

Görüntü İşleme Teknikleri - 2

Oğuzhan Öztaş

İçerik

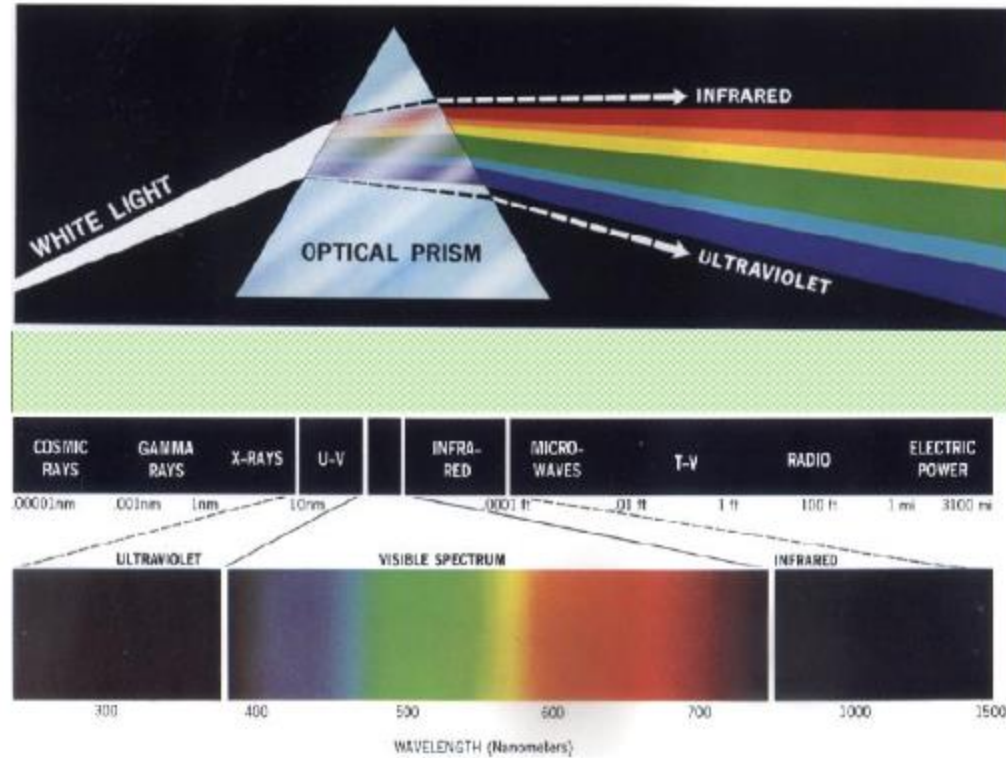
- Renkler
- Aydınlatan ve yansıyan ışık
- Üç renkli renk karışımı
- Renk modelleri
- Gri tonlu imge
- Frekans uzayı
- Fourier dönüşümü

İçerik

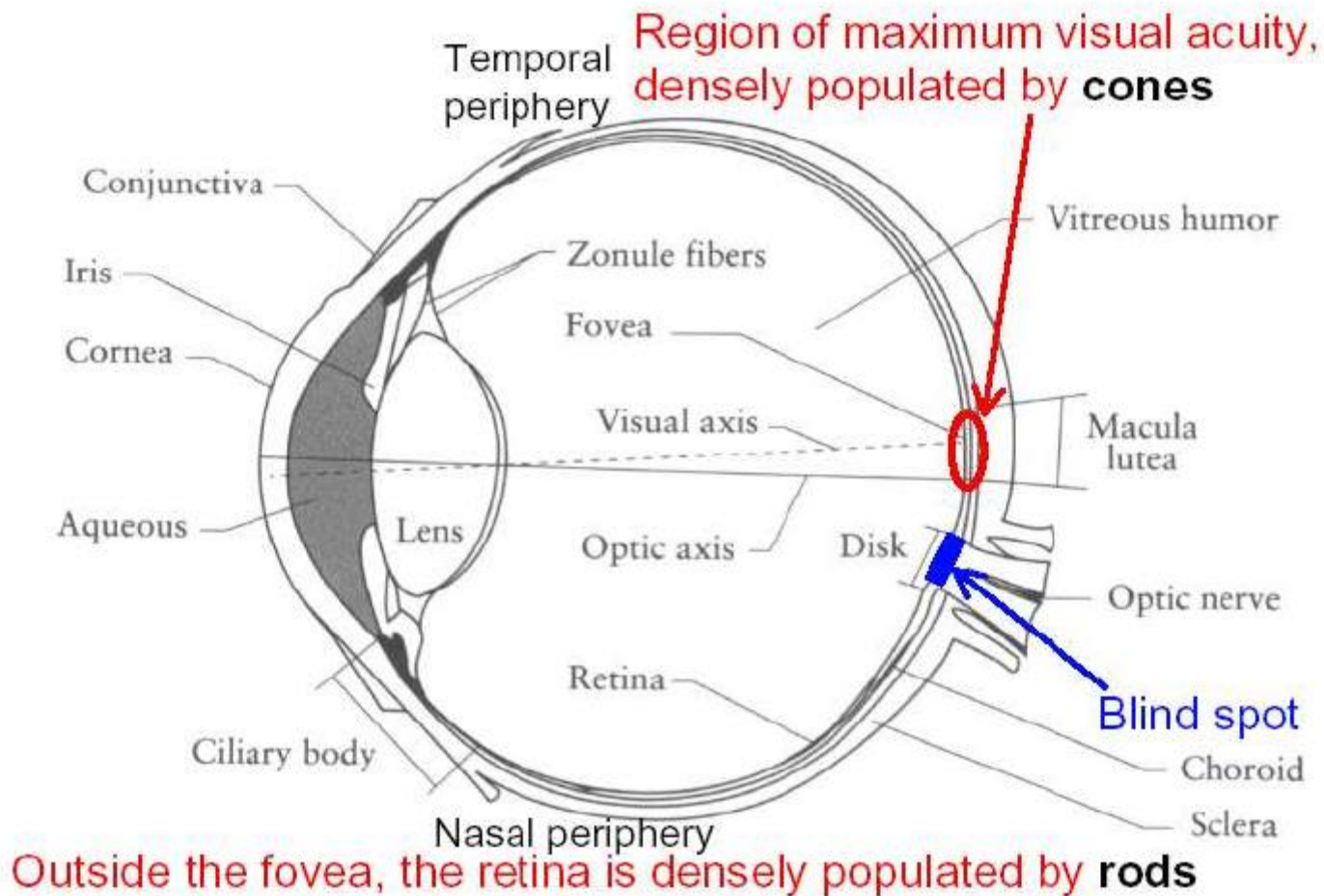
- Morfolojik imge işleme
- Yayma, aşındırma
- Açma, kapama

Renkler

Renk; 400nm-700nm arasında dalga boyuna sahip olan ışığın retinaya düşmesinin algılasal bir sonucudur. Işık EM dalgaların bir parçasıdır.



Renkler: İnsan Gözü



Renkler: İnsan Gözü

Retina koni (cone) ve çubuk (rod) şeklindeki algılayıcıları içerir:

- Koni: Gündüz görüşünü sağlar, renk tonlarını algılayabilir.
- Çubuk: Gece görüşünü sağlar, sadece parlaklığı (brightness) algılayabilir.

Renk Algılaması

- Işıklılık (Luminance) (veya Parlaklık (Brightness))
- Renklilik (Chrominance)
 - Hue : Renk özü, renk tonu
 - Saturation: Rengin saflığı, doygunluk.

Aydınlatan ve Yansıyan Işık

Aydınlatan Kaynaklar

- Işık yayan kaynaklardır (Güneş, Ampül, Monitörler)
- Algılanan renk yayılan frekansı bağlıdır.

Yansıtan Kaynaklar

- Gelen ışığı yansıtılar (Renkli boya, mat yüzeyler, giysiler)
- Algılana renk yansıyan (yayılan-emilen) frekansa bağlıdır.

Üç Renkli Renk Karışımı (Tri-chromatic Color Mixing)

Üç Renkli Renk Karışımı Teorisi:

Herhangi bir renk üç ana rengin doğru oranda karışımı ile elde edilebilir.

Aydınlatan Kaynaklar İçin Ana renkler:

- Kırmızı, Yeşil, Mavi (Red Green Blue-RGB)
- Örneğin renkli monitörler, farklı elektron tabancaları ile kırmızı, yeşil ve mavi fosfor yüzeylerin uyarılması ile çalışır.
- Beyaz üç rengin birleşiminden oluşur

Yansıtan Kaynaklar İçin Ana renkler (İkincil Renkler olarak da bilinir):

- Metal yeşili (Camgöbeği), Morumsu kırmızı (Galibarda), Sarı (Cyan, Magenta, Yellow - CMY)
- Renkli yazıcılar CMYK renklerini kullanırlar.



CMYK özellikle yazıcılar ve matbaalar için geliştirilen bir renk uzayı olmasına karşın, turuncu gibi bazı renklerin basımında tatmin edici sonuçlar vermemektedir. Yeşil gibi ana ve yaygın bir rengin, Cam göbeği ve Sarı bileşenlerinin karışımıyla elde edilmesinden dolayı CMYK "pahalı" bir baskı teknolojisi olarak yorumlanmaktadır. Bu nedenle günümüz matbaacılığında giderek artan bir şekilde altı renkli **Hexacrome** ya da [CMYKOG](#) dediğimiz yeni standart kullanılmaktadır.

CMYKOG Nedir?



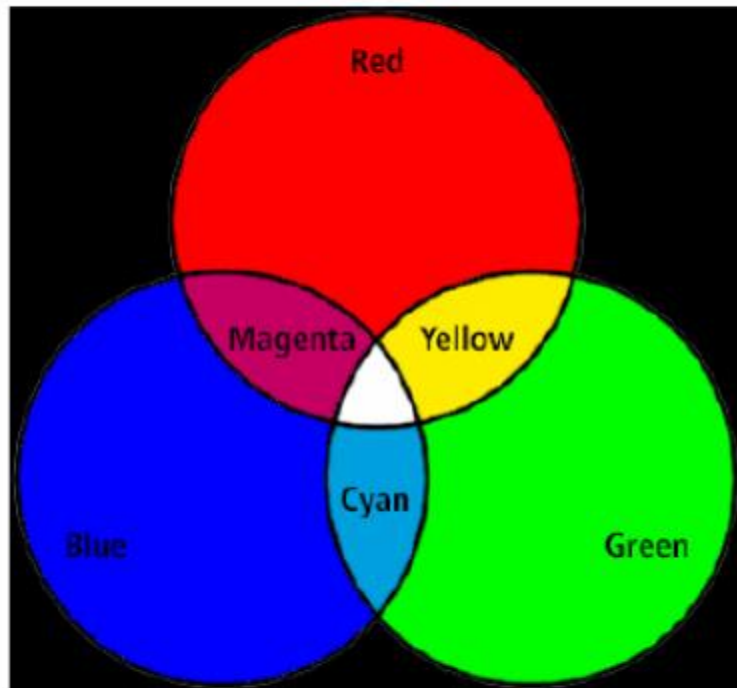
- Orange: Turuncu
- Green: Yeşil

CMYKOG, matbaa ve yeni nesil yazıcılarda giderek daha yaygın bir şekilde kullanılan bir renk uzayıdır. CMYK'ya iki yeni renk ekleyerek, basılabilir renk uzayı genişletilmiştir. CMYKOG uzayındaki temel renkler şunlardır:

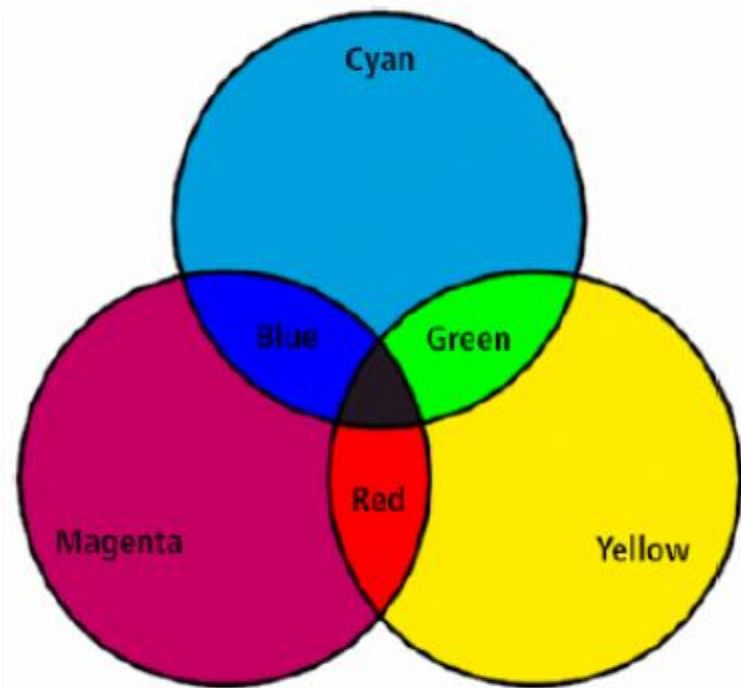
- Cyan: Turkuaz
- Magenta: Kırmızı
- Yellow: Sarı
- Key (black): Siyah

Pantone Inc. tarafından lisanslı ve patentli bir veritabanı olan CMYKOG renk uzayı, altı temel renge dayanmasından ötürü **Hexachrome** olarak da anılır.

RGB ve CMY



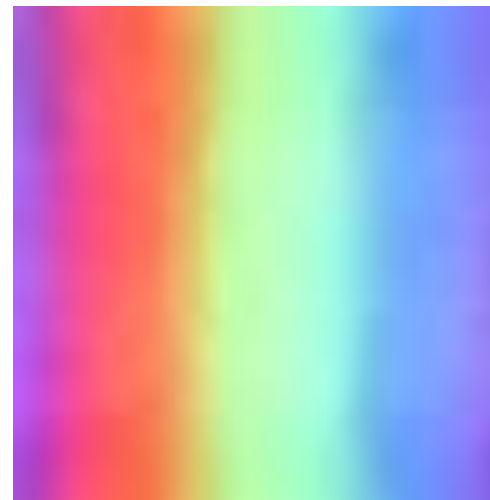
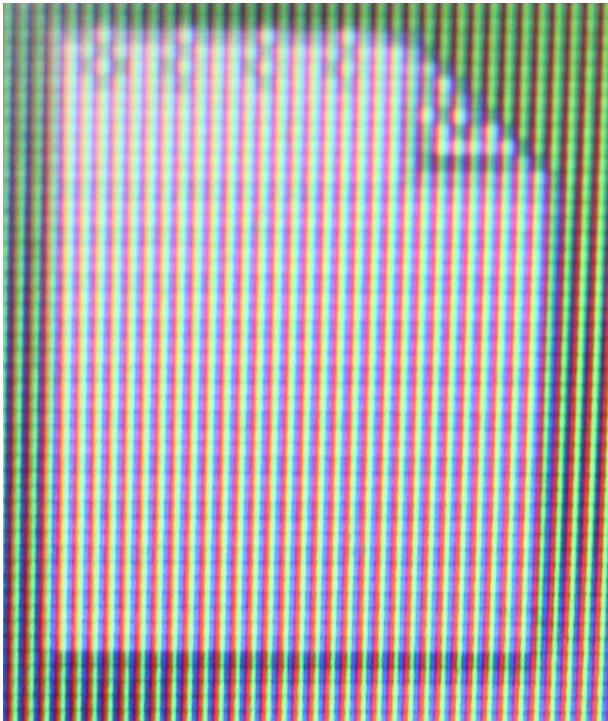
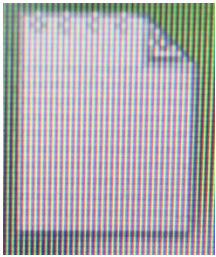
Magenta = Red + Blue
Cyan = Blue + Green
Yellow = Green + Red



Magenta = White - Green
Cyan = White - Red
Yellow = White - Blue

RGB Renk Uzayı





Üç Uyartı Değerleri (Tristimulus Values)

Belirli bir rengi oluşturmak için gerekli olan üç referans rengin (R,G,B gibi) oranları olarak tanımlanır ve X, Y, Z olarak gösterilir.

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z}$$

$x+y+z=1$ olduğundan bir ışığın renkliliğini belirtmek veya belirlemek için iki katsayının bilinmesi yeterlidir.

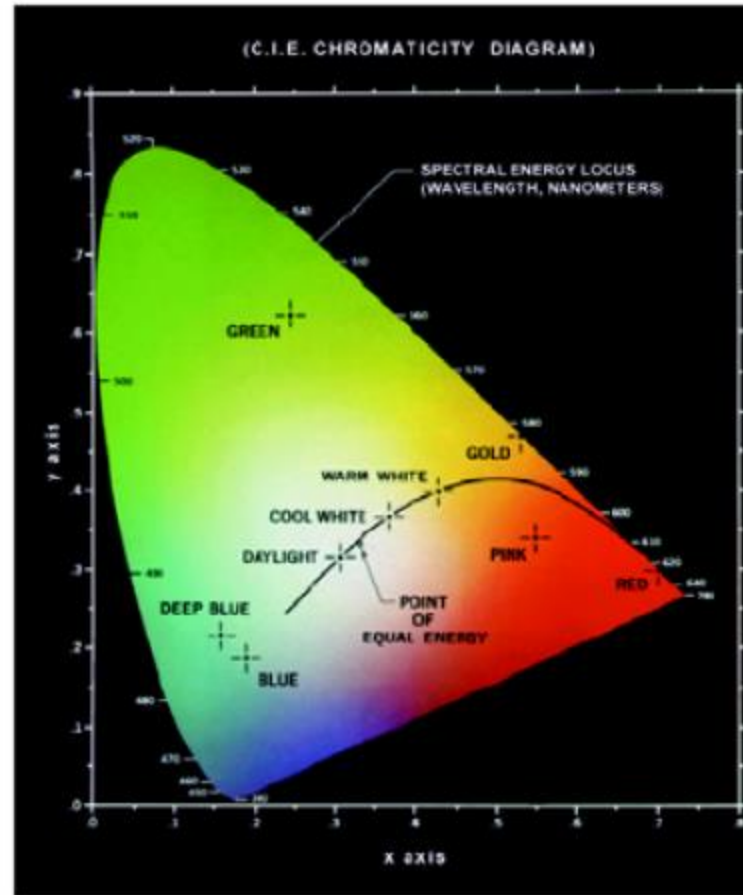
CIE Renksellik Çizgesi (CIE Chromaticity Diagram)

CIE (Commission Internationale de L'Eclairage, International Commission on Illumination) system of color specification

x axis: red
y axis: green

The point marked with GREEN
x: 25%, y: 62%, z: 13%.

FIGURE 6.5
Chromaticity diagram.
(Courtesy of the General Electric Co., Lamp Business Division.)



Renk Modelleri

Bir renk modeli ana veya yardımcı renklerle belirlenebileceği gibi

- Örneğin RGB, CMY.

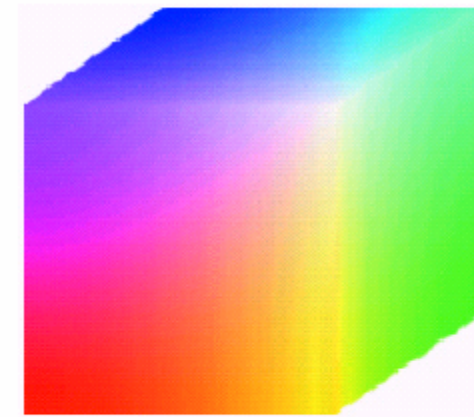
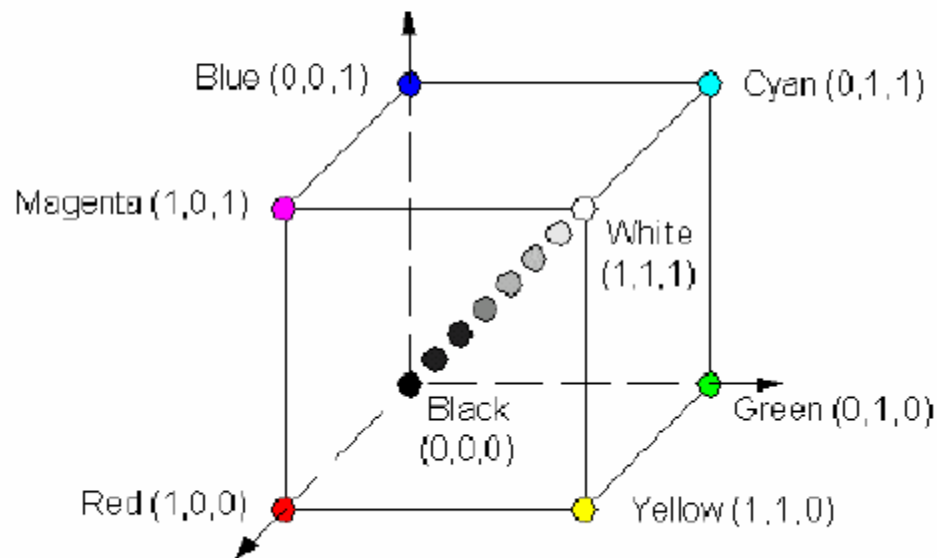
Işıklılık ve renklilik bileşenleri kullanılarak da belirlenebilir:

- HSI, HSB (Hue, Saturation, Intesity or Brightness)
- YIQ: Renkli NTSC yayınında
- YCrCb: Renkli sayısal TV yayınında

Genlik özelliği:

- Her bir renk bileşeni için 8bit kullanılırsa piksel başına 24bit kullanılır. (24bpp)
- 16 milyon renk
- 1000x1000 piksel renkli bir imge yaklaşık 3MB yer kaplar.

RGB Renk Modeli



CMY ve CMYK Renk Modelleri

RGB ve CMY renk modelleri arasındaki dönüşümler aşağıdaki şekilde yapılır.

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

CMY renk modelinde, siyah ancak C,M, ve Y bileşenin en yüksek oranda kullanılması ile elde ediliyordu. Ancak pratikte elde edilen siyah bulanık olduğu için bu modele siyah bileşeni de eklenerek CMYK renk modeli elde edilmiştir. Renkli yazıcılarda kullanılan CMYK renk modeli kullanılır.

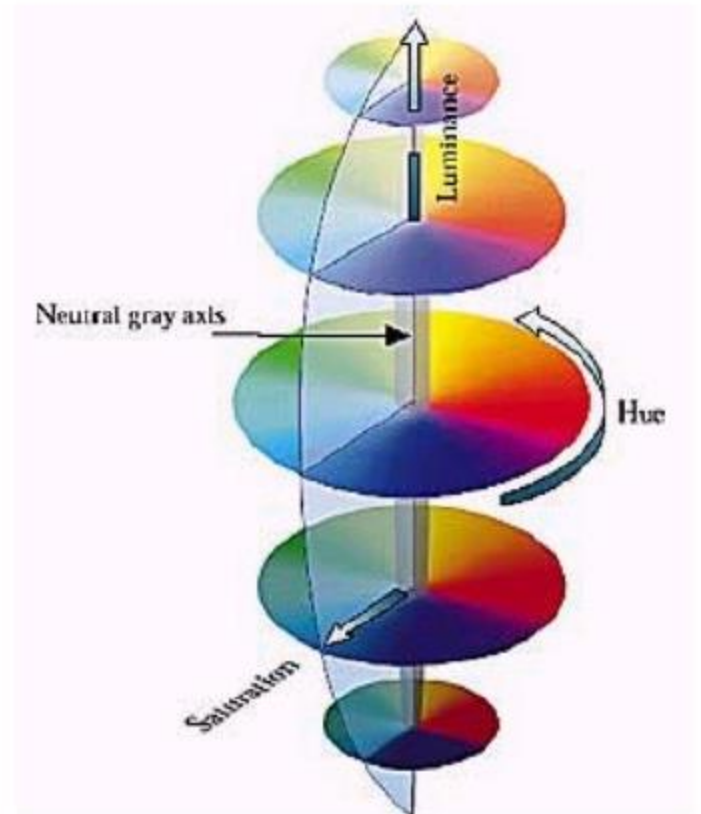
HSI Renk Modeli

Renk Özü (Hue): Baskın rengi gösterir ve baskın dalgaboyu ile ilgili bir özneliktir. Açısal olarak ifade edilir: $0^{\circ} - 360^{\circ}$

Doygunluk (Saturation): Saf bir rengin beyaz ışıkla ne kadar seyreltildiğini gösterir, rengin göreceli saflığı olarak değerlendirilebilir. Yarıçapa karşılık gelir: 0-1. Açık-Koyu ton değişimi için doygunluk ile oynamak yeterlidir.

Işıklılık (Intensity): Işıklılık miktarını gösterir.

Özellikle ışıklılık değişimlerine karşı duyarsızdır. Örneğin gölge etkisinden kurtulmada kullanılabilir.



RGB ve HSI Renk Modelleri Arasındaki Dönüşümler

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}, \quad \text{with } \theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{\left[(R-G)^2 + (R-B)(G-B) \right]^{\frac{1}{2}}} \right\}$$

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

$$I = \frac{1}{3} [R+G+B]$$

RG sector ($0 \leq H < 120$)

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

$$R = I \left[1 + \frac{S \cos H}{\cos(60-H)} \right]$$

$$G = 1 - (R+B)$$

GB sector ($120 \leq H < 240$)

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

$$G = I \left[1 + \frac{S \cos(H-120)}{\cos(60-(H-120))} \right]$$

$$B = 1 - (R+G)$$

BR sector ($240 \leq H < 360$)

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

$$B = I \left[1 + \frac{S \cos(H-240)}{\cos(60-(H-240))} \right]$$

$$R = 1 - (G+B)$$

YUV, genellikle [video](#) görüntülerini kaydetmek için kullanılan bir renk sistemidir. **Y**: *Luminance*, **U**: *Chrominance1*, **V**: *Chrominance2* sözcüklerini n baş harflerinden oluşan kısaltmadır.

Sistemde Y işareti siyah-beyaz, U (**Cb**:Chrominance blue) ve V (**Cr**:Chrominance red) işaretleri ise mavi ve kırmızı renk bilgilerini temsil ederler.

$$Y = 0.2215 * R + 0.7154 * G + 0.0721 * B$$

$$Cb = -0.1145 * R - 0.3855 * G + 0.5000 * B$$

$$Cr = 0.5016 * R - 0.4556 * G - 0.0459 * B$$

YUV ve YCrCb Renk Modeli

YUV renk modeli, PAL TV yayın sisteminde renkleri ifade etmek için kullanılmaktadır. Y ışıklılık, U ile V ise renk bileşenlerine karşılık gelir.

Bilgisayarlarda kullanılan YCrCb renk modeline oldukça benzerdir. Cr ve Cb bileşenleri, U ve V bileşenlerinin skalalandırılmış halidir.

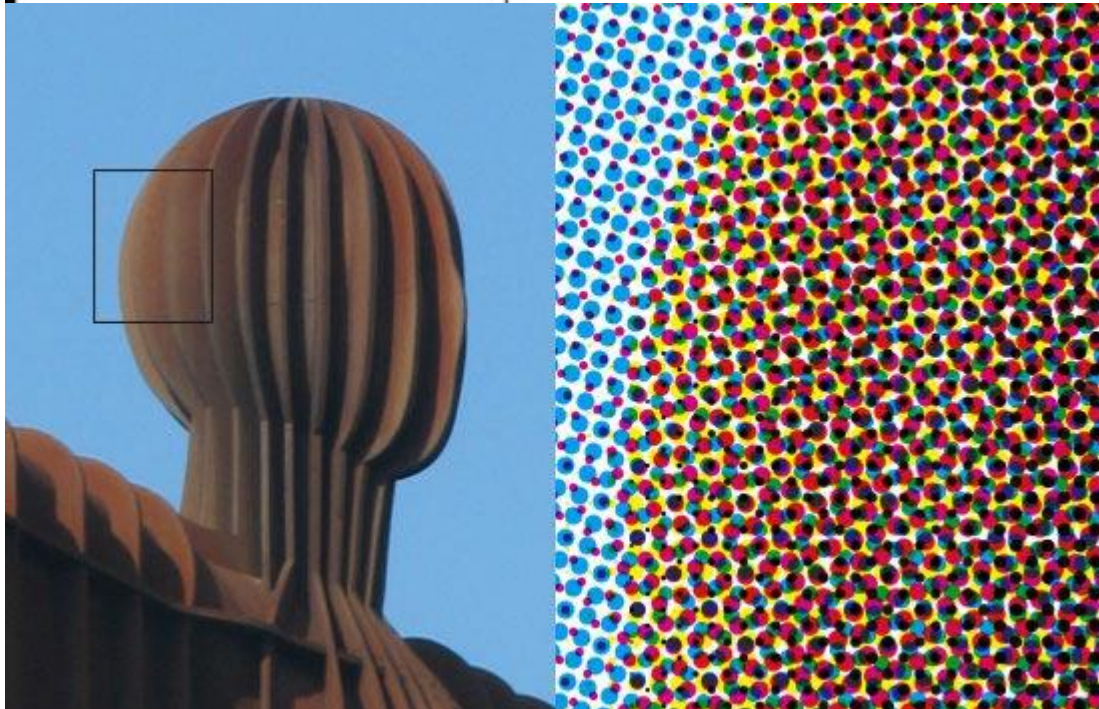
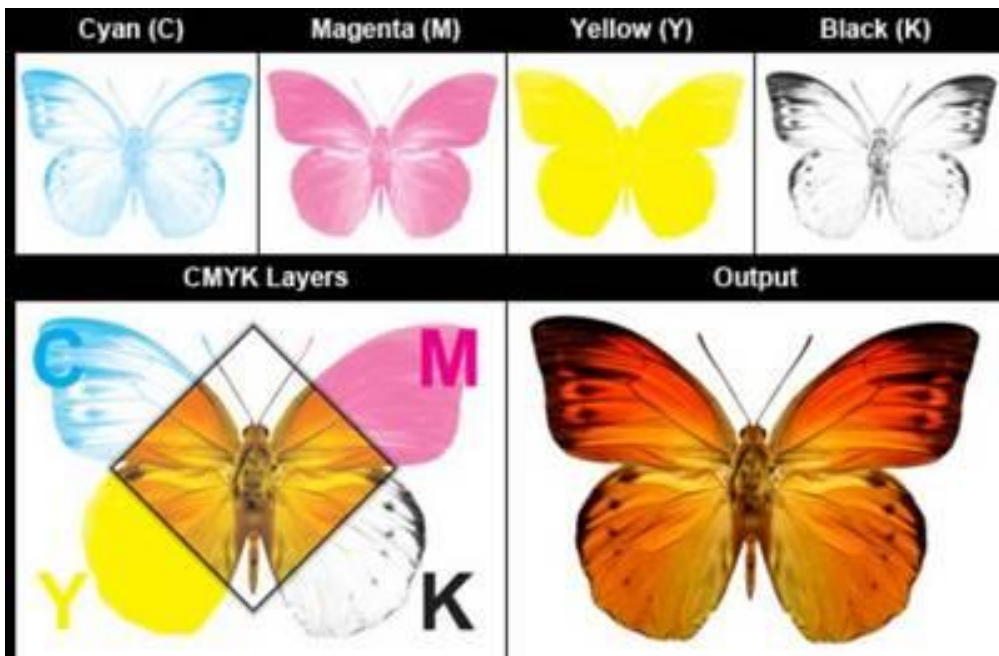
$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.378 \\ 0.578 & -0.587 & -0.114 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

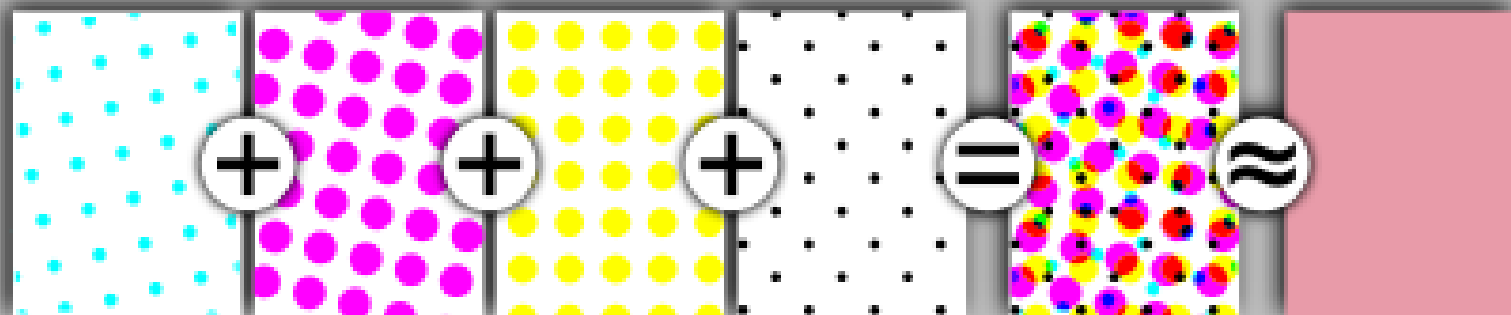
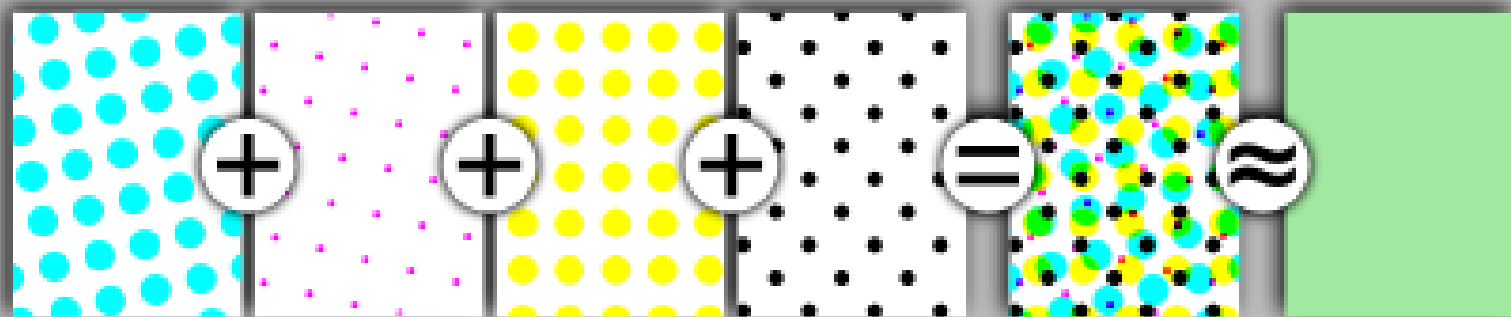
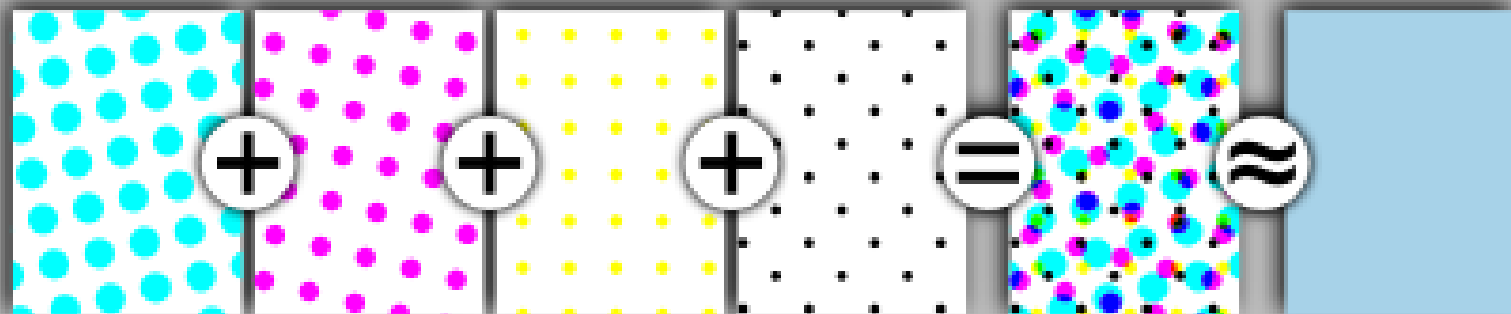
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 1.140 & 0.0 \\ 1.0 & -0.394 & -0.581 \\ 1.0 & 0.0 & 2.028 \end{bmatrix} \cdot \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

Renk Modeli Seçimi

Kullanılacak renk modeli uygulamaya bağlı olarak seçilir.

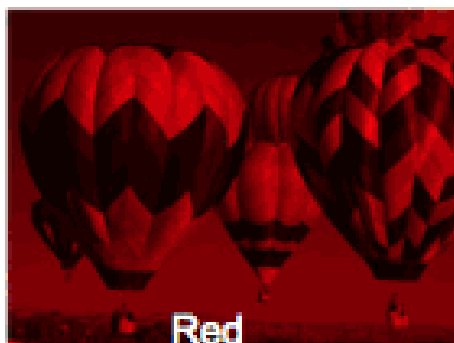
- Gösterge ve yazıcılarda daha fazla renk gösterebilmek için ana renkler tercih edilmektedir. Örneğin göstergeler için RGB, yazıcılar için CMYK.
- Renk farklarının analitik analizi için HSI daha uygundur. İnsanın görsel algılamasına benzer bir yapıya sahiptir.
- İmge/Video aktarımı ve saklaması için ise daha az artıklık sağlayan YIQ, YUV gibi renk modelleri daha uygundur.



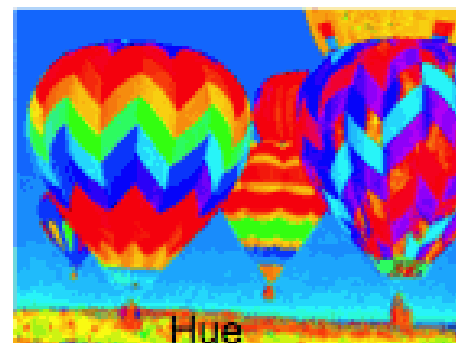




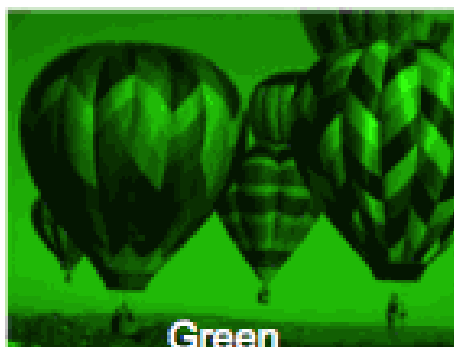
Original



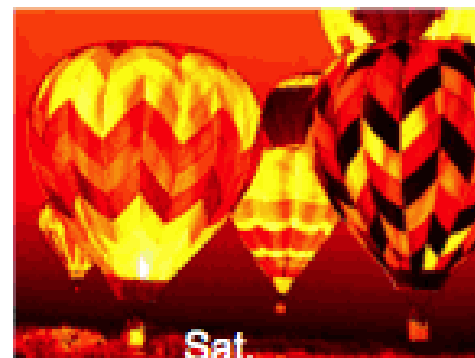
Red



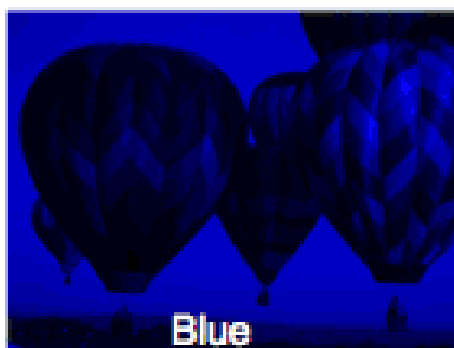
Hue



Green



Sat.



Blue



Int.

Farklı Renk Uzaylarının Karşılaştırılması



Full color



Cyan



Magenta



Yellow



Black



Red



Green



Blue



Hue



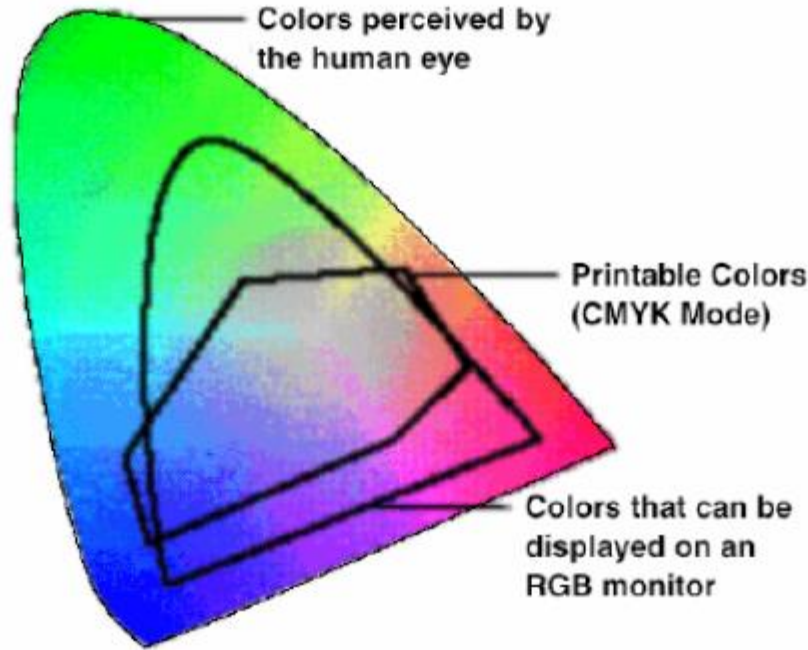
Saturation



Intensity

Renk Uzaylarının İfade Edebildiği Renkler

Bir renk uzayının gösterebildiği farklı renk aralığına “color gamut” denir. Aşağıda görüldüğü gibi RGB renk uzayı CMYK’dan daha geniş bir gamut’a sahiptir. Bu nedenle ekranda görülen renkler (CMYK renk uzayını kullanan) bir yazıcıda elde edilemez.



Renk İmgelerden Gri Tonlu İmgelere Geçiş

NTSC Standardına göre: $Y = 0.299R + 0.587G + 0.114B$ 'dir.

Bunun yerine $Y = 0.33R + 0.33G + 0.33B$ 'de kullanılabilir.

Matlab'da `rgb2gray` fonksiyonunda NTSC standardını kullanır. (flowers.tif)



Frekans Uzayı

The Frequency Domain

Önceki derslerde;

- Piksel temelli (kontrast yayma, eşikleme v.b.)
- Piksel komşuluk temelli (filtreleme, korelasyon v.b.)

Bu derste;

- Frekans uzayında (filtreleme, korelasyon v.b.)

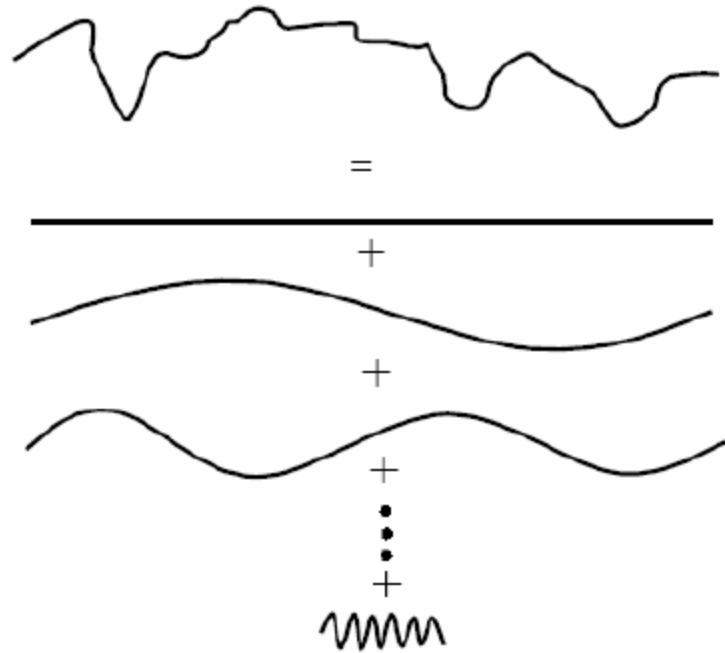
İmgeler frekans uzayında, ışıklılık veya renk değişimlerine bağlı olarak gösterilirler.

İmgede frekans, yatay ve düşey doğrultulardaki pikseller arasındaki ışıklılık değişimleri ile ilişkilidir. Yani, belli uzamsal mesafede piksellerin değişim hızı ile belirlenir.

Örneğin;

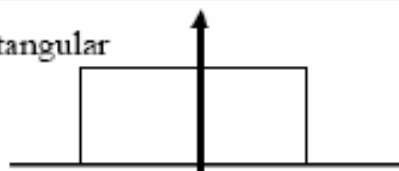
İmgedeki kenar geçişleri, yüksek uzamsal frekans bileşenlerini içerir.

Fourier dönüşümü, bir işareti ya da imgeyi farklı frekanslardaki sinüs ve kosinüs bileşenleri ile tanımlamaya yarar.



Spatial Domain

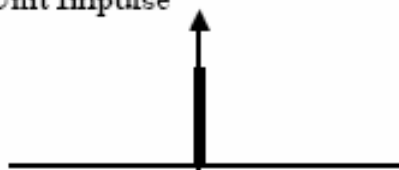
Rectangular



Triangle

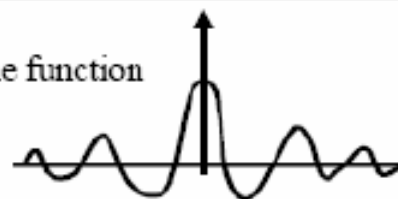


Unit Impulse

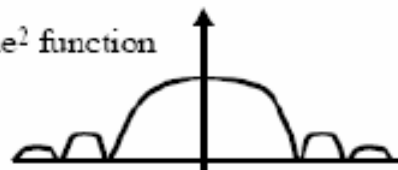


Frequency Domain

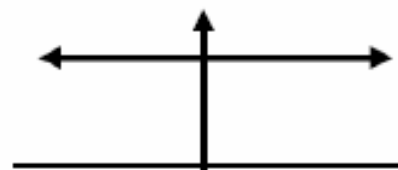
Sine function



Sine² function



Uniform



Sürekli zamanda iki boyutlu FD;

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

$$e^{-j\pi(ux + vy)} = \cos 2\pi(ux + vy) + j \sin 2\pi(ux + vy)$$

Ayrık zamanda iki boyutlu FD (DFT);

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

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

Elde edilen Fourier katsayıları karmaşık sayılardan oluşur:

$$H(u,v) = R(u,v) + jI(u,v)$$

Genlik ve faz;

$$M(u,v) = \text{sqrt}(R(u,v)^2 + I(u,v)^2)$$

$$\theta = \tan^{-1}[I(u,v) / R(u,v)]$$

Eğer; $f(x, y)$ dizisi reel ise, bunun fourier dönüşümü genellikle kompleks sayılardır. Dönüştürülmüş bir resmi frekans uzayında görmek (görsel analizi) onun spetrumunu ifade etmektir.

$R(u, v)$ ve $I(u, v)$ sırasıyla; $F(u, v)$ 'nin reel ve imajiner kısmını gösteriyorsa; Fourier ve güç spektrumu aşağıdaki gibi tarif edilir. Artık frekans uzayında bir görüntüyü, fourier spektrumu veya güç spektrumu ile ifade edebiliriz.

$F(u, v)$ nin genliği

$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)}$$

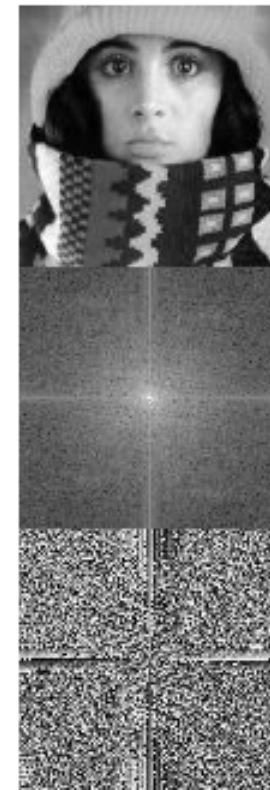
$F(u, v)$ nin faz açısı

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

Güç Spektrumu

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

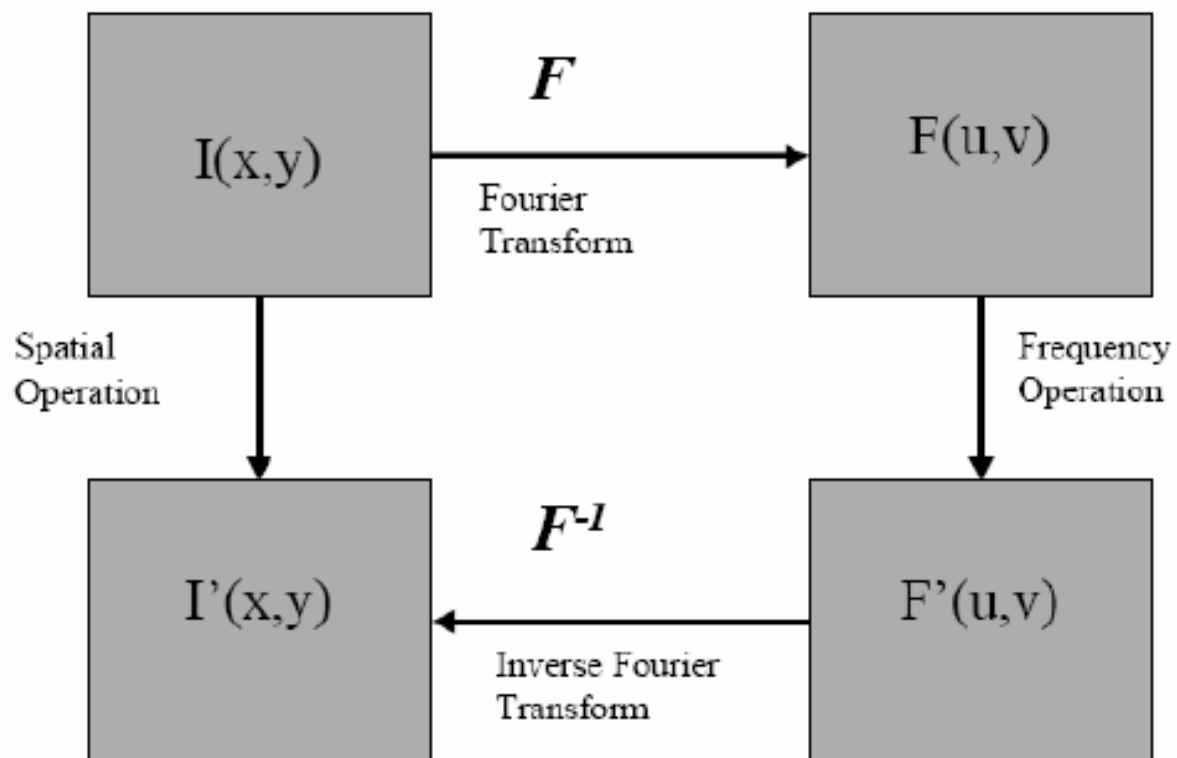
Görüntünün tekrar orijinaline dönüştürülmesi için, hem genlik hem de faz bilgisine ihtiyaç vardır.

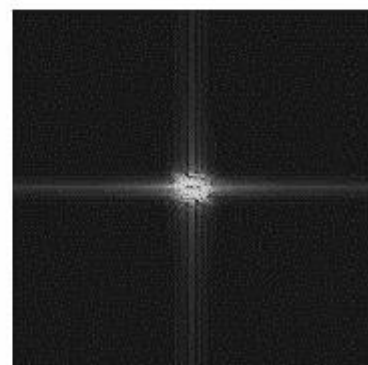
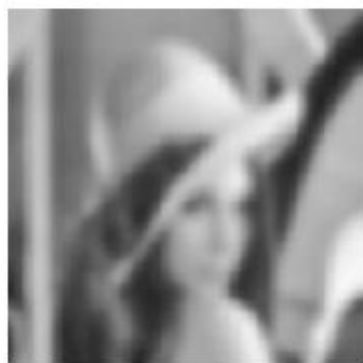
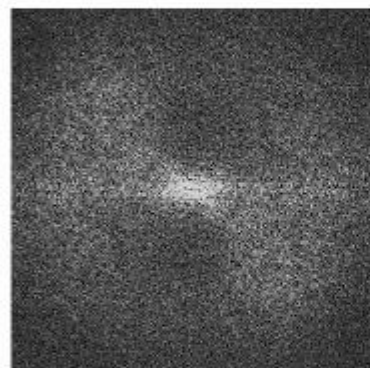
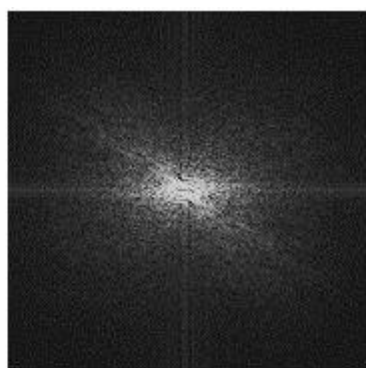


Orijinal

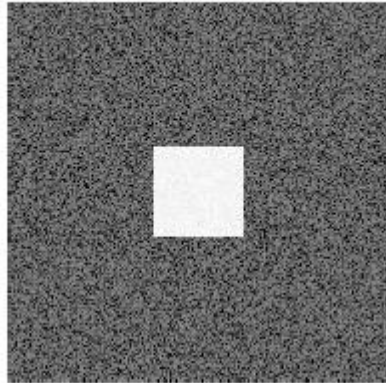
Genlik spk.

Faz spk.





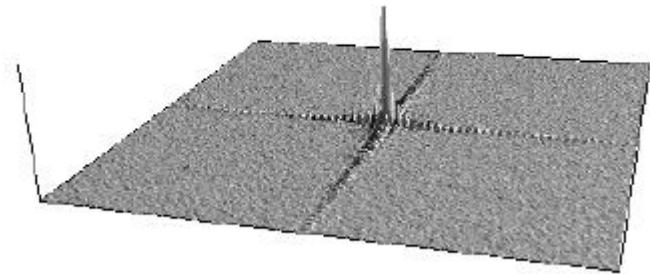
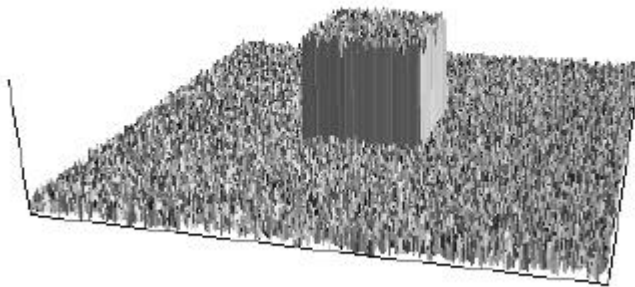
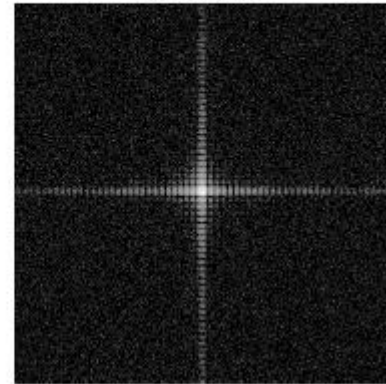
imge



DFT

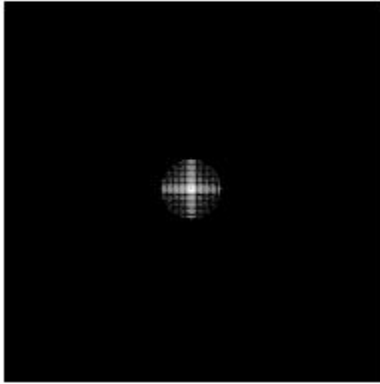


genlik

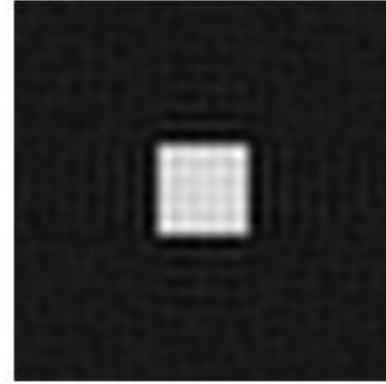


Alçak geçiren filtre

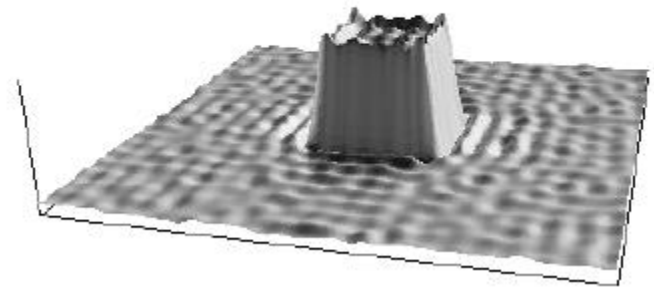
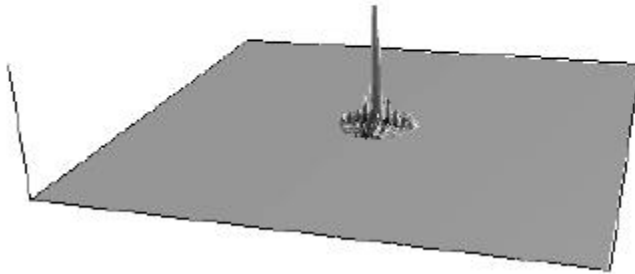
genlik



imge

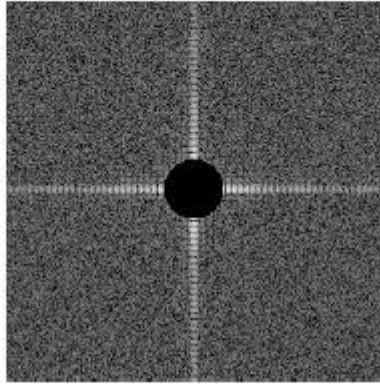


IDFT

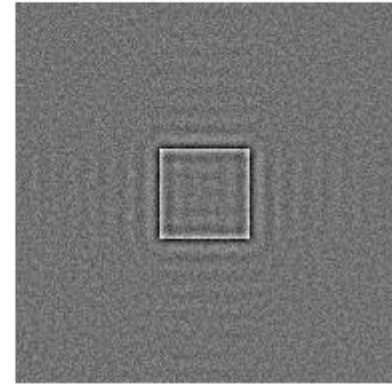


Yüksek geçiren filtre

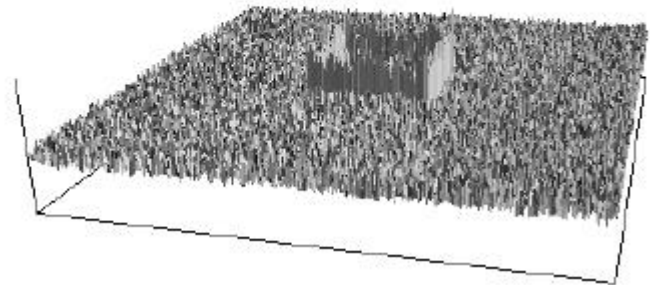
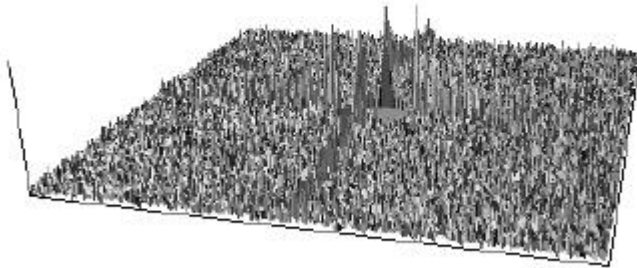
genlik



imge



IDFT



$$f(x) = \frac{a_0}{2} + a_1 \cos(x) + b_1 \sin(x) + a_2 \cos(2x) + b_2 \sin(2x) + \dots$$

bu seriyi daha kısa olarak,

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

Eşitliğin her iki tarafının da $(-\pi, \pi)$ aralığında integrali alınırsa,

$$\int_{-\pi}^{\pi} \frac{a_0}{2} dx = \pi a_0$$

$$\int_{-\pi}^{\pi} \cos(nx) dx = \left| \frac{\sin(nx)}{n} \right|_{-\pi}^{\pi} = 0$$

$$\int_{-\pi}^{\pi} \sin(nx) dx = \left| -\frac{\cos(nx)}{n} \right|_{-\pi}^{\pi} = 0$$

$$\int_{-\pi}^{\pi} f(x) dx = \pi a_0$$

bu denklemden a_0 çekilirse,

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) dx \quad \text{bulunur.}$$

$n \neq k$ ise,

$$\int_{-\pi}^{\pi} \cos(nx) \cos(kx) dx = 0$$

$$\int_{-\pi}^{\pi} \cos(nx) \sin(kx) dx = 0$$

$$\int_{-\pi}^{\pi} \sin(nx) \sin(kx) dx = 0$$

$n = k$ ise,

$$\int_{-\pi}^{\pi} \cos^2(kx) dx = \pi$$

$$\int_{-\pi}^{\pi} \sin(kx) \cos(kx) dx = 0$$

$$\int_{-\pi}^{\pi} \sin^2(kx) dx = \pi$$

Her iki taraf $\cos(kx)$ ile çarpılıp integre edilirse,

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(kx) dx$$

Her iki taraf $\sin(kx)$ ile çarpılıp integre edilirse,

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(kx) dx$$

$$e^{ix} = \cos x + i \sin x$$

$$e^{-ix} = \cos x - i \sin x$$

$$(\cos x + i \sin x)^n = \cos nx + i \sin nx$$

Tek Değişkenli Ayrık Fourier Dönüşümü ve Tersi:

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

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

2D Ayrık Fourier Dönüşümü ve Tersi :

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

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

Tek Değişkenli Ayırık Fourier Dönüşümü ve Tersi:

$$\begin{aligned} F(0) &= \sum_{x=0}^3 f(x) = [f(0) + f(1) + f(2) + f(3)] \\ &= 1 + 2 + 4 + 4 = 11 \end{aligned}$$

$$\begin{aligned} F(1) &= \sum_{x=0}^3 f(x) e^{-j2\pi(1)x/4} \\ &= 1e^0 + 2e^{-j\pi/2} + 4e^{-j\pi} + 4e^{-j3\pi/2} = -3 + 2j \end{aligned}$$

$$F(2) = -(1 + 0j)$$

$$F(3) = -(3 + 2j)$$

$$\begin{aligned}
f(0) &= \frac{1}{4} \sum_{u=0}^3 F(u) e^{j2\pi u(0)/4} \\
&= \frac{1}{4} \sum_{u=0}^3 F(u) \\
&= \frac{1}{4} [1 + 1 - 3 + 2j - 1 - 3 - 2j] \\
&= \frac{1}{4} [4] = 1
\end{aligned}$$

$$\begin{bmatrix} 77 & 63 & 53 \\ 69 & 66 & 60 \\ 61 & 77 & 63 \end{bmatrix}$$

$$F(0,0) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) = 589$$

$$F(0,1) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(y/3)}$$

$$F(0,2) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(2y/3)}$$

$$F(1,0) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(x/3)}$$

$$F(1,1) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(x/3+y/3)}$$

$$F(1,2) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(x/3+2y/3)}$$

$$F(2,0) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(2x/3)}$$

$$F(2,1) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(2x/3+y/3)}$$

$$F(2,2) = \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) e^{-j2\pi(2x/3+2y/3)}$$

Morfolojik İmge İşleme

Morphologic Image Processing

Morfoloji:

Canlıların yapıları ve şekilleri ile ilgilenen bir biyoloji dalıdır.

Matematiksel Morfoloji:

İmge işlemede sıkça kullanılan, temel küme işlemlerine dayanan yöntemlerdir.

Genellikle ikili imgeler üzerinde kullanılırlar.



Morfolojik işlemler ikili / iki değerli (binary) imgelerin analizinde: kenar bulma, gürültü giderme, pekiştirme ve bölütleme gibi uygulamalarda kullanılmaktadır.

Morfolojik filtreleme, inceltme (thinning) ve budama (pruning) gibi ön/son işlemlerde de sıkça kullanılır.

Morfolojik imge işlemede temel olarak kullanılan iki işlem vardır:

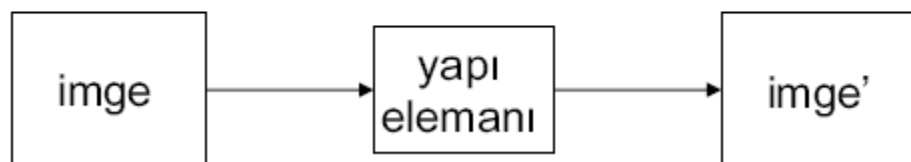
- Yayma (dilation)
- Aşındırma (erosion)

Diğer morfolojik işlemler, bu temel iki işlem kullanılarak yapılmaktadır. Örn; açma (opening), kapama (closing).

Yayma, ikili imgedeki nesneyi büyötmeye ya da kalınlaştırmaya yarayan morfolojik işlemdir.

$$A \oplus B$$

Kalınlaştırma işleminin nasıl yapılacağını yapı elemanı (structure element) belirler.



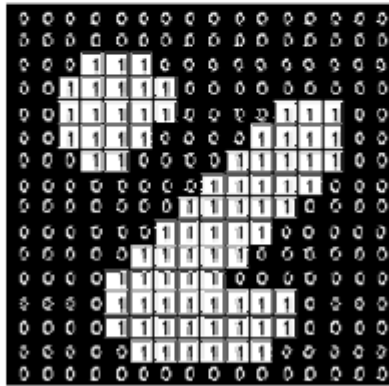
Yapı elemanı

1	1	1
1	1	1
1	1	1

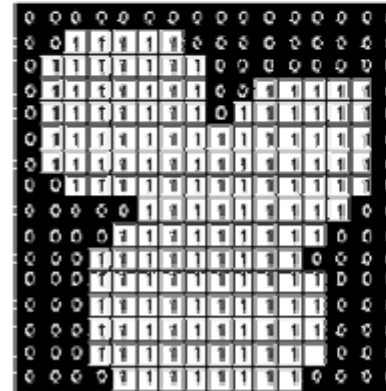
merkez
nokta

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),
(-1, 0), (0, 0), (1, 0),
(-1, 1), (0, 1), (1, 1) }

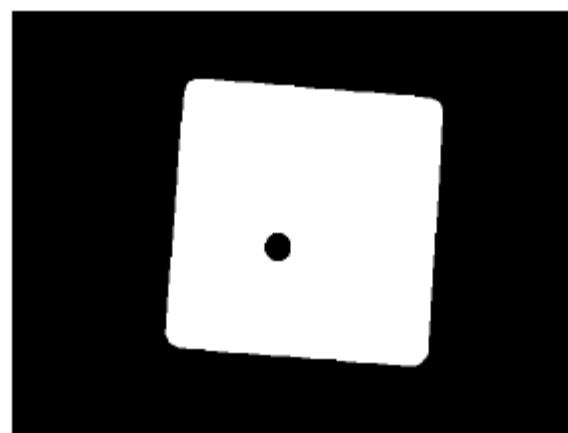
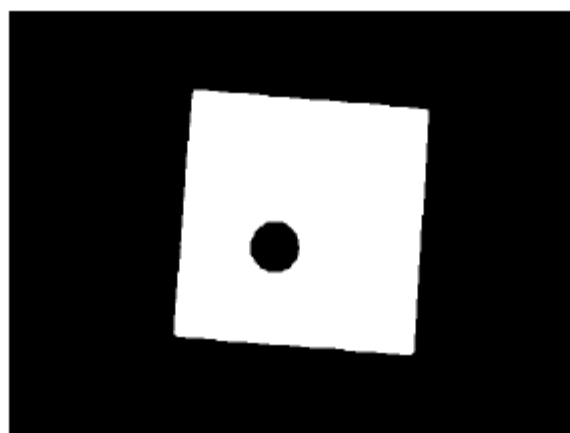


İki değerli imge

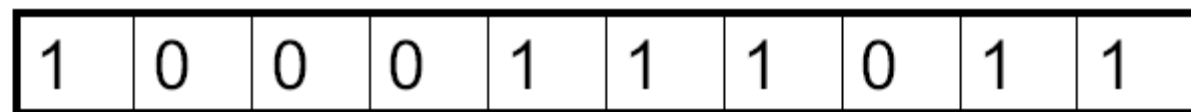


Yayma işlemi sonrası
elde edilen imge

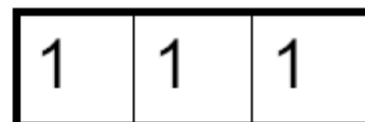
İmdilate(A,B)



Input image



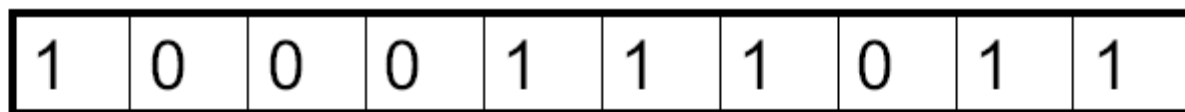
Structuring Element



Output Image



Input image



Structuring Element



Output Image



Input image

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



Structuring Element

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



Output Image

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

Input image

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



Structuring Element

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



Output Image

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

Input image

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



Structuring Element

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



Output Image

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

Input image

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



Structuring Element

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



Output Image

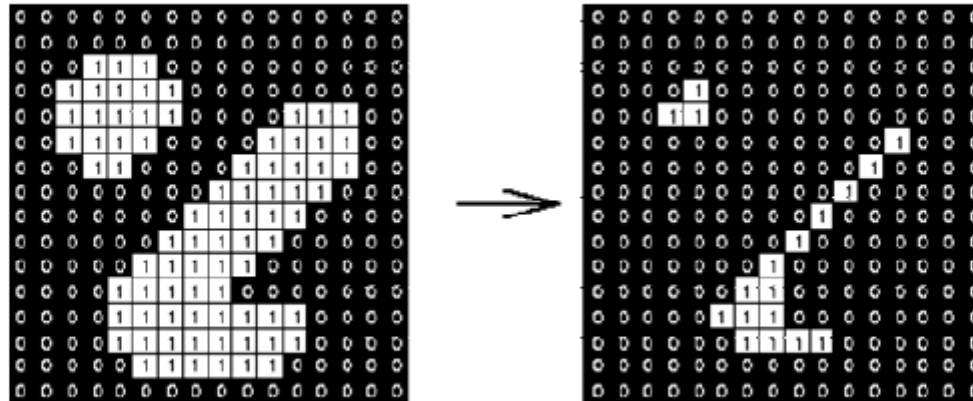
	1	0	1	1	1	1	1	1	
--	---	---	---	---	---	---	---	---	--

Aşındırma, ikili imgedeki nesneyi küçültmeye ya da inceltmeye yarayan morfolojik işlemdir. $A \ominus B$

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),
 (-1, 0), (0, 0), (1, 0),
 (-1, 1), (0, 1), (1, 1) }



$\text{İmerode}(A,B)$

Input image

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



Structuring Element

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



Output Image

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

Input image

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



Structuring Element

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



Output Image

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

Input image

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



Structuring Element

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



Output Image

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

Input image

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



Structuring Element

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



Output Image

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

Input image

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



Structuring Element

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



Output Image

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

Açma ve kapama, yayma ve aşındırma işlemlerinin iki değerli imgeye ardışıl uygulanmasıyla yapılan işlemlerdir.

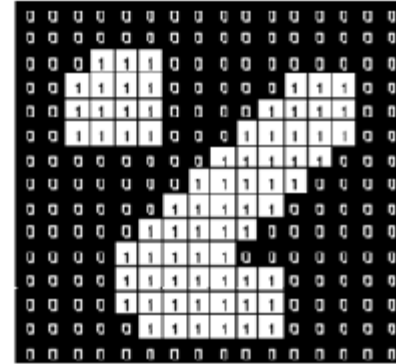
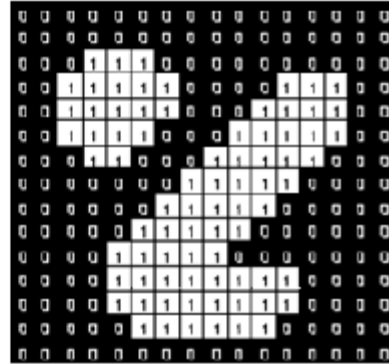
Açma işlemi:

$$A \circ B = (A \ominus B) \oplus B \quad \text{imopen}(A,B)$$

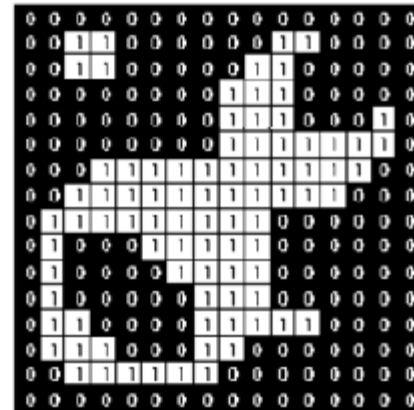
Kapama işlemi:

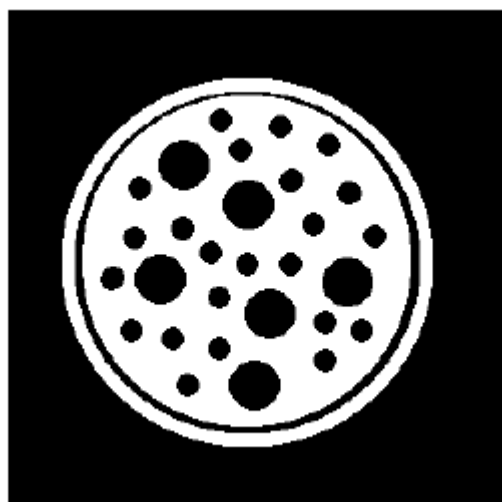
$$A \bullet B = (A \oplus B) \ominus B \quad \text{imclose}(A,B)$$

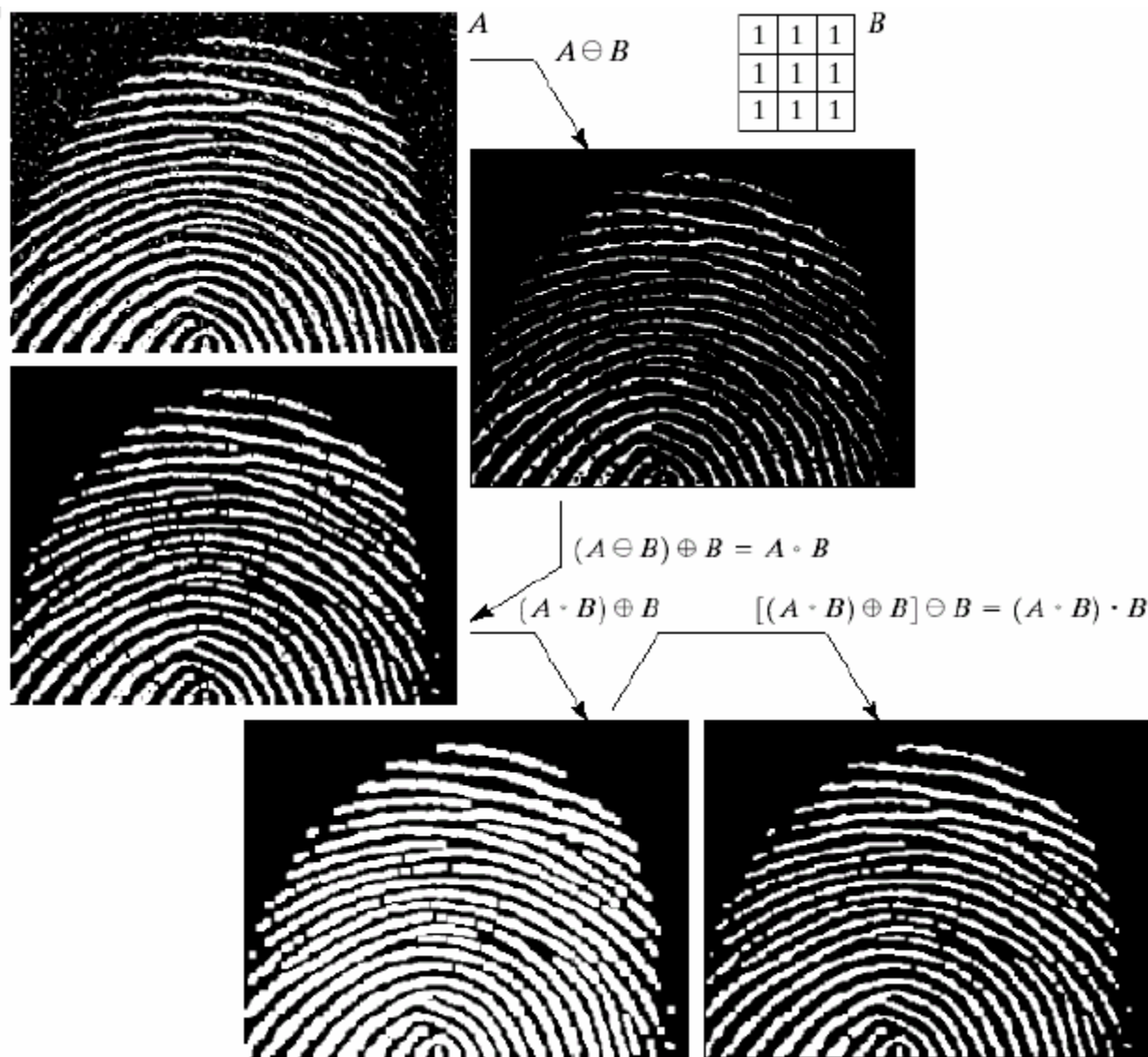
Açma



Kapama







a	b
d	c
e	f

FIGURE 9.11

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

LİNEER UZAYSAL FİLTRELER İÇİN MATLAB FONKSİYONLARI

Burada f giriş görüntüsü, w filtre maskesi(kernel), g filtrelenmiş sonuçtur. Diğer parametreler ise tabloda gösterilmektedir.

```
g=imfilter(f,w,filtering_mode,...  
           boundary_options,size_options)
```

Options	Description
<i>Filtering Mode</i>	
'corr'	Filtering is done using correlation. This is the default.
'conv'	Filtering is done using convolution.
<i>Boundary Options</i>	
P	The boundaries of the input image are extended by padding with a value, P. This is the default, with value 0.
'replicate'	The size of the image is extended by replicating the values in its outer border.
'symmetric'	The size of the image is extended by mirror-reflecting it across its border.
'circular'	The size of the image is extended by treating the image as one period a 2-D periodic function.
<i>Size Options</i>	
'full'	The output is of the same size as the extended (padded) image.
'same'	The output is of the same size as the input. This is the default.

MEDIAN FİLTRELEMeye ÖRNEK

`G=ORDFİLTER2(F, MEDIAN (1:M*N), ONES(M,N))`

Median Filtre maske altındaki piksel gri seviye değerlerini azdan çoğa doğru bu değerlerin ortadakini almaktı. Yukarıdaki fonksiyonda order yerine (1:m*n) değerleri arasında %50 büyüklükteki değerin alınmasıdır. Median filtreler görüntüdeki salt-pepper(tuz-Biber) gürültülerini filtreler.

```
>> B=[2 14 26 37 99;34 90 67 56 100;100 200 56 255 45;  
      10 100 56 178 35; 99 29 0 150 200]  
B =
```

2	14	26	37	99
34	90	67	56	100
100	200	56	255	45
10	100	56	178	35
99	29	0	150	200

```
>> c=ordfilt2(d,median(1:3*3),ones(3,3))
```

c =

0	14	26	37	0
14	56	56	56	45
34	67	90	56	45
29	56	100	56	45
0	10	29	35	0

MEDIAN FİLTREYE ÖRNEK

```
>> a=imread('cameraman.tif');  
>> imshow(a)  
>> b=imnoise(a,'salt & pepper',0.2);  
>> imshow(b)  
>> c=ordfilt2(b,median(1:3*3),ones(3,3));  
>> imshow(c)  
>> d=ordfilt2(b,median(1:9*9),ones(9,9));  
>> imshow(d)
```



- ▶ Burada $m \times n$ uzunlukta bir komşuluk ilişkisi tanımlanır ve onun piksel değerlerinin ortancası ile hesaplar yapılır.
- ▶ Padopt üç olası sınır padding seçeneklerden birini belirtir:
- ▶ Sıfır (varsayılan-default): kenarlara 0 dolgusu yapılır.
- ▶ Simetrik: kenarlar ayna-yansımali simetriklik özelliği ile doldurulur.
- ▶ 'endeksleme' ise, veri double ise kenarlar 1 ile değil ise 0'la doldurulur.
Örnek default fonksiyon
 $g = \text{medfilt2}(f)$

3x3 maske ile medyan hesaplar. Kenarlar 0 ile doldurulur.

MEDIAN FİLTRE İÇİN BAŞKA BİR FONKSİYON

$G = \text{MEDFILT2}(F, [M \ N], \text{PADOPT})$

ÖRNEK:medfilt2

```
I = imread('eight.tif');  
J = imnoise(I,'salt & pepper',0.02);  
K = medfilt2(J);  
imshow(J), figure, imshow(K)
```



Yapısal Eleman Örnekleri

Morfolojik işlemler 'yapısal elemanlar' a bağlı olarak tanımlanmaktadır. Bir yapısal eleman, görüntü yapısını araştırmak için kullanılan bir alt kümedir.

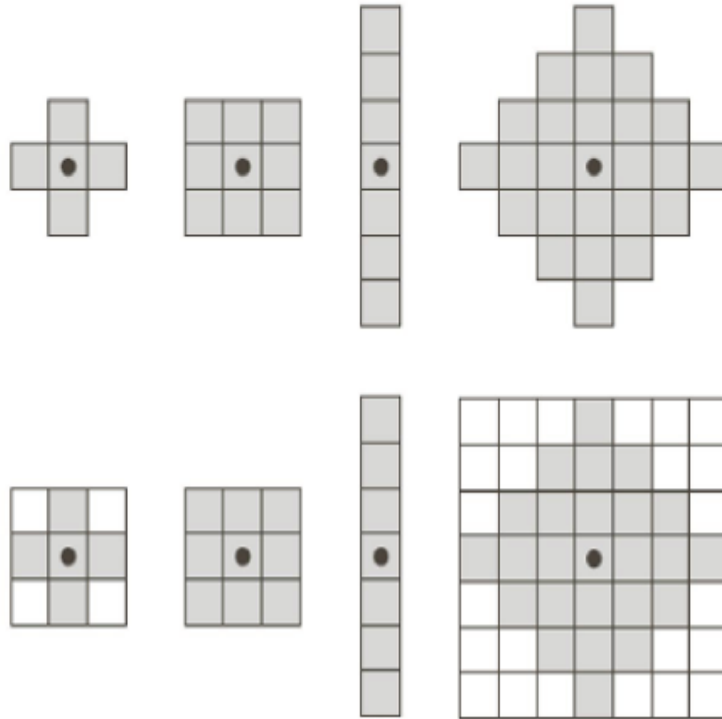


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

```
SE = strel(shape, parameters)
SE = strel('arbitrary', NHOOD)
SE = strel('arbitrary', NHOOD, HEIGHT)
SE = strel('ball', R, H, N)
SE = strel('diamond', R)
SE = strel('disk', R, N)
SE = strel('line', LEN, DEG)
SE = strel('octagon', R)
SE = strel('pair', OFFSET)
SE = strel('periodicline', P, V)
SE = strel('rectangle', MN)
SE = strel('square', W)
```

Örnek 1

```
bw = imread('rice.png');  
lvl=graythresh(bw);  
bw=im2bw(bw,lvl);  
se = strel('line',11,70);  
se1 = strel('square',8);  
bw2 = imdilate(bw,se);  
bw3 = imerode(bw,se1);  
subplot(1,3,1), imshow(bw), title('Original')  
subplot(1,3,2), imshow(bw2), title('Dilate sonucu')  
subplot(1,3,3), imshow(bw3), title('erode sonucu')
```



Örnekte; rice.png görüntüsü sonucun daha anlaşılır gözükmesi için binary moda çevrildi. İşlemler gri seviyede de yapılabilir. se ve se1 olarak iki tane maske belirlendi. Maskelerin parametreleri isteğe göre değiştirilebilir. Mesela; açma işleminin daha belirgin olması için se maskesinin parametreleri artırılmalı. se1 maskesinde 8 sayısı 12 yapıldığında beyaz noktaların tamamen kaybolduğu görülür.


```

f =
    -0.5    0.5
     0.75    1.5

    Performing the conversion
>> g = im2uint8(f)
yields the result

g =
     0    128
    191    255

    >> h = uint8([25 50; 128 200]);
    Performing the conversion
    >> g = im2double(h);
    yields the result

g =
    0.0980    0.1961
    0.4706    0.7843

```

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

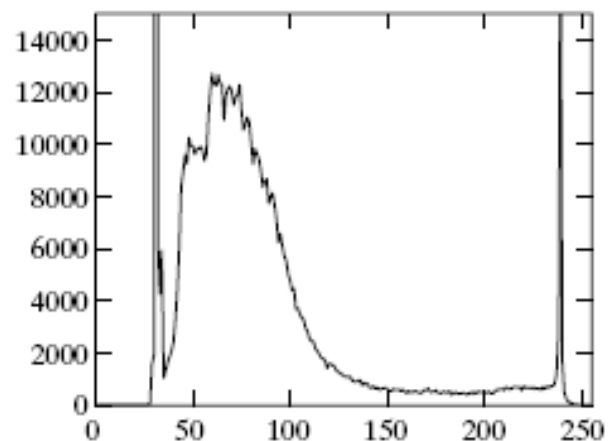
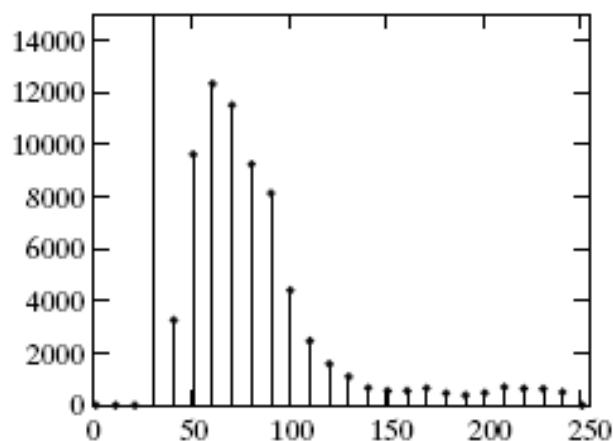
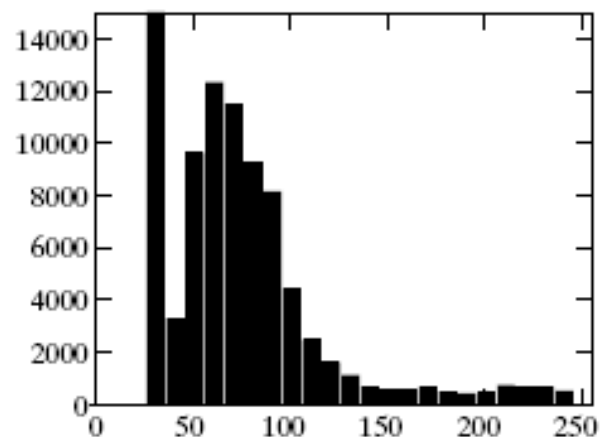
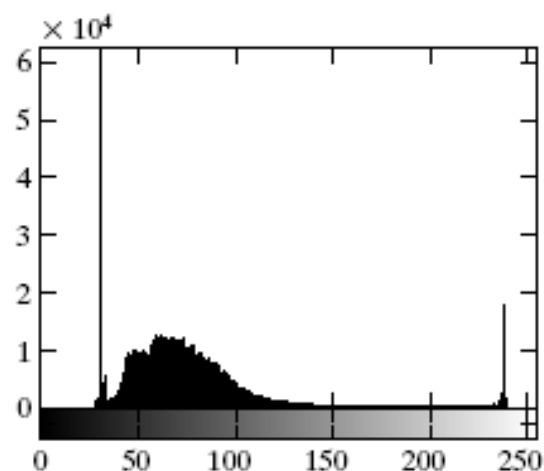
```
clear all;
close all;
clc;
w=256;
h=256;
im=imread('manzara.bmp');
im22=imresize(im,[w,h]);
im2=rgb2gray(im22);
im4=fft2(im2);
im5=real(im4);
im6=imag(im4);
im7=fftshift(sqrt(power(im5,2)+power(im6,2)));
figure,imshow(mat2gray(log(1+abs(im7)))),title('Genlik Spektrumu');
fi=fftshift(atan(im6/im5));
figure,imshow(mat2gray(log(1+abs(fi)))),title('Faz Spektrumu');
```

a b
c d

FIGURE 3.7

Various ways to
plot an image
histogram.

(a) `imhist`,
(b) `bar`,
(c) `stem`,
(d) `plot`.



```
clear all;close all;clc;
w=256;
h=256;
im=imread('manzara.bmp');
im2=imresize(im,[w,h]);
im3=rgb2gray(im2);
figure,imshow(im3);
for n=1:255
    hdizi(n)=0;
end
for i=1:256
    for j=1:256
        for n=1:255
            if (im3(i,j)==n)
                hdizi(n)=hdizi(n)+1;
            end
        end
    end
end
end

bar(hdizi);
grid;
xlabel('bar');
title('Histogram dizisi');
figure,imhist(im3),title('Histogram');
```

```
x=1:255;
figure,bar(x,hdizi);grid;xlabel('bar');title('Histogram dizisi');
figure,imhist(im3),title('Histogram');
figure,stem(hdizi);xlabel('stem');title('Histogram dizisi');
figure,plot(hdizi);xlabel('plot');title('Histogram dizisi');
```

```
clear all;close all;clc;
im=imread('manzara.bmp');
im2=im2bw(im,0.3);
figure;imshow(im2);
im3=im2bw(im,0.4);
figure;imshow(im3);
im4=im2bw(im,0.5);
figure;imshow(im4);
im5=im2bw(im,0.6);
figure;imshow(im5);
im6=im2bw(im,0.7);
figure;imshow(im6);
esikdegeri=graythresh(im)
im7=im2bw(im,esikdegeri);
figure;imshow(im7);
```

```
clear all;close all;clc;
w=200;
h=200;
im=imread('kamereman.jpg');
im2=imresize(im,[w,h]);
im3=rgb2gray(im2);
figure,imshow(im3);
t=125;
for i=1:200
    for j=1:200
        if im3(i,j)<t im4(i,j)=0;
        end
        if im3(i,j)>t im4(i,j)=255;
        end
    end
end
figure,imshow(im4);
```