a);
subplot(2,2,2);
b=imresize(a,[256 256]);
imshow(b);
subplot(2,2,3);
c=rgb2gray(b);
imshow(c);
subplot(2,2,4);
d=im2bw(c);
imshow(d);
```



RGB Komponentleri

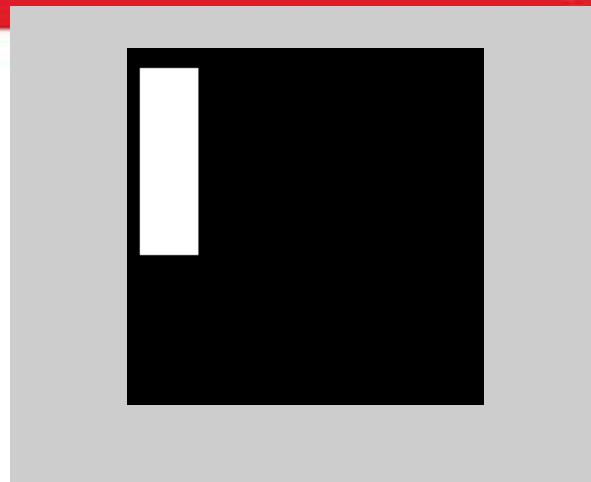
```
a=imread('flowers.tif');
subplot(2,2,1);
imshow(a);
R=a;
G=a;
B=a;
R(:,:,2:3)=0;
subplot(2,2,2);
imshow(R);
G(:,:,1)=0;
G(:,:,3)=0;
subplot(2,2,3);
imshow(G);
B(:,:,1)=0;
B(:,:,2)=0;
subplot(2,2,4);
imshow(B);
```



Temel İşlemler

Binary Images

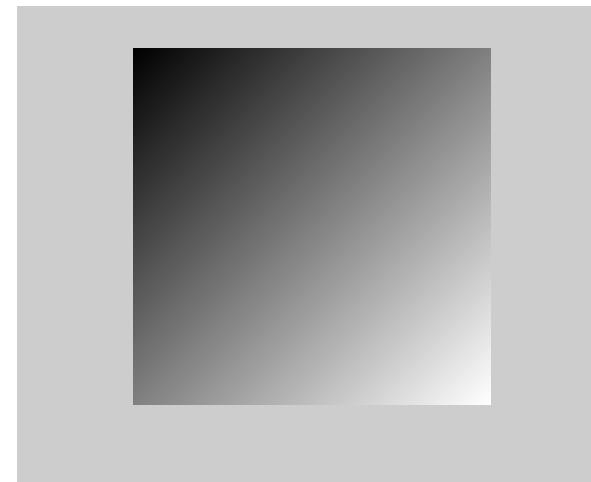
```
r = zeros(250,250);  
r(15:145,10:50) = 1;  
imshow(r);
```



Intensity Images

```
% uint8 [0 256]  
% uint16 [0 65535]  
% double [0.0 1.0]
```

```
for i=1:250  
for j=1:250  
r(i,j) = i/500+j/500;  
end  
end  
imshow(r);
```



Temel İşlemler



% uint8 [0 255]
% uint16 [0 65535]
% double [0.0 1.0]

Name	Converts Input to:	Valid Input Image Data Classes
im2uint8	uint8	logical,uint8,uint16, and double
im2uint16	uint16	logical,uint8,uint16, and double
mat2gray	double (in range [0, 1])	double
im2double	double	logical,uint8,uint16, and double
im2bw	logical	uint8,uint16, and double

Yoğunluk değerlerini tamsayıya dönüştürmek:

A = im2uint8(r);
display(A);

Yapılan işlem: r'deki 1 değerini 255'e eşlemektir. round(r*255);
Eğer r'nin maksimum değeri 1'den büyük ise bütün bu değerler de 255'e eşitlenir.

>> r(1:5,1:5)

ans =

0.0040 0.0060 0.0080 0.0100 0.0120
0.0060 0.0080 0.0100 0.0120 0.0140
0.0080 0.0100 0.0120 0.0140 0.0160
0.0100 0.0120 0.0140 0.0160 0.0180
0.0120 0.0140 0.0160 0.0180 0.0200

>> A(1:5,1:5)

ans =

1 2 2 3 3
2 2 3 3 4
2 3 3 4 4
3 3 4 4 5
3 4 4 5 5

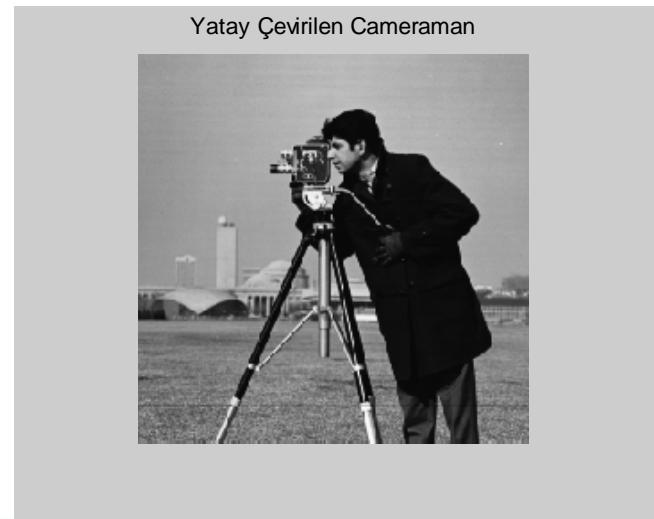
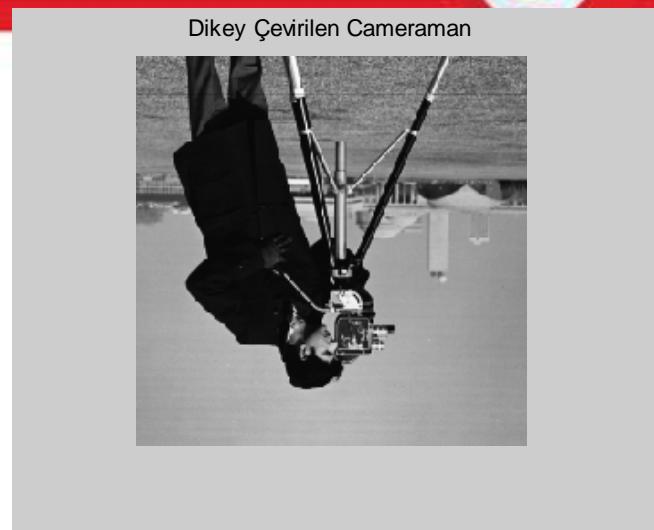
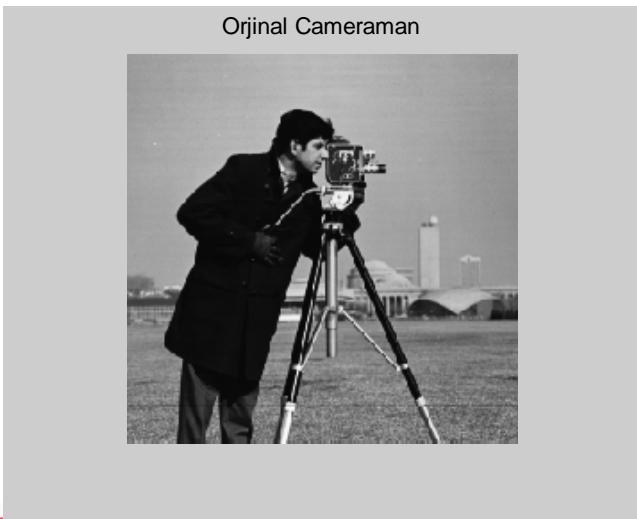
Temel İşlemler

Resmi Ters Çevirme

```
cMan = imread('cameraman.tif');  
imshow(cMan),title('Orjinal Cameraman')
```

```
cManVFlipped=cMan(end:-1:1,:);  
imshow(cManVFlipped)  
title('Dikey Çevirilen Cameraman')
```

```
cManHFlipped=cMan(:,end:-1:1);  
imshow(cManHFlipped)  
title('Yatay Çevirilen Cameraman')
```



Temel İşlemler



Belli açıyla resmi çevirme

```
I = imread('ic.tif'); J= imrotate(I, 35, 'bilinear');  
imshow(I),figure, imshow(J)
```

Orjinal Cameraman



35 derece Döndürülmüş İmge



Alt Örnek Elde Etme



Subsampling (Seyreltme, Altörnekleme)

- Alt örnekleme yaparak spatial çözünürlüğünü azaltabiliriz
 - Daha az sayıda örnek

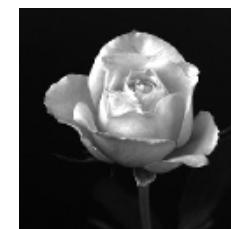
Orjinal İmge



Downsampled By 2



Downsampled By 4



```
imshow(gul,[]),title('Orjinal İmge')
```

```
N=2;imshow(gul(1:N:end,1:N:end),[]),title('Downsampled By 2')
```

```
N=4;imshow(gul(1:N:end,1:N:end),[]),title('Downsampled By 4')
```

Alt Örnek Elde Etme



```
imshow(gul,[],title('Orjinal İmge')
```

```
N=2;imshow(gul(1:N:end,1:N:end),[]),title('Downsampled By 2')
```

```
N=4;imshow(gul(1:N:end,1:N:end),[]),title('Downsampled By 4')
```

Downsample komutu da kullanılabilir

Input matrisinin kolonları boyunca alt örnekleme yapmamızı sağlar

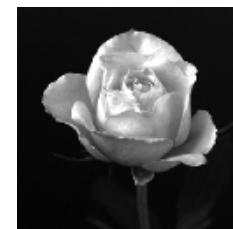
Orjinal İmge



Downsampled By 2



Downsampled By 4



```
imshow(gul,[],title('Orjinal İmge')
```

```
N=2;imshow(gul(1:N:end,1:N:end),[]),title('Downsampled By 2')
```

```
N=4;imshow(gul(1:N:end,1:N:end),[]),title('Downsampled By 4')
```

Alt Örnekten Geri Elde Etme



```
function Y=upsampleby2(X)
Y=zeros(size(X,1)*2,size(X,2)*2,3);
for n=1:3
    Y(1:2:end,1:2:end,n)=X(:,:,n);
    Y(2:2:end,2:2:end,n)=X(:,:,n);
    Y(1:2:end,2:2:end,n)=X(:,:,n);
    Y(2:2:end,1:2:end,n)=X(:,:,n);
end
Y=uint8(Y);
Ya da
yenilmge=im_resize(imge,2*size(imge));
```

1/8 Oranında Örneklenmiş İmge



Boyut: 64x64

```
g2=im_resize(g,8*size(g));imshow(g2,[])
```

1/8 Oranında Örneklenmiş İmgeden
Orjinal İmgenin Elde Edilişi

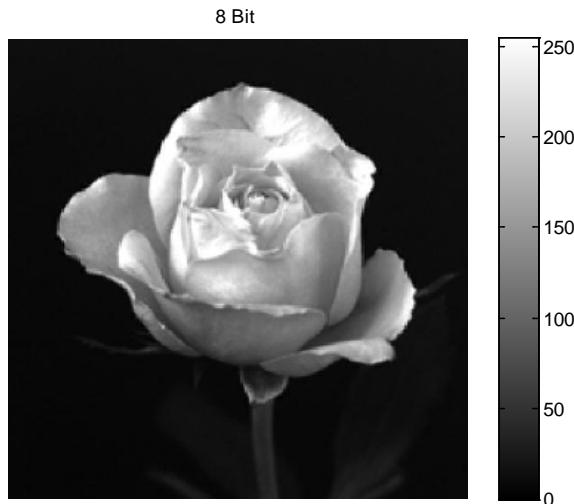


Boyut: 512x512

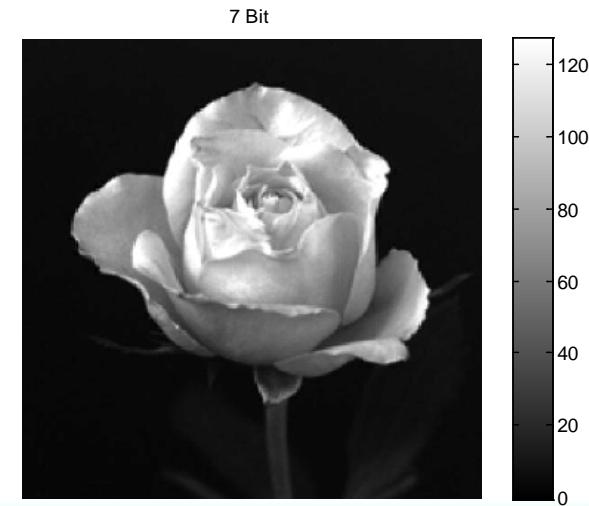
Bit Kalitesini Değiştirme



```
gul=imread('gulGray.jpg');
N=256;
map=colormap(gray(N));
imshow(gul,map)
title([ num2str(log2(N)) ' Bit'])
colorbar
```



```
N=N/2;
map=colormap(gray(N));
gul=floor(gul/2);
imshow(gul,map)
title([ num2str(log2(N)) ' Bit'])
colormap
```



Bit Kalitesini Değiştirme



gray2ind :

[X,MAP] = GRAY2IND(I,N) converts the intensity image I to an indexed image X with colormap GRAY(N). If N is omitted, it defaults to 64.

```
[cMan,map]=imread('cameraman.tif');  
imshow(cMan)  
title('Orjinal İmge - 8 Bit')
```

```
[x,map]=gray2ind(cMan,16);  
imshow(x,[])  
title('4 Bit ile Temsili')
```

Orjinal İmge - 8 Bit



4 Bit ile Temsili



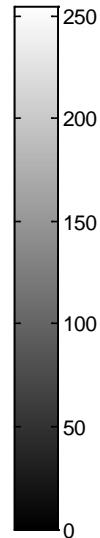
Bit Kalitesini Değiştirme



$[Y, \text{newmap}] = \text{imapprox}(X, \text{map}, n)$

- *imapprox* fonksiyonu X imgesindeki renkleri ve map renk haritasını minimum varyans kuanlatalama kullanarak değiştirir.
- Yeni imge Y içinde en fazla n renk bulunur

Bit Kalitesini Değiştirme



```
gul=imread('gulGray.jpg');
imshow(gul)
colorbar
map=colormap;
```

```
[Y, newmap] = imapprox(gul, map, 8);
imshow(Y,newmap)
colorbar
```

Temel İşlemler

Resmin Negatifini Elde Etme

```
r2 = imcomplement(r1);
```

```
figure,imshow(r2);
```

veya

```
r2 = imadjust(r1,[0 1], [1 0]);
```

```
figure,imshow(r2);
```

Yapılan işlem:

Siyah (0) <= Beyaz (255)

Beyaz (255) <=Siyah (0)

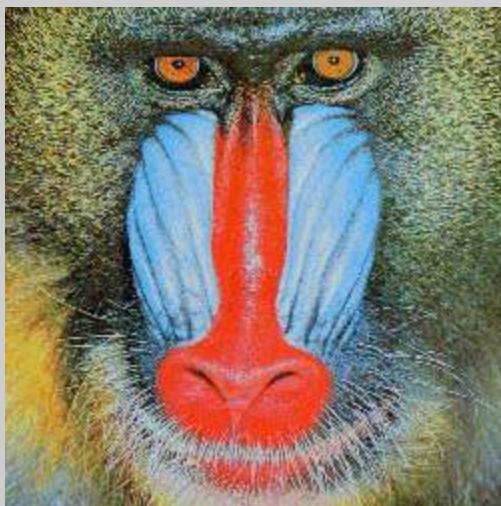
$r2=255-r1;$ %uint 8 olduğu için



Temel İşlemler



Renkli imgelerde negatif

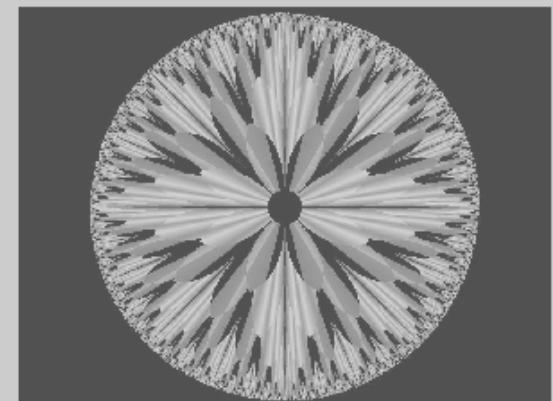
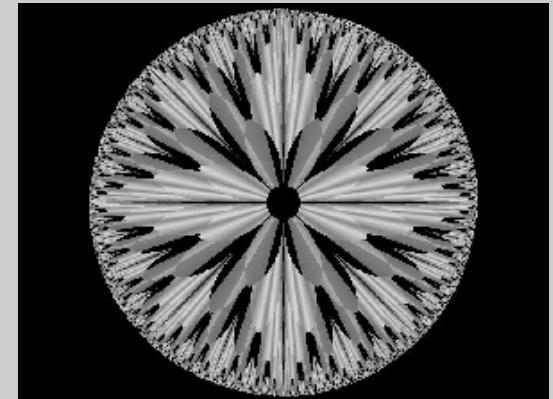


Temel İşlemler



İmgeyi transparent yapma

```
f = imread('fractal-iris.tif');  
h=imshow(f);  
set(h,'AlphaData',0.6)  
%ya da alpha(0.5)  
%Current axis'teki tüm imgelerin ayarı için  
% alpha(h,0.5)  
%sadece handle'ı h olan için
```



İki İmgeyi Üst Üste Gösterme

```
gul=imread('gulGray.jpg')
cMan = imread('cameraman.tif');
imshow(cMan)
hold
g=imshow(gul(1:2:end,1:2:end));
alpha(g,0.5)
```



Aritmetik İşlemler



Operator	Name	MATLAB Function	Comments and Examples
+	Array and matrix addition	plus(A, B)	$a + b$, $A + B$, or $a + A$.
-	Array and matrix subtraction	minus(A, B)	$a - b$, $A - B$, $A - a$, or $a - A$.
.*	Array multiplication	times(A, B)	$C = A.*B$, $C(I, J) = A(I, J).*B(I, J)$.
.*	Matrix multiplication	mtimes(A, B)	$A*B$, standard matrix multiplication, or $a*A$, multiplication of a scalar times all elements of A .
./	Array right division	rdivide(A, B)	$C = A./B$, $C(I, J) = A(I, J)/B(I, J)$.
.\	Array left division	ldivide(A, B)	$C = A.\B$, $C(I, J) = B(I, J)\./A(I, J)$.
/	Matrix right division	mrdivide(A, B)	A/B is roughly the same as $A*inv(B)$, depending on computational accuracy.
\	Matrix left division	mldivide(A, B)	$A\B$ is roughly the same as $inv(A)*B$, depending on computational accuracy.
.^	Array power	power(A, B)	If $C = A.^B$, then $C(I, J) = A(I, J)^B(I, J)$.
.^	Matrix power	mpower(A, B)	See online help for a discussion of this operator.
.'	Vector and matrix transpose	transpose(A)	$A.'$. Standard vector and matrix transpose.
'	Vector and matrix complex conjugate transpose	ctranspose(A)	A' . Standard vector and matrix conjugate transpose. When A is real $A.^* = A'$.
+	Unary plus	uplus (A)	$+A$ is the same as $0 + A$.
-	Unary minus	uminus (A)	$-A$ is the same as $0 - A$ or $-1*A$.
:	Colon		Discussed in Section 2.8.

Function	Description
imadd	Adds two images; or adds a constant to an image.
imsubtract	Subtracts two images; or subtracts a constant from an image.
immultiply	Multiplies two images, where the multiplication is carried out between pairs of corresponding image elements; or multiplies a constant times an image.
imdivide	Divides two images, where the division is carried out between pairs of corresponding image elements; or divides an image by a constant.
imabsdiff	Computes the absolute difference between two images.
imcomplement	Complements an image. See Section 3.2.1.
imlincomb	Computes a linear combination of two or more images. See Section 5.3.1 for an example.

Resimler için aritmetik operatörler

İşlemler



- imabsdiff
- imadd
- imcomplement
- imdivide
- imlincomb
- immultiply
- imsubtract
- imabsdiff
 - Mutlak fark
- imlincomb
 - İmgelerin lineer kombinasyonu
 - imlincomb(k_1, I_1, k_2, I_2)

Toplama

```
gul=imread('gulGray.jpg');  
gul=gul(1:2:end,1:2:end);  
cMan = imread('cameraman.tif');  
imshow(gul+cMan)  
%ya da imshow(imadd(gul,cMan))
```



Dikkat:

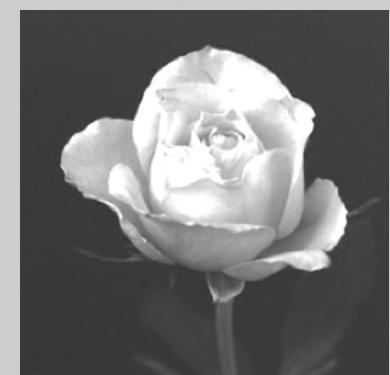
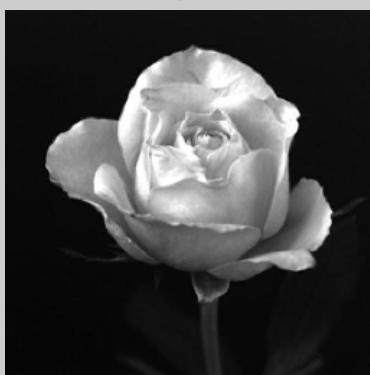
uint8 tipindeki değişkenler üzerinde yapılan işlemlerde 255'ten büyük tüm değerlerin yerine 255 atanır.

Bunun yerine:

```
A=double(gul)+double(cMan);  
A = A/max(max(A)); A = im2uint8(A);  
imshow(A)
```



Toplama ve Çıkarma



Çarpma

İki resmin çarpılması

```
gD=double(gul);cD=double(cMan);
gDc=gD.*cD;
gDcGray=mat2gray(gDc);
imshow(gDcGray)
```



İki resmin bölünmesi

```
gD=double(gul);cD=double(cMan);
gDc=gD./cD;
gDcGray=mat2gray(gDc);
imshow(gDcGray)
```



Arkaplan



- Arka plan aydınlanması hesaplama
clear, close all

```
I = imread('rice.tif');
```

```
imshow(I)
```

```
background = imopen(I, strel('disk', 15));
```

```
imshow(background)
```

Arka Plan



strel : Morfolojik yapısal element yaratır

SE = strel(*shape*, parameters)

Desteklenen yassı şekiller:

'arbitrary'	'pair'
'diamond'	'periodicline'
'disk'	'rectangle'
'line'	'square'
'octagon'	

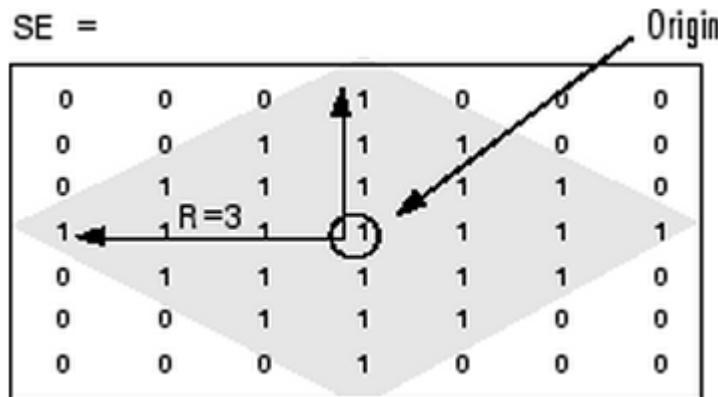
Desteklenen yassı olmayan şekiller:

'arbitrary'	'ball'
-------------	--------

strel

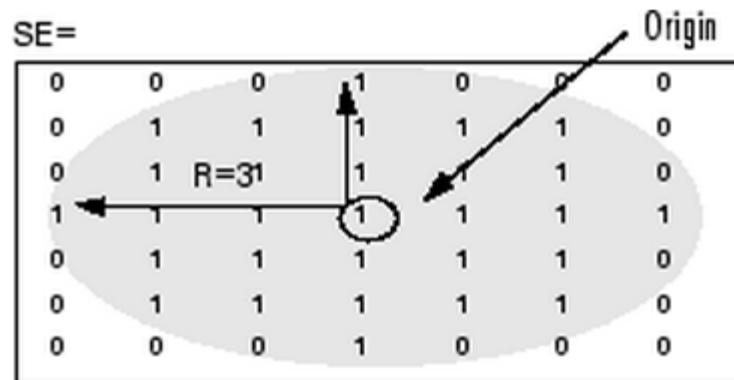


SE =



SE = strel('diamond', 3)

SE =



SE = strel('disk', 3, 0)

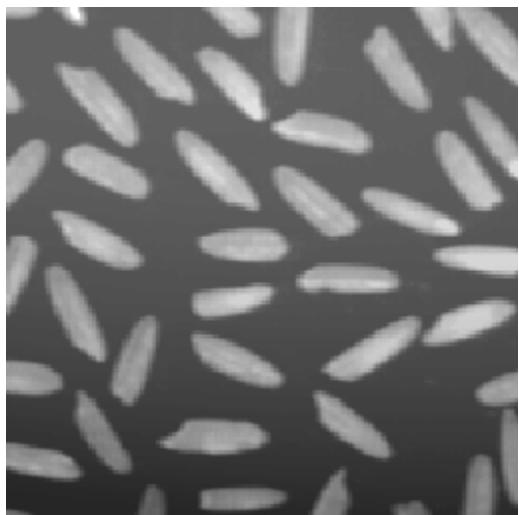
Arkaplan



Arka plan aydınlanması hesaplama

```
clear, close all  
I = imread('rice.tif');  
imshow(I), title('rice.tif')  
background = imopen(I, strel('disk', 15));  
imshow(background), title('rice.tif için arka plan')  
imshow(I-background), title('rice.tif - arkaplan')
```

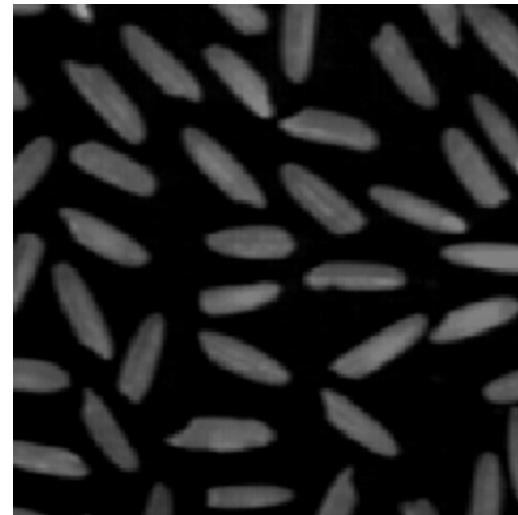
rice.tif



rice.tif için arka plan



rice.tif - arkaplan

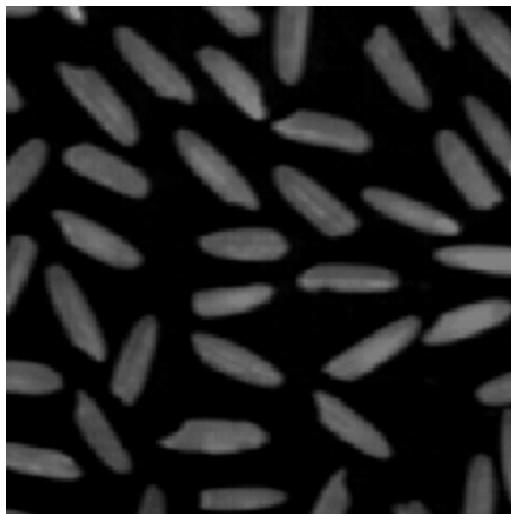


İmge Kontrast Ayarı

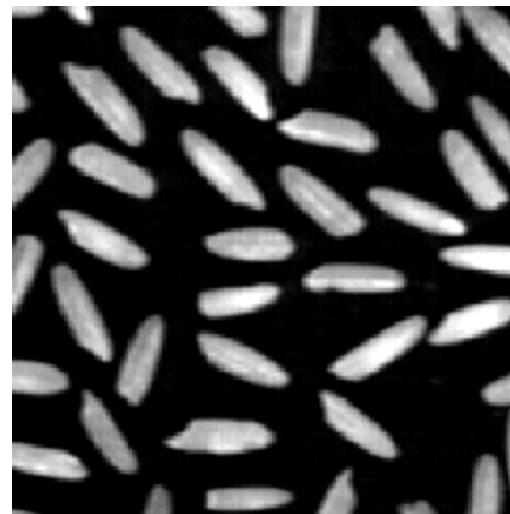


- İmge kontrastlarının ayarını yapmak için imadjust fonksiyonu kullanılır
- $j = \text{imadjust}(I, [\text{low}; \text{high}], [\text{low_o}; \text{high_o}])$
- İmgeyi yeni low_o ve high_o değerlerine getirir

rice.tif - arkaplan



rice.tif - arkaplan : Adjusted



```
I3 = imadjust(I2, stretchlim(I2), [0 1]);
figure, imshow(I3)
title('rice.tif - arkaplan : Adjusted')
```

İmgeye Threshold Uygulama

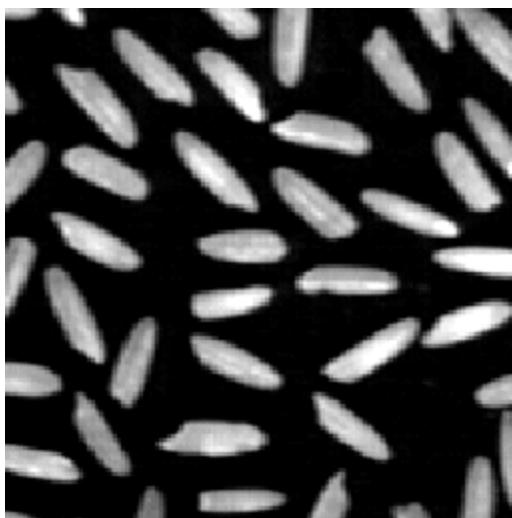


- Bir imgeyi binary imge haline getirmek için kullanılır:
`level = graythresh(I3);
bw = im2bw(I3, level);
figure, imshow(bw)`
- `graythresh`: Otsu metodu ile imge için global bir threshold belirlenir
- `im2bw` ile siyah beyaz imge elde edilir. Sonuçta logical (0 ve 1 değerlerine sahip) bir imge elde edilir.

İmgeye Threshold Uygulama



rice.tif - arkaplan : Adjusted



Siyah Beyaz İmge



Birbirine Bağlı Elementlerin Etiketlenmesi

- $[L, \text{num}] = \text{bwlabel}(\text{bw}, N);$
 - N: 4 ya da 8 olabilir
 - 4 ya da 8 komşuluğ göz önünde bulundurur
 - Default değer=8
- 2D siyah beyaz imgede birbirine bağlı olan elementler etiketlenir
- Dönüş L bw ile aynı boyuttadır.
- Num ise toplamda bulunan bağlı elements sayısını gösteriri
- Her bir bağlı elemana başka numara verilir.
 - L içinde bulunacak olan max sayı num içindeki değerdir.



Birbirine Bağlı Elementlerin Etiketlenmesi



```
[labeled, numObjects] = bwlabel(bw, 4);  
numObjects %42 değerine sahiptir  
max(labeled(:)) %42 değerine sahiptir
```

bwconncomp fonksiyonu da benzer iş yapar.

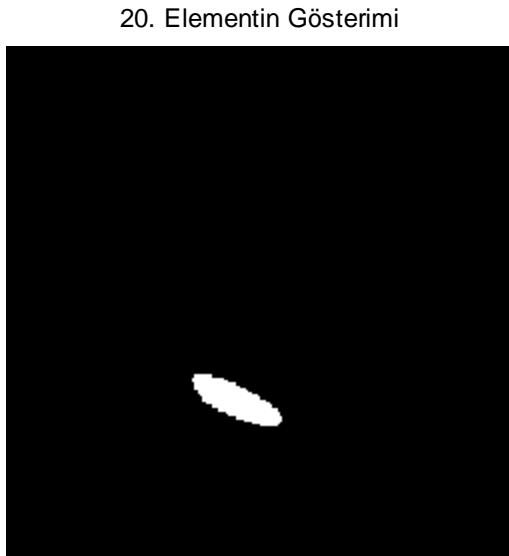
```
cc = bwconncomp(bw, 4)  
cc =  
    Connectivity: 4  
    ImageSize: [256 256]  
    NumObjects: 42  
    PixelIdxList: { 1x42 cell}
```

Birbirine Bağlı Elementlerin Etiketlenmesi



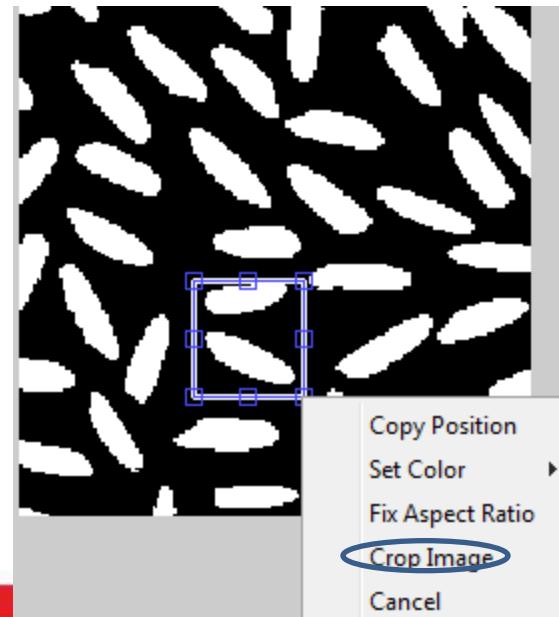
Tek bir bağlı elementi göstermek isterseniz:

```
grain = false(size(bw));
grain(cc.PixelIdxList{0}) = true;
figure, imshow(grain);
title('20. Elementin Gösterimi')
```



İnteraktif seçim:

```
grain = imcrop(labeled);
imshow(grain)
title('Cropped Image')
```



Nesne Özellikleri

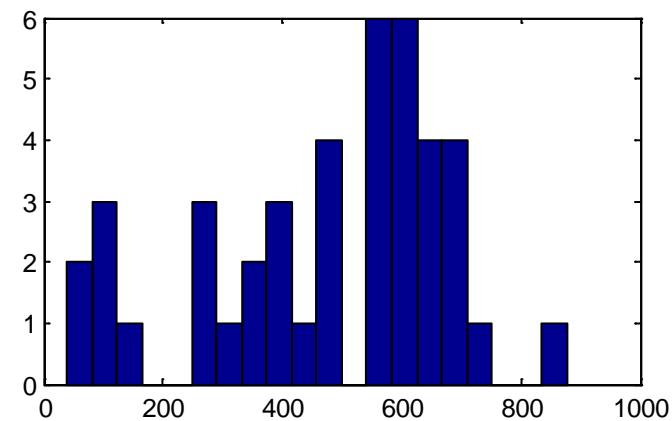


- Nesnelerin özellikleri için
 - stats = regionprops(bw,properties)
 - graindata = regionprops(labeled, 'basic');
 - graindata =
 - 42x1 struct array with fields:
 - Area
 - Centroid
 - BoundingBox
 - allgrains = [graindata.Area];
% Tüm nesnelerin alanlarının tek vektör içinde gösterimi

İstatistiksel özellikler



- allgrains değişkeni içinde tüm nesnelerin alanları tutuluyordu
- Buradan istatistiksel özellikler alınabilir
 - max,min,mean vs.
- Alanların dağılımına bakmak için:
 - hist(allgrains, 20)



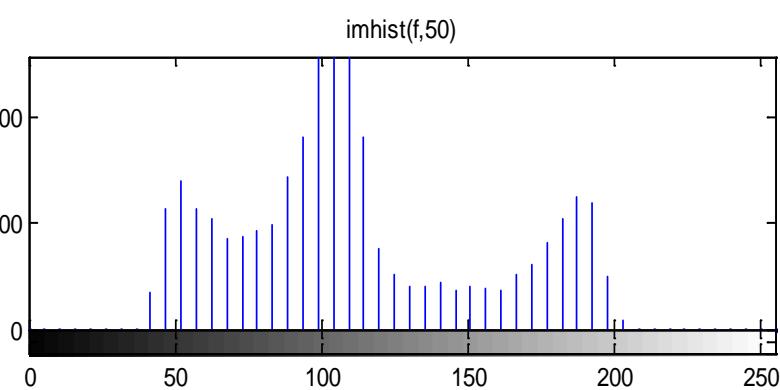
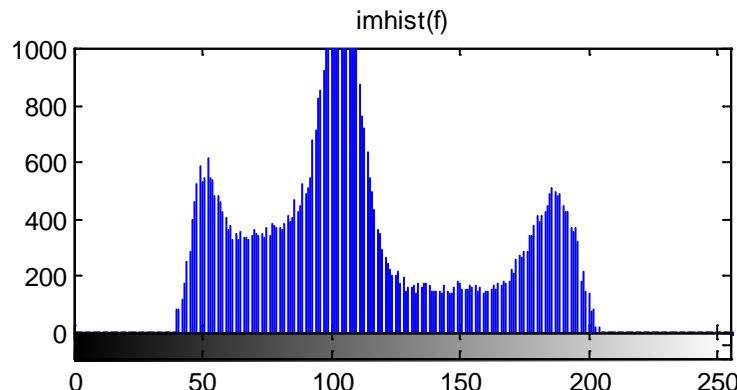
Histogram



```
f=imread('rice.png');
```

```
h=imhist(f);%f imgesinin her renkte kaç değer olduğu bilgisini verir
```

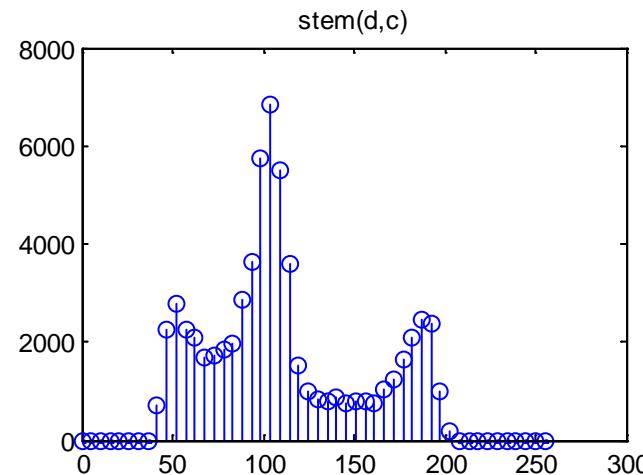
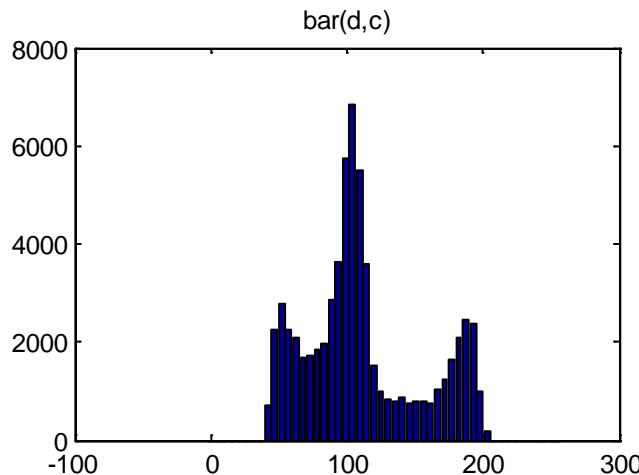
256 renk varsa h 1x256'lık vektördür
Dönüş değeri olmazsa çizim yapılır.



Histogram



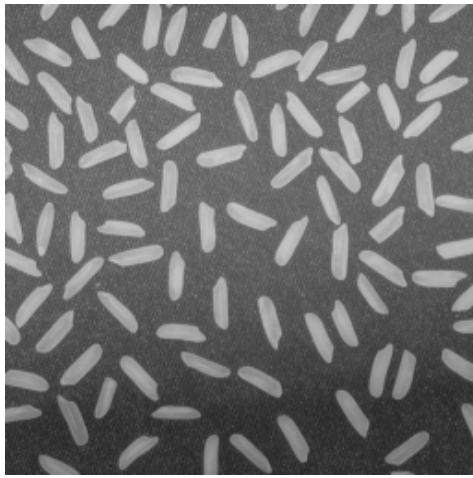
- `[c,d]=imhist(f,50);`
 - c: f imgesinde her aralıkta kaç adet değer var
 - d: Aralıkları gösterir
 - Burada aralıklar-> 0: (255/49):255 olacak şekildedir
- Sonuçlar `bar(c)`, `stem(c)` kullanılarak gösterilebilir



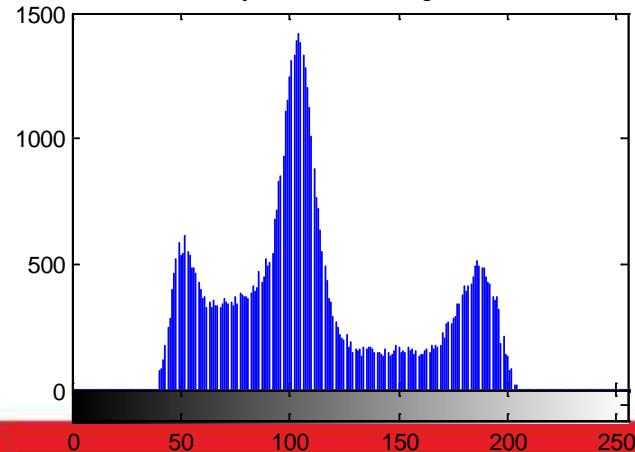
Histogram Eşitleme



Orjinal Resim

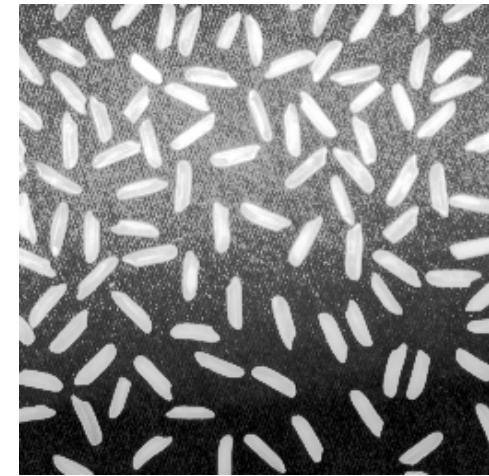


Orjinal Resim Histogramı

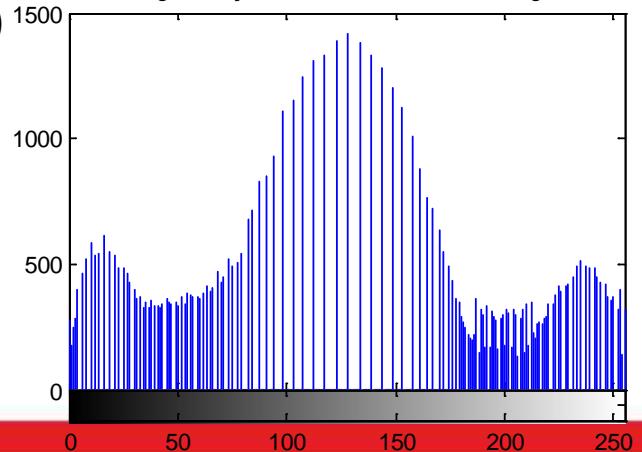


```
f=imread('rice.png');
figure,imshow(f);title('Orjinal
Resim')
figure;
imhist(f);title('Orjinal Histogram')
ylim('auto');
g=histeq(f,256);
figure,imshow(g);
title('Eşitleme Sonrası Resim')
figure;imhist(g);
title('Eşitleme Sonrası Histogram ')
ylim('auto');
```

Histogram Eşitleme Sonrası Resim



Histogram Eşitleme Sonrası Resim Histogramı



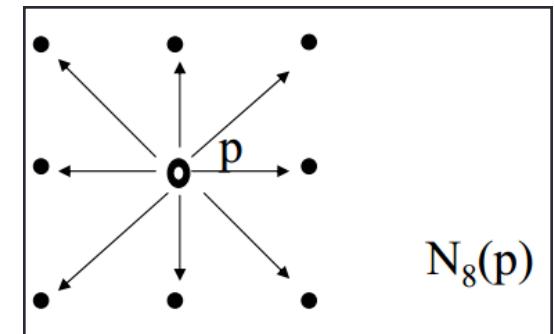
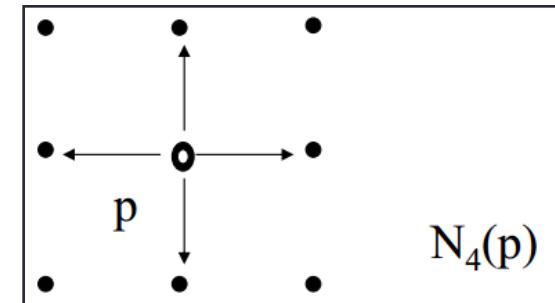
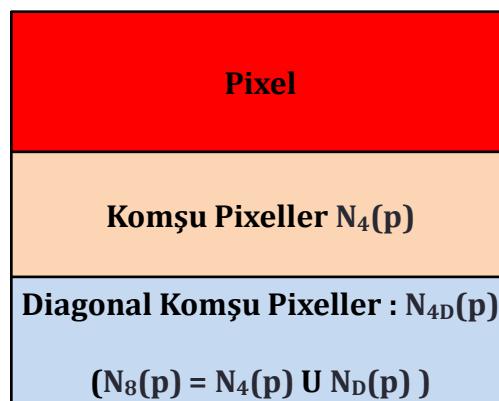
SAYISAL GÖRÜNTÜ İŞLEME

Pixeller Arasındaki Temel İlişkiler

Komşuluk

- (x,y) noktasındaki pixel p , 4 yatay ve dikey komşuya sahiptir
 - $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$
 - Bu pixeller p 'nin 4 komşusu olarak bilinir ve $N_4(p)$ ile gösterilir
 - Her komşu bir birim uzaklıktadır.
 - Köşegenlerdeki komşular $N_8(p)$ komşulardır

$(x-1,y-1)$	(x,y)	$(x+1,y-1)$
$(x,y-1)$	(x,y)	$(x,y+1)$
$(x+1,y-1)$	$(x+1,y)$	$(x+1,y+1)$



Adjacency, Connectivity, Regions, and Boundaries

- Adjacency (Bitişiklik)
 - Connectivity (Bağlılık)
 - Regions (Bölgeler)
 - Boundaries (Sınırlar)
-
- ***Pixeller arası bağlılık*** bölge ve sınır gibi temel sayısal imge kavramlarını kolaylaştırır
 - İki pixelin *bağlı* olup olmadığını anlamak için
 - İki pixel komşu olmalıdır
 - Grilik seviyeleri belli bir kriteri sağlamalıdır.
 - Örneğin binary bir imgede iki pixel komşu olabilir ancak ikisinin de değeri birbirine eşitse (1-1 veya 0-0) bağlı oldukları söylenir.

Adjacency (Bitişiklik)

- V 'nin bitişikliği test etmek için tanımlanmış değerler setini simgelediğini varsayalım
 - Örneğin, Binary imge için $V=\{1\}$ olabilir
 - Gri-seviyeli imge için 0-255 arası herhangi bir değeri içerebilir, ya da V bu 256 değerin herhangi bir alt kümesi olabilir
- 3 çeşit bitişiklik tanımı vardır:
 - **4-adjacency:** V kümesine ait değerlere sahip iki pixel, p ve q , eğer $q \in N_4(p)$ kümesi içindeyse 4-bitmiş kabul edilir
 - **8-adjacency:** V kümesine ait değerlere sahip iki pixel, p ve q , eğer $q \in N_8(p)$ kümesi içindeyse 8-bitmiş kabul edilir
 - **m -adjacency :**(mixed adjacency): V kümesine ait değerler sahip iki pixel, p ve q , aşağıdaki şartları taşırsa m -bitmiş kabul edilir
 - $q \in N_4(p)$ içinde ya da
 - $q \in N_D(p)$ içinde ve $[N_4(p) \cap N_4(q)]$ kümesinin içerisindeki değerlere sahip pixel yok]

Adjacency (Bitişiklik)

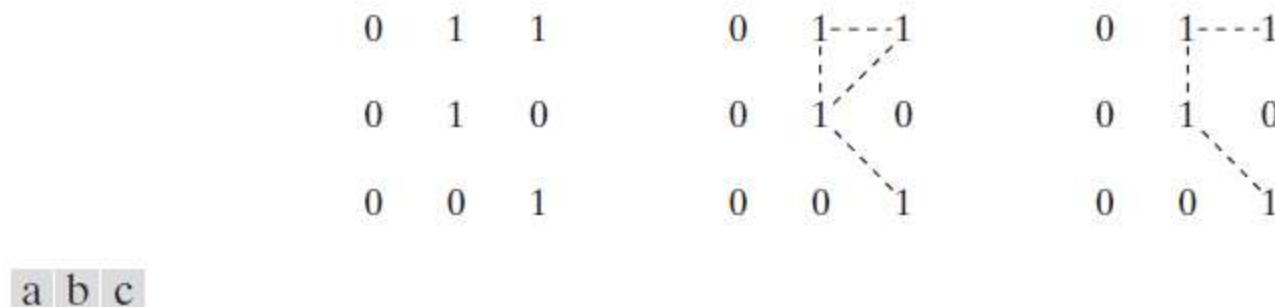
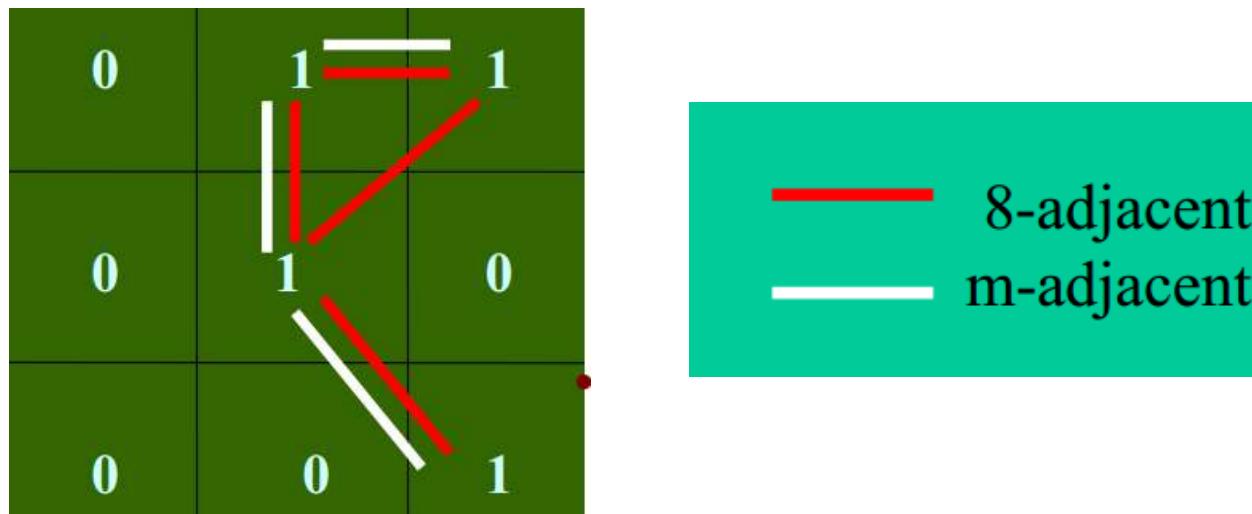


FIGURE 2.26 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) m -adjacency.

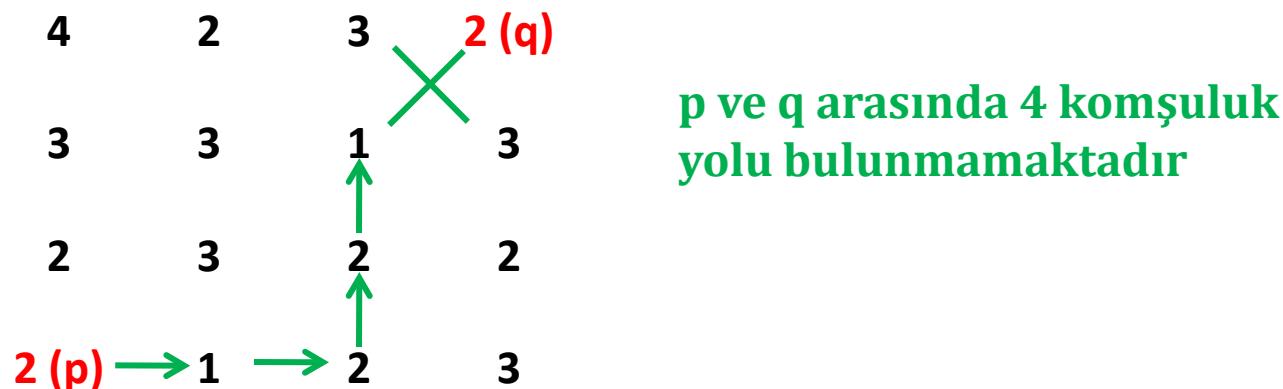


Sayısal Yol (Path)

- Koordinatları (x,y) olan pixel p'den koordinatları (s,t) olan pixel q'ya yol koordinatları aşağıda verilen birbirinden ayrık pixeller dizisidir:
 - $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ ve aşağıdakiler sağlanır:
 - $(x_0, y_0) = (x, y)$ $(x_n, y_n) = (s, t)$
 - (x_i, y_i) (x_{i-1}, y_{i-1}) $0 \leq i \leq n$ bitişik pixellerdir
 - n: yolun uzunluğudur
 - $(x_0, y_0) = (x_n, y_n)$ ise Kapalı yol (Closed Path) adını alır
 - Bitişiklik kavramı 3 tane tanımdan herhangi biri olabilir

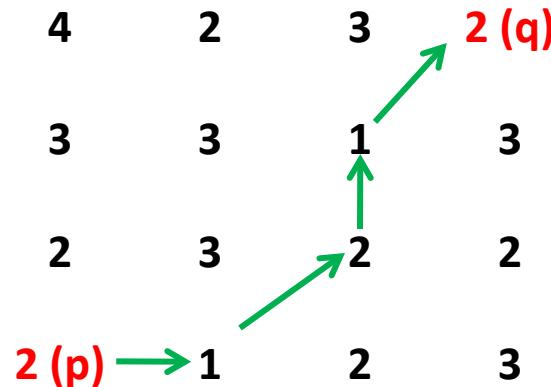
Sayısal Yol (Path)

- Örnek: Aşağıdaki imge parçasını düşünelim. Bu parça için p ve q arasındaki enkısa-4, enkısa-8, ve enkısa-m yol uzunluklarını bulunuz. $V=\{1,2\}$



Sayısal Yol (Path)

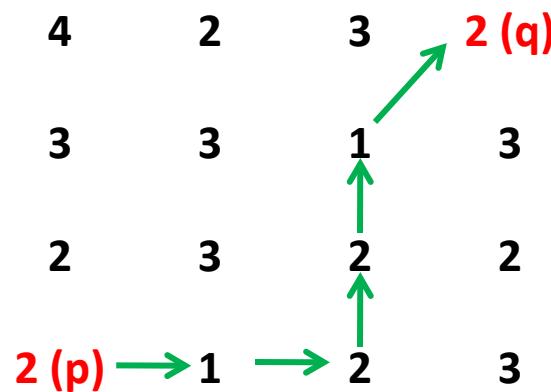
- Örnek: Aşağıdaki imge parçasını düşünelim. Bu parça için p ve q arasındaki enkisa-4, enkisa-8, ve enkisa-m yol uzunluklarını bulunuz. $V=\{1,2\}$



p ve q arasında kurulan bu
yolun uzunluğu 4 olduğu için
bu yol en kısa-8'dir.

Sayısal Yol (Path)

- Örnek: Aşağıdaki imge parçasını düşünelim. Bu parça için p ve q arasındaki enkisa-4, enkisa-8, ve enkisa-m yol uzunluklarını bulunuz. $V=\{1,2\}$



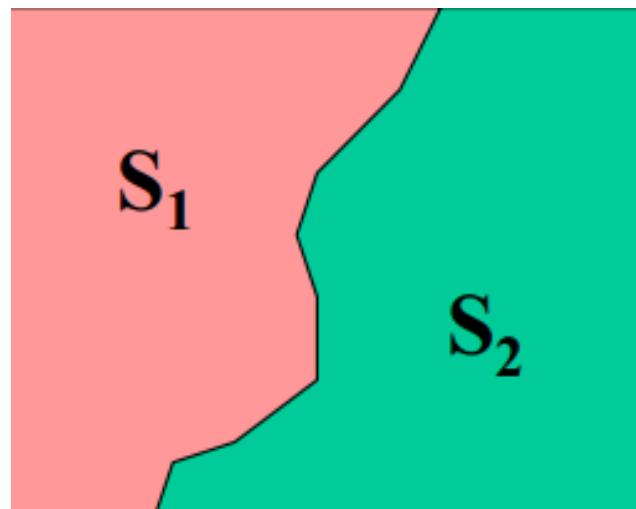
p ve q arasında enkisa-m yol
uzunluğu 5'tir.

Adjacency (Bitişiklik)/Connectvity (Bağlılık)

- S bir imgenin alt kümesi olsun
- Pixel p ile pixel q arasında S 'nin bütün pixellerini içeren bir bir yol bulunabiliyorsa bunlar S içinde birbirine bağlıdır.
- S içinde her pixel için, S içinde bu pixele bağlı olan pixeller seti S 'nin bağlı elemanları (***connected components***) olarak bilinir
- Eğer S sadece bir tane connected component'e sahipse ***connected set*** olarak adlandırılır

Adjacency (Bitişiklik)/Connectvity (Bağlılık)

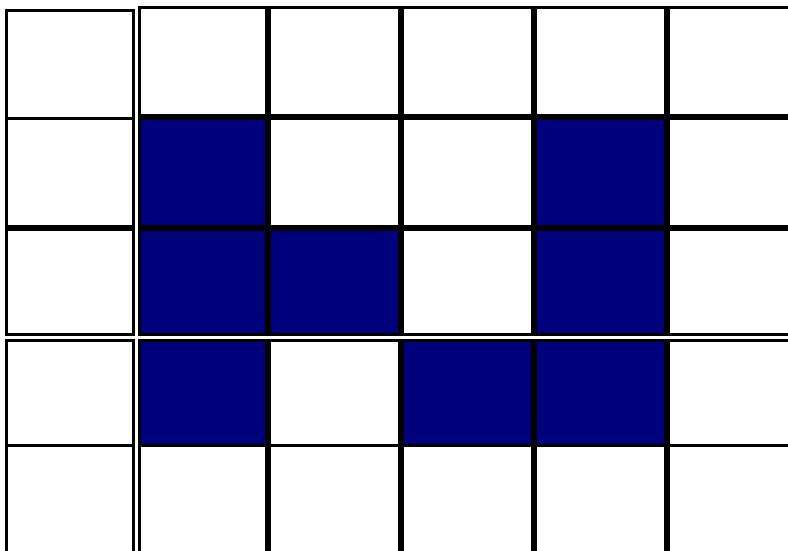
- S1 ve S2 alt imgeleri, eğer eğer S1'deki bazı pixeller S2deki bazı pixellerle bitişik ise bu iki alt imge bitişiktir.



Region

- R bir imgenin alt kümesi olsun
- R connected set ise imgenin bir bölgesi (region) olarak adlandırılır
- Bir bölgenin sınırı (boundary, border ya da contour) bu R içerisinde olan ve en az bir komşusu R dışında olan pixeller setidir.
- Eğer R tüm imgenin (rectangular set of pixels) sınır ilk ve son satır ve sütunlardan oluşur.

Region



4'lü komşuluk düşünülürse 2 bölge
8'li komşuluk düşünülürse 1 bölge

Uzaklık Ölçüleri (Distance Measures)

- p, q, ve z, with koordinatları sırasıyla (x, y), (s, t), ve (v, w), olan pixeller için uzaklık fonksiyonu , D aşağıdaki koşulları sağlar:
 - $D(p, q) \geq 0$ ($D(p, q)=0$ iff $p=q$)
 - $D(p, q)=D(q, p)$, ve
 - $D(p, z) \leq D(p, q)+D(q, z)$.
- p ve q arasında *Euclidean distance*
 - $D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$
- p ve q arasında D_4 uzaklık (*city-block distance*)
 - $D_4(p, q) = |x - s| + |y - t|$
- p ve q arasında D_8 uzaklık (*chessboard distance*)
 - $D_8(p, q) = \max(|x - s|, |y - t|)$

Uzaklık Ölçüleri (Distance Measures)

D₄ uzaklıkları

D₄ = 1 (x,y)'nin 4 komşularıdır

			2	
2	2	1	2	
2	1	0	1	2
2	1	2		
			2	

D₈ uzaklıkları (chessboard distance)

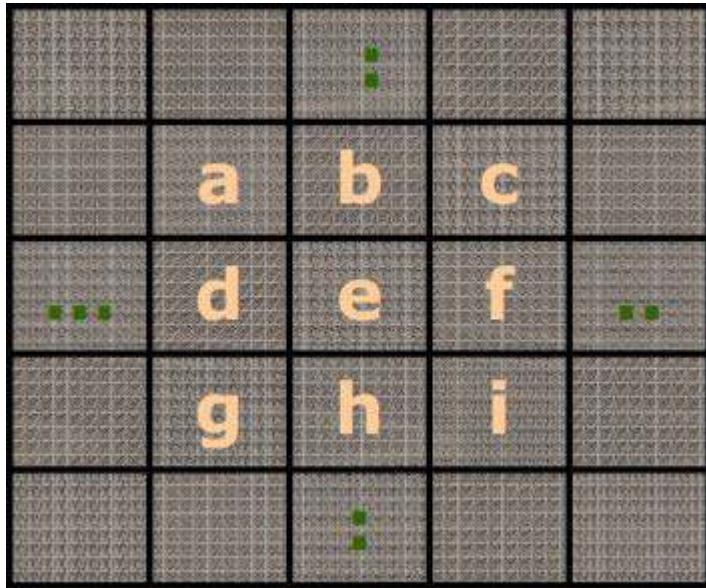
D₈ = 1 (x,y)'nin 8- komşularıdır

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Pixel Bazında İmge İşlemleri

- Arithmetic/Logic Operations
 - Toplama: $p + q$
 - Çıkarma: $p - q$
 - Çarpma: $p * q$
 - Bölme: p/q
 - AND: $p \text{ AND } q$
 - OR : $p \text{ OR } q$
 - Complement: $\text{NOT}(q)$
- Komşuluk Bazlı aritmetik İşlemler

Komşuluk Bazlı Aritmetik İşlemler



w₁	w₂	w₃
w₄	w₅	w₆
w₇	w₈	w₉

$$\begin{aligned} p &= (w_1\mathbf{a} + w_2\mathbf{b} + w_3\mathbf{c} + w_4\mathbf{d} + w_5\mathbf{e} + w_6\mathbf{f} + w_7\mathbf{g} + w_8\mathbf{h} + w_9\mathbf{i}) \\ &= \sum w_i f_i \end{aligned}$$

Arithmetric/Logic İşlemler

- Komşuluk göz önünde bulundurularak gerçekleştirilen işlemler:
 - Smoothing / averaging
 - Noise removal / filtering
 - Edge detection
 - Contrast enhancement
- Önemli noktalar:
 - w_i seçimi
 - Pencere genişliğinin, N, seçimi
 - Sınırarda hesaplamalar:
 - Sınırarda hesap yapma
 - İmgeyi 0'lar ekleyerek genişlet (sınır noktalarında zero padding)
 - İmgeyi periyodik olarak varsayıarak sınırlara ekleme yap

SAYISAL GÖRÜNTÜ İŞLEME

Hafta3: Uzaysal Domainde Görüntü iyileştirme

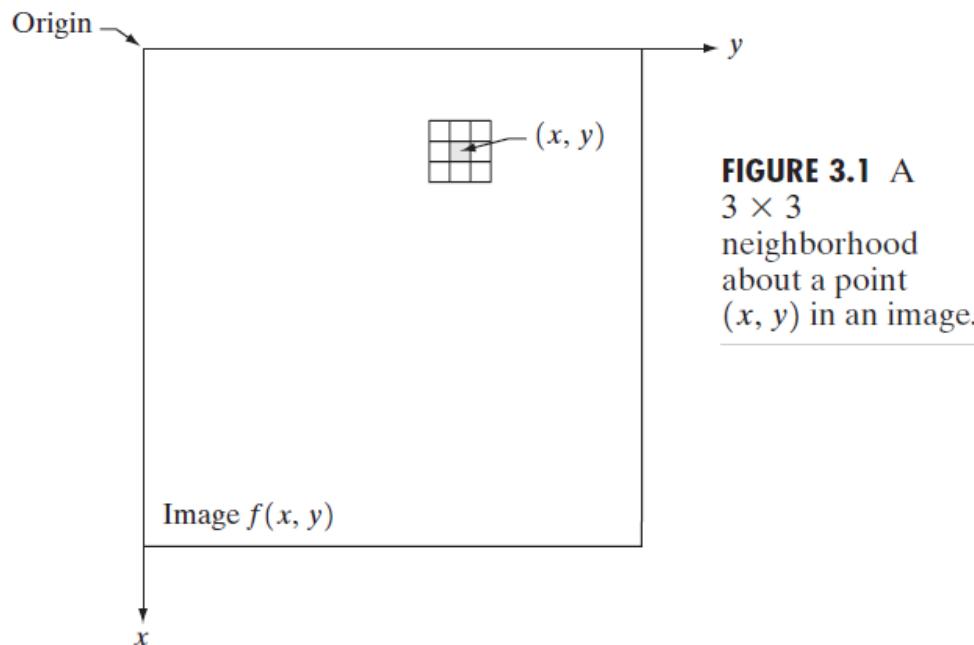
Görüntü İyileştirme

- Amaç
 - İmgeler, iyileştirilerek elde edilen sonucun özel bir uygulama için daha uygun olması için işlenir
- İki domainde iyileştirme metotları bulunur:
 - Uzaysal (Spatial) Domainde iyileştirme metotları
 - Pixellerin direk manipülasyonu
 - Frekans Domain Domainde iyileştirme metotları
 - İmgenini Fourier transformu üzerinde yapılan işlemler

Spatial Domainde Görüntü İyileştirme

- İşlem şu şekilde simgelenenecek:

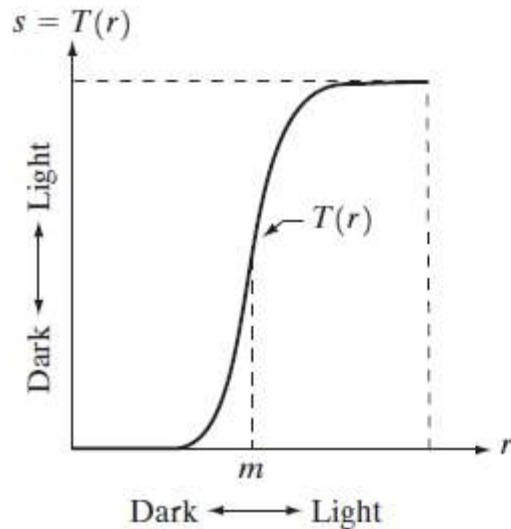
- $g(x,y) = T[f(x,y)]$
 - $f(x,y)$: Input image
 - $g(x,y)$: İşlenmiş (Output) imge
 - T : f üzerinde belli bir (x,y) 'nin belli komşuluğunda tanımlı bir fonksiyon



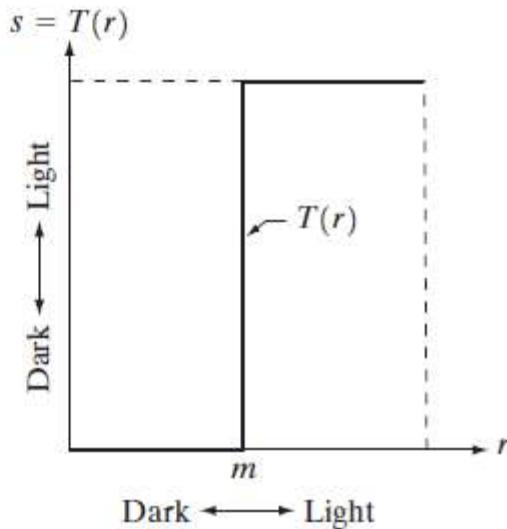
Komşuluk 1 olarak seçilirse işlem her pixel üzerinde yapılır.
En basit fonksiyon bu şekilde elde edilir. (**gray-level (intensity or mapping) transformation function**)

$$s = T(r)$$

Kontrast İyileştirme



Contrast streching :
m'den yüksek değerler
açılırken, diğerleri
koyulaştırılır.



thresholding function:
Binary İmge oluşur.

a b

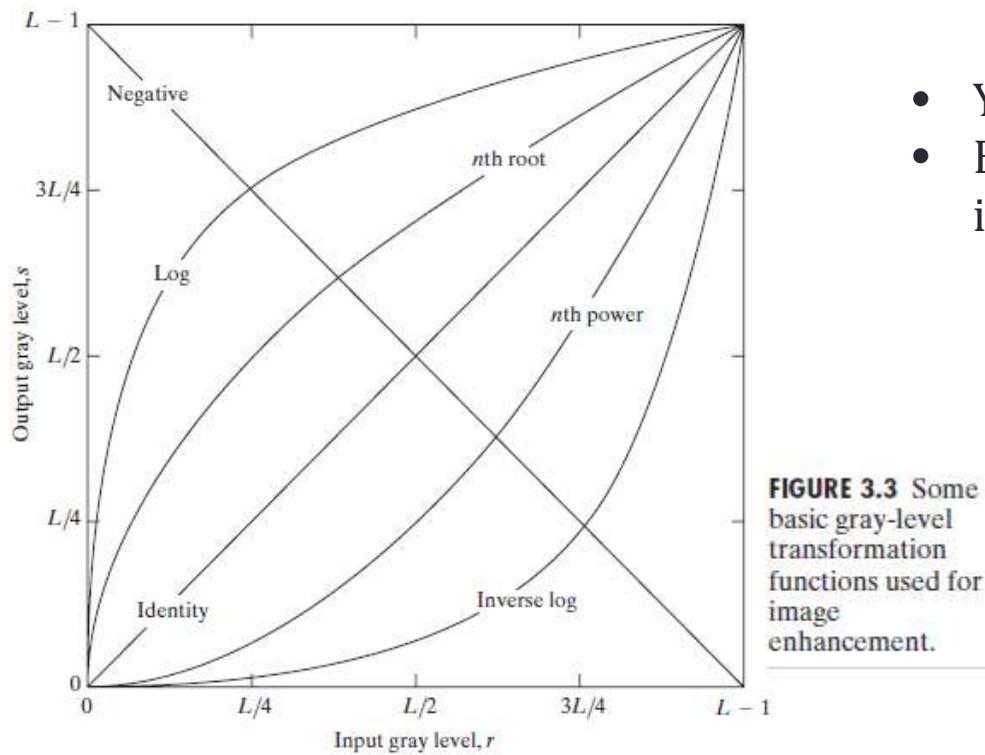
FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

Gri Seviyeli İmgeler İçin Temel Dönüşümler

- Image Negatives
- Log Transformations
- Power-Law Transformations
- Piecewise-Linear Transformation Functions

Image Negatives

- $[0, L-1]$ aralığındaki bir imgenin negatifi için şekilde gösterilen fonksiyon kullanılır:



- Yoğunluk seviyeleri tersine çevrilir
- Beyaz ya da siyah ayrıntıların iyileştirilmesi gereğinde uygundur

$$s = L - 1 - r.$$

Image Negatives

Original Image



Negative Image

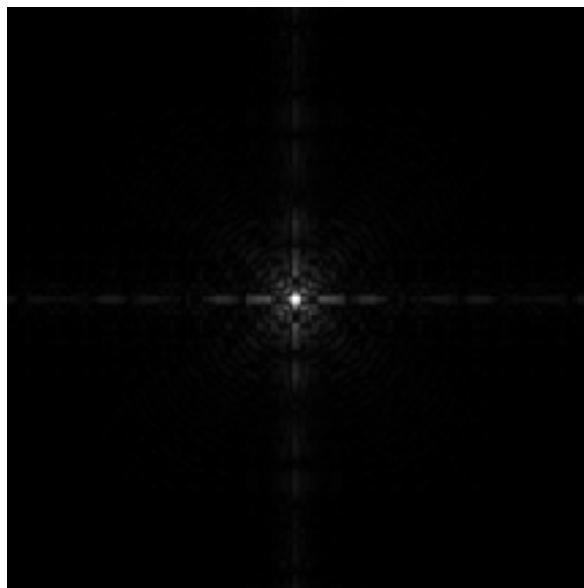


Log Transformations

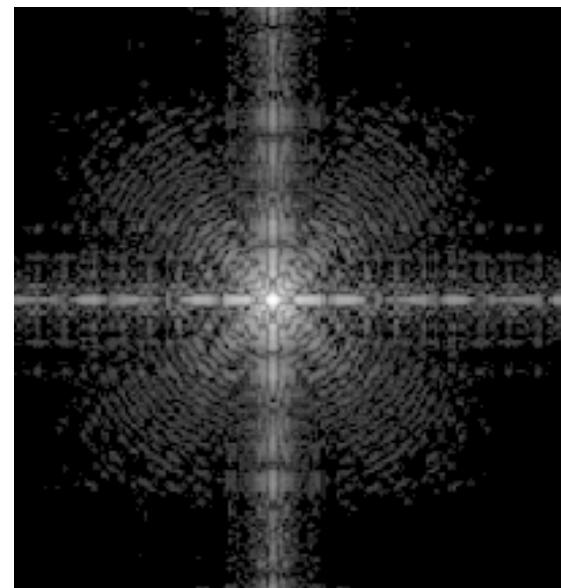
- Genel formu:
 - $s = c \log(1 + r)$
 - c : sabit, $r \geq 0$
- Bu dönüşüm az aralıklı düşük gri seviyeli değerleri (siyaha yakın değerler) yüksek aralıklı hale getirir.
 - Yüksek seviyeli (beyaza yakın) değerler için tersi geçerli
- Koyu renkli pixelleri genişletmek için kullanılabilir
- Açık renklileri genişletmek için ters log dönüşüm kullanılır

Log Transformations

Original Image



Log Transform Sonucu



Power-Law Transformations

- Genel formu:

- $s = cr^\gamma$ ya da input 0 olduğunda $c(r + \varepsilon)^\gamma$
- c ve γ sabit sayılar
- Gamma correction olarak da bilinir

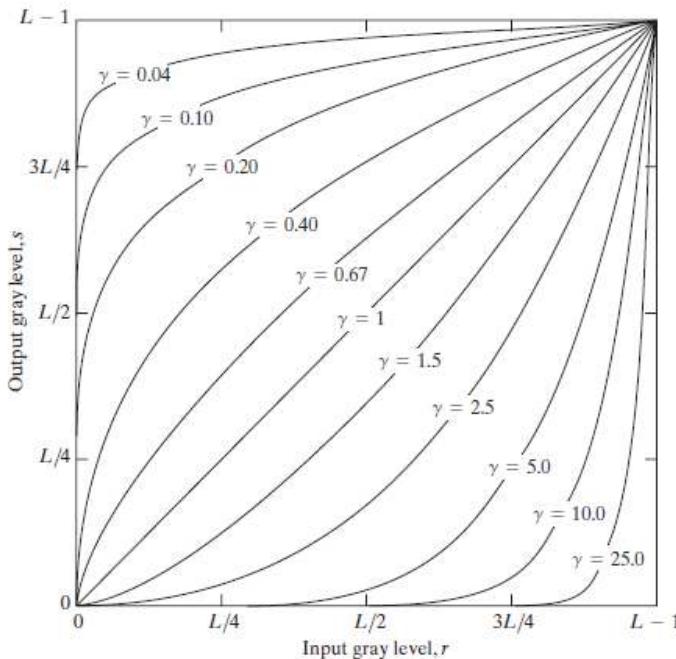


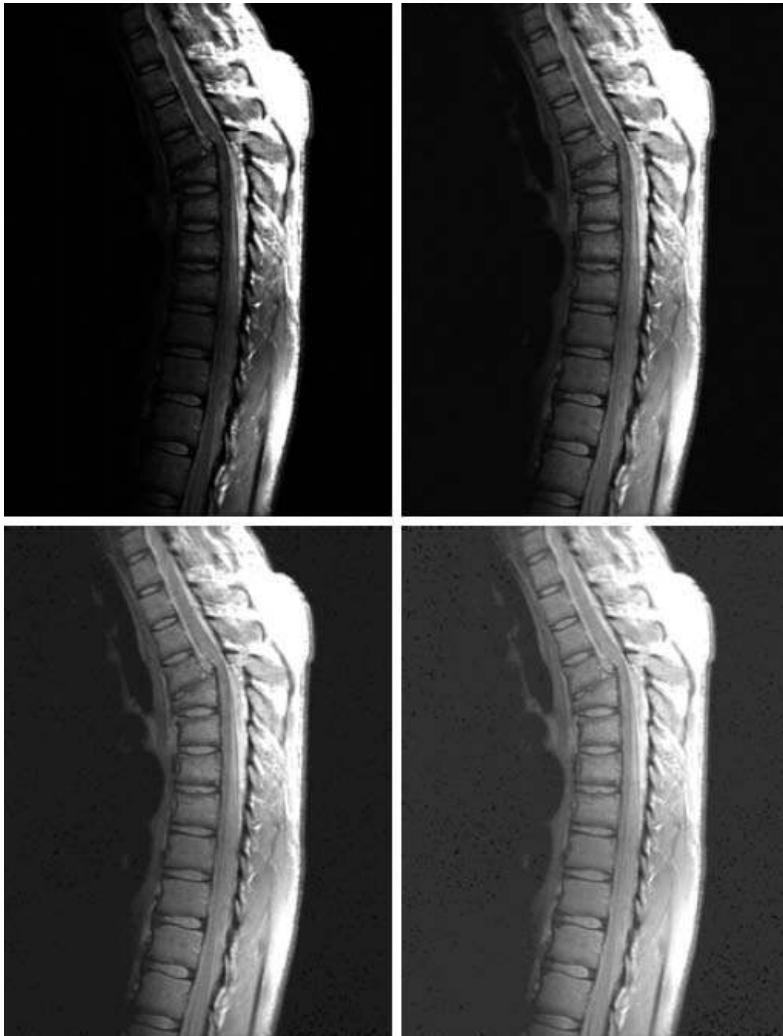
FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases).

Log fonksiyonuna benzer. Ancak burada çok daha farklı curve elde edilebilir.

- Dar pencereli koyu renkli giriş değerlerini daha geniş bir çıkış aralığına haritalar.

$\gamma < 1$ ve $\gamma > 1$ durumları birbirinin tersi eğimler oluşturmaktadır

Power-Law Transformations



a b
c d

FIGURE 3.8
(a) Magnetic resonance (MR) image of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3 , respectively.
(Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Power-Law Transformations

a
b
c
d

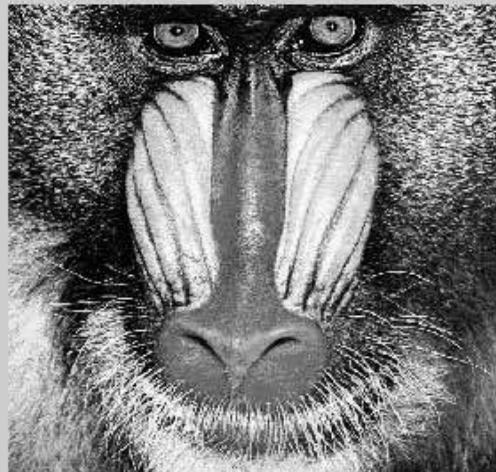
FIGURE 3.9

(a) Aerial image.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0 , respectively.
(Original image for this example courtesy of NASA.)

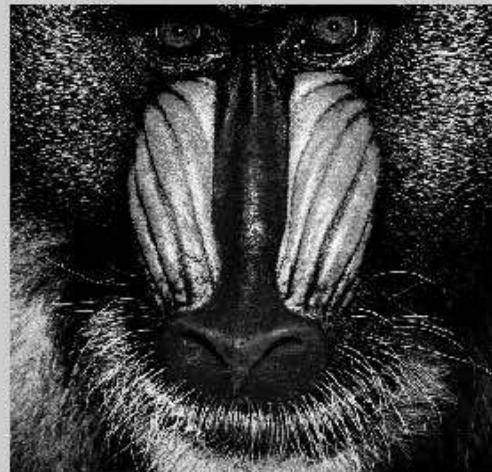


Power-Law Transformations

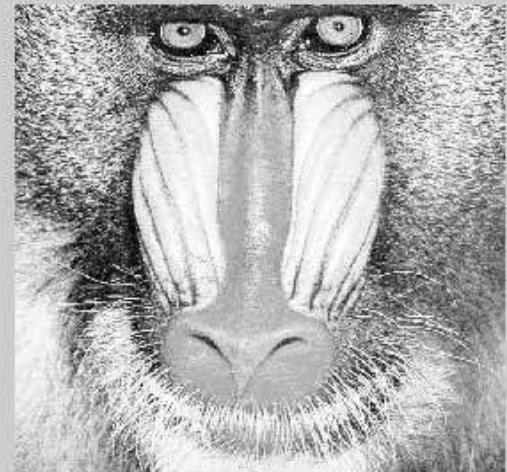
Original Image



Gamma = 3



Gamma = 0.33



Piecewise-Linear Transformation Functions

- Önceki metotları tamamlayıcı bir yaklaşım
- En önemli avantajı
 - Fonksiyon sabit değil
- Dezavataj
 - Çalışırken daha çok bilgi girilmesi gereklidir
- En basit fonksiyon
 - **Contrast stretching**

Contrast Stretching

- En basit piecewise-linear fonksiyonlardandır
- Düşük kontrastlı imgeler nedenleri:
 - Kötü aydınlanma
 - Görüntüleme sensöründeki dinamik aralık eksikliği
 - Görüntü yakalama esnasında lens aparatlarının yanlış kurulması
- Amaç: Gri seviyelerindeki dinamik aralığı artırmak

Contrast Stretching

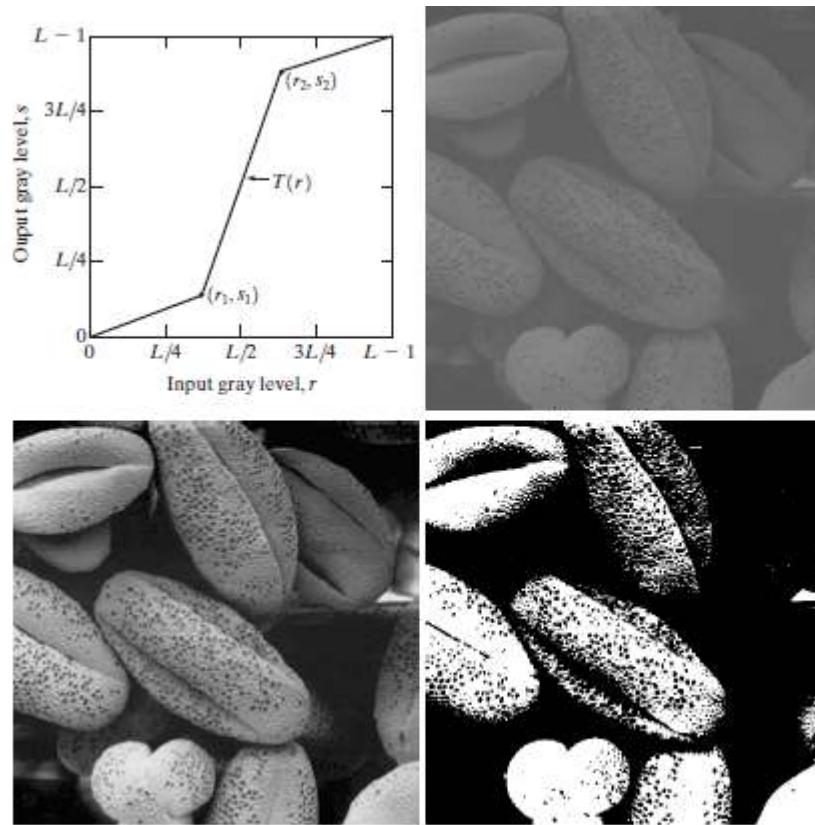
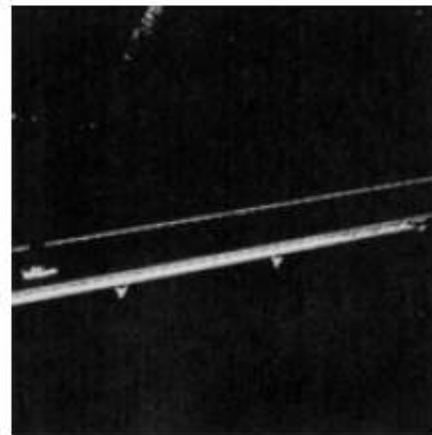
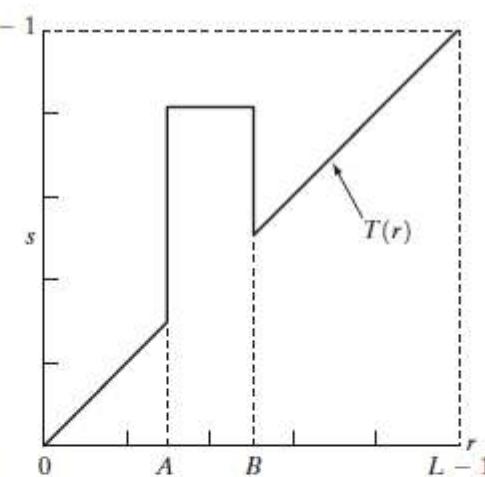
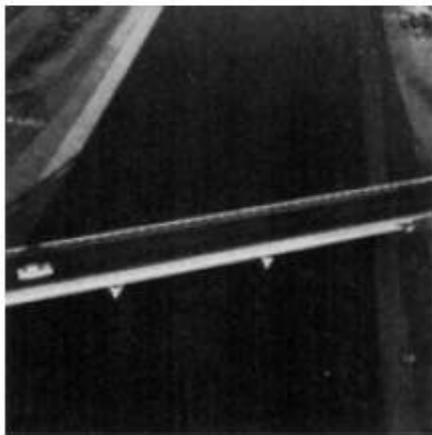
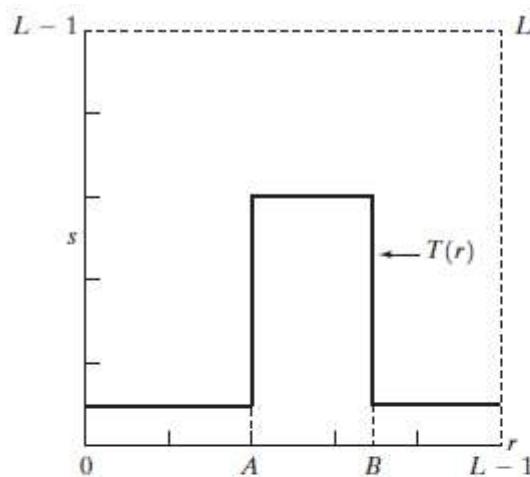


FIGURE 3.10
Contrast
stretching.
(a) Form of
transformation
function.
(b) A
low-contrast
image.
(c) Result
of contrast
stretching.
(d) Result of
thresholding.
(Original image
courtesy of
Dr. Roger Heady,
Research School
of Biological
Sciences,
Australian
National
University,
Canberra,
Australia.)

Gray Level Slicing

- Gri seviyelerinin belli aralıklarına öncelik vermek genelde istenen bir durumdur
- Uygulamalar:
 - Uzaydan görüntülemede su kütlelerinin öne çıkarılması
 - X-ray hatalarının düzeltilmesi
- Farklı yöntemler uygulanabilir. Ancak temelde 2 öntem:
- 1. yöntem istenen aralıktaki gri seviyelerini daha yüksek bir seviyeye çekmektir
 - Bu binary imgé oluşturur
- 2. yöntemde istenen aralıklar aydınlatılır, ancak kalan pixeller (background) ve bunların gri seviyeleri korunur

Gray Level Slicing



a
b
c
d

FIGURE 3.11
(a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level.
(b) This transformation highlights range $[A, B]$ but preserves all other levels.
(c) An image.
(d) Result of using the transformation in (a).

Bit Plane Slicing

- Bütün gri seviye aralıkları ile uğraşmak yerine belli bitlerin imgeye katkısı ön planda tutulur.
- Her pixelin 8 bit ile temsil edildiğini varsayıyalım
 - İmgenin 8 farklı 1 bitlik imgeden oluştuğunu düşün
 - Bu durumda LSB bit plane-0
 - MSB Bit plane-7

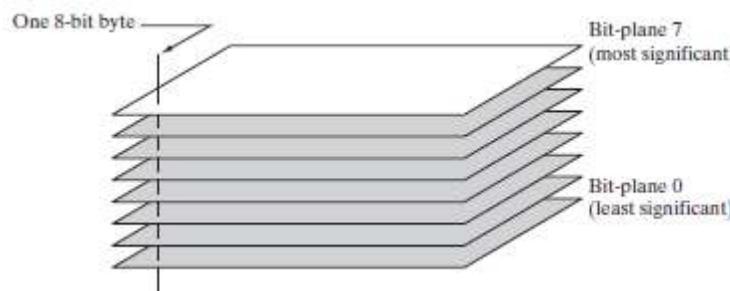


FIGURE 3.12
Bit-plane
representation
of
an 8-bit image.

Bit Plane Slicing

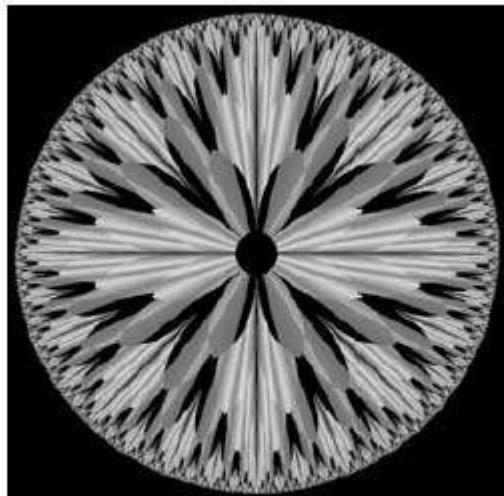


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

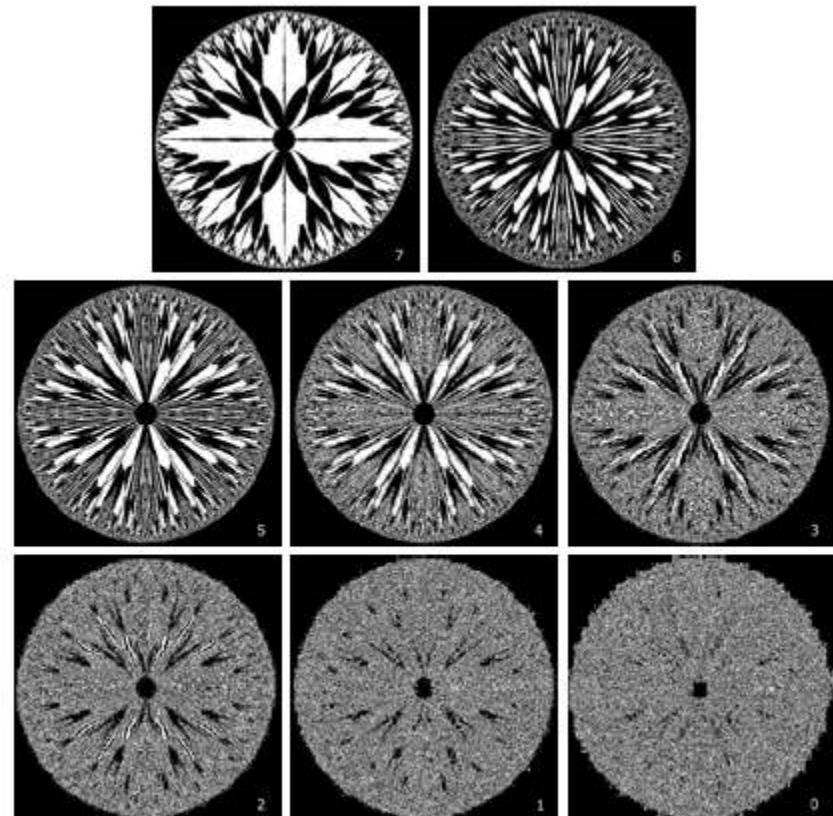


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom right of each image identifies the bit plane.

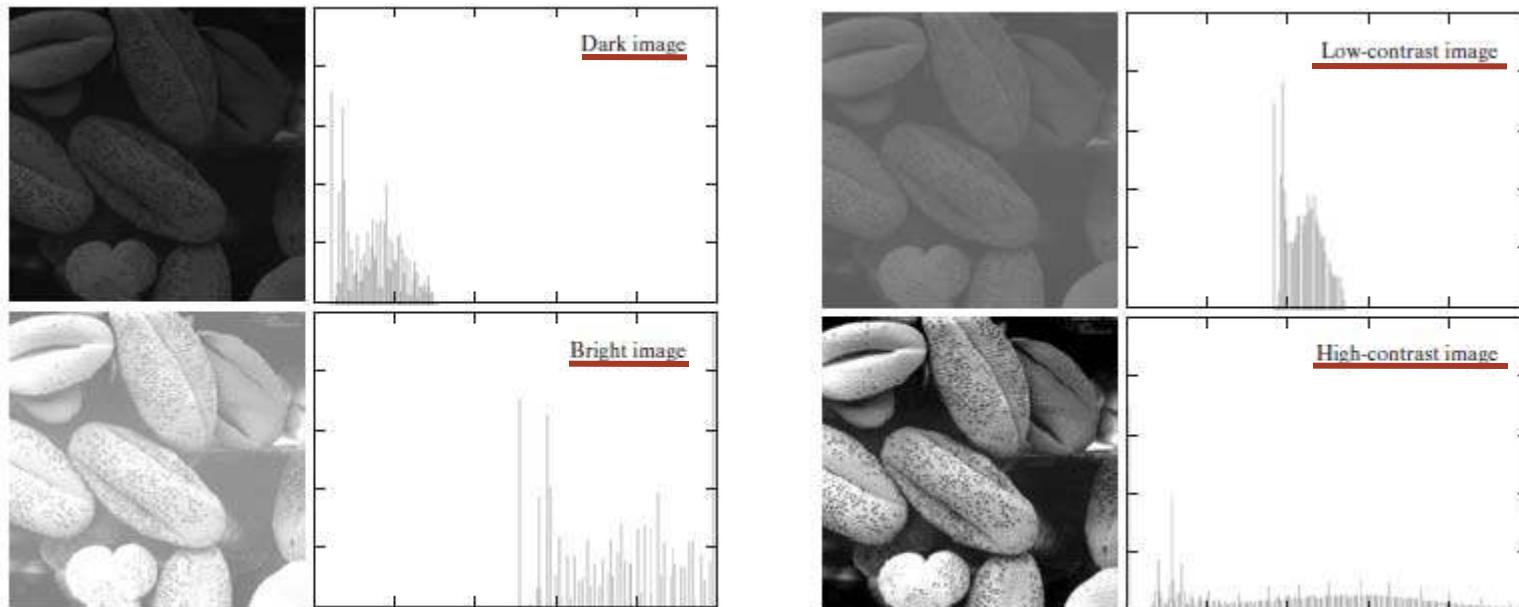
Bit Plane Slicing

- Görsel olarak önemli bilgiler üst sıralardaki bit (özellikle 4-7 arası) düzlemlerindedir.
- Diğer düzlemler daha çok ayrıntıları içeriri
- İmge sıkıştırma için kullanılması yararlıdır
- Bit plane-7'nin imgenin direk eşikleme yöntemi ile bulunmasıdır.
 - Değeri >129 olan pixellerin 1 diğerlerinin 0 olduğu açıktır

Histogram İşleme

- $[0 L-1]$ arasındaki değerlere sahip imgenin histogramı ayrık bir fonksiyondur
 - $h(r_k) = n_k$
 - r_k k. gri seviyesi
 - n_k ise k. gri seviyesine sahip pixel sayısı
- Histogramın normalize edilmesi
 - $p(r_k) = n_k / n$, n: number of pixels in the image
 - $p(r_k)$ sonuç olarak r_k gri seviyesinin ortaya çıkma olasılığıdır
- Histogramlar uzaysal imgé işleme teknikleri için çok önemlidir ve genel işleme tekniklerinde de etkili olarak kullanılır
 - İmge geliştirme
 - İmge bölütleme
 - İmge sıkıştırma

Histogram İşleme



a b

FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Histogram Eşitleme

- r : İyileştirilecek imgenin gri seviyeleri
 - $r \in [0, 1]$ 0-> siyah 1-> beyaz
- pixel değerleri $\in [0, L-1]$
- Dönüşüm
 - $s = T(r)$, $0 \leq r \leq 1$
- Dönüşüm aşağıdaki şartları sağlamalı:
 - $T(r)$ tek değerlidir
 - Böylelikle bir geri dönüşümün varlığı garanti altına alınır
 - $0 \leq r \leq 1$ aralığında monotonik olarak artan bir fonksiyondur
 - Siyadır beyaza yoğunluk artışı korunmuş olur
 - $0 \leq r \leq 1$ için $0 \leq T(r) \leq 1$
 - Çıkış imgesi de giriş seviyeleri ile aynı olur

Histogram Eşitleme

- Ters dönüşüm
 - $r = T^{-1}(s), 0 \leq s \leq 1$

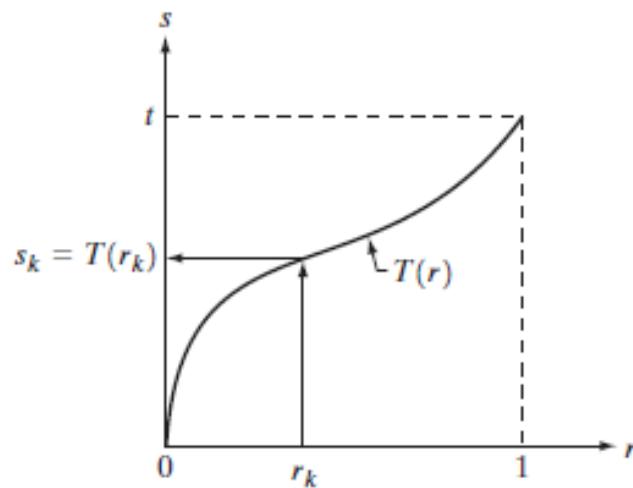


FIGURE 3.16 A gray-level transformation function that is both single valued and monotonically increasing.

Histogram Eşitleme

- Bu dönüşümde gri seviyeleri rastgele değişkenler olarak düşünülebilir
- Rastgele değişkenler için en önemli tanımlayıcı olasılık yoğunluk fonksiyonudur (PDF)
 - $p_r(r)$ ve $p_s(s)$ bu rastgele değişkenlerin PDF'leri olsun

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right|.$$

- Bu durumda

$$s = T(r) = \int_0^r p_r(w) dw$$

- Yukarıdaki eşitlikte sağ taraf r için CDF olarak bilinir

$$\begin{aligned}\frac{ds}{dr} &= \frac{dT(r)}{dr} \\ &= \frac{d}{dr} \left[\int_0^r p_r(w) dw \right] \\ &= p_r(r).\end{aligned}$$

Histogram Eşitleme

- Amaç $p_s(s)$ 'nin düzgün dağılıma sahip olmasınadır
- Sonuç olarak:

$$p_r(r_k) = \frac{n_k}{n} \quad k = 0, 1, 2, \dots, L - 1$$

$$\begin{aligned} s_k &= T(r_k) = \sum_{j=0}^k p_r(r_j) \\ &= \sum_{j=0}^k \frac{n_j}{n} \quad k = 0, 1, 2, \dots, L - 1. \end{aligned}$$

Histogram Eşitleme

- Bu dönüşüm histogram eşitleme ya da histogram doğrusallaştırma olarak bilinir
- Sonuç olarak elde edilen imge daha geniş bir gri seviyesi aralığını içerecektir

$$\begin{aligned}s_k &= T(r_k) = \sum_{j=0}^k p_r(r_j) \\ &= \sum_{j=0}^k \frac{n_j}{n} \quad k = 0, 1, 2, \dots, L - 1.\end{aligned}$$

Histogram Eşitleme

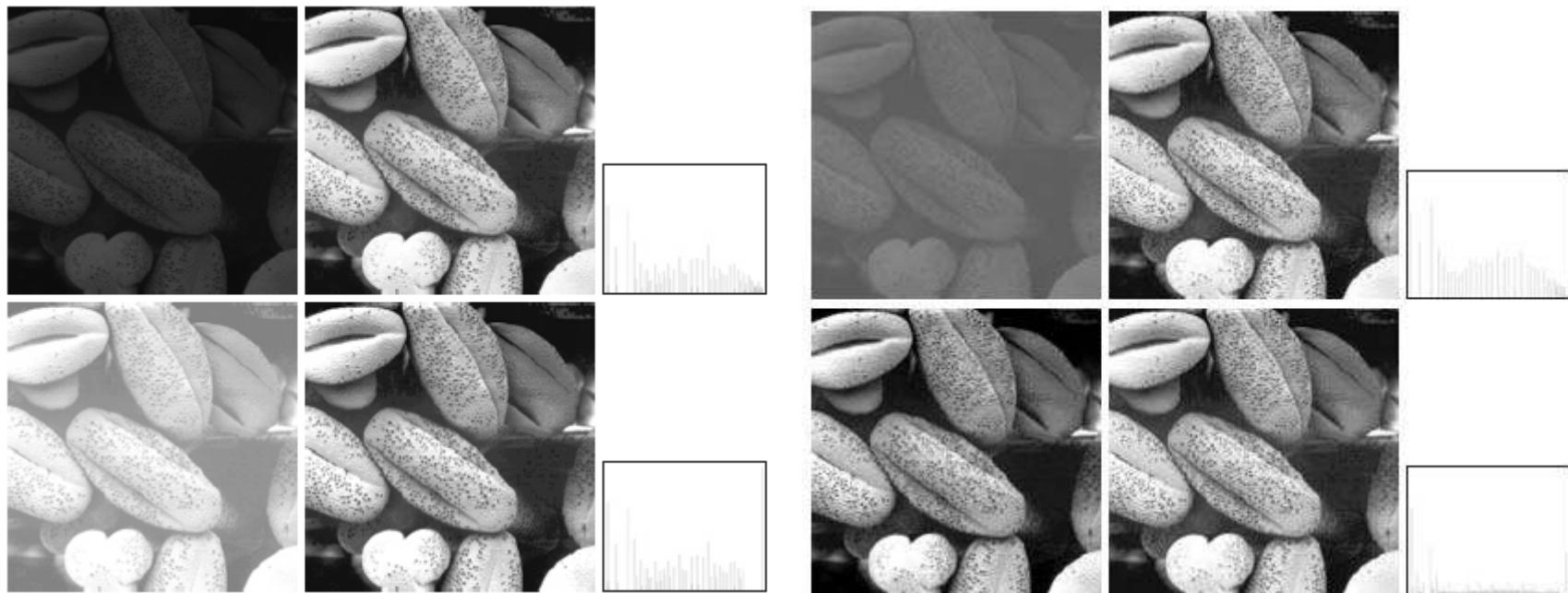
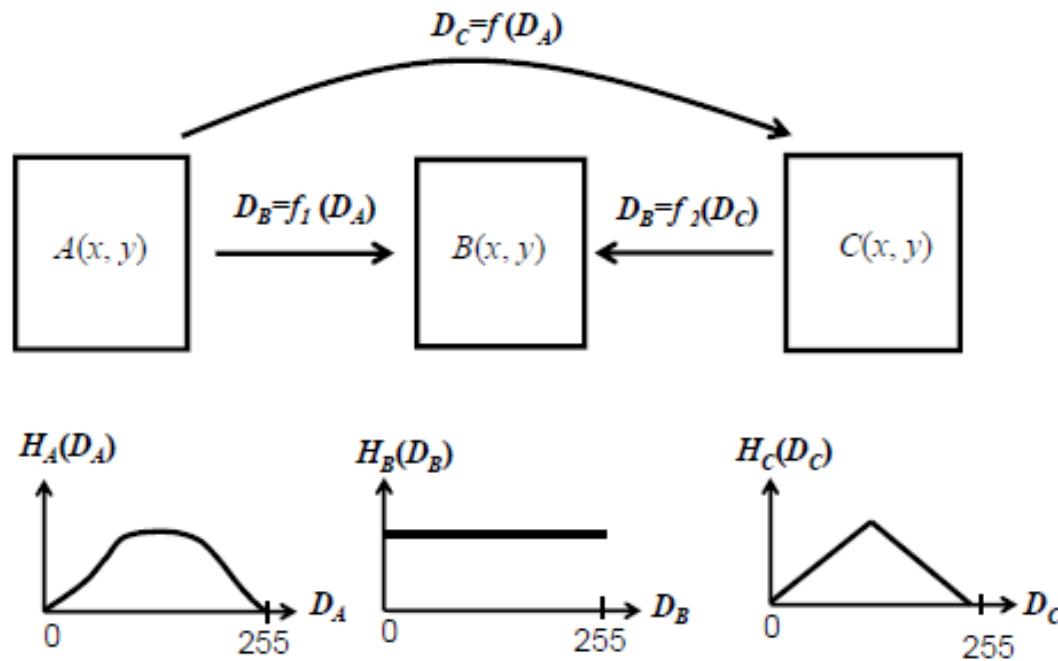


FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

Histogram Matching (Specification)

- Histogram eşitlemede dönüşüm fonksiyonu bellidir
 - Amaç çıkış imgesindeki histogramı düzgün dağılımlı yapmaktır
 - Otomatik düzeltmelerde faydalıdır ve uygulaması çok basittir
- Düzgün dağılıma sahip çıkış histogramı istenmeyen durumlarda özel bir histogram belirtilerek buna ulaşmak istenebilir
 - Histogram Eşleme ya da Histogram Tanımlama olarak bilinir

Histogram Matching (Specification)



Histogram Matching (Specification)

- Uygulanacak adımlar:

- İmgenin histogramını çıkart
- Histogram eşitleme yaparak s_k değerlerini hesapla
- Önceden verilen $p_z(z_i)$ 'yi ve s_k değerlerini kullanarak G dönüşüm fonksiyonunu elde et

$$v_k = G(z_k) = \sum_{i=0}^k p_z(z_i) = s_k \quad k = 0, 1, 2, \dots, L - 1.$$

- Her s_k değeri için z_k değerlerini hesapla. Bunun için aşağıdaki koşulu sağlayan en küçük \hat{z} değerini bul

$$(G(\hat{z}) - s_k) \geq 0 \quad k = 0, 1, 2, \dots, L - 1.$$

- Orijinal imgedeki değeri r_k ile gösterilen her pixel için s_k ve z_k değerlerini hesapla.



MATLAB İLE GÖRÜNTÜ İŞLEME

TEMEL İŞLEMLER

Yoğunluk Dönüşüm Fonksiyonları

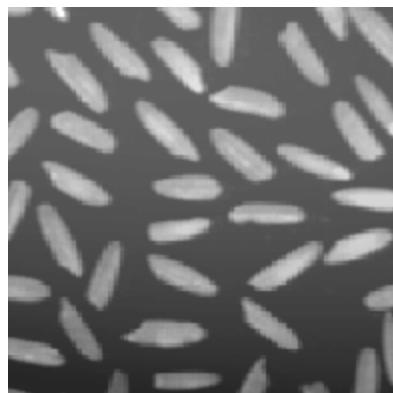


- `stretchlim` : imgenin kontrast ayarını yapmak için gerekli 2 elemanlı vektörü geri yollar
 - düşük ve yüksek gri seviyeleri
 - En alttaki ve en üstteki %1 pixel değerleri
 - Bu değerler `imadjust` fonksiyonunda kullanılarak imgenin kontrasti artırılır

Yoğunluk Dönüşüm Fonksiyonları

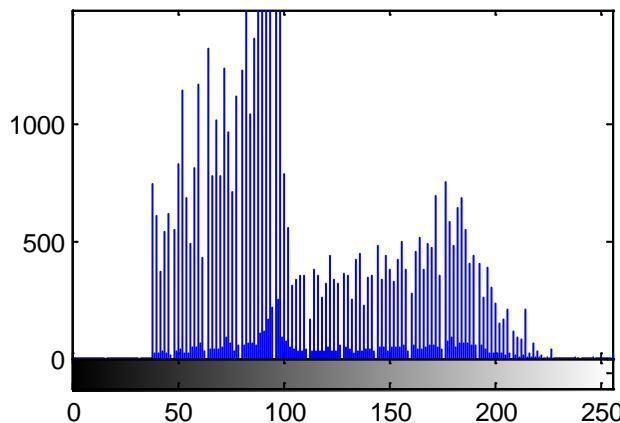


rice.tif - Orjinal imge



```
[rice,map]=imread('rice.tif');
imhist(rice),title('rice.tif - Histogram')
imshow(rice), title('rice.tif - Orjinal imge')
>> stretchlim(rice)
ans =
    0.1490
    0.8157
```

rice.tif - Histogram



```
>> riceSirali=sort(rice(:));length(riceSirali)
ans =
    65536
>>[double(riceSirali(655))/255
double(riceSirali(64881))/255,]
ans =
    0.1490  0.8157
```

Yoğunluk Dönüşüm Fonksiyonları



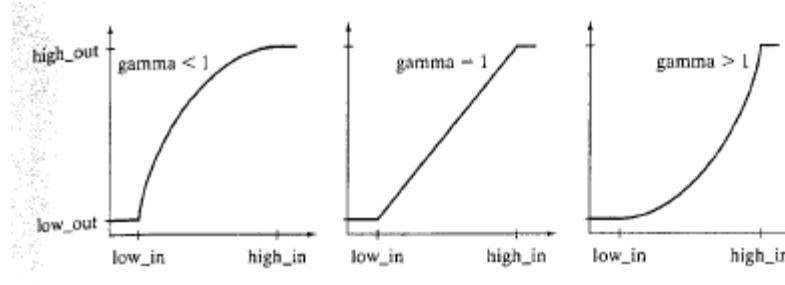
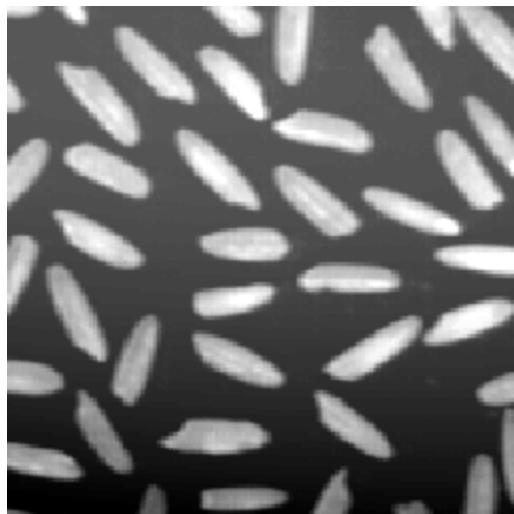
```
g = imadjust(f, [low_in high_in], [low_out high_out], gamma)
```

```
riceAdjusted=imadjust(rice);figure,imshow(riceAdjusted)
```

%imadjust(rice,stretchlim(rice)); ile aynı işlem yapılıyor

```
title('rice.tiff - Kontrast Ayarlı imge')
```

rice.tiff - Kontrast Ayarlı imge

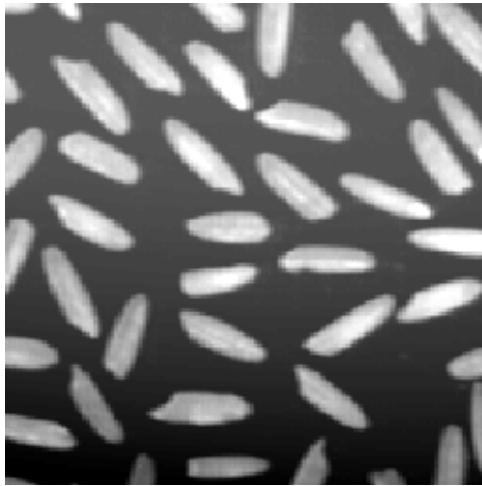


a b c
FIGURE 3.2 The various mappings available in function `imadjust`.

Parameter gamma specifies the shape of the curve that maps the intensity values in f to create g . If gamma is less than 1, the mapping is weighted toward higher (brighter) output values, as Fig. 3.2(a) shows. If gamma is greater than 1, the mapping is weighted toward lower (darker) output values. If it is omitted from the function argument, gamma defaults to 1 (linear mapping).

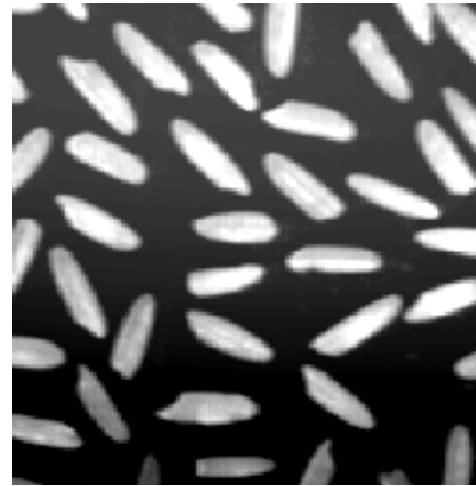
Yoğunluk Dönüşüm Fonksiyonları

rice.tif - Kontrast Ayarlı imge

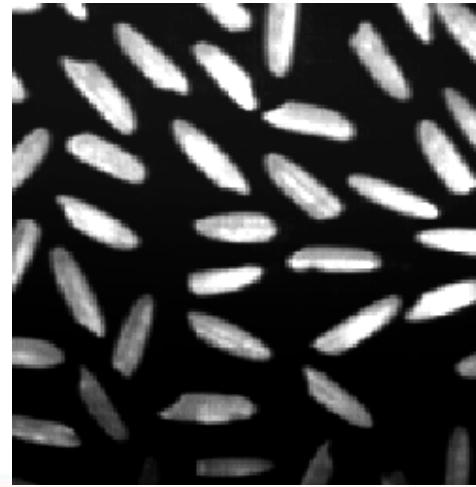
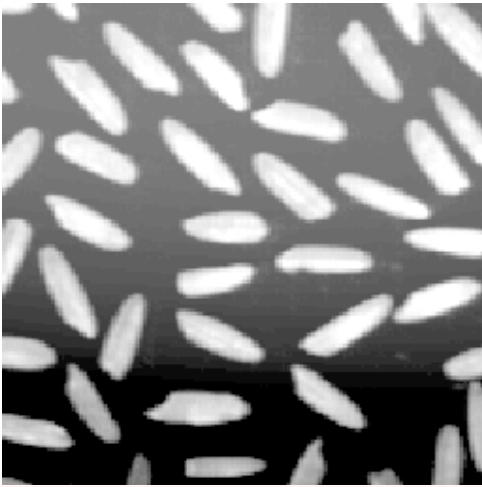


rice.tif - imadjust(rice,[0.25 0.75],[0 1])

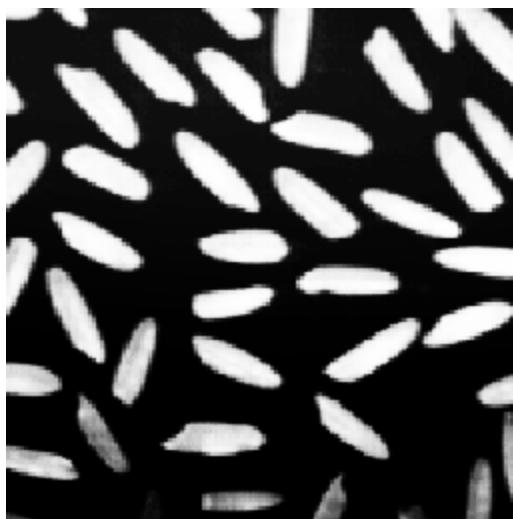
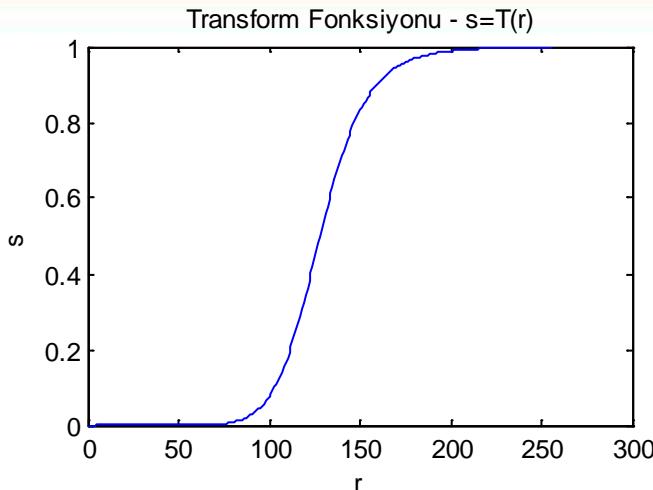
rice.tif - imadjust(rice,[0.25 0.75],[0 1])



rice.tif - imadjust(rice,[0.25 0.75],[0 1],0.5)



Contrast Stretching Transformation



$$m=127; E=10;$$

```
s=1./(1+(m./double(rice)+eps).^E);
```

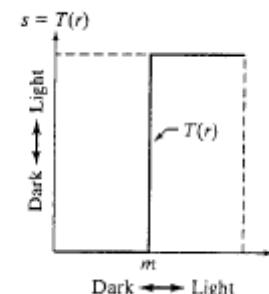
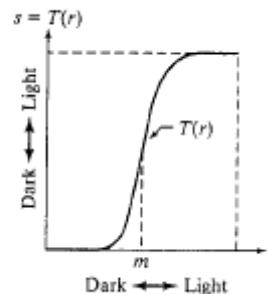
```
figure,imshow(s);
```

The function shown in Fig. 3.4(a) is called a *contrast-stretching* transformation function because it compresses the input levels lower than m into a narrow range of dark levels in the output image; similarly, it compresses the values above m into a narrow band of light levels in the output. The result is an image of higher contrast. In fact, in the limiting case shown in Fig. 3.4(b), the output is a binary image. This limiting function is called a *thresholding* function, which, as we discuss in Chapter 10, is a simple tool used for image segmentation. Using the notation introduced at the beginning of this section, the function in Fig. 3.4(a) has the form

$$s = T(r) = \frac{1}{1 + (m/r)^E}$$

where r represents the intensities of the input image, s the corresponding intensity values in the output image, and E controls the slope of the function. This equation is implemented in MATLAB for an entire image as

```
g = 1./(1 + (m./double(f) + eps).^E)
```



a b

FIGURE 3.4
(a) Contrast-stretching transformation.
(b) Thresholding transformation.

Logarithmic Transformations



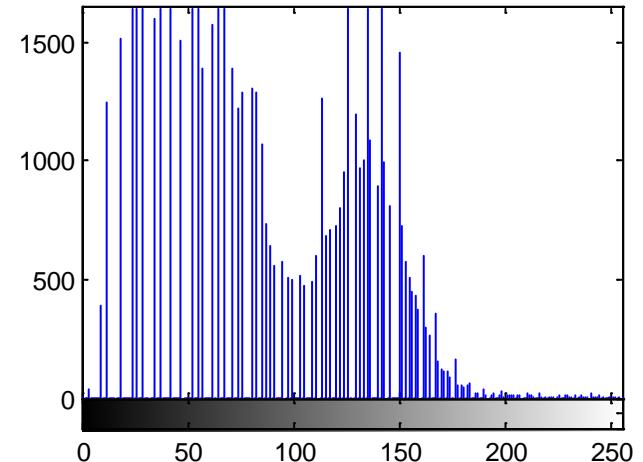
pout.tif - Orjinal imge



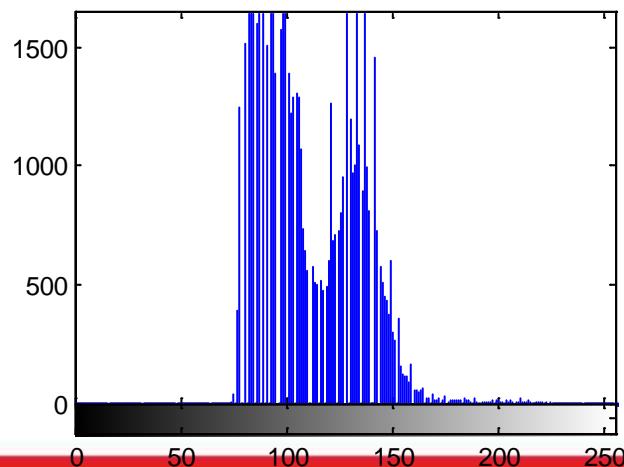
pout.tif (log transform Sonrası)



pout.tif (log transform Sonrası)- imhist



pout.tif - imhist

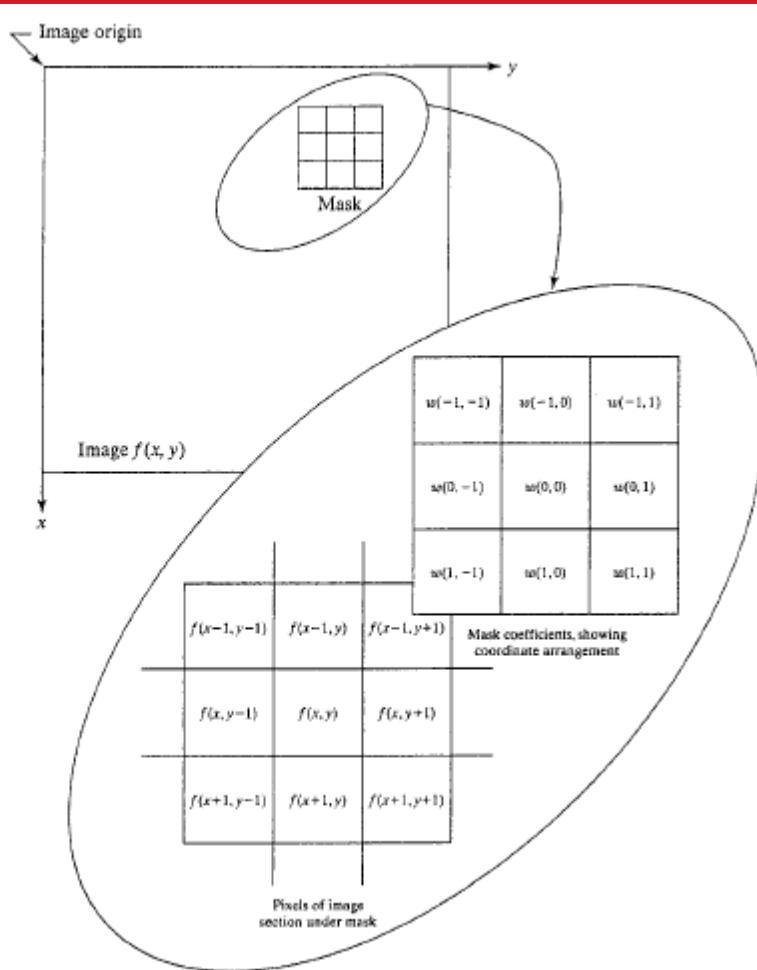


```
I = imread('pout.tif');
figure,imshow(I); title('pout.tif - Orjinal imge')
figure,imhist(I), title('pout.tif - imhist')
```

```
g=im2uint8(mat2gray(log(1+double(I))));
figure,imshow(g); title('pout.tif (log transform Sonrası)')
figure,imhist(g),title('pout.tif (log transform Sonrası)- imhist')
```

FİLTRELER

FIGURE 3.12 The mechanics of linear spatial filtering. The magnified drawing shows a 3×3 mask and the corresponding image neighborhood directly under it. The neighborhood is shown displaced out from under the mask for ease of readability.



`f = im2double(imread('rice.tif'));
imshow(f);title('rice.tif - Orjinal imge')`

`w1=ones(3);
g1=imfilter(f,w1,'replicate');
figure, imshow(g1,[]);
title('rice.tif - 8 Komşu')`

`w2=ones(10);
g2=imfilter(f,w2,'replicate');
figure, imshow(g2,[]);
title('rice.tif - 99 Komşu')`

Filter, mask, kernel, window

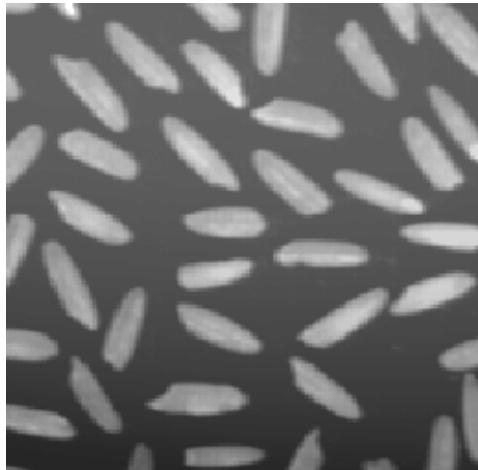
Çarpımlar toplamı hesaplanır, komşu pikseller dikkate alınır



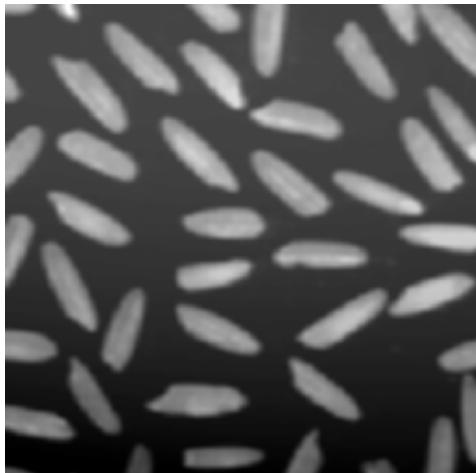
FİLTRELER



rice.tif - Orjinal imge



rice.tif - 8 Komşu



$$w = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

```
f = im2double(imread('rice.tif'));
imshow(f);title('rice.tif - Orjinal imge')
```

```
w1=ones(3); g1=imfilter(f,w1,'replicate');
figure, imshow(g1,[]); title('rice.tif - 8 Komşu')
```

```
w2=ones(10); g2=imfilter(f,w2,'replicate');
figure, imshow(g2,[]); title('rice.tif - 99 Komşu')
```

rice.tif - 99 Komşu



$$w = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots \\ 1 & 1 & 1 & 1 & \dots \\ 1 & 1 & 1 & 1 & \dots \\ 1 & 1 & 1 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

1. Standard Linear Spatial Filters

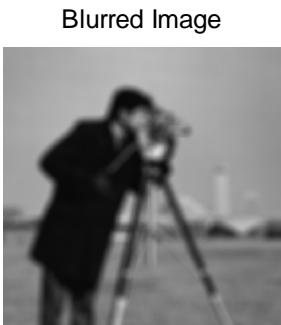
Doğrusal Uzaysal Filtreler



Type	Syntax and Parameters
'average'	<code>fspecial('average', [r c])</code> . A rectangular averaging filter of size $r \times c$. The default is 3×3 . A single number instead of $[r c]$ specifies a square filter.
'disk'	<code>fspecial('disk', r)</code> . A circular averaging filter (within a square of size $2r + 1$) with radius r . The default radius is 5.
'gaussian'	<code>fspecial('gaussian', [r c], sig)</code> . A Gaussian lowpass filter of size $r \times c$ and standard deviation sig (positive). The defaults are 3×3 and 0.5. A single number instead of $[r c]$ specifies a square filter.
'laplacian'	<code>fspecial('laplacian', alpha)</code> . A 3×3 Laplacian filter whose shape is specified by $alpha$, a number in the range $[0, 1]$. The default value for $alpha$ is 0.5.
'log'	<code>fspecial('log', [r c], sig)</code> . Laplacian of a Gaussian (LoG) filter of size $r \times c$ and standard deviation sig (positive). The defaults are 5×5 and 0.5. A single number instead of $[r c]$ specifies a square filter.
'motion'	<code>fspecial('motion', len, theta)</code> . Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of len pixels. The direction of motion is $theta$, measured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.
'prewitt'	<code>fspecial('prewitt')</code> . Outputs a 3×3 Prewitt mask, wv , that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: $wh = wv'$.
'sobel'	<code>fspecial('sobel')</code> . Outputs a 3×3 Sobel mask, sv , that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: $sh = sv'$.
'unsharp'	<code>fspecial('unsharp', alpha)</code> . Outputs a 3×3 unsharp filter. Parameter $alpha$ controls the shape; it must be greater than 0 and less than or equal to 1.0; the default is 0.2.

1. Standard Linear Spatial Filters

Doğrusal Uzaysal Filtreler



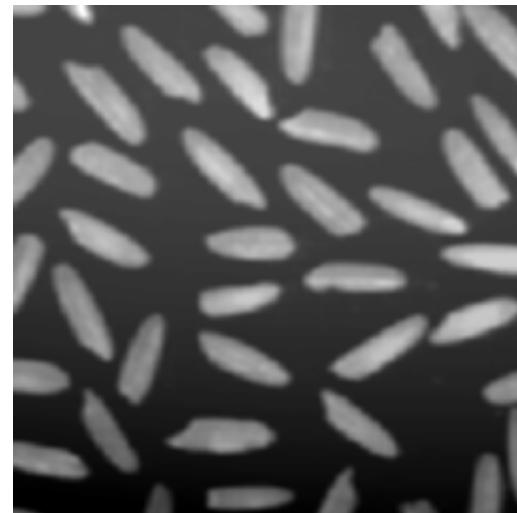
```
I = imread('cameraman.tif');
subplot(2,2,1);imshow(I);
title('Original Image');
H = fspecial('motion',20,45);
MotionBlur = imfilter(I,H,'replicate');
subplot(2,2,2);
imshow(MotionBlur);
title('Motion Blurred Image');
H = fspecial('disk',10);
blurred = imfilter(I,H,'replicate');
subplot(2,2,3);
imshow(blurred);title('Blurred Image');
H = fspecial('unsharp');
sharpened = imfilter(I,H,'replicate');
subplot(2,2,4);imshow(sharpened);
title('Sharpened Image');
```

fspecial -average filter



```
H = FSPECIAL('average',HSIZE)  
hsiz: filtre boyutu
```

rice.tif - average filter



```
>> H=fspecial('average',3)
```

```
H =
```

```
0.1111  0.1111  0.1111  
0.1111  0.1111  0.1111  
0.1111  0.1111  0.1111
```

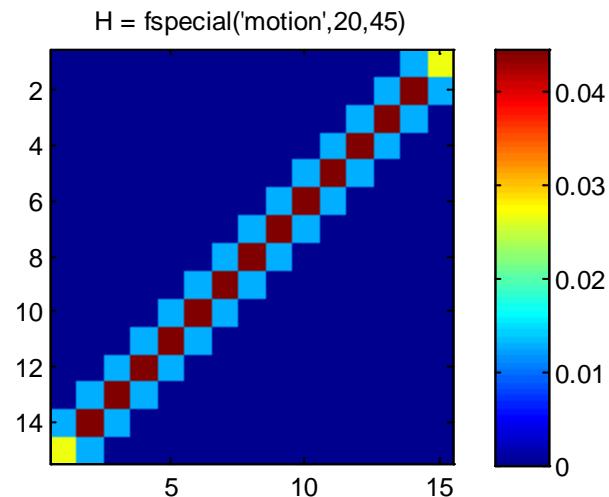
fspecial -motion filter



`H = FSPECIAL('motion',LEN,THETA)`

len: filtre boyutu

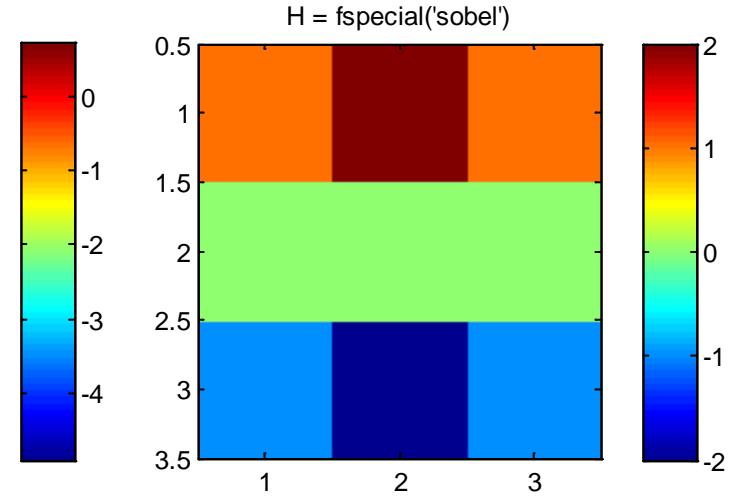
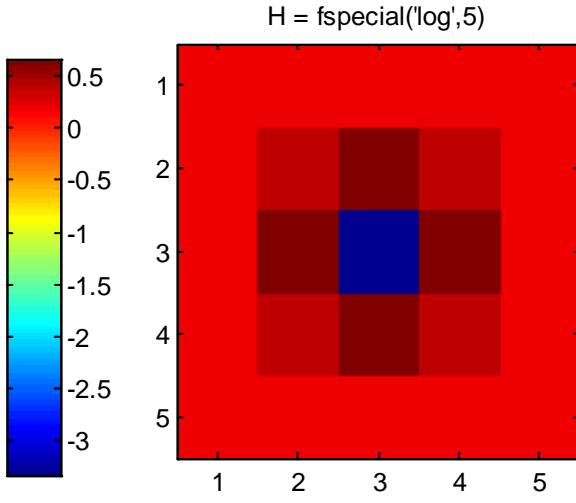
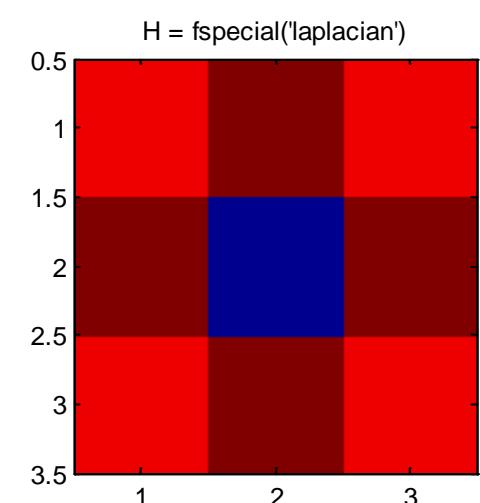
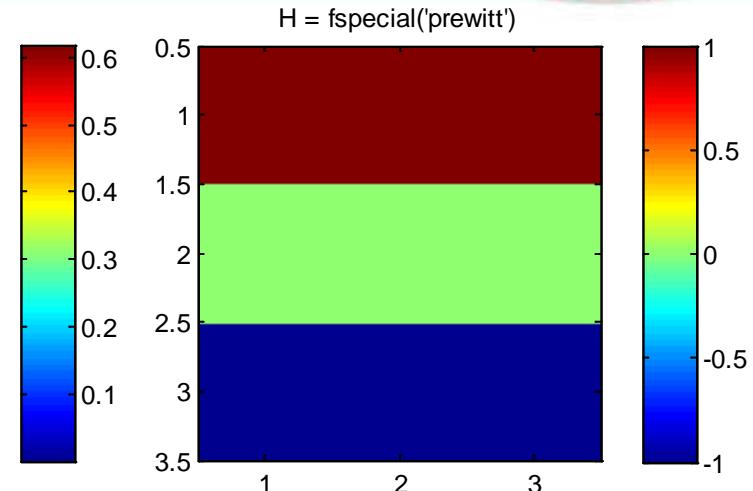
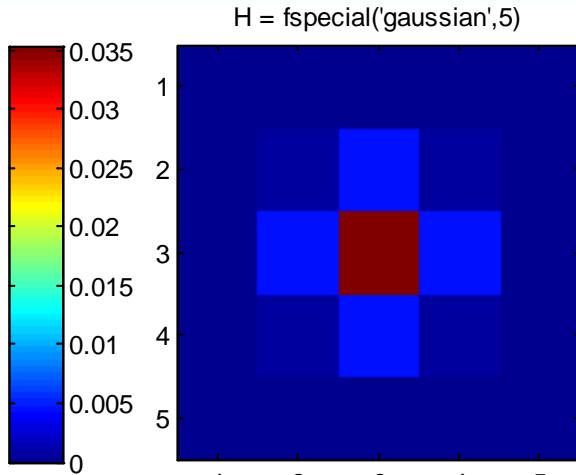
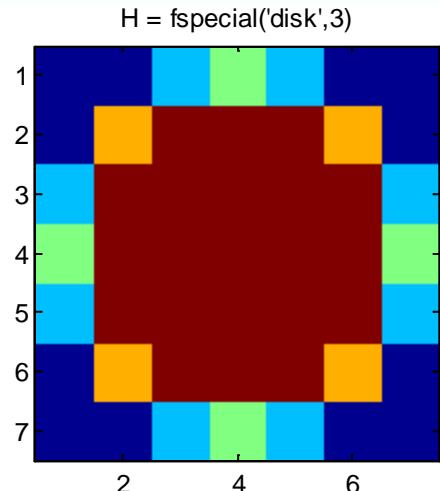
theta: Eğim açısı



Motion Blurred Image



fspecial



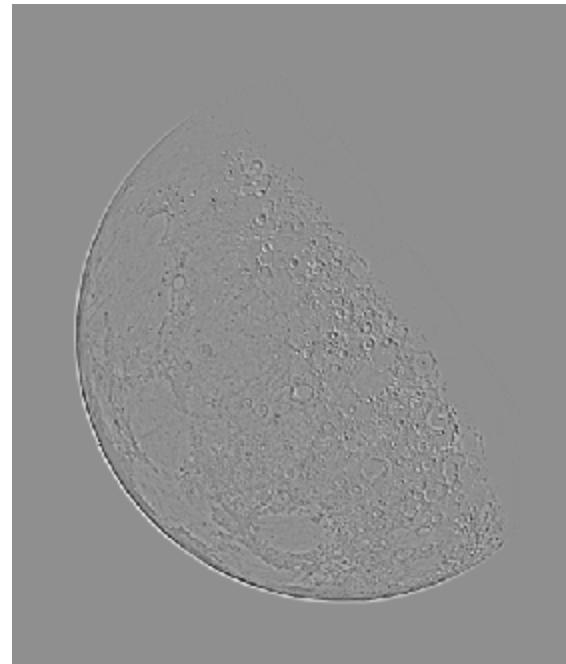
Laplacian Filter



moon.tif - Orjinal imge



moon.tif - Laplacian Filter



```
moon=imread('Fig0338(a)(blurry_moon).tif');
w4=fspecial('laplacian',0);
f=im2double(moon);
g4=imfilter(f,w4,'replicate');
figure,imshow(g4,[]);
```

Laplacian Filter



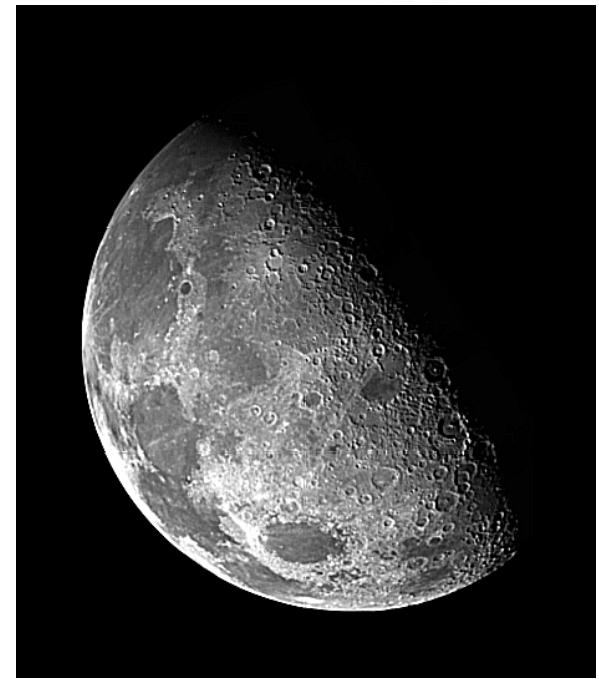
f-imfilter(f,w4,'replicate')



Orjinal İmge



f-imfilter(f,w8,'replicate')



```
w4=fspecial('laplacian',0);w8 = [1 1 1; 1 -8 1; 1 1 1];
f=im2double(moon); g4=f-imfilter(f,w4,'replicate');
g8=f-imfilter(f,w8,'replicate');imshow(f);title('Orjinal İmge')
figure,imshow(g4);title('f-imfilter(f,w4,"replicate")')
figure,imshow(g8);title('f-imfilter(f,w8,"replicate")')
```

$$\begin{matrix} w4 = \\ \begin{matrix} 0 & 1 & 0 \\ 1 & -8 & 1 \\ 0 & 1 & 0 \end{matrix} \end{matrix}$$

$$\begin{matrix} w8 = \\ \begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix} \end{matrix}$$

2. Standard Nonlinear Spatial Filters

order statistic filters or rank filters

- Min filter `g = ordfilt2(f, 1, ones(m, n))`
- Max Filter `g = ordfilt2(f, m*n, ones(m, n))`
- Median Filter `g = ordfilt2(f, median(1:m*n), ones(m, n))`

ordfilt2



$B = \text{ordfilt2}(A, \text{order}, \text{domain})$

Her eleman domain ile belirlenen alandaki 0'dan farklı elemanları sıralayarak order ile belirtilen eleman ile değiştirir.

Bu durumda domain=true(5) ise

Order=1, Bu aralıktaki en küçük pixel ile

Order=25, Bu aralıktaki en büyük pixel ile

Order=13, Bu aralıktaki ortadaki (medyan) pixel ile
değiştirmek demektir.

Medyan filtre



- Salt and pepper tarzı gürültüler için kullanılır.

Salt & Pepper Gürültülü img



Medyan Filtre - IM=medfilt2(IN)

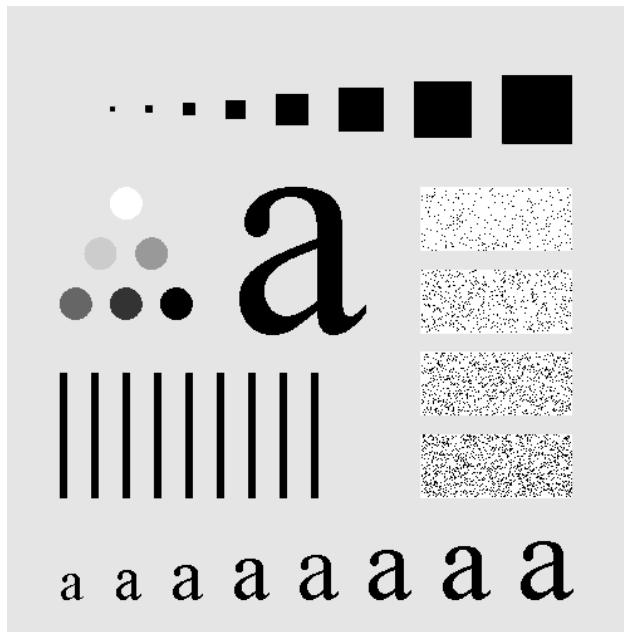


```
I=imread('livingroom.tif');
IN=imnoise(I,'salt & pepper',0.02);
IM=medfilt2(IN);
```

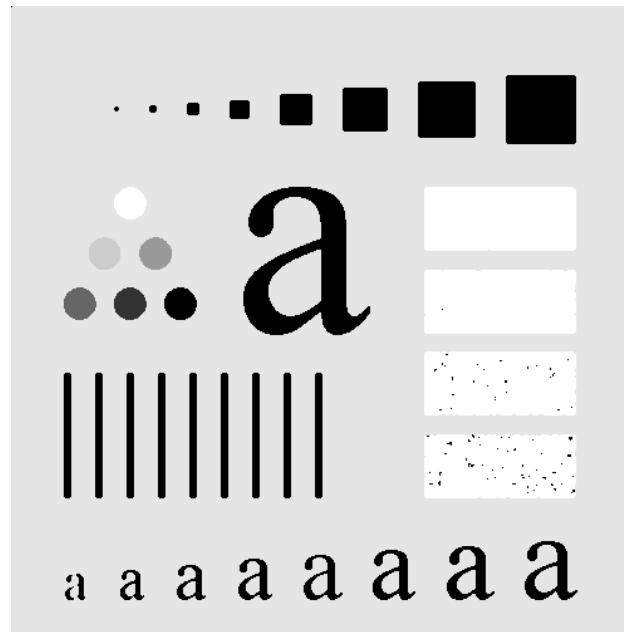
Medyan filtre



Orjinal imge



Medyan Filtre - $fM = \text{medfilt2}(f)$



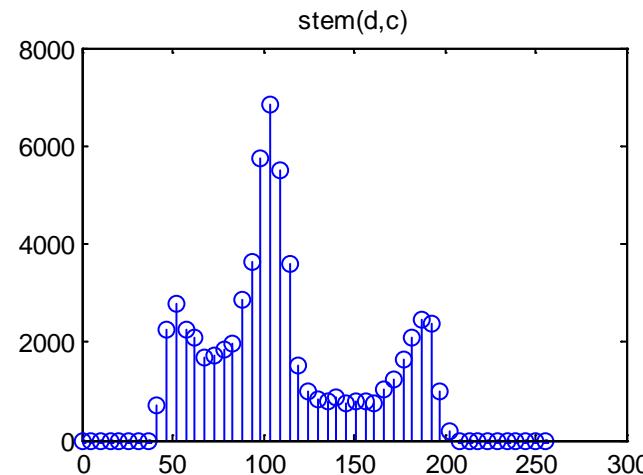
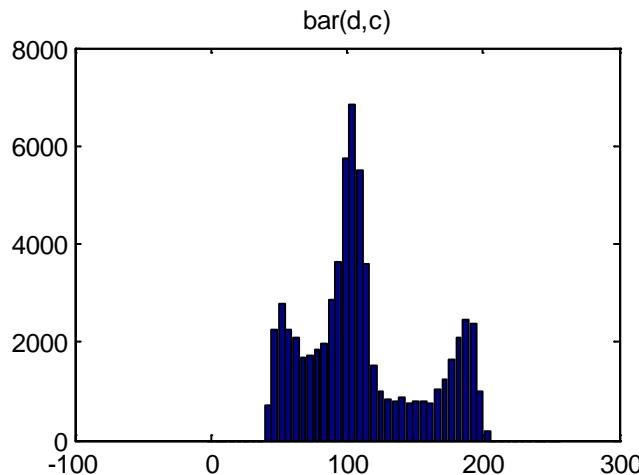
Histogram Processing

HISTOGRAM İŞLEME

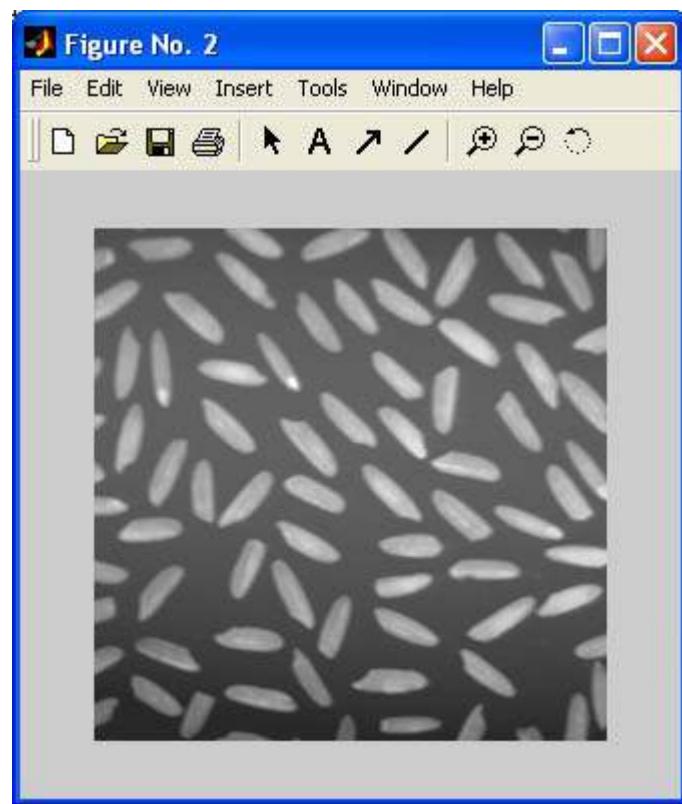
Histogram



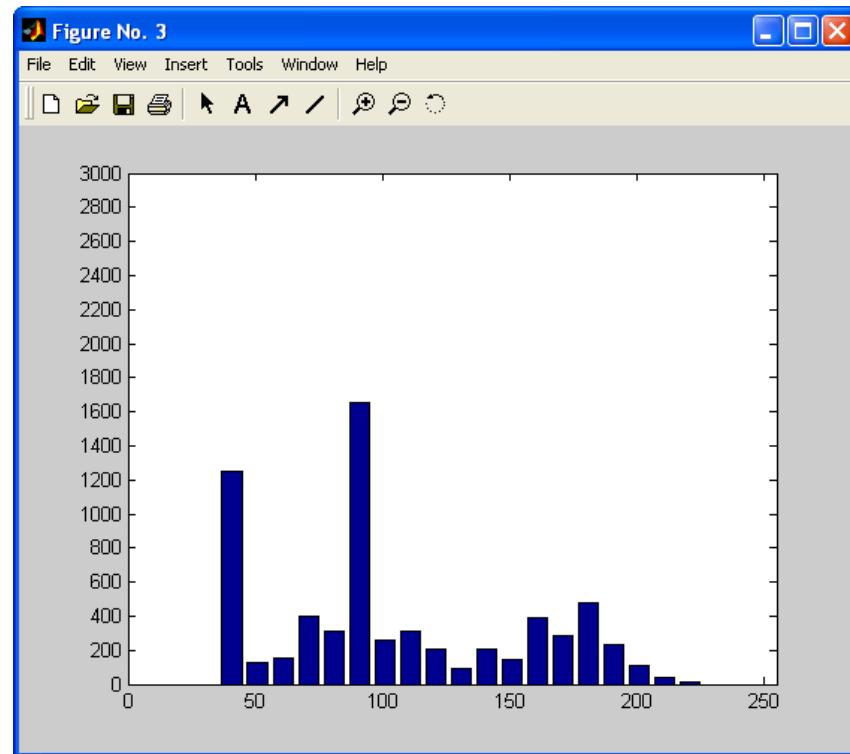
- `[c,d]=imhist(f,50);`
 - c: f imgesinde her aralıkta kaç adet değer var
 - d: Aralıkları gösterir
 - Burada aralıklar-> 0: (255/49):255 olacak şekildedir
- Sonuçlar `bar(c)`, `stem(c)` kullanılarak gösterilebilir



Örnek 1 : Resim Histogramları : Generating and Plotting Image Histograms



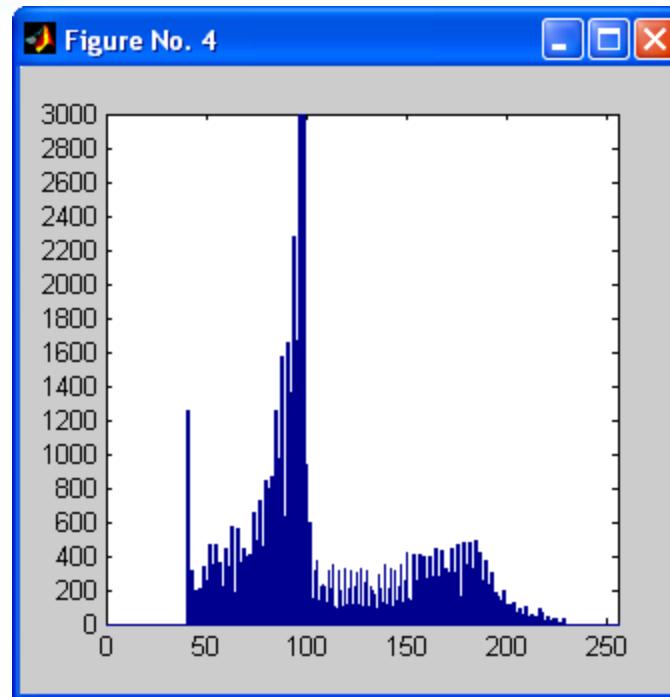
```
f=imread('rice.png');
h=imhist(f);
h1=h(1:10:256);
horz=1:10:256;
bar(horz,h1);
axis([0 255 0 3000]);
set(gca,'xtick',0:50:255);
set(gca,'ytick',0:200:3000);
```



Tüm Yoğunluk Değerleri İçin



```
f=imread('rice.png');  
h=imhist(f);  
h1=h(1:256);  
horz=1:256;  
bar(horz,h1);  
axis([0 255 0 3000]);  
set(gca,'xtick',0:50:255);  
set(gca,'ytick',0:200:3000);
```



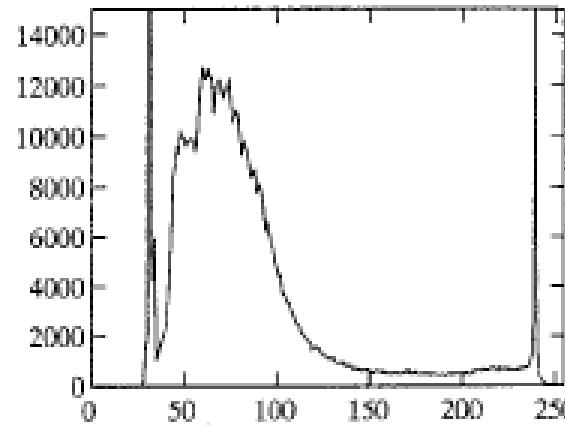
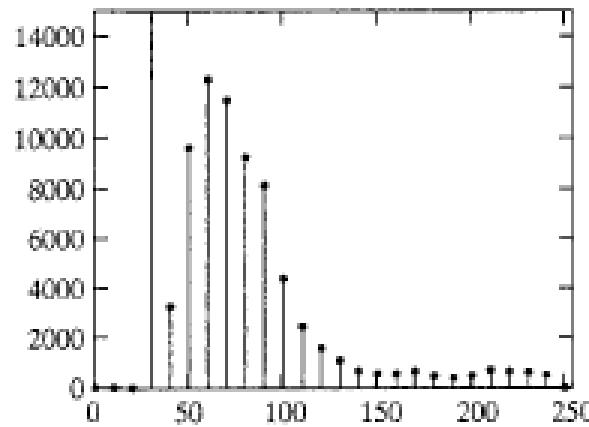
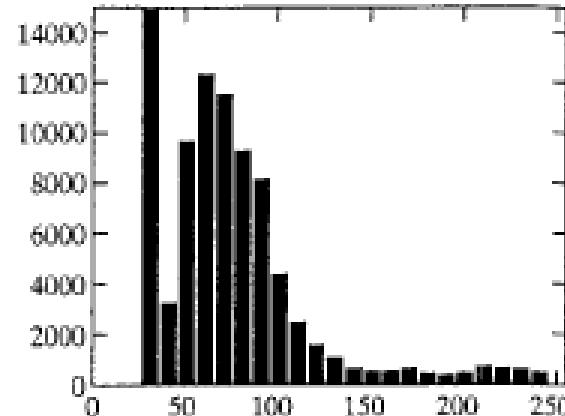
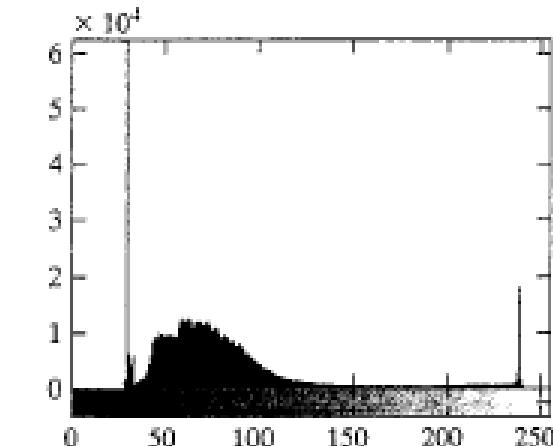
Değişik Histogram Çizdirme Yolları

a b
c d

FIGURE 3.7

Various ways to plot an image histogram.
(a) `imhist`,
(b) `bar`,

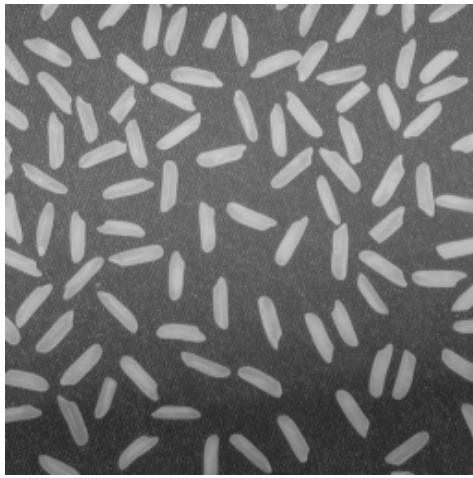
(c) `stem`,
(d) `plot`.



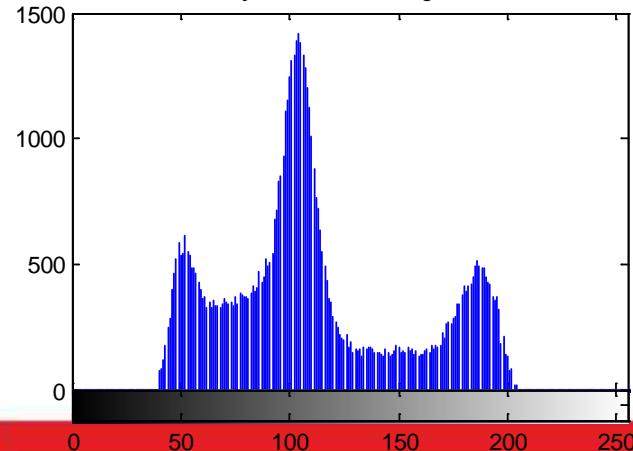
Histogram Eşitleme



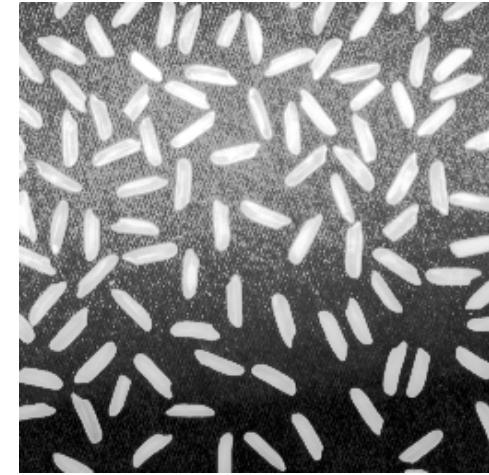
Orjinal Resim



Orjinal Resim Histogramı

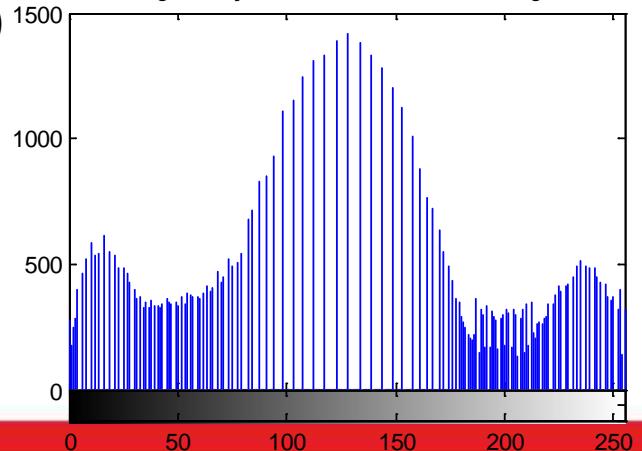


Histogram Eşitleme Sonrası Resim



```
f=imread('rice.png');
figure,imshow(f);
title('Orjinal Resim')
figure;
imhist(f);title('Orjinal Histogram')
ylim('auto');
g=histeq(f,256);
figure,imshow(g);
title('Eşitleme Sonrası Resim')
figure;imhist(g);
title('Eşitleme Sonrası Histogram ')
ylim('auto');
```

Histogram Eşitleme Sonrası Resim Histogramı



Örnek 4 : Karakter Boyu Bulma

```
r1 = imread('c:\\plaka2.jpg');
imshow(r1);

K = imfinfo('c:\\plaka2.jpg');
for i=1:K.Height
    satir_toplam(i)=sum(r1(i,:));
end

display(K.Height); display(K.Width);
display(satir_toplam);

for i=K.Height/2:-1:1
    if(satir_toplam(1)-satir_toplam(i)<2500)
        break;
    end;
end
low = i;
display(low);

for i=K.Height/2:1:K.Height
    if(satir_toplam(1)-satir_toplam(i)<2500)
        break;
    end;
end
high = i;
display(high);

r2 = imcrop(r1,[1 low K.Width high-low]);
figure, imshow(r2);
```



SAYISAL GÖRÜNTÜ İŞLEME

GÖRÜNTÜDE FILTRELEME OPERASYONLARI

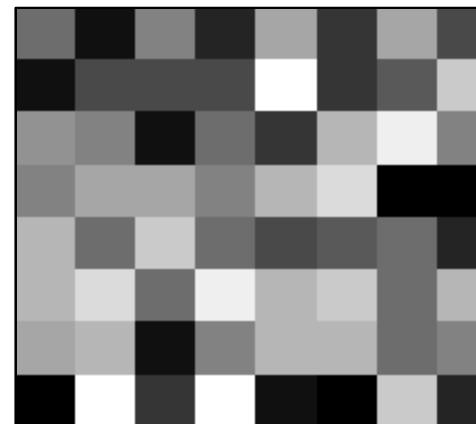
GÖRÜNTÜDE FILTRELEME OPERASYONLARI

- Filtreler görüntü zenginleştirme amacı ile de uygulanan, adından da anlaşılacağı gibi görüntüde belirli ayrıntıların ayıklanması ya da daha belirgin hale getirilmesi vb. gibi operasyonları gerçekleştiren operatörlerdir.
- Farklı amaçlar için farklı filtreleme operatörleri vardır. Bunlara:
 - Kenar keskinleştirme
 - Kenar yakalama
 - Görüntü yumusatma ve bunun gibi daha bir çok amaçla kullanılan filtreler örnek verilebilir.

GÖRÜNTÜDE FILTRELEME OPERASYONLARI

- Bilindiği görüntüyü oluşturan pikseller konumları ve gri değerleri ile tanımlanabilmektedir.
- Daha doğrusu bir görüntü matris formuna sahiptir.
- Aşağıdaki şekilde 8x8 lik bir görüntüyü oluşturan piksellerin gri değerleri verilmiştir

	0	1	2	3	4	5	6	7
0	87	82	88	83	90	84	90	85
1	82	85	85	85	95	84	86	92
2	89	88	82	87	84	91	94	88
3	88	90	90	88	91	93	81	81
4	91	87	92	87	85	86	87	83
5	91	93	87	94	91	92	87	91
6	90	91	82	88	91	91	87	88
7	81	95	84	95	82	81	92	83



GÖRÜNTÜDE FILTRELEME OPERASYONLARI

- Filtreler çekirdek matris formundadır ve boyutları $3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9, 11 \times 11$ şeklinde olabilir.
- Filtre matrisi tanımlandığı amaca yönelik olarak görüntüde işleme sokulur.

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

Örneğin yandaki filtre matrisi ile görüntüyü filtrelediğimizde bu matrisi tüm görüntü üzerinde 3×3 lük pikseller şeklinde uygularız

i-1,j-1	i,j-1	i+1,j-1
i-1,j	i,j	i+1,j
i-1,j+1	i,j+1	i+1,j+1

GÖRÜNTÜDE FILTRELEME OPERASYONLARI

- Buna göre bu örnek filtre matrisini uygulayacak olursak:

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

$$g'(i,j) = (-1 * g_{i,j-1}) + (-1 * g_{i-1,j}) + (5 * g_{i,j}) + (-1 * g_{i+1,j}) + (-1 * g_{i,j+1})$$

FİLTRE

Örneğin 1,1 koordinatlı pikselin滤relenmiş değerini bulmak istersek

i-1,j-1	i,j-1	i+1,j-1
i-1,j	i,j	i+1,j
i-1,j+1	i,j+1	i+1,j+1

Komşuluk İlişkisi

0	87	82	88	83	90	84	90	85
1	82	85	85	85	95	84	86	92
2	89	88	82	87	84	91	94	88
3	88	90	90	88	91	93	81	81
4	91	87	92	87	85	86	87	83
5	91	93	87	94	91	92	87	91
6	90	91	82	88	91	91	87	88
7	81	95	84	95	82	81	92	83

$$g'_{1,1} = -82-82+5*85-85-88 = 88$$

$$g'_{4,5} = -93-85+5*86-87-92 = 73$$

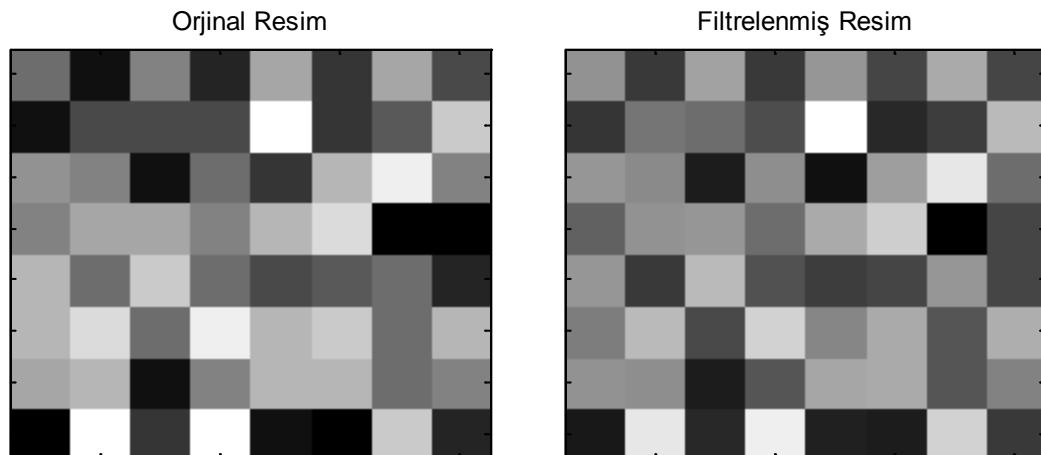
GÖRÜNTÜDE FILTRELEME OPERASYONLARI

- İlk ve son satır ve sütunlar, sınır noktaları bütün komşulara sahip olmadığı için özel işleme tabii tutulurlar:

	0	1	2	3	4	5	6	7
0	87	82	88	83	90	84	90	85
1	82	85	85	85	95	84	86	92
2	89	88	82	87	84	91	94	88
3	88	90	90	88	91	93	81	81
4	91	87	92	87	85	86	87	83
5	91	93	87	94	91	92	87	91
6	90	91	82	88	91	91	87	88
7	81	95	84	95	82	81	92	83

Sınır İşlemleri

87	87	82	88
87	87	82	88
82	82	85	85
89	89	88	82



$$H =$$

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$$

`f=imfilter(img,H,'replicate')`

Görüntü yumuşatma operatörleri

- Görüntüdeki gürültünün (bozucu etkinin) yok edilmesi veya indirgenmesinde kullanılır.
 - Eğer görüntüde mevcut gürültü yüksek frekansta ise lowpass-alçak geçirgenli filtre ile indirgenir.
- Çözünürlüğün indirgenmesi işleminde kullanılır.
 - Eğer görüntünün çözünürlüğü çok yüksek ise ön görüntüleme işlemleri için tüm çözünürlüğe gereksinimim duyulmaz.
 - Çözünürlüğün indirgenmesi görüntü piramitlerinin oluşturulmasında da kullanılır.

Averaj Operatörü

- Ortalama operatörü:

$$h = \frac{1}{(2N+1)(2M+1)} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}$$

$$g_s(x, y) = \sum_{i=-N}^N \sum_{j=-M}^M g(x-i, y-j)h(i, j)$$

- Burada $(2N+1, 2M+1)$ filtrenin boyutunu tanımlar.
- Yumuşatılmış g_s görüntüsü orijinal görüntünün katlamasından (convolving) elde edilmiştir
- Genelde $N=M$ alınır.
- Bazı durumlarda yukarıda verilen filtre katsayıları değiştirilebilir.

Katlama (ConvolutionFiltering)

- Katlama filtreleme bir görüntünün mekansal frekans karakteristiğinin değiştirilmesinde kullanılır.
- Filtre çekirdeği kullanılarak filtreleme işlemi gerçekleştirilir.
- Zamanda yapılan konvolusyon işlemi frekans ortamında çarpmaya karşılık gelir.
- H konvolusyon kerneli olarak adlandırılır:

$$g(x, y) * h(x, y) = \sum_{i=-N}^N \sum_{j=-M}^M g(x-i, y-j)h(i, j)$$

$$F\{g(x, y) * h(x, y)\} = G(u, v)H(u, v)$$

Katlama (ConvolutionFiltering)

- Matlab'de konvolusyon için **conv2** (1D işaretler için **conv**) kullanılır.
- Filtreleme işlemlerinde de zaman domeninde konvolusyon uygulanmaktadır.
- Konvolusyon işlemi sonucunda sınır bölgeleri için de işlem yapıldığından orijinal boyuttan daha büyük bir boyut elde edilir
 - İmge Boyutu + Filtre boyutu -1

i =

87	82	88	83
82	85	85	85
89	88	82	87
88	90	90	88

>> conv2(i,h)

21.75	42.25	42.50	42.75	20.75
42.25	84	85	85.25	42
42.75	86	85	84.75	43
44.25	88.75	87.50	86.75	43.75

>> conv2(i,h,'same')

(imfilter(i,h) ile aynı sonucu verecektir)

h =

0.2500	0.2500
0.2500	0.2500

22	44.50	45	44.50	22
----	-------	----	-------	----

44.50	45	44.50	22
-------	----	-------	----

SAYISAL GÖRÜNTÜ İŞLEME

Frekans Ortamı İşlemleri

Fourier Dönüşümü

Fourier dönüşümü (FD), herhangi bir sinyali zaman dizisi yerine, frekans dizisi ile ifade eder; sinyali, farklı frekansların meydana getirdiği bir sinyal şeklinde gösterir.

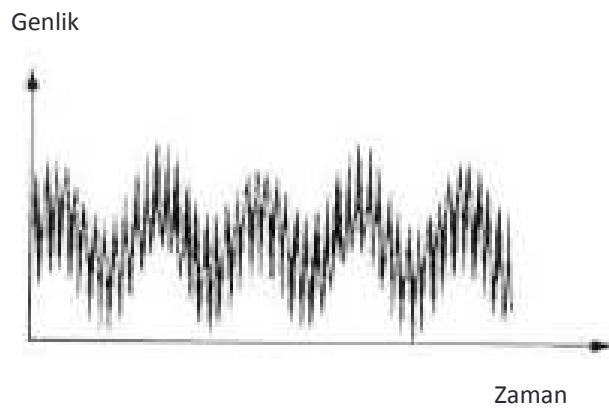
Bir çok sinyal, genellikle içindeki şiddet değişimleri olarak temsil edilir. Bu durumu iki boyutlu bir grafik ile gösterecek olursak, eksenlerden biri zamanı diğer ise sinyalin şiddetini (genliğini) temsil eder.

Fourier Dönüşümü

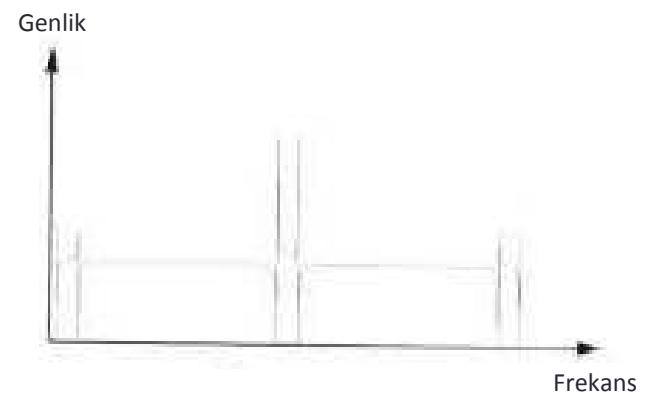
Burada zaman bağımsız bir değişken iken genlik bağımlı değişkendir.

Sinyal Fourier dönüşümünden geçtikten sonra, sinyalin zaman ekseni frekans ile yer değiştirir. Böylece sinyal zaman ortamından, frekans ortamına taşınmış olur. Artık bağımsız değişken frekans, bağımlı değişken ise yine frekansa bağlı genektir.

Fourier Dönüşümü



Fourier
Dönüşümü
(FD)



Fourier dönüşümü ile zaman ortamından frekans ortamına geçiş

Fourier Dönüşümü

- Fourier dönüşümü ile zaman ortamından frekans ortamına geçildiğinde zaman bilgisi kaybolur.
- Böyle bir durumda sinyal üzerinde herhangi bir durum ortaya çıktığında, bu durumun ne zaman gerçekleştiği belirlenemez.
- Fakat, ters dönüşüm mevcuttur. Eğer istenilirse frekans ortamından zaman ortamına geri dönülebilir ve zaman bilgisi geri kazanılabilir.

Dönüşüm matematiksel olarak sinyali, iki temel fonksiyon olan sinüs ve kosinüs fonksiyonlarının doğrusal bileşimi şeklinde ayırtırır.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cos(\omega t) dt + j \int_{-\infty}^{\infty} f(t) \sin(\omega t) dt$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2j\pi\omega t} dt$$

t; zaman,

w; frekans,

F; frekans ortamında sinyal genliğiidir.

f(t) ise sinyalin zaman ortamındaki genliği (şiddeti) dir.

Dönüşümün tersi ise;

$$f(t) = \int_{-\infty}^{\infty} F(\omega) e^{2j\pi\omega t} d\omega$$

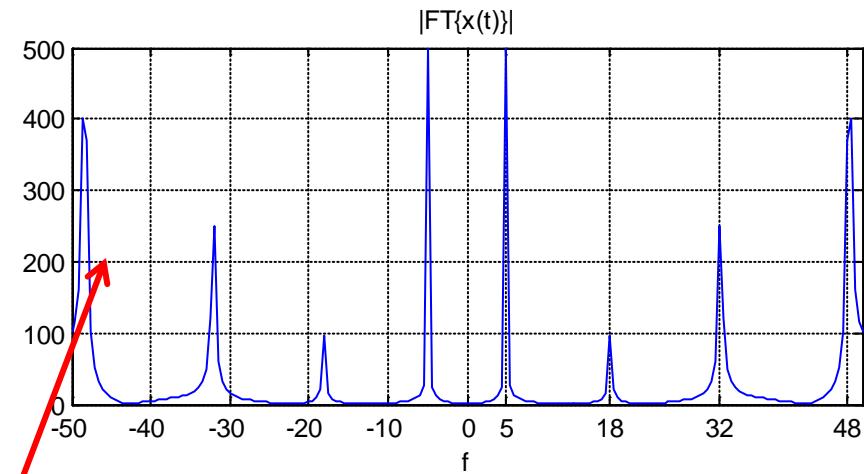
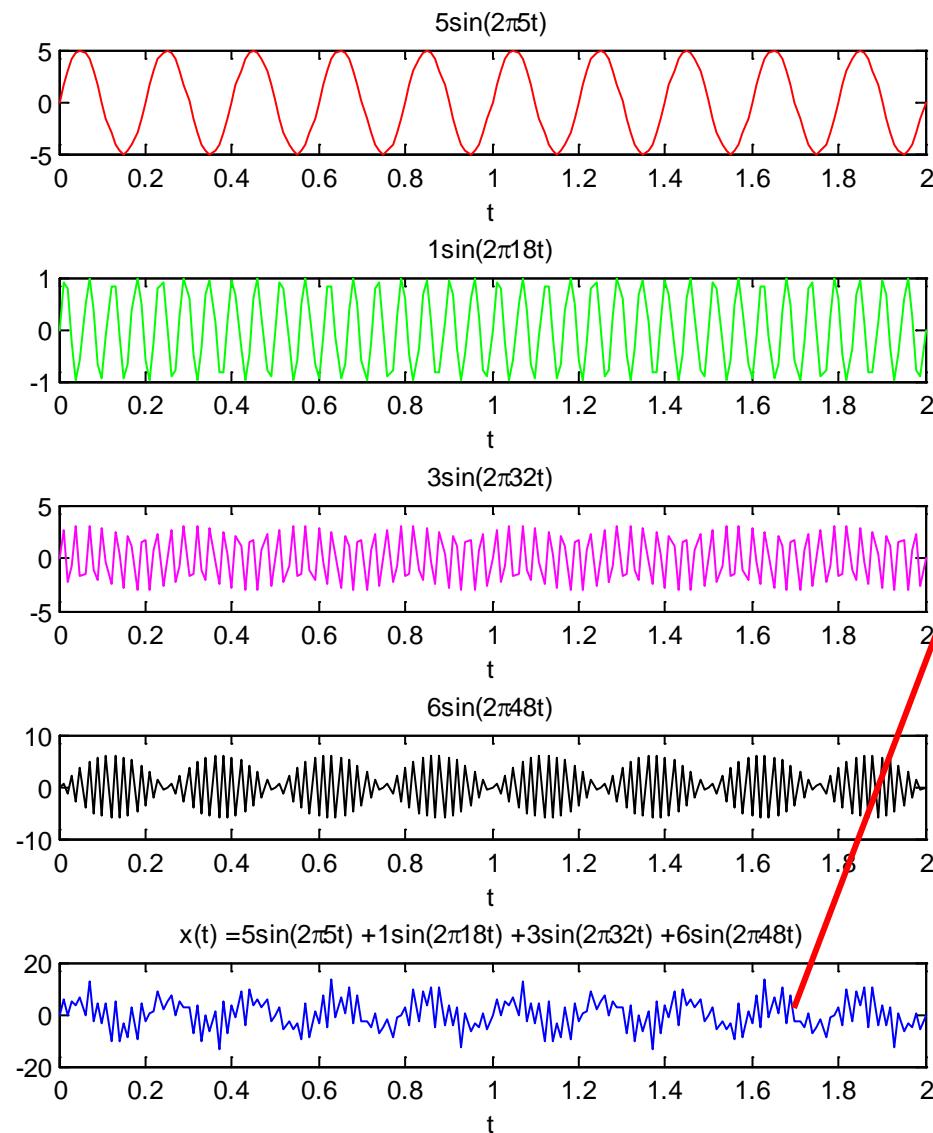
Eğer sinyal zamana bağlı olarak değişim gösteriyorsa, bu dönüşümle belirli bir zaman dilimde (pencerede) sinyalin bir kısmına bakılabilir.

Bu tekniğe ‘pencereleme’ (*windowing*) ismi verilir. Bu teknik, sinyali, zamanın ve frekansın iki boyutlu fonksiyon gösterimi şeklinde ifade eder.

Böylece bir sinyal üzerinde gerçekleşen bir değişimin, hangi zamanda ve frekansta meydana geldiği görülebilir.

Fakat bu metod seçilecek pencerenin boyutu ile sınırlıdır. Bir sinyal bu şekilde bir pencerelenmiş dönüşümden geçiyorsa buna Kısa Zamanlı Fourier Dönüşümü (KSFD) adı verilir.

Zaman ve frekans bilgisini aynı anda inceleyebilmek için Wavelet (Dalgacık) Dönüşümü kullanılabilir.



Fourier Transform

- $-T/2$ ile $T/2$ arasında değer alır
- 0 'a göre simetriktir

Örneklem frekansı: 100 Hz
2 saniyelik veri

Görüntü Üzerinde Fourier Dönüşümü

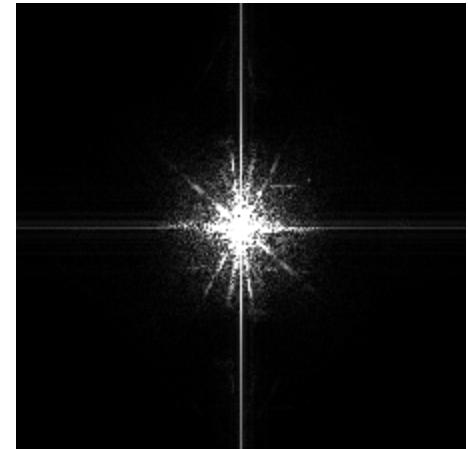
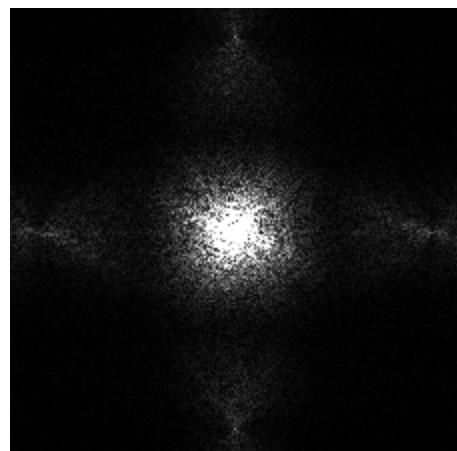
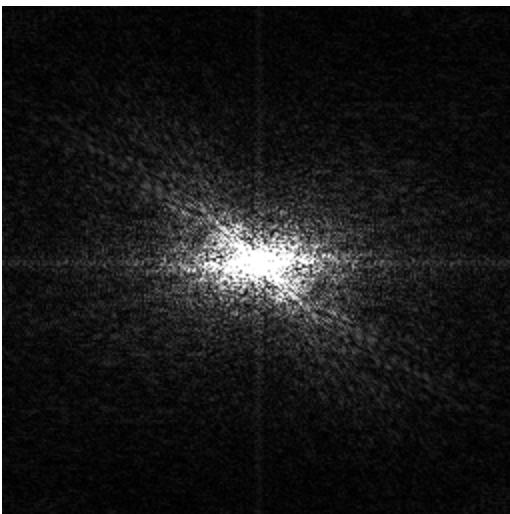
Sayısal görüntülerin gösterimi mekansal alanın gösteriminde kullanılan x ve y koordinatlarından oluşmaktadır. Bu görüntü Fourier dönüşümünden geçtikten sonra frekans alanında temsil edilmiş olur.

Görüntü Üzerinde Fourier Dönüşümü

- Dönüşüm sonucundaki gösterimde, görüntüdeki **yüksek frekanslar kenarlarda, alçak frekanslar ise orta kısımlarda** toplanır.
- Yersel ortamda yatay olan çizgiler dikey; dikey olan çizgiler ise yatay olarak görüntülenir.
- İstenirse görüntü işleme programındaki fonksiyonlar kullanılarak tekrar yersel ortama geçiş sağlanabilir.



Dijital imgeler ve 2-D DFT'leri



2 Boyutlu Fourier Dönüşümü

2D Fourier Dönüşümü

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

2D Ters Fourier Dönüşümü

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

Kesikli Fourier Dönüşümü

- İmgeler sayısallaştırılmış, ayrık sayı dizileridir.
- Bu nedenle imgelerde dönüşüm için ayrık zamanlı bir işlem gereklidir
 - Bu işlemde düzgün aralıklı verilerin işlenerek frekans ortamında eşit aralıklı Fourier dönüşüm değerlerinin geri gönderilmesi gereklidir.
- Kesik zamanlı işlemlerde integral yerini toplama bırakır, sonuç Ayrık Fourier Dönüşümü (DFT) olarak adlandırılır.
- Aşağıdaki formülde sıkılıkla $M=N$ olur (imge boyutları)

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

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

2D Ayrık Fourier Dönüşümü

2D Ters Ayrık Fourier Dönüşümü

Fourier Transform - Özellikler

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)}$
3) Polar representation	$F(u, v) = F(u, v) e^{j\phi(u, v)}$
4) Spectrum	$ F(u, v) = [R^2(u, v) + I^2(u, v)]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$
6) Power spectrum	$P(u, v) = F(u, v) ^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$

TABLE 4.2
Summary of DFT definitions and corresponding expressions.

Fourier Transform - Özellikler

TABLE 4.2
(Continued)

Name	Expression(s)
8) Periodicity (k_1 and k_2 are integers)	$F(u, v) = F(u + k_1M, v) = F(u, v + k_2N)$ $= F(u + k_1M, v + k_2N)$ $f(x, y) = f(x + k_1M, y) = f(x, y + k_2N)$ $= f(x + k_1M, y + k_2N)$
9) Convolution	$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$
10) Correlation	$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n)h(x + m, y + n)$
11) Separability	The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11.1.
12) Obtaining the inverse Fourier transform using a forward transform algorithm.	$MNf^*(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v)e^{-j2\pi(ux/M+vy/N)}$ <p>This equation indicates that inputting $F^*(u, v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x, y)$. Taking the complex conjugate and dividing by MN gives the desired inverse. See Section 4.11.2.</p>

Fourier Transform - Özellikler

TABLE 4.3

Summary of DFT pairs. The closed-form expressions in 12 and 13 are valid only for continuous variables. They can be used with discrete variables by sampling the closed-form, continuous expressions.

Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$
3) Translation (general)	$f(x, y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M+vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$
5) Rotation	$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$ $x = r \cos \theta \quad y = r \sin \theta \quad u = \omega \cos \varphi \quad v = \omega \sin \varphi$
6) Convolution theorem [†]	$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$ $f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$

(Continued)

Fourier Transform - Özellikler

Name	DFT Pairs
7) Correlation theorem [†]	$f(x, y) \star h(x, y) \Leftrightarrow F^*(u, v)H(u, v)$ $f^*(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$
8) Discrete unit impulse	$\delta(x, y) \Leftrightarrow 1$
9) Rectangle	$\text{rect}[a, b] \Leftrightarrow ab \frac{\sin(\pi ua)}{(\pi ua)} \frac{\sin(\pi vb)}{(\pi vb)} e^{-j\pi(ua+vb)}$
10) Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $j \frac{1}{2} [\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)]$
11) Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $\frac{1}{2} [\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)]$
The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.	
12) Differentiation (The expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$)	$\left(\frac{\partial}{\partial t}\right)^m \left(\frac{\partial}{\partial z}\right)^n f(t, z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu, \nu)$ $\frac{\partial^m f(t, z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu, \nu); \frac{\partial^n f(t, z)}{\partial z^n} \Leftrightarrow (j2\pi\nu)^n F(\mu, \nu)$
13) Gaussian	$A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

TABLE 4.3
(Continued)

[†] Assumes that the functions have been extended by zero padding. Convolution and correlation are associative, commutative, and distributive.

Görüntü Üzerinde Filtreleme İşlemleri

- Orijinal görüntü frekans ortamında temsil edildikten sonra üzerinde filtreleme işlemleri uygulanabilir.
- Filtreleme işlemlerinin ardından tekrar yersel ortama geri dönüldüğünde görüntüdeki bazı unsurların ön plana çıkarıldığı bazılarının ise yumuşatıldığı görülür.
- Filtre çeşitleri
 - Alçak geçiren filtre (Low Pass Filter – LPF)
 - Yüksek geçiren filtre (High Pass Filter – HPF)
 - Bant geçiren filtre
 - Bant durdurucu filtre

Alçak Geçiren Filtre

- LPF belli bir frekanstan küçük frekansa ait bilgileri serbest bırakarak diğerlerini durdurun filtredir
- İdeal bir LPF için transfer fonksiyonu aşağıdaki gibidir.

$$H_{LP}(u, v) = \begin{cases} 1 & r(u, v) \leq r_0 \\ 0 & r(u, v) > r_0 \end{cases}$$

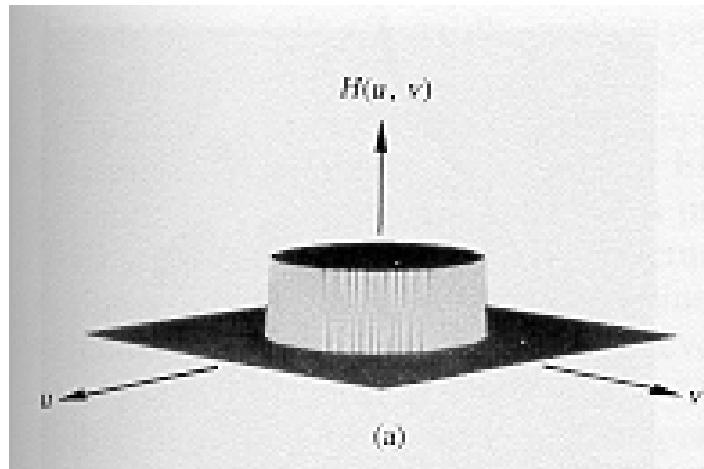
r_0 : Negatif olmayan bir değer, (Cut-off frekansı)

$r(u, v)$: Frekans ortamında, (u, v) noktasından orjine uzaklık

$$r(u, v) = \sqrt{u^2 + v^2}$$

Alçak Geçiren Filtre

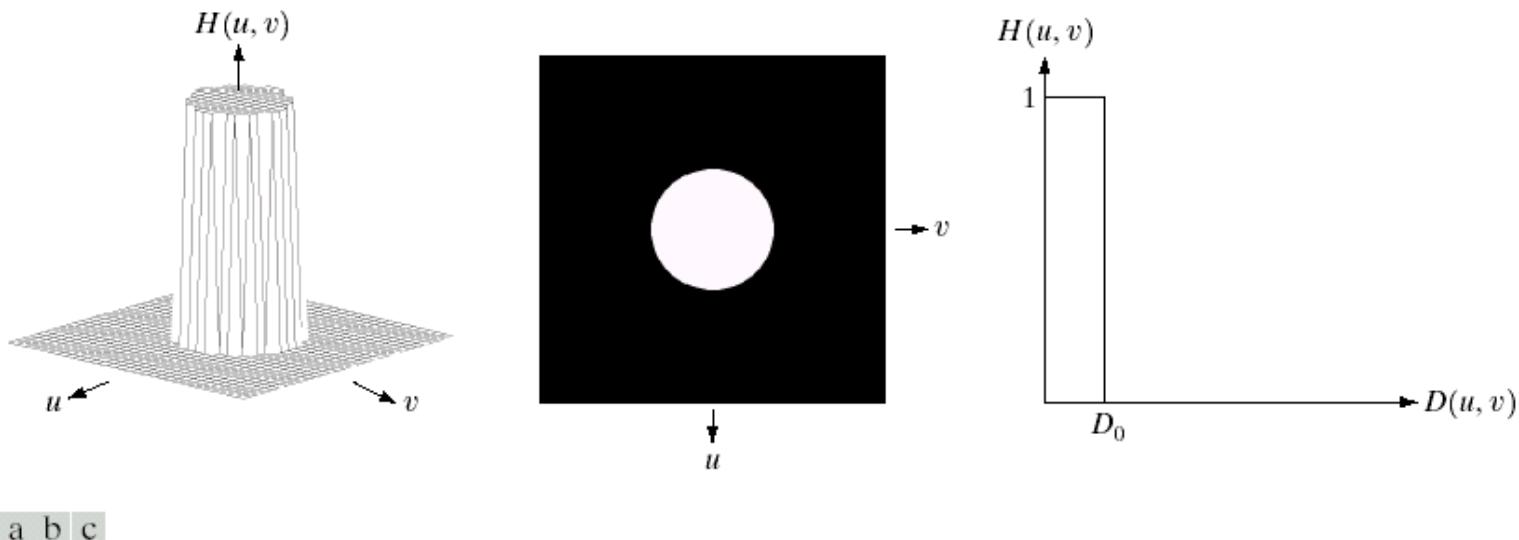
- 2D için cut-off frekansları bir çember içinde toplanır



$$H(u, v) = \begin{cases} 1 & \text{if } u^2 + v^2 \leq D_0^2 \\ 0 & \text{otherwise} \end{cases}$$

Alçak Geçiren Filtre

- İdeal LPF

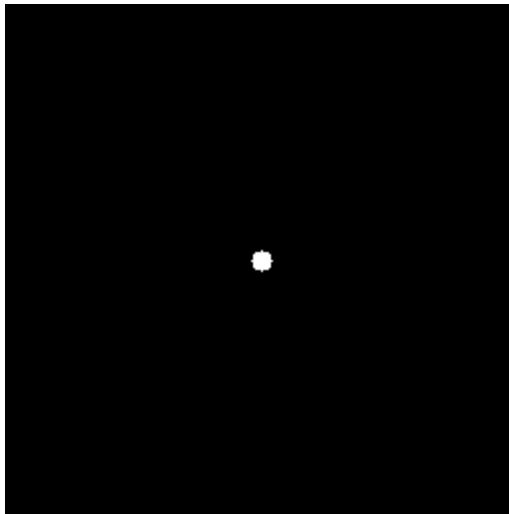


a b c

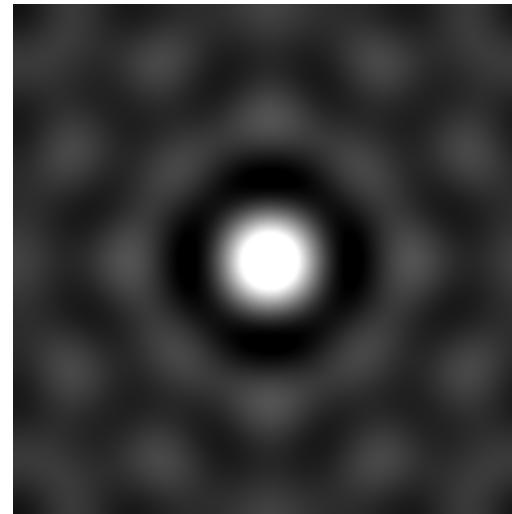
FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Alçak Geçiren Filtre

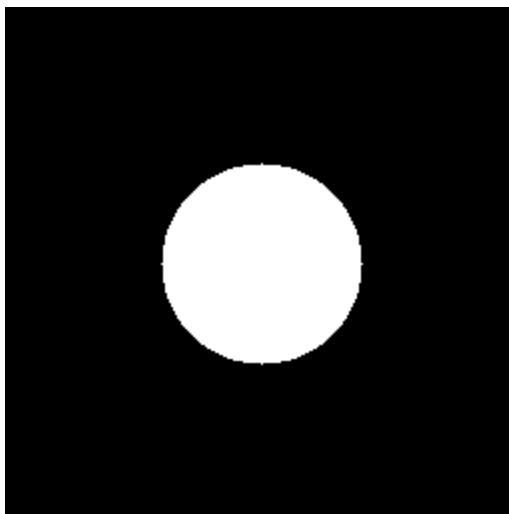
N = 5



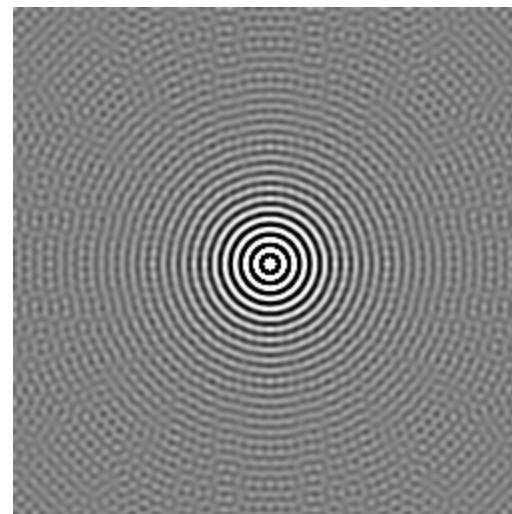
Ters FT - N = 5



N = 50



Ters FT - N = 50



Alçak Geçiren Filtre

- Alçak geçirgen filtreleme işleminde, filtre fonksiyonu
 - şeklin merkezinde toplanan frekansların geçişine izin verirken,
 - kenarlarda toplanan yüksek frekansların geçişine izin vermeyip süzmektedir.
- Böylece yersel ortamda piksellerdeki ani geçişler ortadan kalkar ve görüntü yumuşar, bulanıklaşır.

Alçak Geçiren Filtre

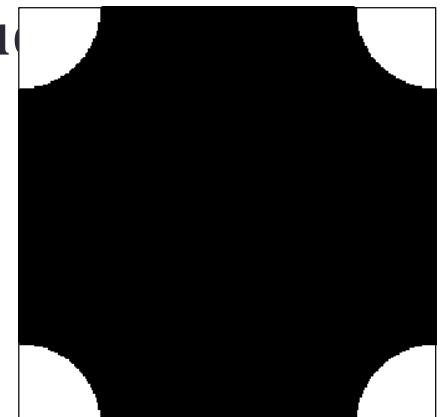
- Düşük freksanslı komponentler imgenin yavaş değişen karakteristiklerinden sorumludur
 - Genel kontrast ve ortalama yoğunluk gibi
- İmgeye uygulanması durumunda imge bulanıklaşır, çünkü imgenin yüksek frekanslı elemanlarına katkıda bulunan sınırları ve imgedeki diğer keskin detayları azaltır.

İşlemler

- DFT işlemi bir imge için 2 boyutlu olacaktır. Bu nedenle **fft2** fonksiyonu kullanılır
- H için aşağıdaki kuralı sağlayan bir filtre tasarlanır.

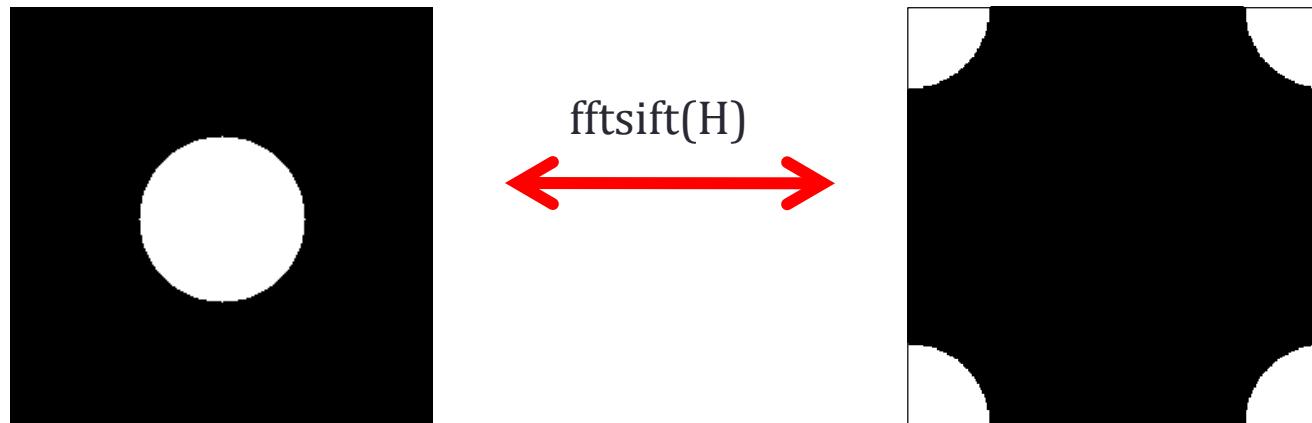
$$H(u, v) = \begin{cases} 1 & \text{if } u^2 + v^2 \leq D_0^2 \\ 0 & \text{otherwise} \end{cases}$$

- Burada merkez noktanın (0,0) noktası olması gerektiğini ve ancak işlem esnasında H'nin merkezleştirilmemesi gerektiğini unutmayın. Kısaca oluşan H aşağıda olmalıdır.



İşlemler

- F ve H eleman eleman çarpılır (\cdot^* işlemi)
- Elde ettiğiniz sonucun ters FT alarak bunun reel kısmını alın.
- Ters FT : ifft2
- Merkezileştirme işlemi (ya da tersi için)
 - fftshift fonksiyonu kullanılır



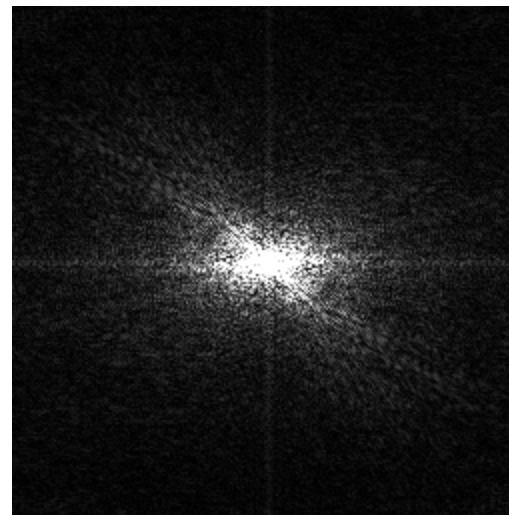
Hatırlatma

- Fourier Transform işlemi komplex bir işlemidir.
- Elde edilen sonuçlar karmaşık sayıdır
- Çizim yapılırken genliği (abs ya da real kısmı) ya da fazı (angle) çizdirilir.

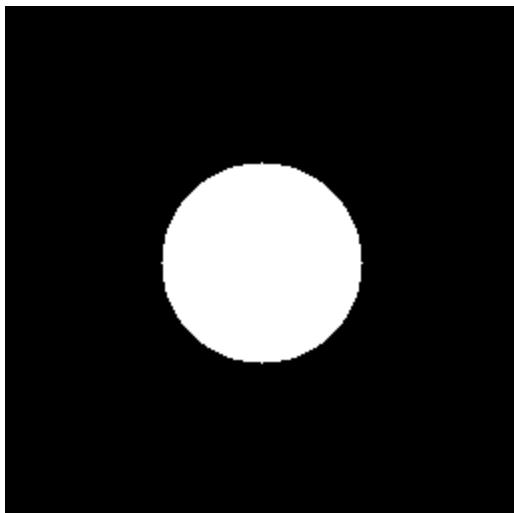
Original Lena Image



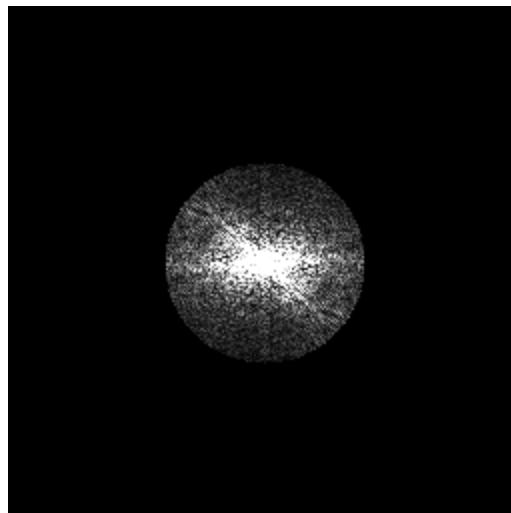
FT of Lena Image



LPF : N = 50



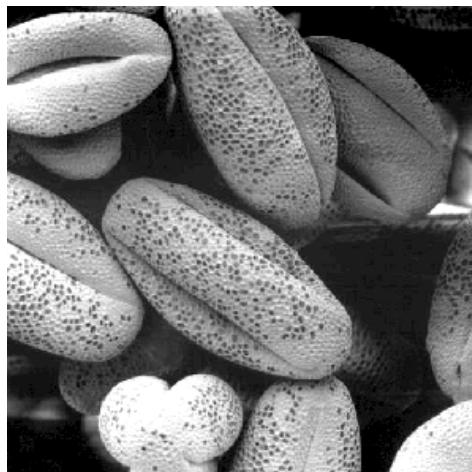
FT of Low passed Filtered Lena Image



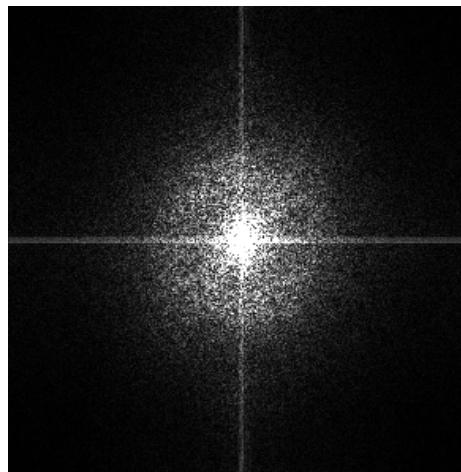
Reconstructed Lena Image After LPF



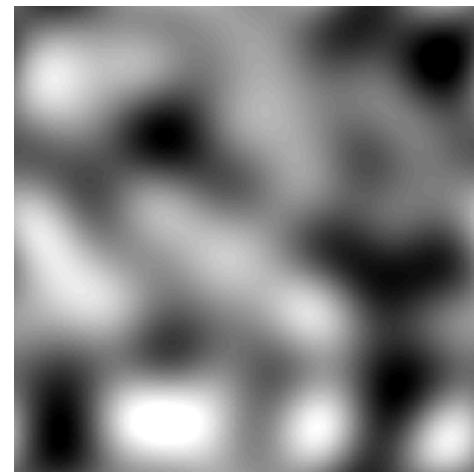
Orjinal Img



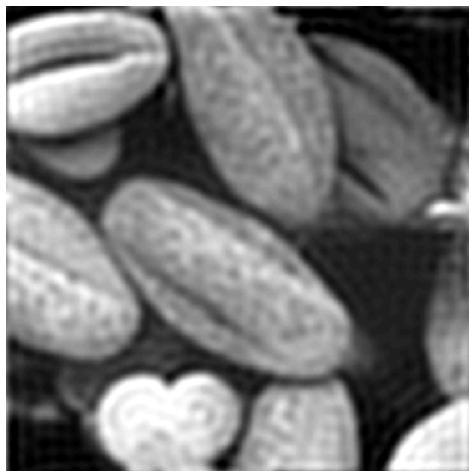
İmgenin Fourier Transformu



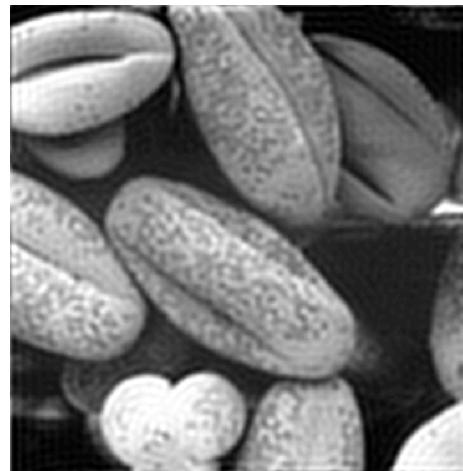
Reconstructed Image After LPF N = 5



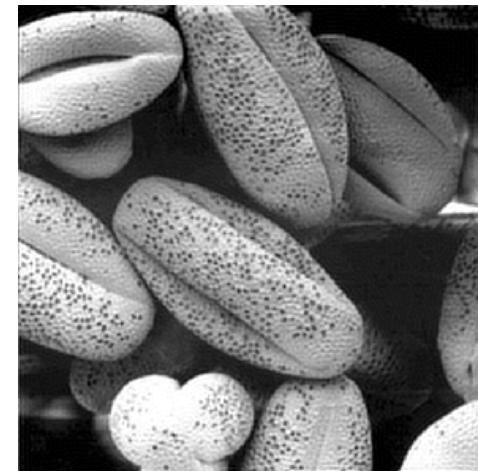
Reconstructed Image After LPF N = 30



Reconstructed Image After LPF N = 50



Reconstructed Image After LPF N = 100



Filtreleme İşlemi : İzlenecek Adımlar

1. $f(x,y)$ imgesinin boyutları $M \times N$ 'den sıfır ekleme (zero padding) yapılarak $P \times Q$ 'ya çıkarılır. (Zero padding işlemi)
2. $f_p(x,y)$ imgesi $(-1)^{(x+y)}$ ile çarpılarak şekil ortalanır
3. 2. adımda elde edilen İmgenin ($f_p(x,y)$) DFT'si ($F(u,v)$) hesaplanır
4. Filtreye de sıfır ekleme işlemi yapılır şekil ortalanır ve filtrenin DFT'si (H) hesaplanır
 - H ve F aynı boyutta olmalı
5. F ve H çarpılır (element by element)
 - $G(u,v)=H(u,v)F(u,v)$
6. Ters DFTnin **real** kısmı alınır
 - $g_p(x,y)=\text{real}\{ F^{-1}\{ G(u,v) \} \} (-1)^{(x+y)}$
 - Son olarak elde edilen imgenin boyutları tekrar $M \times N$ 'e çekilir

Zero Padding işlemi

- Konvolusyon işlemi sınırlı sayıda örnekli işaretler için çembersel olarak yapılır.
 - 0 noktası N noktası ile komşu kabul edilir
 - Bu wraparound error adı verilen bozulmalara yol açar
 - İki nokta arasında büyük fark varsa yüksek frekanslı komponentler oluşur
- İmgelerin etrafına sıfır ekleme işlemi çok yaygın olarak kullanılır

$$f_p(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A - 1 \text{ and } 0 \leq y \leq B - 1 \\ 0 & A \leq x \leq P \text{ or } B \leq y \leq Q \end{cases}$$

$$h_p(x, y) = \begin{cases} h(x, y) & 0 \leq x \leq C - 1 \text{ and } 0 \leq y \leq D - 1 \\ 0 & C \leq x \leq P \text{ or } D \leq y \leq Q \end{cases}$$

KURAL: $P \geq A + C - 1$
 $Q \geq B + D - 1$

Filtreleme İşlemi : İzlenecek Adımlar 1 ve 2

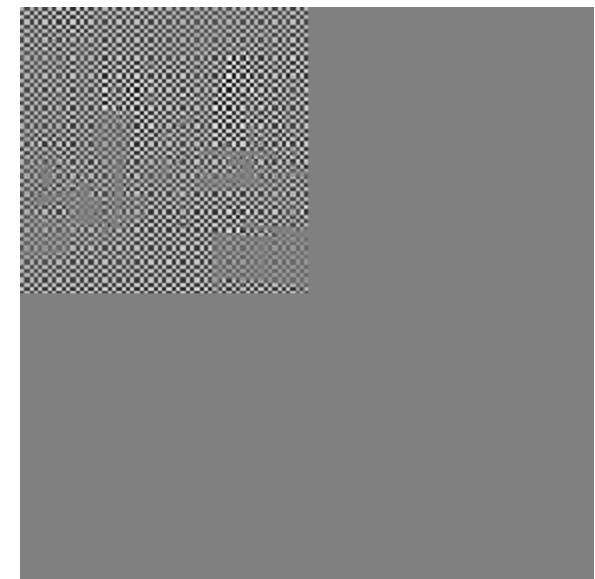
Orjinal İmge: $f(x,y)$



$f_p(x,y)$: Sıfır Eklendi

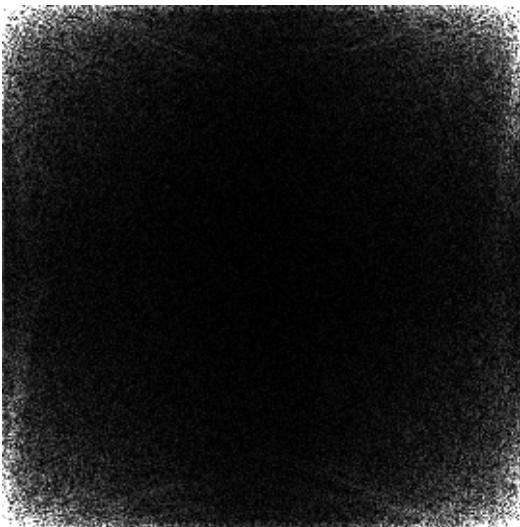


$f_p(x,y)(-1)^{(x+y)}$: Çarpım Yapıldı

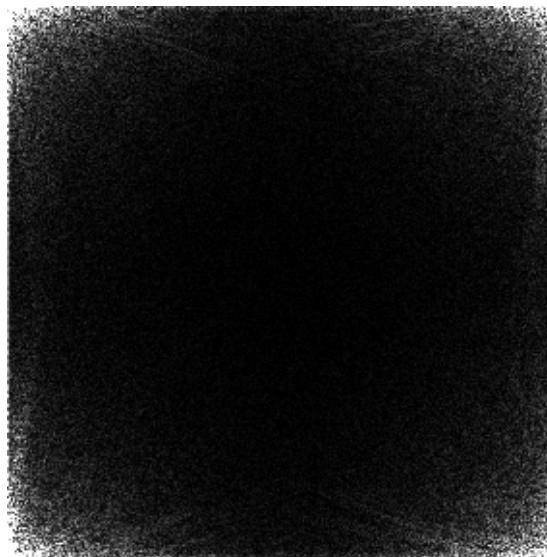


Filtreleme İşlemi : İzlenecek Adımlar 3

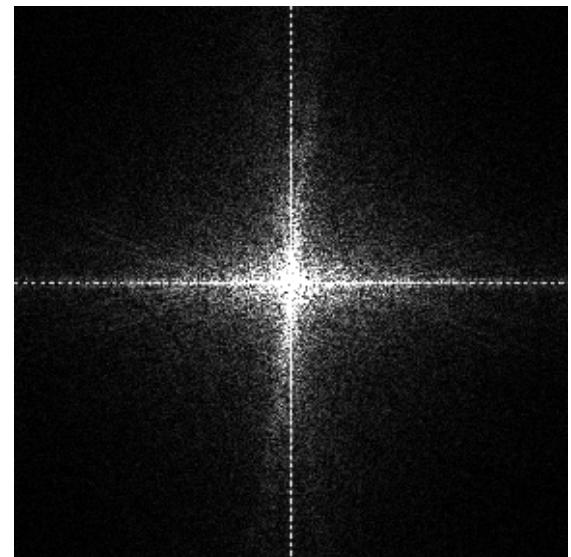
Orjinal Spectrum: $F(u,v)$



Zero-Padded Imgae: $F_p(u,v)$

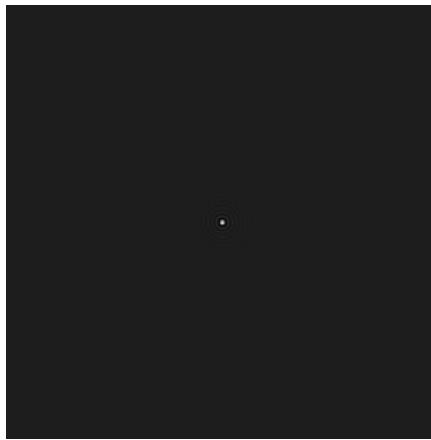


Ortalanmış Spectrum: $F^{-1}(f_p(x,y)(-1)^{(x+y)})$



Filtreleme İşlemi : İzlenecek Adımlar 4

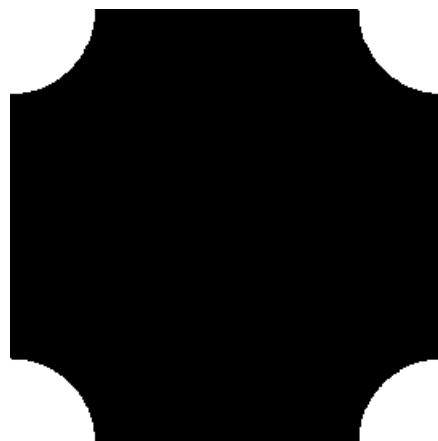
$h(x,y)$



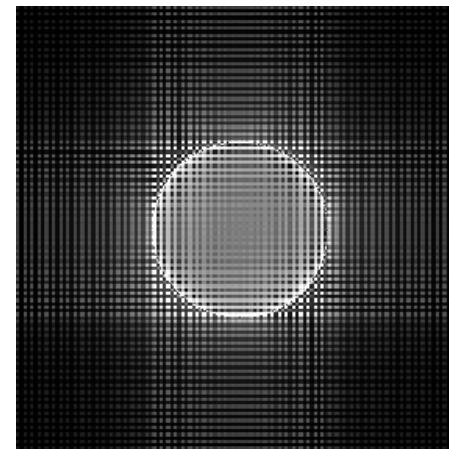
$h(x,y)$: Sıfır Eklendi



$H(u,v)$: Fourier Dönüşümü Sonucu

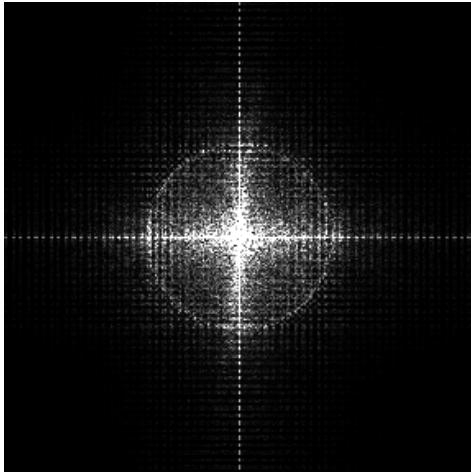


$$H(u,v) = F \{ h_p(x,y)(-1)^{(x+y)} \}$$

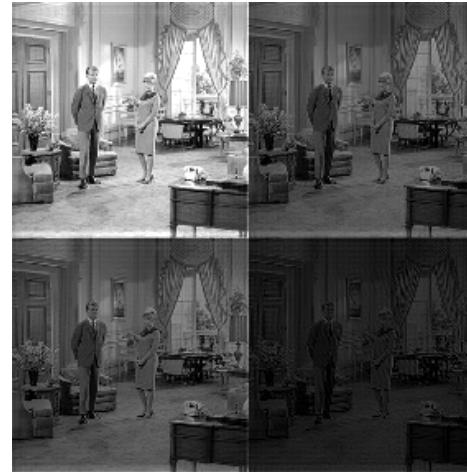


Filtreleme İşlemi : İzlenecek Adımlar 5,6

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

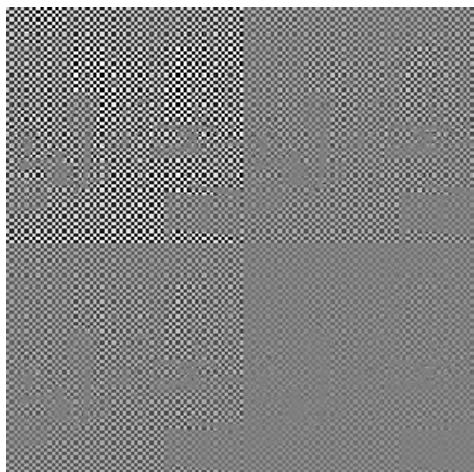


$$G(u,v) = \text{real}(F^{-1}(F(u,v)H(u,v)))(-1)^{(x+y)}$$



LPF filtrelenmiş İmge

$$G(u,v) = \text{real}(F^{-1}(F(u,v)H(u,v)))$$



Örnekler

Orjinal İmge: $f(x,y)$



LPF filtrelenmiş İmge



Orjinal İmge: $f(x,y)$



LPF filtrelenmiş İmge



Orjinal İmge: $f(x,y)$



LPF filtrelenmiş İmge



Cut-off frekansı ve Smoothing

- Hatırlatma: Frekans ortamında çarpmaya zaman ortamında konvolusyon işlemi demektir (tersi de geçerlidir)

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

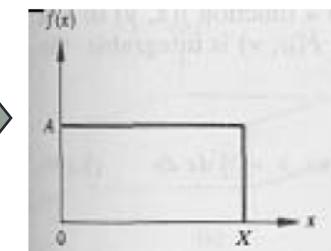


$$f(x, y) * h(x, y) = g(x, y)$$

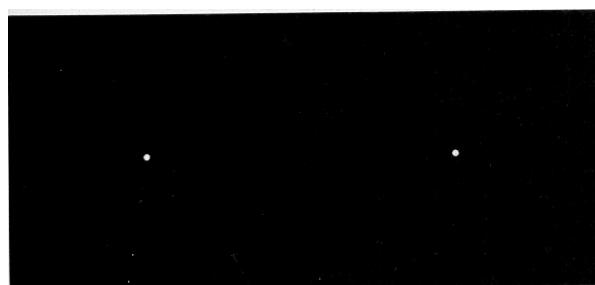
zaman ortamı



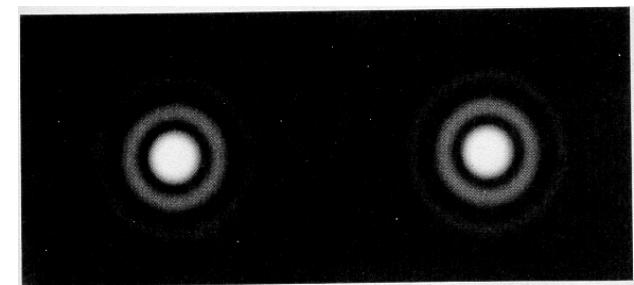
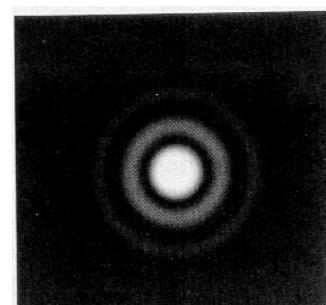
Frekans ortamı



$$f(x, y) = \delta(x + x_0, y) + \delta(x - x_0, y)$$

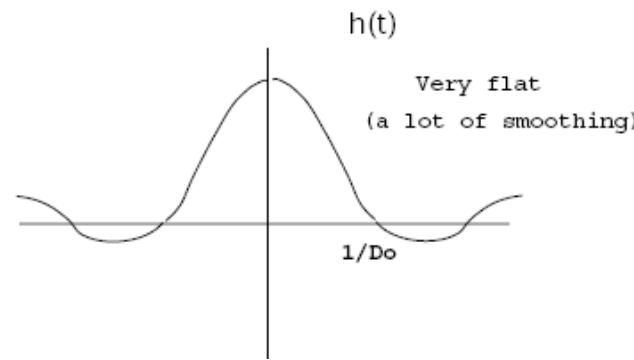
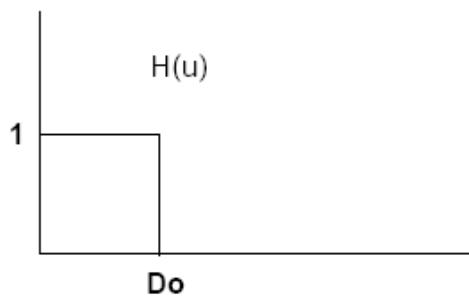
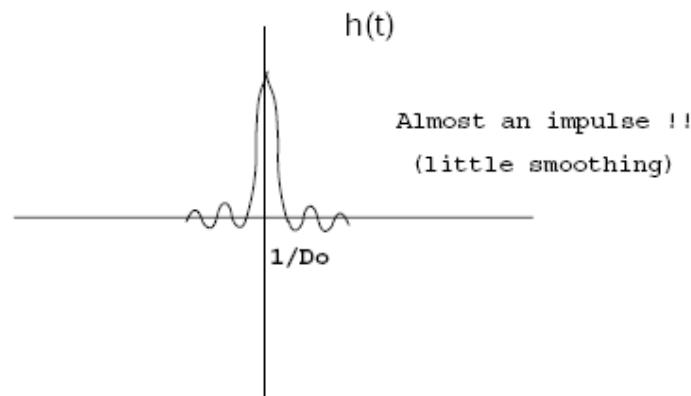


*



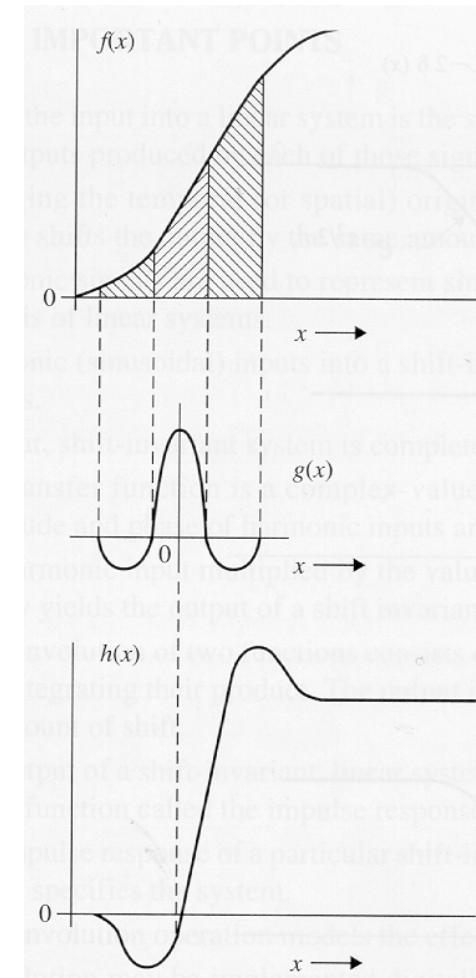
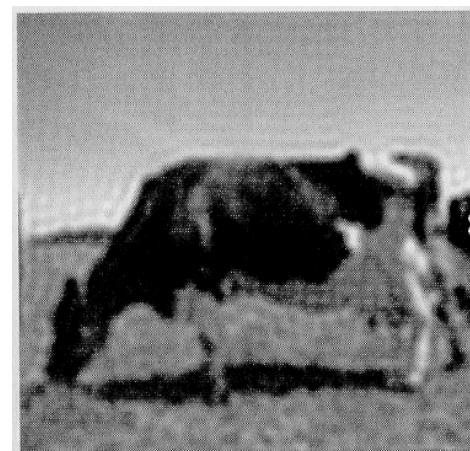
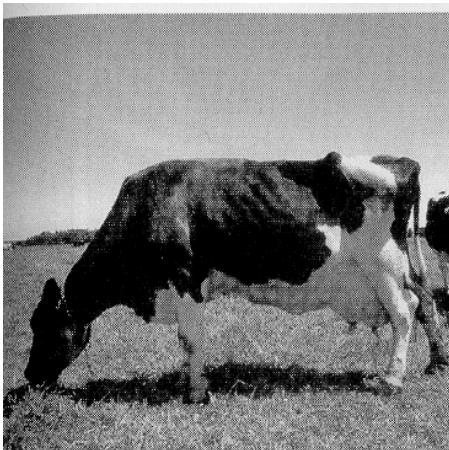
Cut-off frekansi ve Smoothing

- Blurring D_0 ile kontrol edilir



Ringing Effect

- Çok keskin kesim frekansları imgenin bu frekans etrafındaki noktalarında aşırıya gidilmesine neden olur (**ringing effect**).



Diğer Filtreler

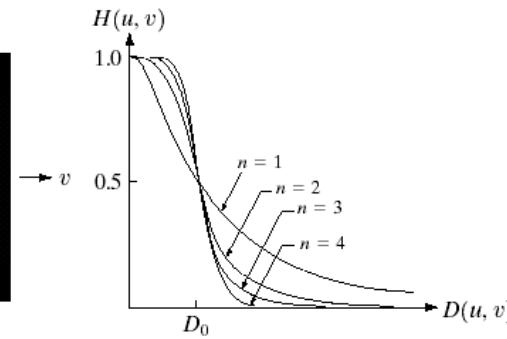
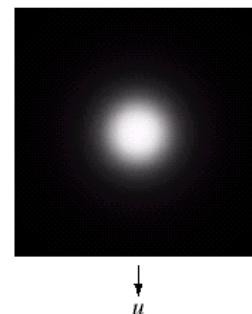
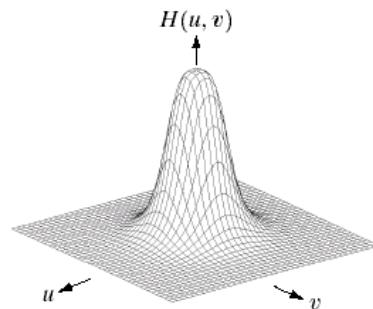
- Ideal filtrelerin dışında aşağıdaki filtreler de kullanılabilir.
 - Butterworth
 - Gaussian

Butterworth LPF

- BLPF keskin geçişlere sahip değildir
- BLPF için kesim frekansı transfer fonksiyonunun maksimum değerinin yarısına eşit olduğu yerdeki frekanstır
- n. dereceden D_0 kesim frekanslı Butterworth LPF için transfer fonksiyonu aşağıdaki gibidir:

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$

$D(u, v)$: (u, v) 'nin orijine uzaklığı

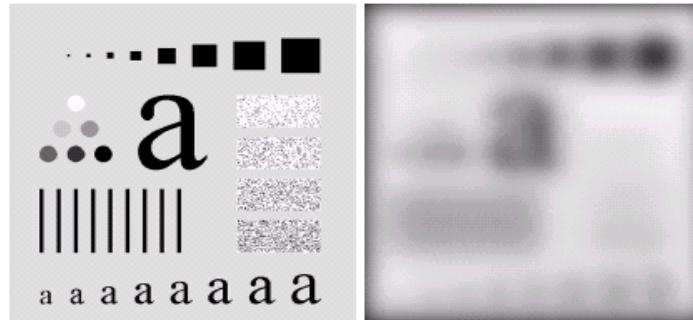


a b c

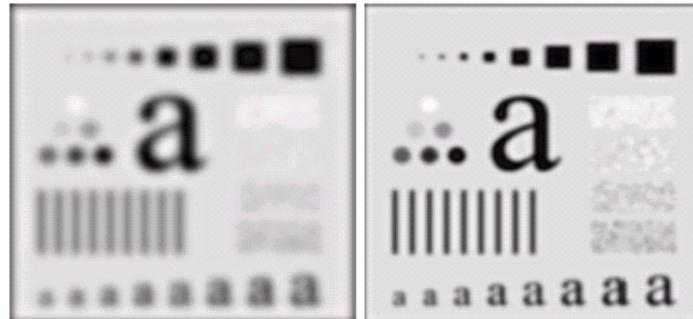
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Butterworth LPF

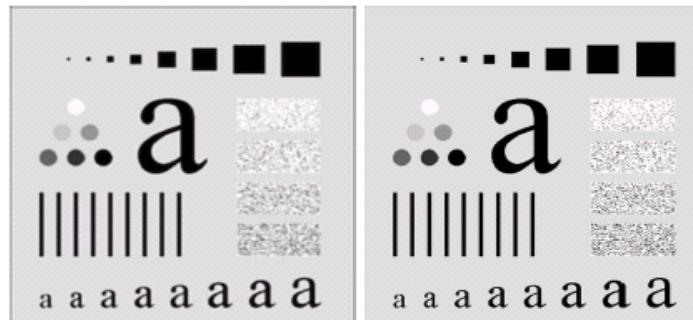
Orijinal Image



BLPF
Order=2
cutoff yarıçapı= 15



BLPF
Order=2
cutoff yarıçapı= 80

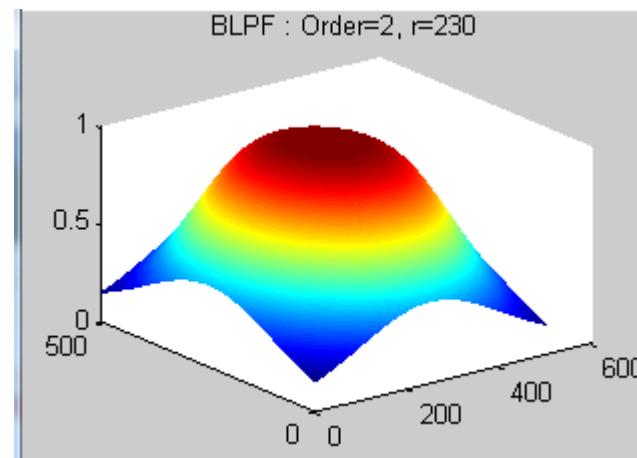
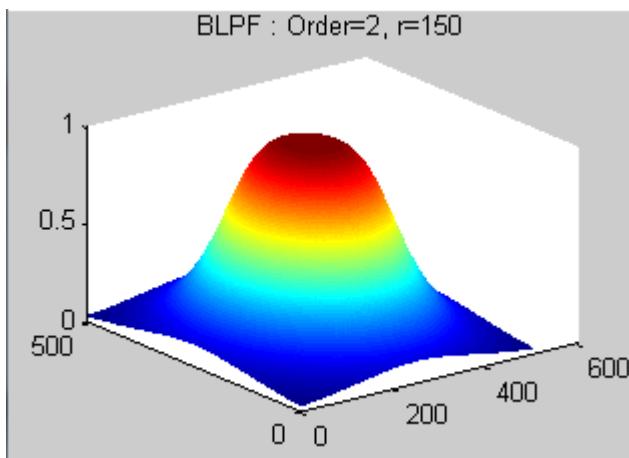
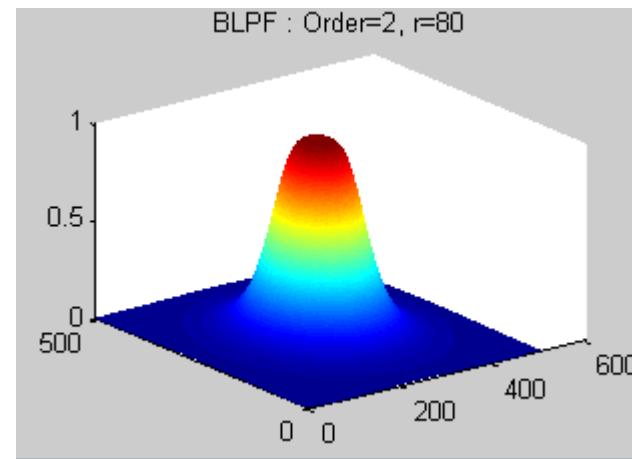
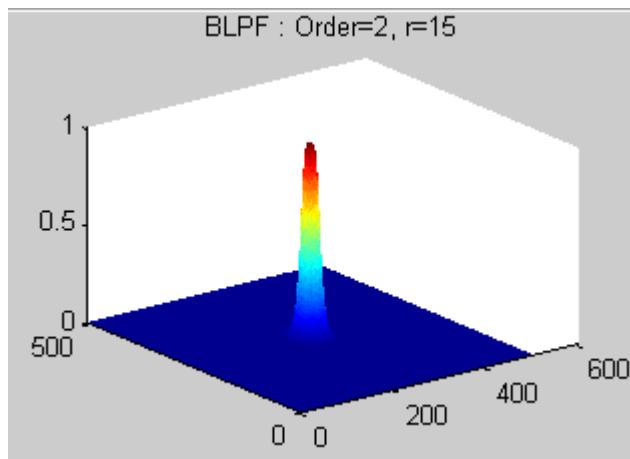


BLPF
Order=2
cutoff yarıçapı= 5

BLPF
Order=2
cutoff yarıçapı= 30

BLPF
Order=2
cutoff yarıçapı= 230

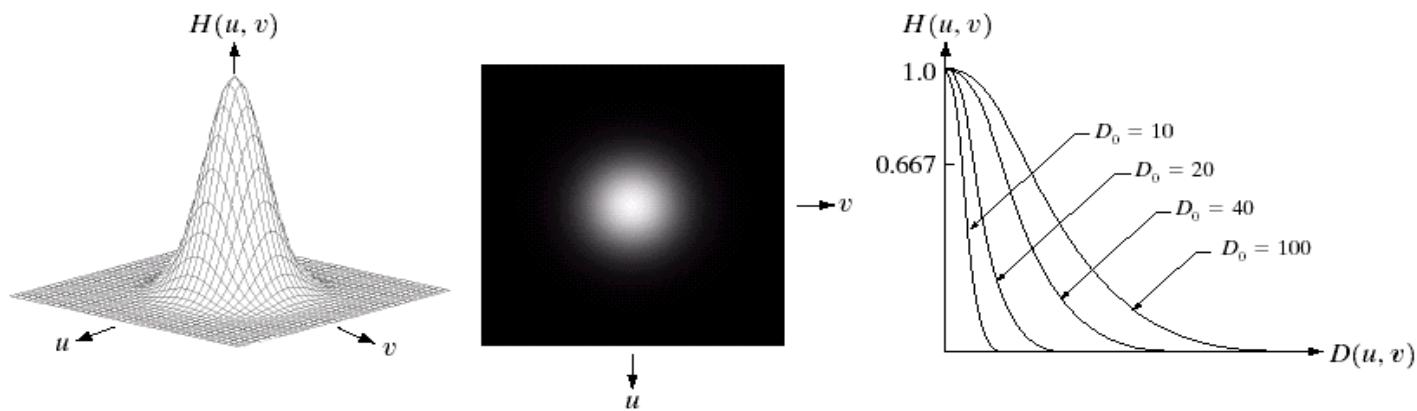
Butterworth LPF : Transfer Fonksiyonları



Gaussian LPF

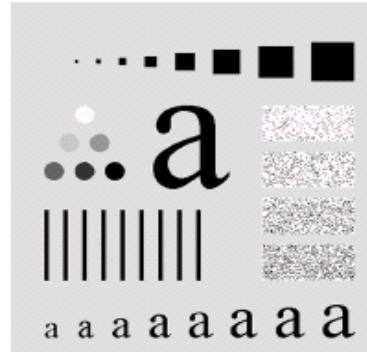
- D_0 kesim frekanslı Gaussian LPF için transfer fonksiyonu aşağıdaki gibidir:

$$H(u, v) = e^{-D^2(u,v)/2D_0^2}$$

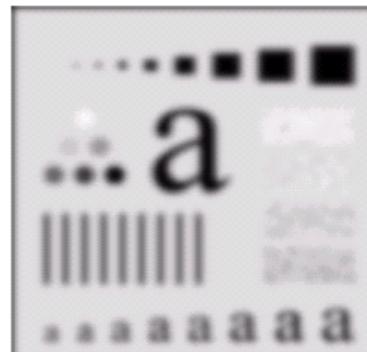


Gaussian LPF

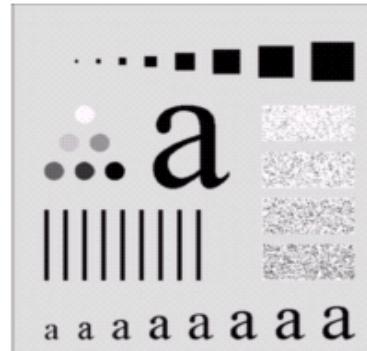
Orijinal Image



Gaussian LPF
cutoff yarıçapı= 15



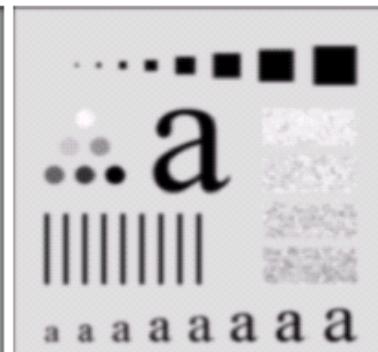
Gaussian LPF
cutoff yarıçapı= 80



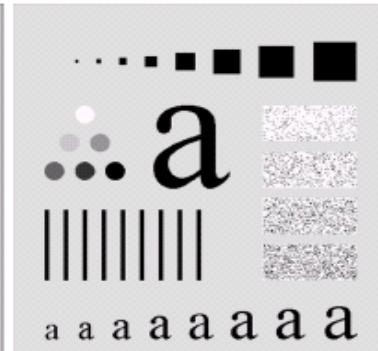
Gaussian LPF
cutoff yarıçapı= 5



Gaussian LPF
cutoff yarıçapı= 30



Gaussian LPF
cutoff yarıçapı= 230



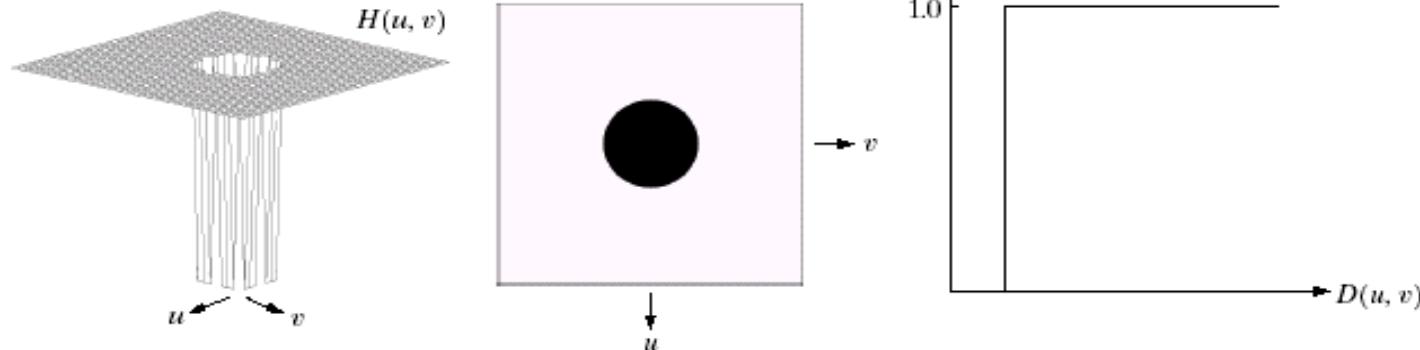
Yüksek Geçiren Filtre

- HPF belli bir frekanstan büyük frekansa ait bilgileri serbest bırakarak diğerlerini durdurun filtredir
- İdeal bir HPF için transfer fonksiyonu aşağıdaki gibiidir.

$$H_{HP}(u, v) = \begin{cases} 0 & r(u, v) \leq r_0 \\ 1 & r(u, v) > r_0 \end{cases} = 1 - H_{LP}(u, v) \quad r(u, v) = \sqrt{u^2 + v^2}$$

r_0 : Negatif olmayan bir değer, (Cut-off frekansı)

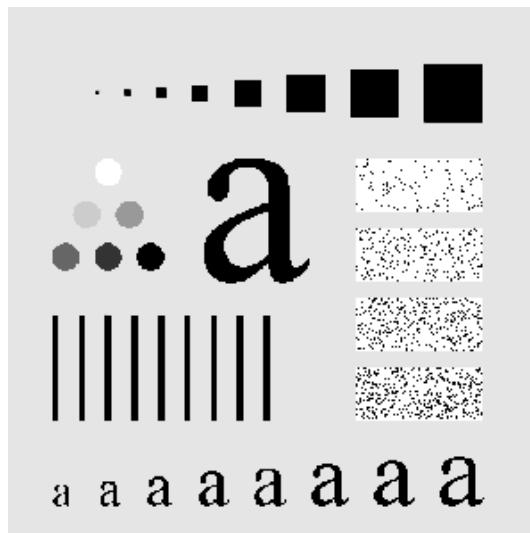
$r(u, v)$: Frekans ortamında, (u, v) noktasından orjine uzaklık



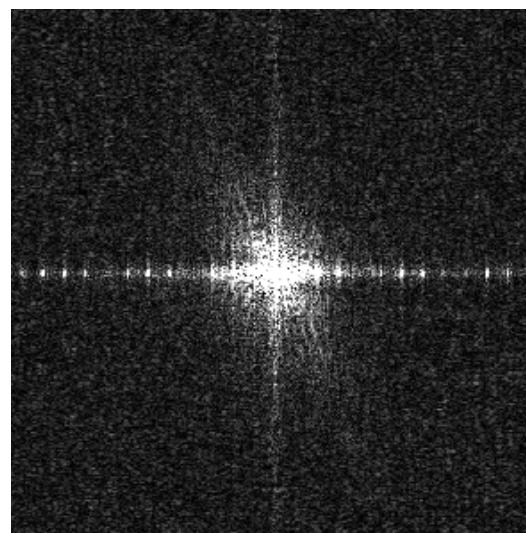
Görüntü Üzerinde Fourier Dönüşümü

- Yüksek geçirgen filtreleme işlemi LPF işleminin tam tersi bir yaklaşımı
• şeklin merkezindeki frekansları süzerken
• kenarlardaki yüksek frekansların geçişine izin vermektedir.
- Yersel x,y koordinatlarına geri dönüldüğünde keskin hatların daha da güçlenmesi ve belirgin bir şekilde görülmesi sağlanır.

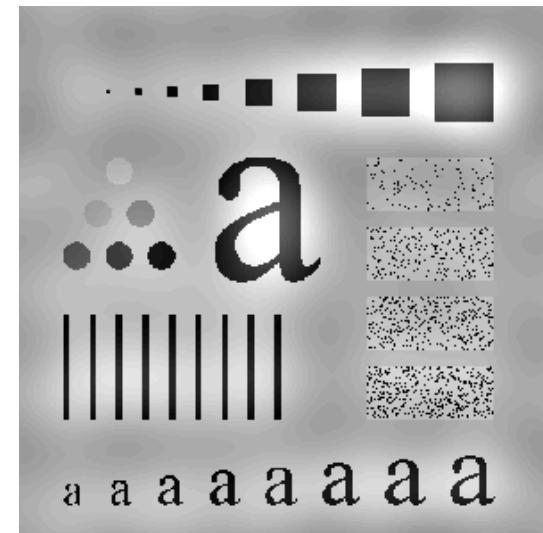
Orjinal Img



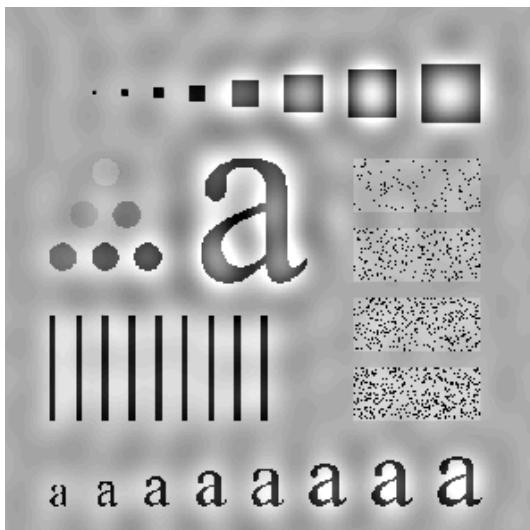
İmgenin Fourier Transformu



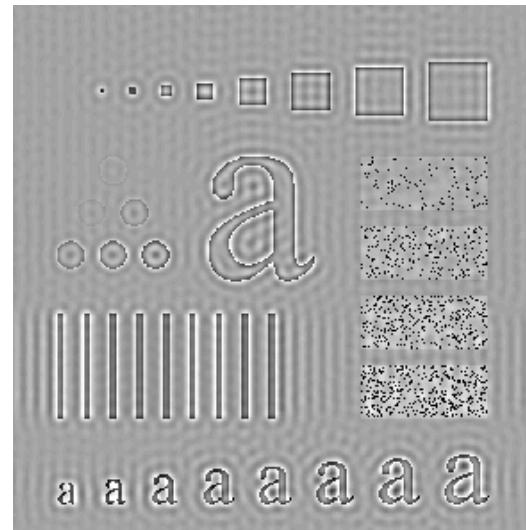
Reconstructed Image After HPF N = 5



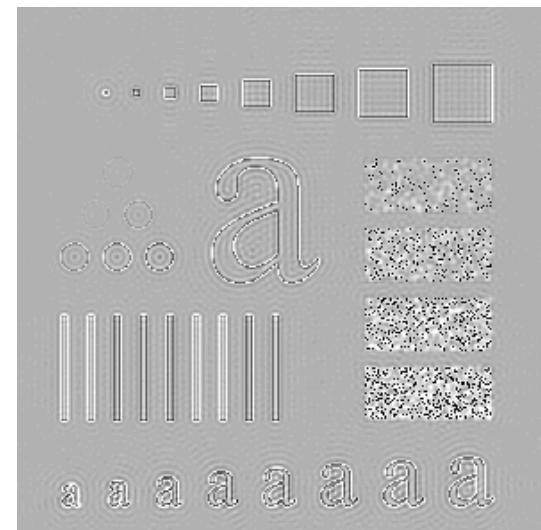
Reconstructed Image After HPF N = 10



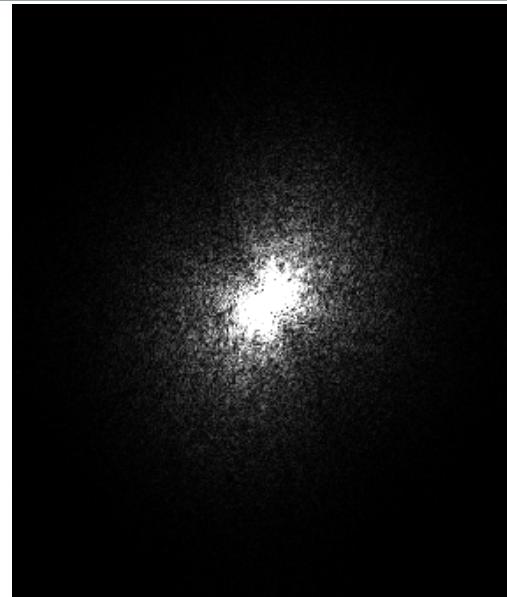
Reconstructed Image After HPF N = 40



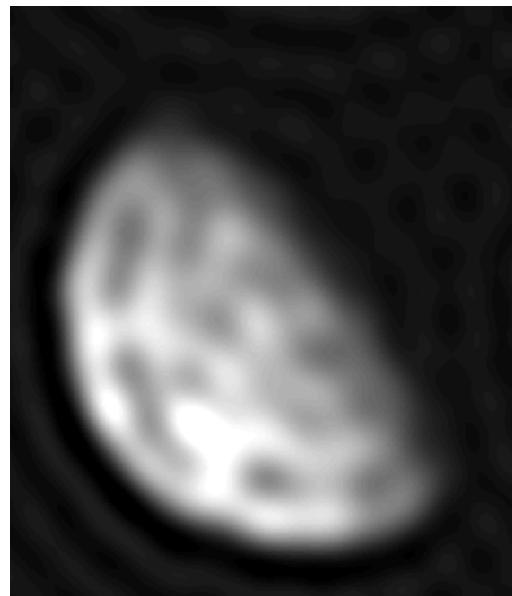
Reconstructed Image After HPF N = 80



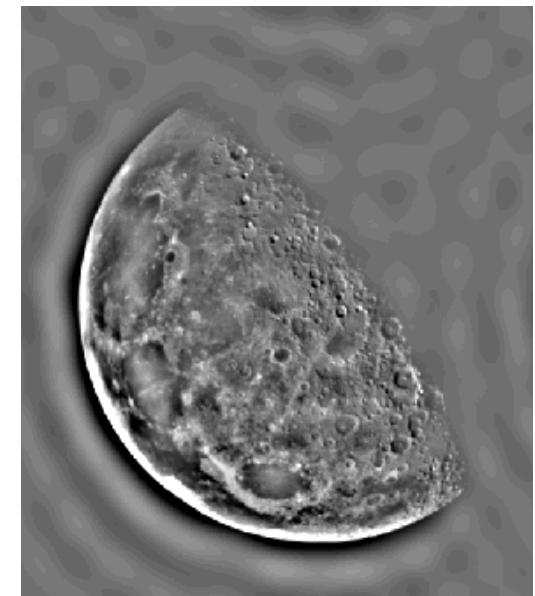
Orjinal İmge



Reconstructed Image After LPF N = 10



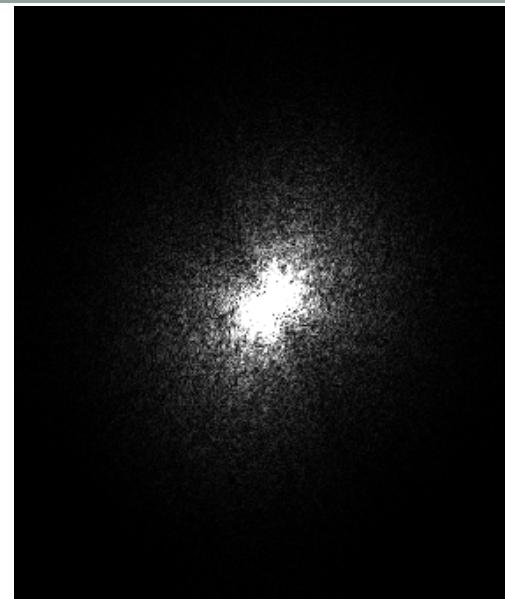
Reconstructed Image After HPF N = 10



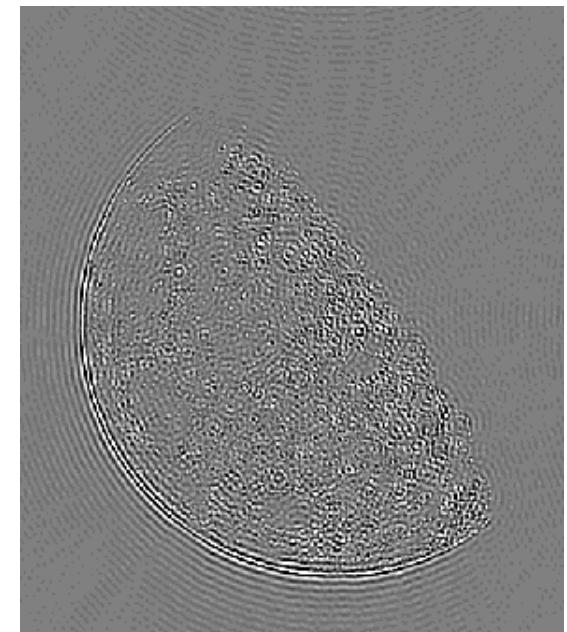
Orjinal İmge



Reconstructed Image After LPF N = 80



Reconstructed Image After HPF N = 80

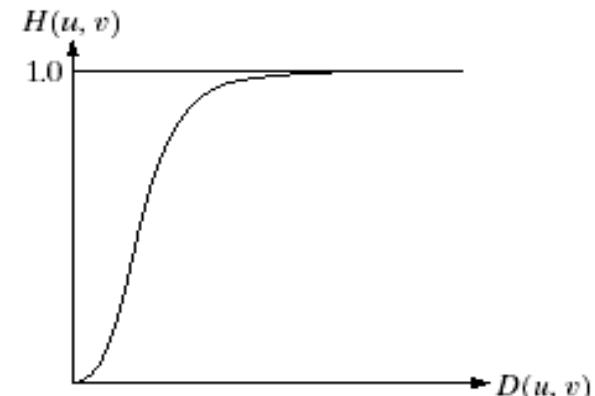
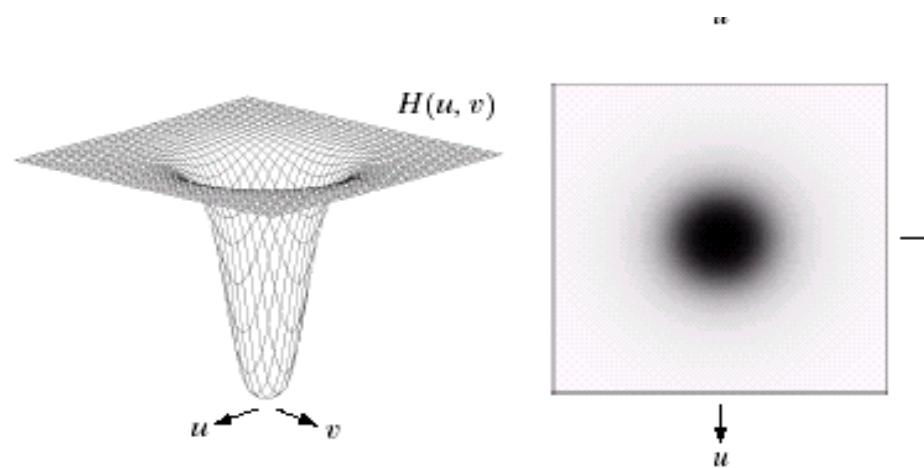


Butterworth HPF

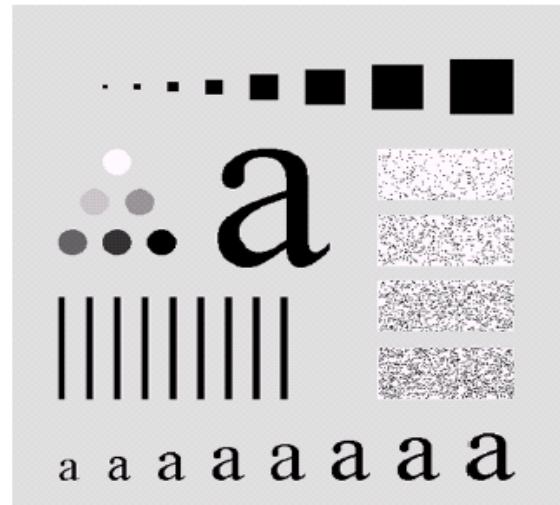
- n. dereceden D_0 kesim frekanslı Butterworth HPF için transfer fonksiyonu aşağıdaki gibidir:

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

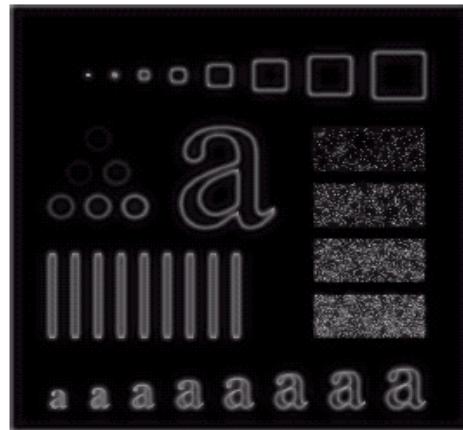
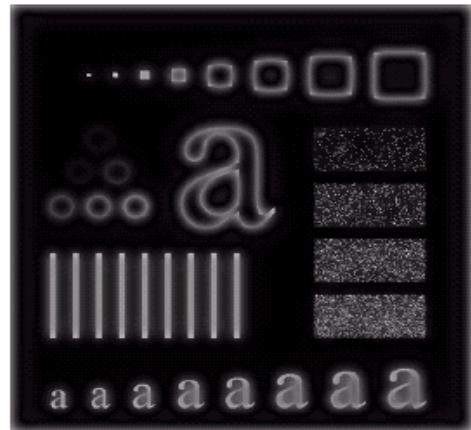
$D(u, v)$: (u, v) 'nin orijine uzaklığı



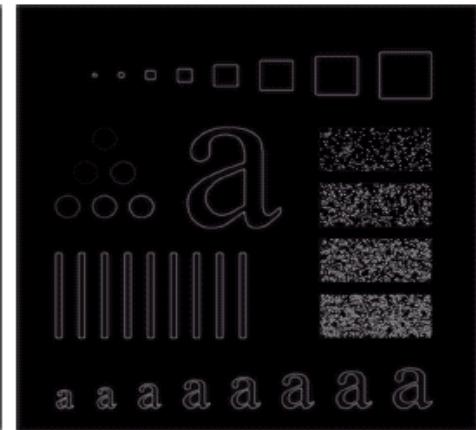
Butterworth HPF



BHPF
Order=1
cutoff = 15



BHPF
Order=1
cutoff = 30

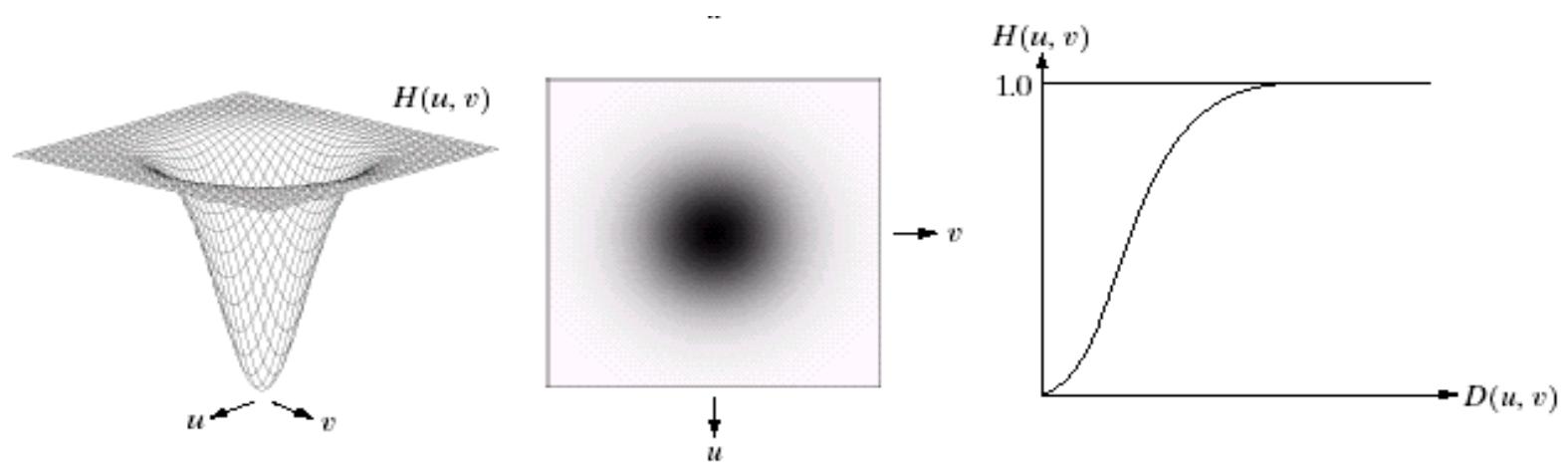


BHPF
Order=1
cutoff = 80

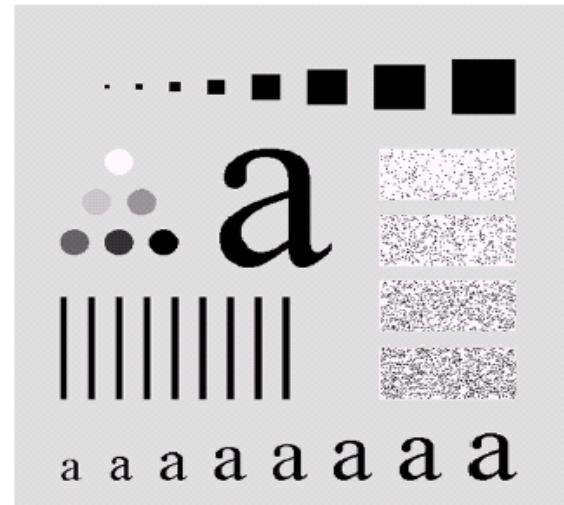
Gaussian HPF

- D_0 kesim frekanslı Gaussian LPF için transfer fonksiyonu aşağıdaki gibiidir:

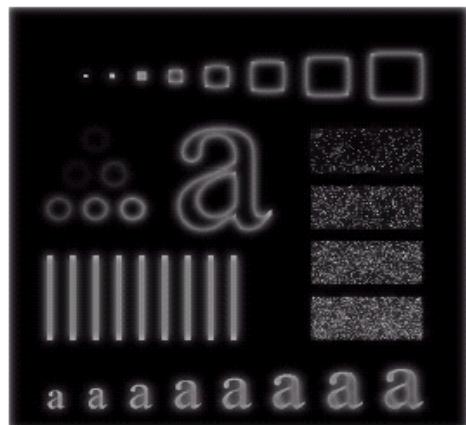
$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$$



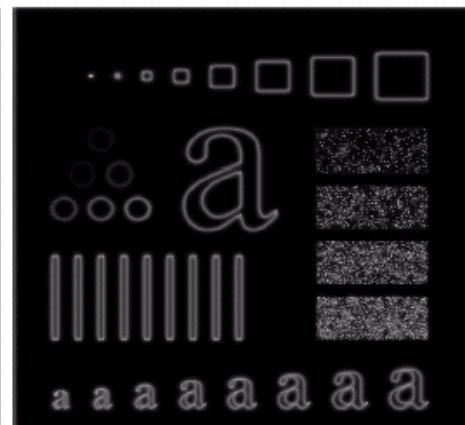
Gaussian HPF



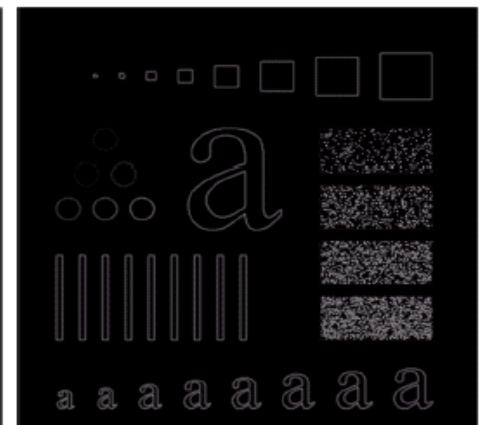
Gaussian HPF
cutoff = 15



GaussianHPF
cutoff = 30



Gaussian HPF
cutoff = 80



Seçici Filtreler

- Daha önceki filtreler tüm frekans düzleminde işlem görünüyordu
- Bazı durumlarda sadece belli bir frekans aralığı üzerinde işlem yapmak gerekebilir
- *Bandreject ya da Bandpass filtreler belli frekans aralıklarında işlem görür*
- *Notch filtre frekans bölgesinin çok küçük bir alanında işlem görür*

Bant Durduran (Band Reject) Filtreler

- Diğer filtreler gibi tasarlanır. Belli bir bant aralığını reddederek diğerlerini geçirmeyi sağlar
 - $D(u,v)$ orijinden olan uzaklık
 - D_0 ise istenen bandın merkezi
 - W ise istenen bant genişliği
- İdeal Bant durdurucu Filtre

Butterworth
Bant durdurucu Filtre

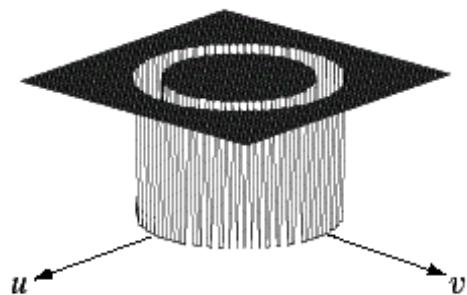
$$H(u, v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$$

$$H(u, v) = \frac{1}{1 + \left[\frac{DW}{D^2 - D_0^2} \right]^{2n}}$$

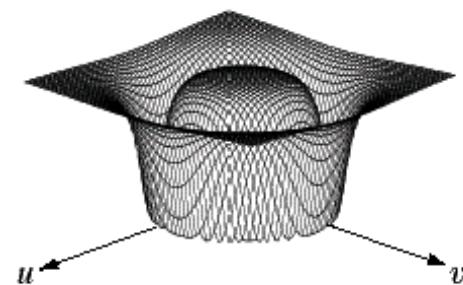
Gaussian Bant durdurucu Filtre

$$H(u, v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW} \right]^2}$$

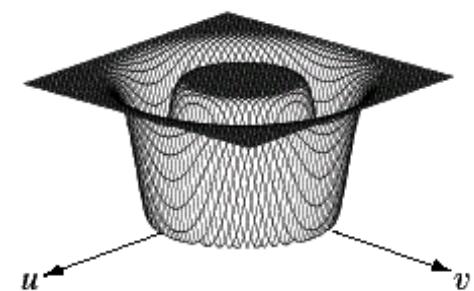
Bant Durduran (Band Reject) Filtreler



Ideal Band
Reject Filter



Butterworth
Band Reject
Filter (of order 1)



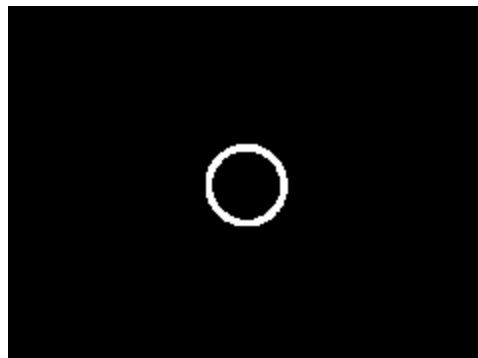
Gaussian
Band Reject
Filter

Bant Geçiren (Band Pass) Filtreler

- Bant durduran Filtreden geliştirilebilir:

$$H_{BP}(u, v) = 1 - H_{BR}(u, v)$$

Band Geçiren Filtre : $D_0=50$, $W = 10$



Band Durduran Filtre : $D_0=50$, $W = 10$

