

Gri Seviye Dönüşümleri ve Uzaysal Filtreleme

BMÜ-357 Sayısal Görüntü İşleme

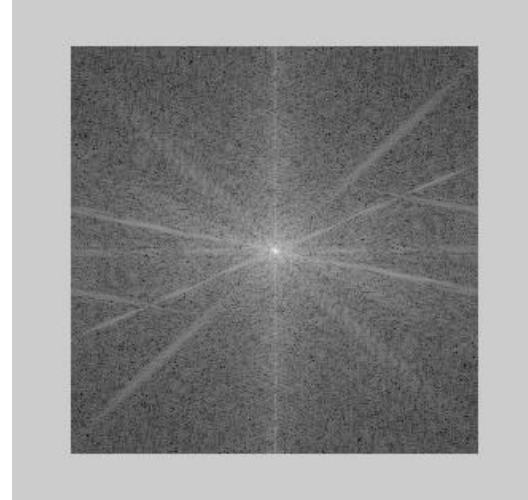
Yrd. Doç. Dr. İlhan AYDIN

GRİ SEVİYE DÖNÜŞÜMLERİ

- Herhangi bir görüntü işleme operasyonu, görüntüdeki pikselin gri seviye değerlerini dönüştürme işlemidir. Ancak, görüntü işleme operasyonları; dönüşümü gerçekleştirmek için, ihtiyaç duyacağı bilgilere göre 3 sınıfa ayrılabilir. Bunlar en zordan en basite göre;
- **1- Transformlar (Dönüşümler):** değişik domainlere dönüşüm yapılarak görüntü işleme işlemidir. (Bu derste uzaysal domain(Spatial domain) ve frekans domaininde (frequency domain) işlemler yapılacaktır.) Çok etkili ve verimli algoritmalar bu şekilde çalıştırılır. Bir dönüşümü kullanarak, tüm görüntünün tek bir büyük blok gibi işlenmiş olduğunu düşünebilirsiniz.
- **2-Komşuluk ilişkili (Neighbourhood processing-Bölgesel) işlemler:** Belirli bir pikselin gri düzeyini değiştirmek için bilmemiz gereken tek şey verilen piksel etrafında küçük bir bölgedeki (komşuluk ilişkisinin olduğu yerde) gri düzeylerinin değeridir.
- **3-Noktasal İşlemler :** Bir pikselin yeni gri seviye değerini, bağımsız olarak, etrafındaki piksel bilgilerine ihtiyaç olmadan elde etme işlemidir. Noktasal işlemler en basit işlemler olmasına rağmen birçok görüntü işleme operasyonlarında kullanılırlar. Özellikle bir görüntünün; ana işlemlerden geçirilmesine hazırlamak üzere kullanılırlar.

Tamamlayıcı Not

- **Uzaysal domain (Spatial Domain):** Günlük hayatta kullandığımız sayısal resimlerin oluşturulduğu domaindir. Bu domaindeki resimlerin pixelleri doğrudan doğruya işlenebilir.



- **Frekans domain(Frequency domain):** Görüntünün birçok farklı frekanslı bileşenden oluştuğu kabul edilir. Uzaysal domaindeki görüntü fourier v.b dönüşümü ile frekans domanine çevrilir. Burada işlenip ters dönüşüm yapılır.

Uzaysal Domain'de görüntü işlemleri

- Uzaysal domain teknikleri , bir görüntünün pikselleri üzerinde doğrudan işlem yapar.
- Bu domendeki işlemler aşağıdaki denklemle ifade edilir.

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

Burada $f(x, y)$ giriş görüntüsüdür. $g(x, y)$ ise çıkış (İşlenmiş) görüntüsüdür.

T ise f 'de belirli bir (x,y) komşuluk ilişkisi bölgesinde işlem yapan bir operatördür.

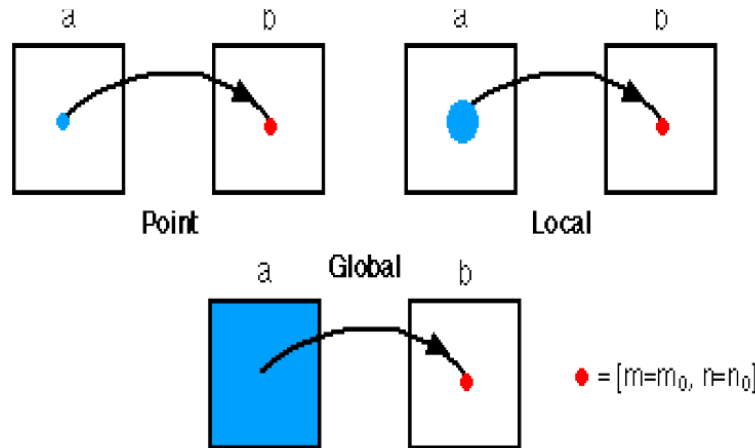
Örneğin T operatörü; K görüntülerinde gürültü azaltmak için, bir görüntü seti işlemi olarak ta çalışabilir.

T ile belirtilen operasyonlar, Noktasal,Lokal(yerel)ve Global olarak yapılabilir.

Noktasal Operasyon: Sadece 1x1 lik bölgede yapılan işlemlerdir. Nokta operasyonlarında, bir resimdeki her pikselin gri seviyesi yalnızca onun orijinal gri seviyesinden(tonundan) hesaplanır. Bu sebeple bu işlemlere “piksel değeri haritalama” veya “gri ton değişikliği” (modification) gibi isimler verilir. Nokta operasyonları genellikle “resim onarımı” (manipulation) için kullanılır. Mesela, bir resmin kontrastının yükseltilmesi gibi. **Nokta operasyonları sıfır hafıza operasyonlarıdır.**

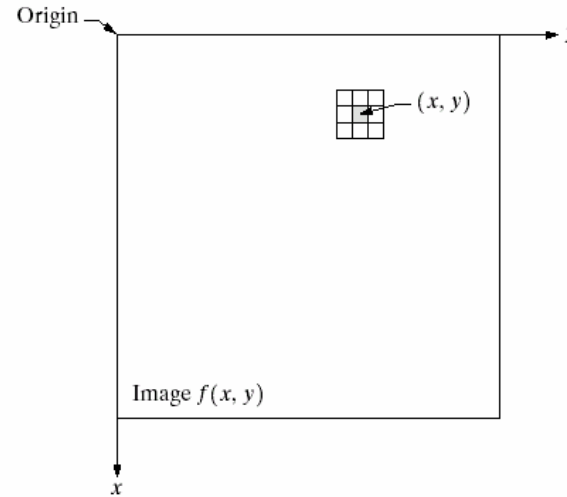
Bölgesel (lokal-Komşuluk ilişkili) işlemlerde merkez pikselin değeri komşu piksellerin değeri ile belirlenir. Filtreleme işlemlerinde çok kullanılır.

Global işlemlerde ise Domain dönüşümü (uzaysaldan frekans domanine veya tersi) yapılarak image üzerinde işlem yapılır.



Bir piksel (x,y) 'in komşuluk bölgesi veya komşuluk ilişkisi için; merkezi (x,y) olan kare, dikdörtgen tanımlama kullanılır.

FIGURE 3.1 A
 3×3
neighborhood
about a point
 (x, y) in an image.



Bu bölgenin merkezi, pikselden piksele hareket ettirilerek (Her yöne), etrafındaki farklı komşuları içine alır. T operatörü her bir lokasyona (x,y) uygulanarak lokasyonda işlenmiş $g(x,y)$ çıkışı elde edilir.

En küçük lokasyon (komşuluk ilişkisi-bölgesel) resim içindeki 1×1 'lik bölgedir. Bu bölge içinde yapılan işlemlerde (çalışılan pikselde) diğer piksellerin hiçbir etkisi olamaz. Yani o tek piksele yapılan işlemde komşu piksellerin rolü olamaz. Bu tür işlemlere Noktasal işlemler denir.

*** Dikkat Burada T yapılan işlemi belirtir. Yani her piksele; komşuluk ilişkisine göre gezilerek T'nin belirttiği işlem yapılır.

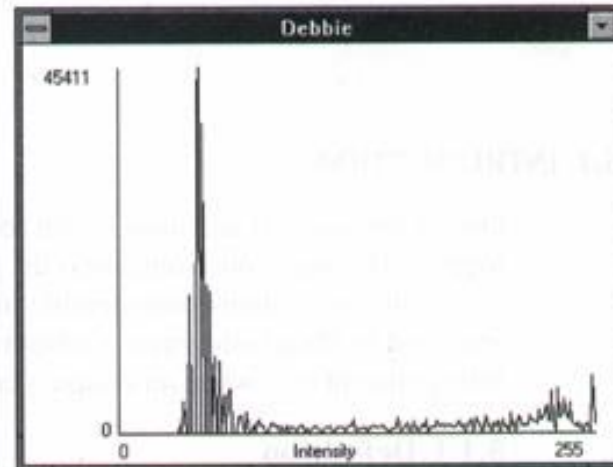
Komşuluk ilişkili operasyonlar

- 1x1'lik komşuluk ilişkisine dayanan (Noktasal) operasyonlar;
 - Kontrast Yayma (contrast stretching)
 - Eşikleme (Thresholding)
 - Aritmetiksel İşlemler
- Büyük çaplı komşuluk ilişkisine dayanan operasyonlar ise (maskeleme- filtreler (kernels, templates, pencereler,...)) v.b.
 - Görüntü netleştirme (Image sharpening)

GRİ SEVİYE HİSTOGRAMIN TARİFİ

Gri seviyesi histogramı, görüntü içindeki her bir gri seviyesine karşılık gelen ilgili gri seviyesindeki piksel sayısını karşılık düşüren bir fonksiyondur. Apsis eksenini gri seviyesini gösterirken ordinat eksenini ise meydana gelme sıklığını göstermektedir. Şekil bir örnek göstermektedir .

Histogram, sayısal bir resim içerisinde her renk değerinden kaç adet olduğunu gösteren grafiklerdir. Bu grafiğe bakılarak resmin parlaklık durumu ya da tonları hakkında bilgi sahibi olunabilir.



- Verilen kötü kontrastlı bir görüntüyü iyileştirmek için, görüntünün histogramını geniş bir alana yayabiliriz. Bu iki şekilde gerçekleştirilebilir.

1- Histogram Germe- Kontrast germe (Histogram stretching): Histogram germe, histogram değiştirmenin özel bir halidir.

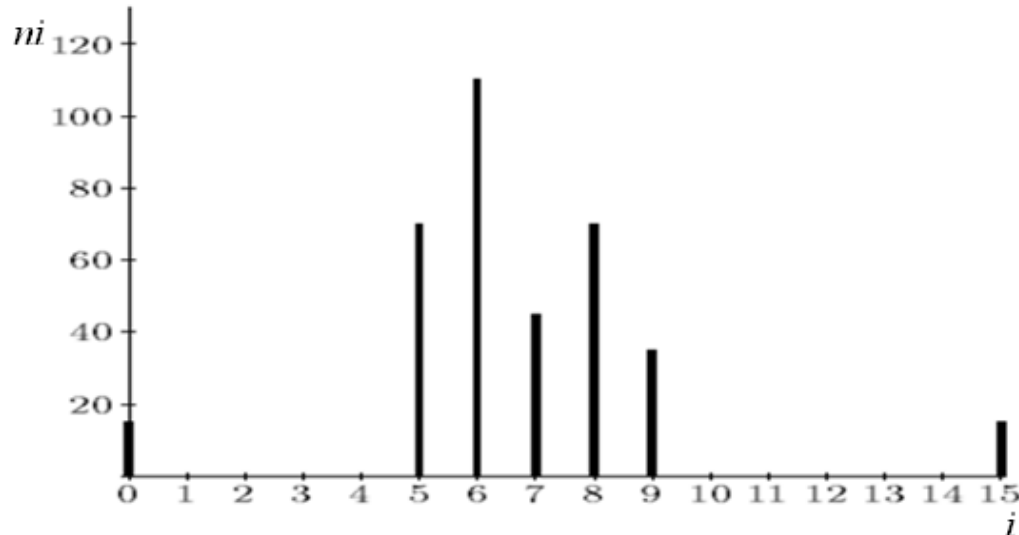
2- Histogram Eşitleme (Histogram equalization)

Histogram Germe (Açma) -(Histogram Stretching)

Kötü kontrastlı resimlerin (histogramı dar bir alana yayılmış görüntüler), kontrastlarını iyileştirmek için bir genel yöntemdir. Temel espirisi histogramı geniş bir bölgeye yaymaktır.

Varsayalım ki histogramı ve ilişkin tablosu aşağıda verilen bir görüntüye sahibiz. Burada; n_i , i.gri seviyedeki piksel sayısı değeridir. Biz bu gri seviye değerlerini, parça parça lineer bir fonksiyon uyguluyarak, orijinal aralığın dışına yayabiliriz.

Grey level i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
n_i	15	0	0	0	0	70	110	45	70	35	0	0	0	0	0	15

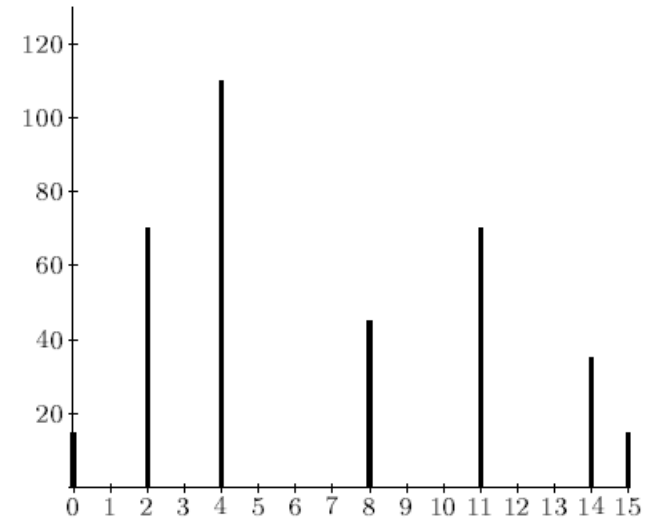
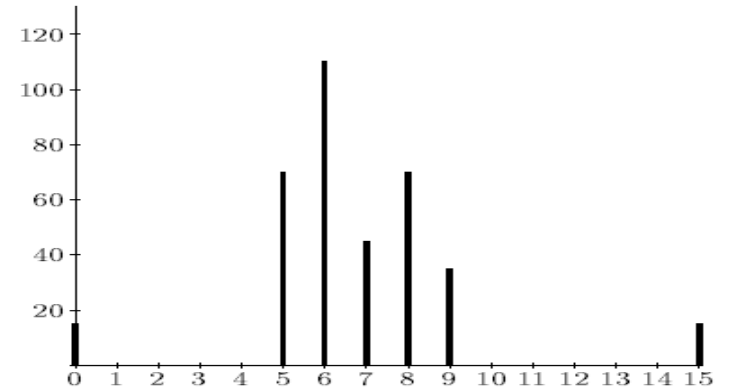
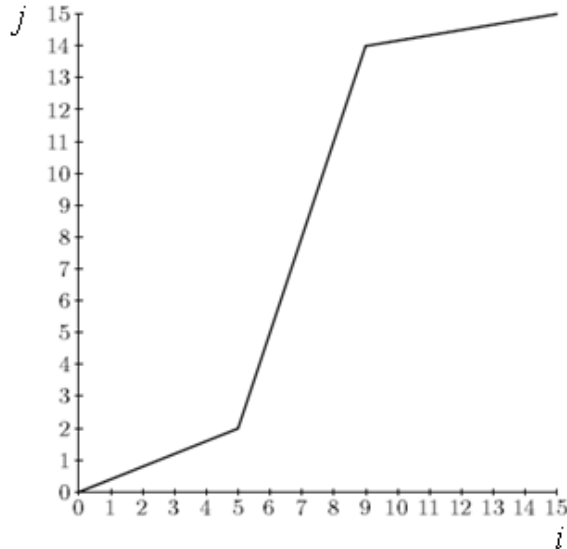


Bu fonksiyon; 5-9 arasındaki gri seviyeli pikselleri 2-14 gri seviyeleri arasına yayar. Aşağıdaki denklemde i , orijinal gri seviye ve j dönüşümden sonraki değerdir. Bu aralığın dışındaki Gri seviyeleri ya (bu durumda olduğu gibi) tek başına kalır ya da aşağıdaki parça parça lineer fonksiyonun işlevine göre değişir. Bu uygulamada tek başına (değiştirilmeden) bırakılmıştır.

Bu şekilde elde edilen histograma göre görüntü kontrastı iyileştirilmiştir. Görüntü germe işlemi için en uygun MATLAB fonksiyonu; **imadjust**

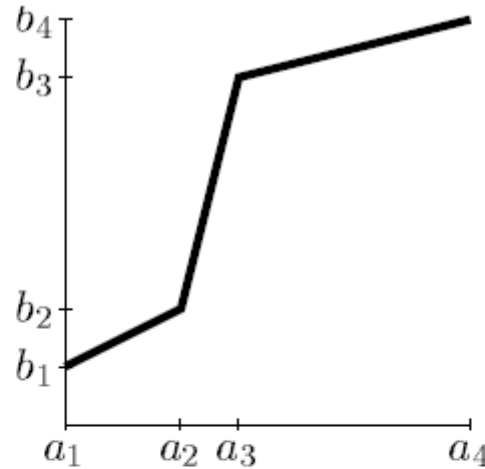
$$j = \frac{14 - 2}{9 - 5}(i - 5) + 2$$

i	5	6	7	8	9
j	2	5	8	11	14



Parça parça Lineer germe fonksiyonu için Bilgi NOTU: Parça parça Lineer germe fonksiyonu

a_i , b_i ve a_{i+1} ve b_{i+1} koordinatları arasında kalan piksel değerlerini bulmak için find komutu kullanılır.



$$y = \frac{b_{i+1} - b_i}{a_{i+1} - a_i}(x - a_i) + b_i$$

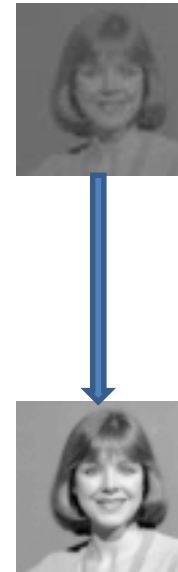
```
pix=find(im >= a(i) & im < a(i+1));  
out(pix)=(im(pix)-a(i))*(b(i+1)-b(i))/(a(i+1)-a(i))+b(i);
```

Burada im giriş görüntüsü, out çıkış görüntüsüdür. Aynı veri sınıfında olmalıdır.

```
>> th=histpwl(t,[0 .25 .5 .75 1],[0 .75 .25 .5 1]);  
>> imshow(th)  
>> figure,plot(t,th,'.'),axis tight
```

Histogram Germe(Histogram streching): Matlab uygulaması

```
A=imread('wom1.png');
A=rgb2gray(A);
[m,n]=size(A);
figure
imshow(A)
B=A;
minp=min(min(A));
maxp=max(max(A));
for i=1:m
    for j=1:n
        B(i,j)=(250/(maxp-minp))*(A(i,j)-minp);
    end
end
figure
imshow(B)
```

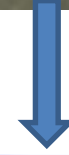
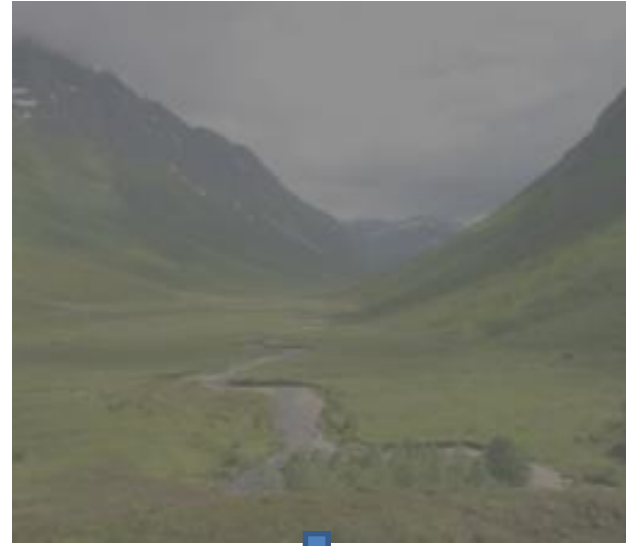


Ödev:

Yanda verilen RGB görüntüsü üzerinde histogram stretching yöntemini uygulayınız?

Not: Hazır Matlab fonksiyonu kullanılmayacaktır.

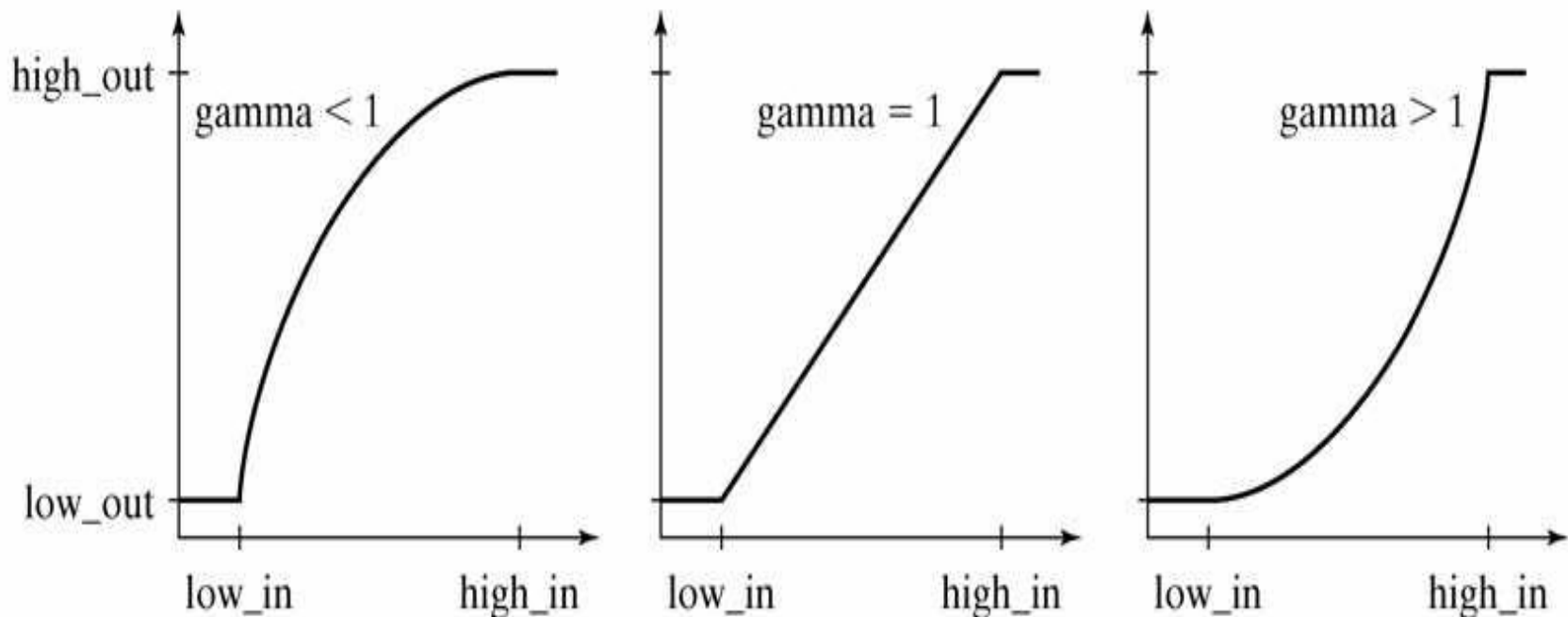
Ödev Teslim tarihi: 19.11.2013



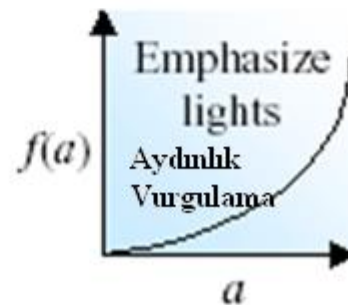
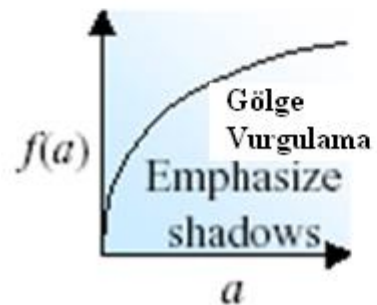
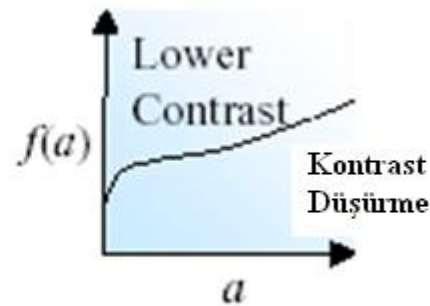
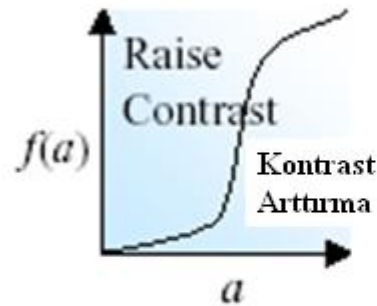
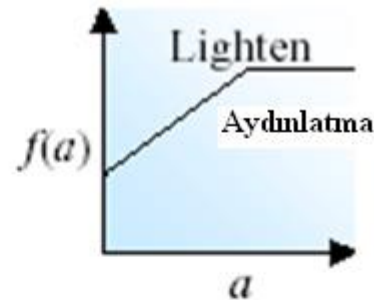
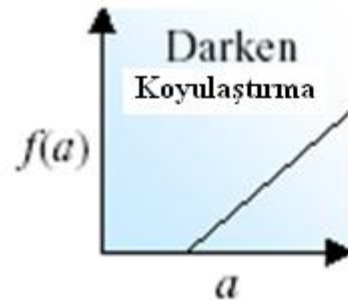
imadjust Fonksiyonu

Temel parlaklık işleme fonksiyonudur. Gri skala görüntülerde parlaklık seviyesi dönüşümü yapar. Giriş image'ı **f** uint8,uint16,double olabilir. g çıkışı da aynı formattadır.

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

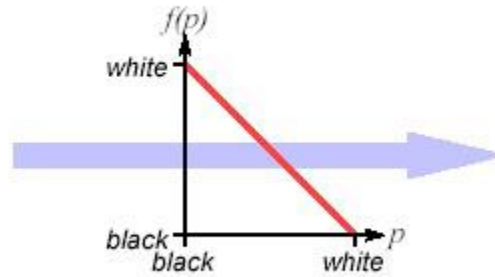


En basit görüntü işlemleri , o noktanın belirli bir fonksiyondan geçirilerek yeni nokta değerlerinin bulunmasına dayanır.



Bir resmin negatifini oluřturmak

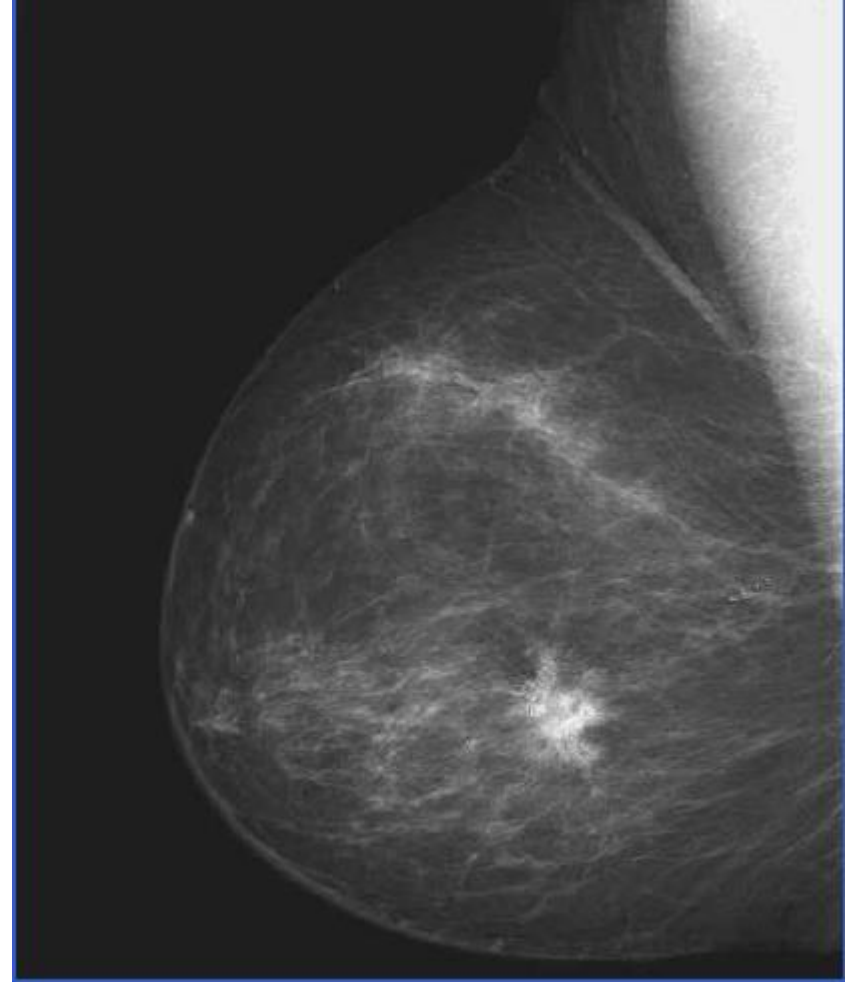
```
>> f = imread('rose_512.tif');  
>> g1=imadjust(f,[0 1],[1 0]);  
>> subplot(1,2,1),imshow(f)  
>> subplot(1,2,2),imshow(g1)
```



Bir radyolojik görüntü üzerine inceleme (Sayısal Mamografi image'ı)

```
>> f=imread('breast.tif');
```

Resimde; göğüs üzerinde
küçük bir lezyon göze
çarpmaktadır.



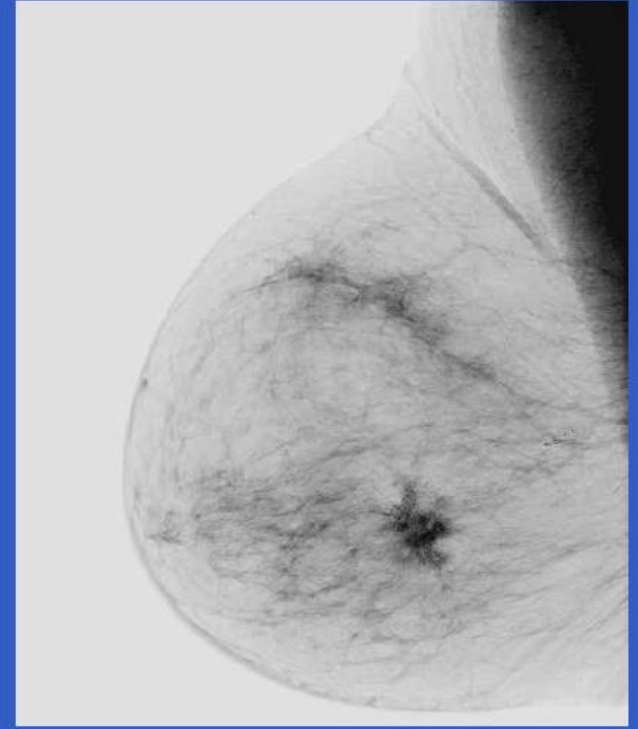
Resimlerin negatifleri birçok uygulamada kullanılmaktadır, bunlara örnek olarak; medikal röntgen, ultrasound görüntüleri verilebilir.

İnsan gözü, yapısı gereği açık tonlar arasındaki farklılığı koyu tonlar arasındaki farklılığa göre daha iyi algılayabilmektedir. Resmin gerektiği durumlarda negatifinin alınması algılamanın gücünü artırabilmektedir.

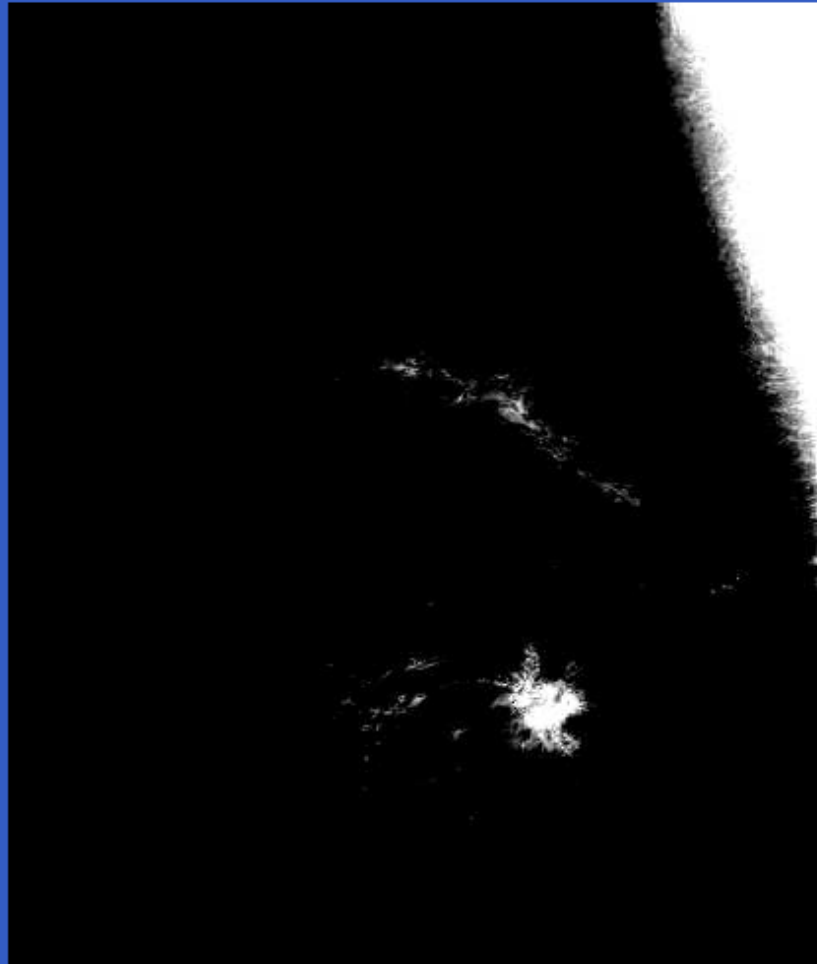
Bu işlem giriş resimdeki gri-seviye değerlerinin ters çevrilmesiyle yapılmaktadır. Örneğin orijinal resimdeki bir pikselin değeri 255 (beyaz) ise negatifinde aynı pikselin değeri 0 (siyah) olacaktır.

Gri-seviye resim için pikselin maksimum alabileceği değer 255 'dir. Bu resmin negatifini almak için resimdeki piksellerin renk değerlerini maksimum değer olan 255'den çıkarılıp ortaya çıkan sonuç aynı piksele atanırsa ve bu işlem resmin tüm piksellerine uygulanırsa orijinal resmin negatifi alınmış olur. Buradan da anlaşılacağı gibi bir resmi kendi negatifi ile toplarsak ortaya beyaz bir resim çıkar, yani resim içerisindeki tüm piksellerin değeri 255 olur.

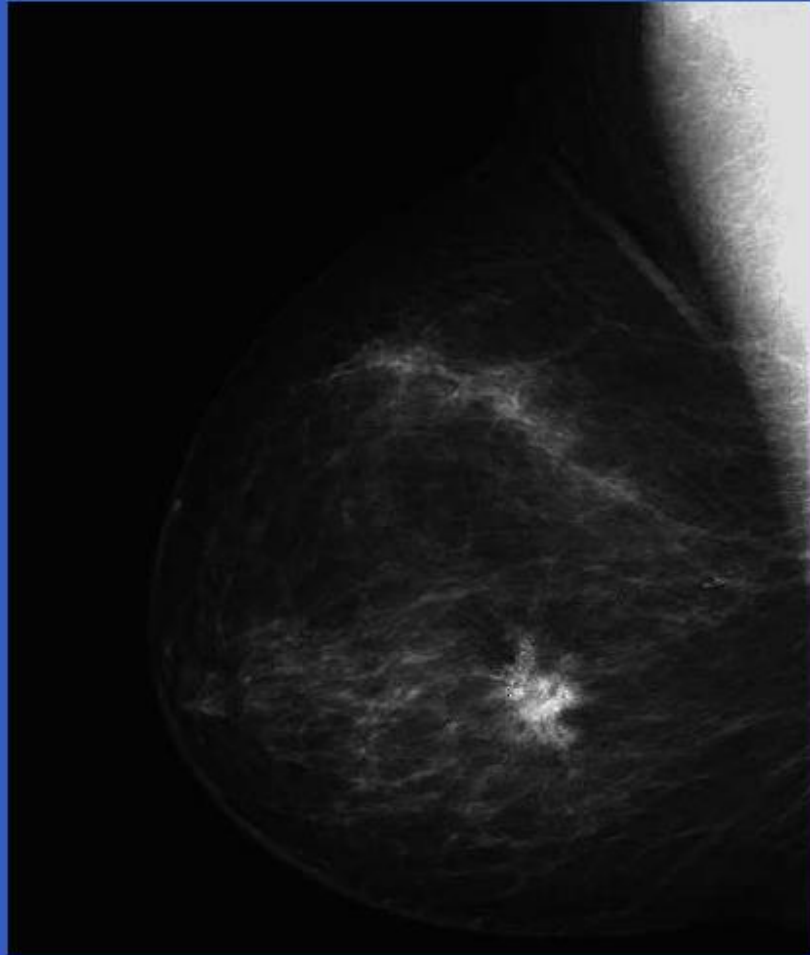
```
>> g1=imadjust(f,[0 1],[1 0]);
```



```
>> g2=imadjust(f,[0.5 0.75],[0 1]);
```



```
>> g3=imadjust(f, [], [], 2);
```



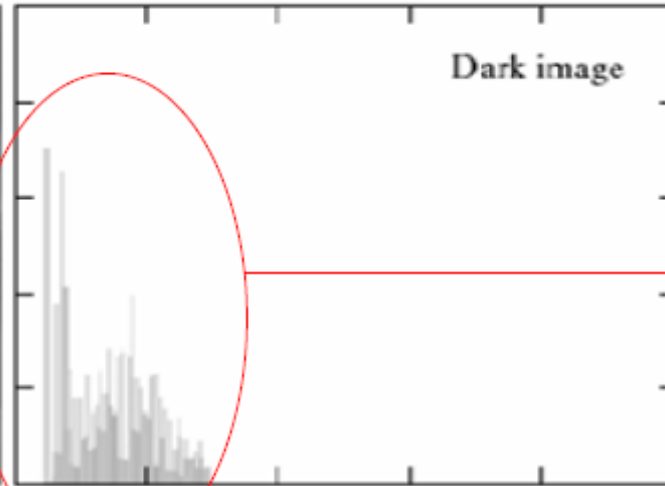
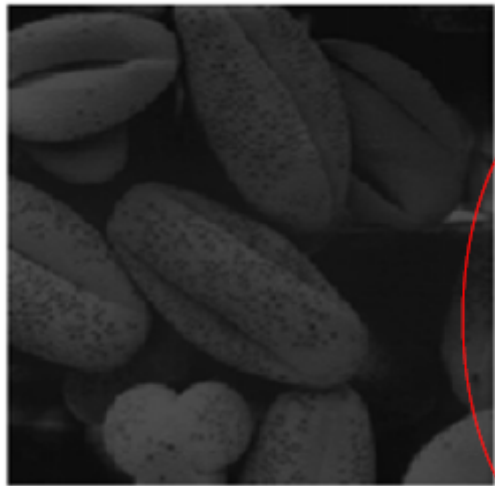
Histogram Processing and Function Plotting

(Histogram İşleme ve fonksiyon çizimi)

- Gri seviye histogramlarından elde edilmiş bilgilere dayalı parlaklık dönüşüm fonksiyonları, görüntü işlemede önemli bir rol oynar.
- Bunlar görüntü iyileştirme, sıkıştırma, segmentasyon, tarif etme v.b.
- Sayısal görüntü işlemede en basit ve en çok kullanılan araçlardan birisi gri seviyesi histogramıdır. Bu fonksiyon bize görüntünün gri seviyesi içeriğini vermektedir.

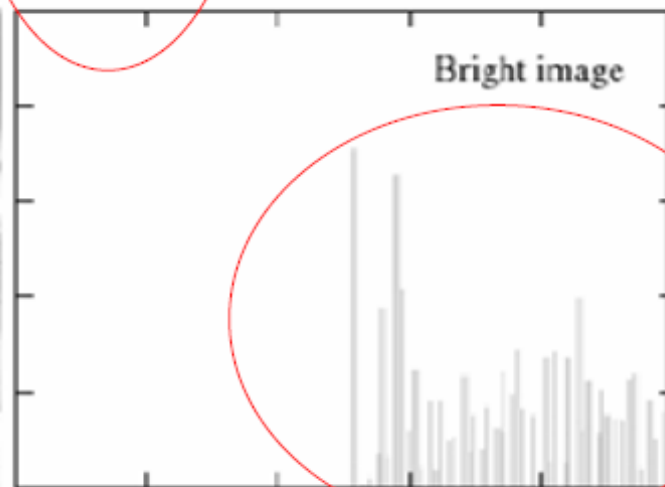
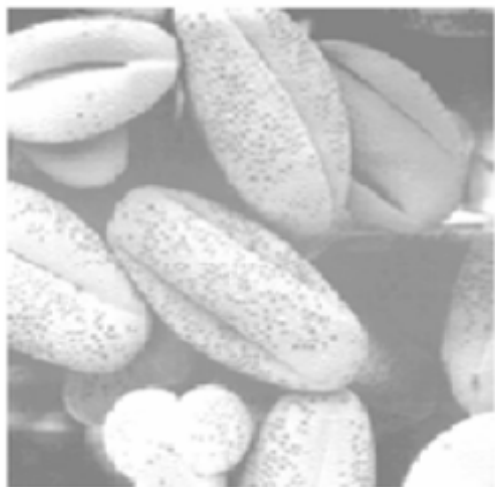
Histogramdan elde edilen bilgiler

- Bir görüntünün histogramı, o görüntü hakkında önemli bilgiler verir. Önemlileri aşağıdadır.
- Koyu (Karanlık) bir görüntünün histogram grafiğinin düşük gri seviye bölgesine yığılacağı açıktır.
- Parlak (Açık renk) düzgün bir görüntünün histogram grafiğinin büyük gri seviye bölgesine yığılacağı açıktır.
- Eğer histogram bir bölgeye yığılmış ise (yani gri seviye ekseninin belirli bir bölgesine) bu görüntünün kontrastı kötüdür denir.
- İyi kontrastlı bir resmin histogram grafiği tüm gri seviye değerlerine eşit yayılmış olduğunu açıklar.



Dark image

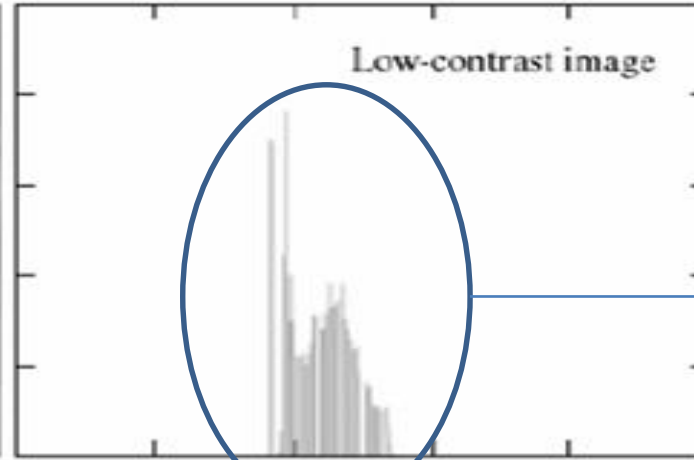
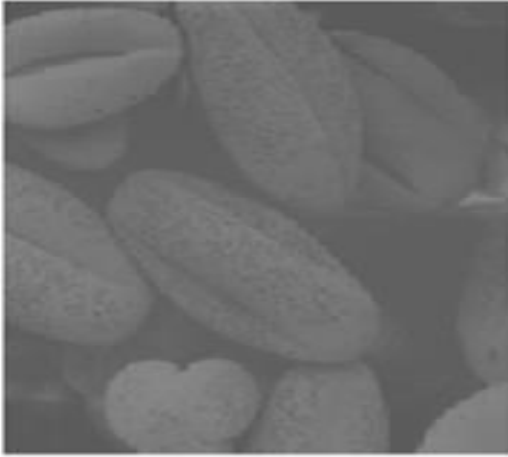
→ Karanlık imge



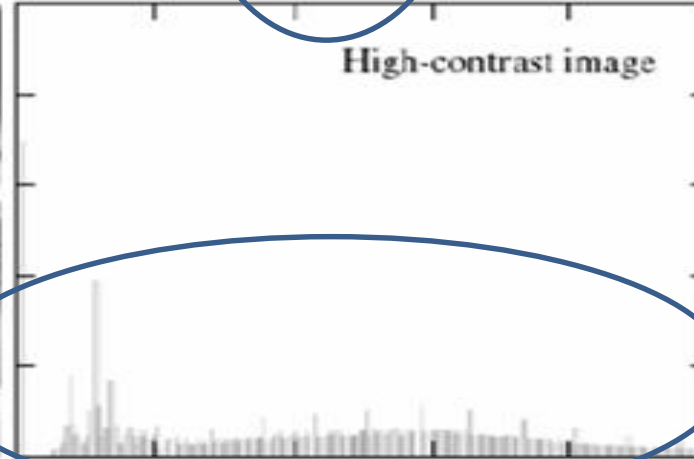
Bright image

→ Parlak imge

- Genelde küçük yayılım gösteren histogramların kontrastı düşüktür, daha geniş alana yayılmış olan histogramların kontrastı daha yüksektir.
- Aralığın aşağı sonunda sıkışmış olan histogram koyu bir görüntüye ait olurken, aynı aralığın yukarı sonunda gruplanmış olan histogram parlak bir görüntüye aittir.
- Histogram aynı zamanda, daraltma, genişletme veya kaydırma işlemleri için bir karşılık düşürme fonksiyonu ile de değiştirilebilir. Histogram genişletme ve daraltma gri seviyesi değişimleri biçimindedir ve bazen histogram boyutlandırma olarakta adlandırılır.



→ Karşıtlığı düşük



→ Karşıtlığı yüksek

Histogram'la ilgili

- Bir $[0, G]$ sahasında, Gri seviye toplam sayısı L olan bir sayısal görüntünün histogramı;

$$h(r_k) = n_k$$

ayrık fonksiyonuyla tanımlanır. Burada r_k , $[0, G]$ aralığındaki grilik seviyesi değeridir. Grilik seviyesi r_k olan piksel sayısı ise n_k dır. (Örneğin 125.gri seviye değerinde ($r_k=125$), 345 piksel olabilir ($n_k=325$))

Buradaki G , uint8'de 255, double görüntü sınıfında 1'e eşittir.

Not: uint8 ve uint16 sınıfında $G=L-1$ alınmalıdır.

Normalize edilmiş histogram

$h(r_k)$ 'nin tüm elemanlarının, görüntüdeki tüm piksel sayısına bölünmesiyle elde edilir. Normalize histogram çoğu uygulamada kullanılır. (r_k .grilik seviyesindeki piksel sayısının toplam piksel sayısına bölünmesi ile o grilik seviyesindeki pikseller normalize edilmiş olur.)

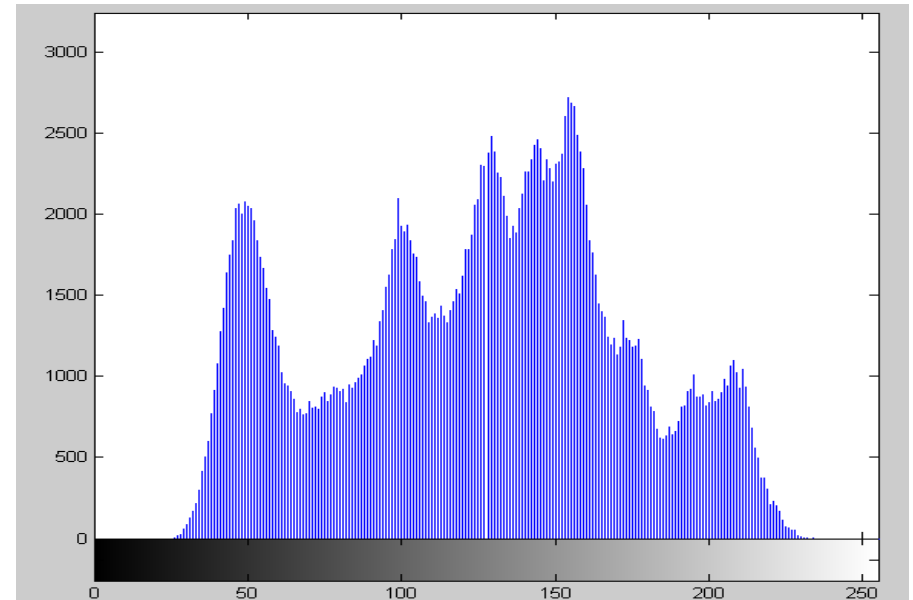
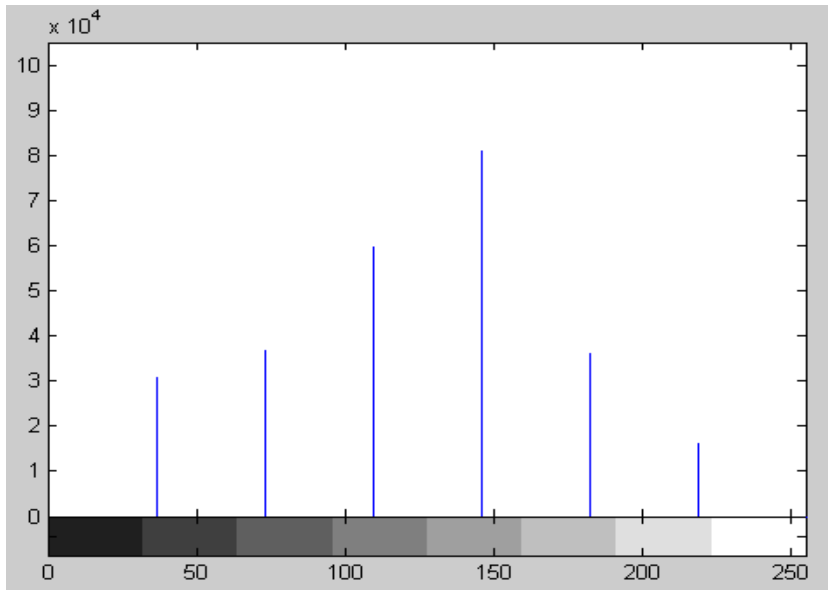
$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}$$

for $k = 1, 2, \dots, L$.

```
h=imhist(f,b)
```

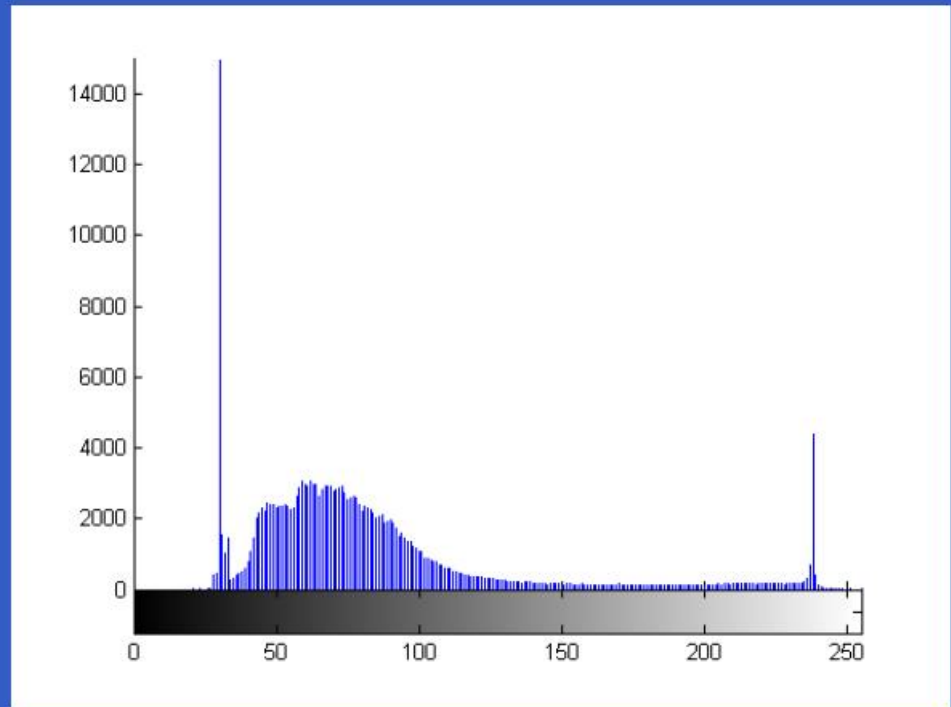
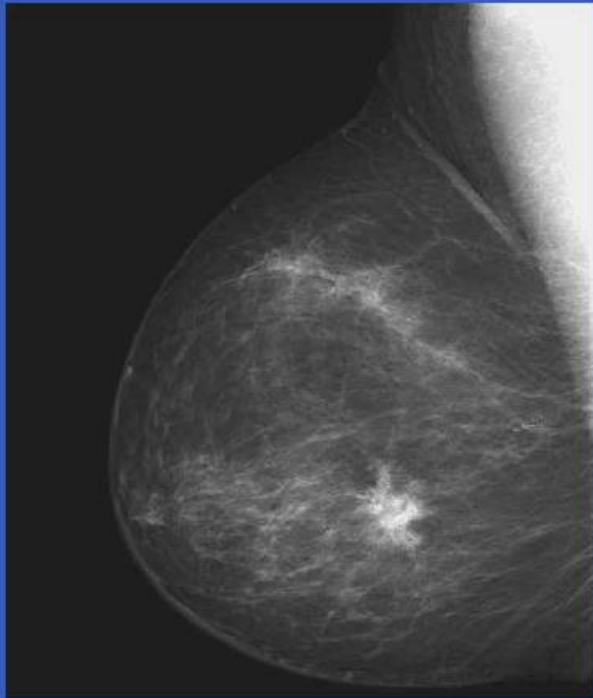
Burada **f**, histogramı istenen görüntü, **h** görüntünün histogramı, **b** ise yatay eksen bölme sayısıdır.

Uint8 sınıfından bir görüntü $b=8$ alınırsa sadece 8 gri seviyeye bölünür. B'yi hiç kullanmazsak /default değerdir) yatay eksen 256 gri seviyeye bölünür.



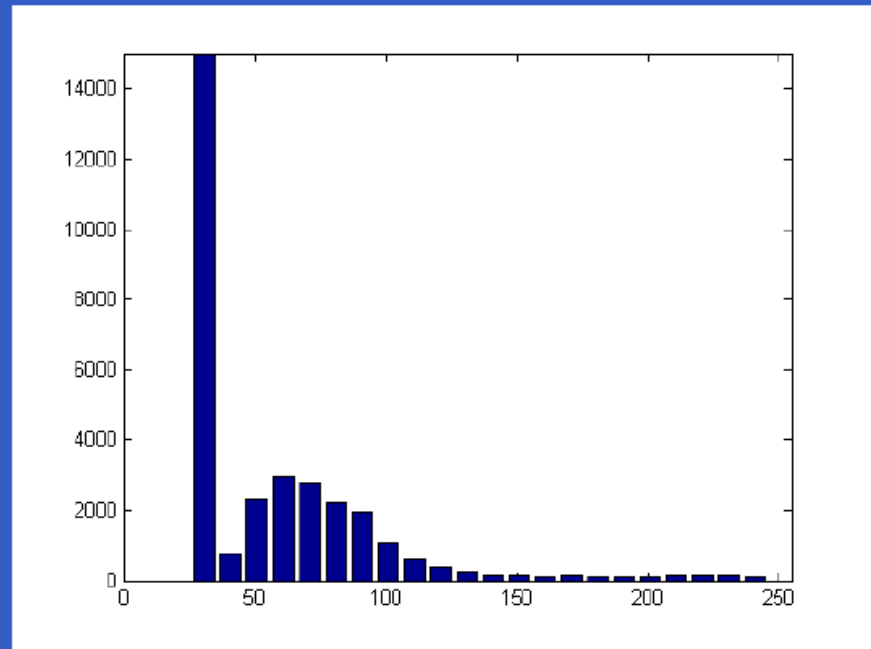
Histogram fonsiyonun çizdirme-örnekler

```
>> f=imread('breast.tif');  
>> imshow(f), imhist(f)
```



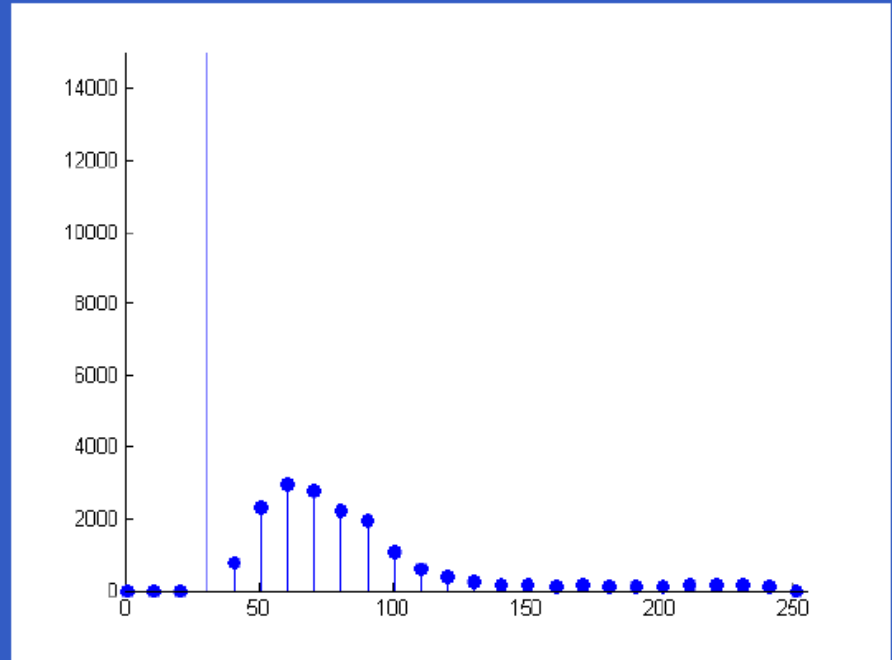
Histogramı BAR formunda göstermek

```
>> h=imhist(f);  
>> h1=h(1:10:256);  
>> horz=1:10:256;  
>> bar(horz,h1)  
>> axis([0 255 0 15000])  
>> set(gca,'xtick',0:50:255)  
>> set(gca,'ytick',0:2000:15000)
```



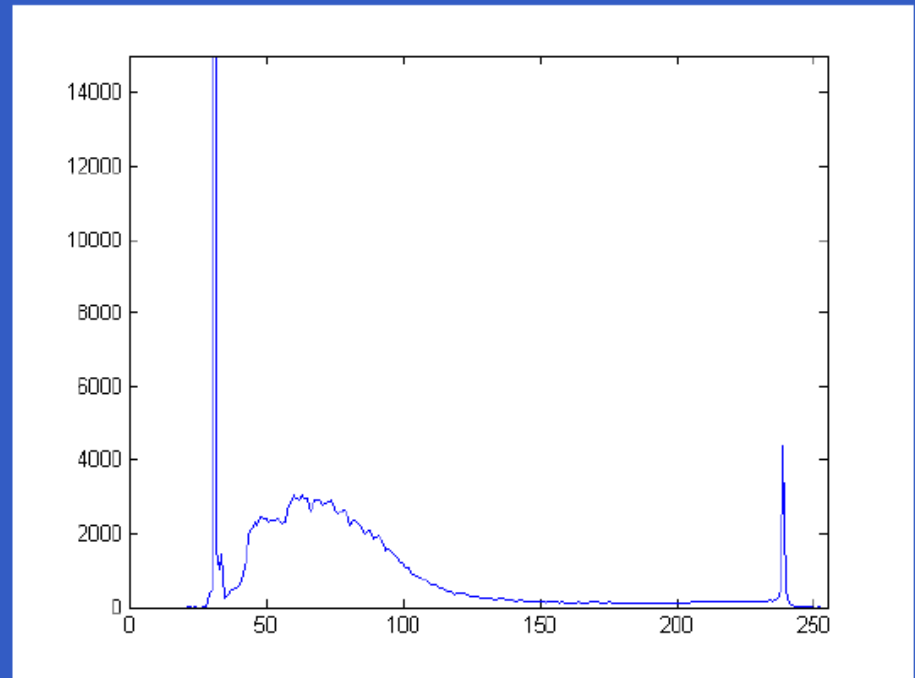
Histogramı STEM formunda göstermek

```
>> h=imhist(f);  
>> h1=h(1:10:256);  
>> horz=1:10:256;  
>> stem(horz,h1,'fill')  
>> axis([0 255 0 15000])  
>> set(gca,'xtick',0:50:255)  
>> set(gca,'ytick',0:2000:15000)
```



Histogramı plot ile(sürekli formda çizdirmek)

```
>> h=imhist(f);  
>> plot(h)  
>> axis([0 255 0 15000])  
>> set(gca,'xtick',0:50:255)  
>> set(gca,'ytick',0:2000:15000)
```



Bazı faydalı çizim fonksiyonları

- `plot(horz,v,'color_linestyle_marker')`
- `bar(horz,v,width)`
- `stem(horz,v,'color_linestyle_marker','fill')`
- `axis([horzmin horzmax vertmin vertmax])`
- `xlabel('text string','fontsize',size)`
- `ylabel('text string','fontsize',size)`
- `text(xloc,yloc,'text string','fontsize',size)`
- `title('titlestring')`

Faydalı çizim fonksiyonları

Symbol	Color	Symbol	Line Style	Symbol	Marker
k	Black	-	Solid	+	Plus sign
w	White	--	Dashed	o	Circle
r	Red	:	Dotted	*	Asterisk
g	Green	-.	Dash-dot	.	Point
b	Blue	none	No line	x	Cross
c	Cyan			s	Square
y	Yellow			d	Diamond
m	Magenta			none	No marker

Histogram Eşitleme

- Historam germenin; bir kullanıcı girişine ihtiyaç duyma gibi bir dezavantajı vardır. Histogram eşitleme ise otomatik bir **presedürdür**. (Aslında histogram eşitleme genel anlamada histogram germe kapsamına sokulabilir. Fakat germede parça-parça linerer veya değişik fonksiyonlar kullanmak gerekirken , eşitlemde otomatik işlem yapılır.)
- İdeal olarak Histogram eşitleme; Giriş histogramını, her gri seviyesinde eşit piksel sayısına sahip bir histograma dönüştürme işlemi gibi düşünülebilir. Bu pratikte mümkün değildir.
- Bu yöntem histogramı dar olan resimler ya da resim içindeki bölgeler için daha iyi sonuç verir. Yani Histogram eşitleme renk değerleri düzgün dağılımlı olmayan resimler için uygun bir görüntü iyileştirme metodudur. Resmin tümüne uygulanabileceği gibi sadece belli bir bölgesine de uygulanabilir. Tüm resme uygulanırsa global histogram eşitleme, resmin belli bir bölgesine uygulandığında ise lokal histogram eşitleme adını alır.

Histogram eşitlemenin özeti

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

- **Uygulanışı:**

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

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

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

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

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

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

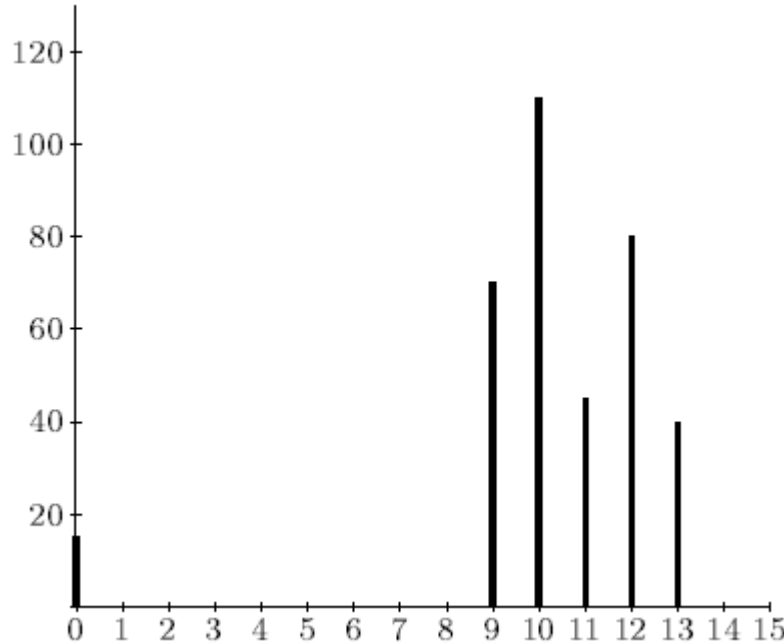
L: mümkün olan (veya istenilen) toplam gri seviye sayısı(8 bit renk derinliğinde 255 v.b)

s_k : Daha iyi kontrastlı bir görüntü elde etmek için gri seviye dönüşüm değeri.

ÖRNEK:

Gri seviye i	: 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(j. Gri değerli piksel sayısı) Π_j	: 15	0	0	0	0	0	0	0	0	70	110	45	80	40	0	0

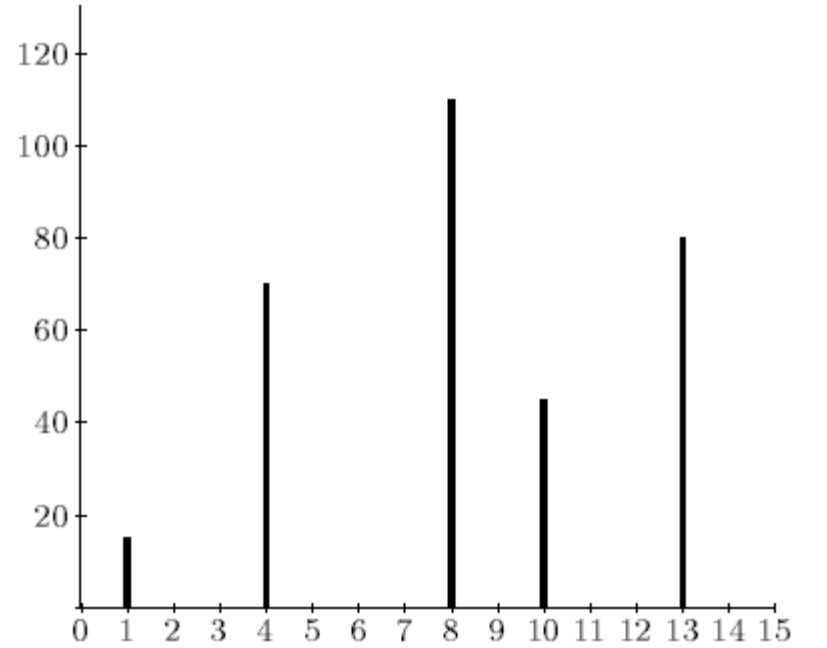
Yukarıda, renk derinliği 4 bit olan gri seviyeli bir görüntünün histogram tablosu görülmektedir. Bunun histogram grafiği ise aşağıdaki şekildedir. (Toplam piksel sayısı $n=360$, istenen gri seviye $L=16$)



Buna göre aşağıdaki denklemden; gri seviye değerlerini tek tek hesaplarsak;

$$\left(\frac{n_0 + n_1 + \cdots + n_i}{n} \right) (L - 1)$$

Grey level i	n_i	Σn_i	$(1/24)\Sigma n_i$	Rounded value
0	15	15	0.63	1
1	0	15	0.63	1
2	0	15	0.63	1
3	0	15	0.63	1
4	0	15	0.63	1
5	0	15	0.63	1
6	0	15	0.63	1
7	0	15	0.63	1
8	0	15	0.63	1
9	70	85	3.65	4
10	110	195	8.13	8
11	45	240	10	10
12	80	320	13.33	13
13	40	360	15	15
14	0	360	15	15
15	0	360	15	15



Original grey level i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Final grey level j	1	1	1	1	1	1	1	1	1	4	8	10	13	15	15	15

Histogram Eşitleme

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

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



Örnek:

Görüntü
bloğu

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

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

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

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

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

histogram

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

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

cdf

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

Ödev:

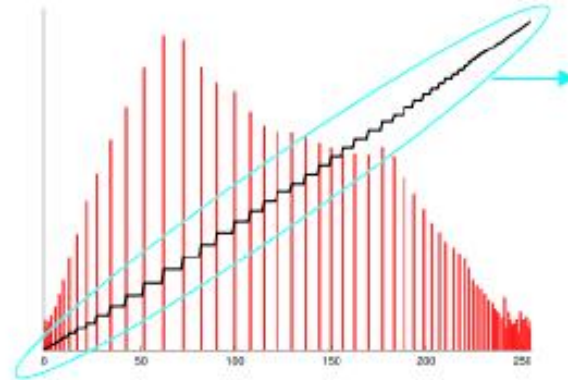
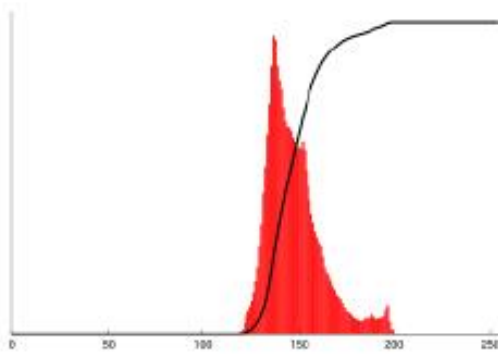
Ödev Teslim Tarihi:
19.11.2013

- Gri bir görüntü üzerinde histogram eşitleme işlemini gerçekleştiren MATLAB kodunu yazınız?



Histogram Eşitleme

- İmgenin olasılık dağılım fonksiyonu doğrusallaştırılmaktadır.



Doğrusallaştırılmış
cdf

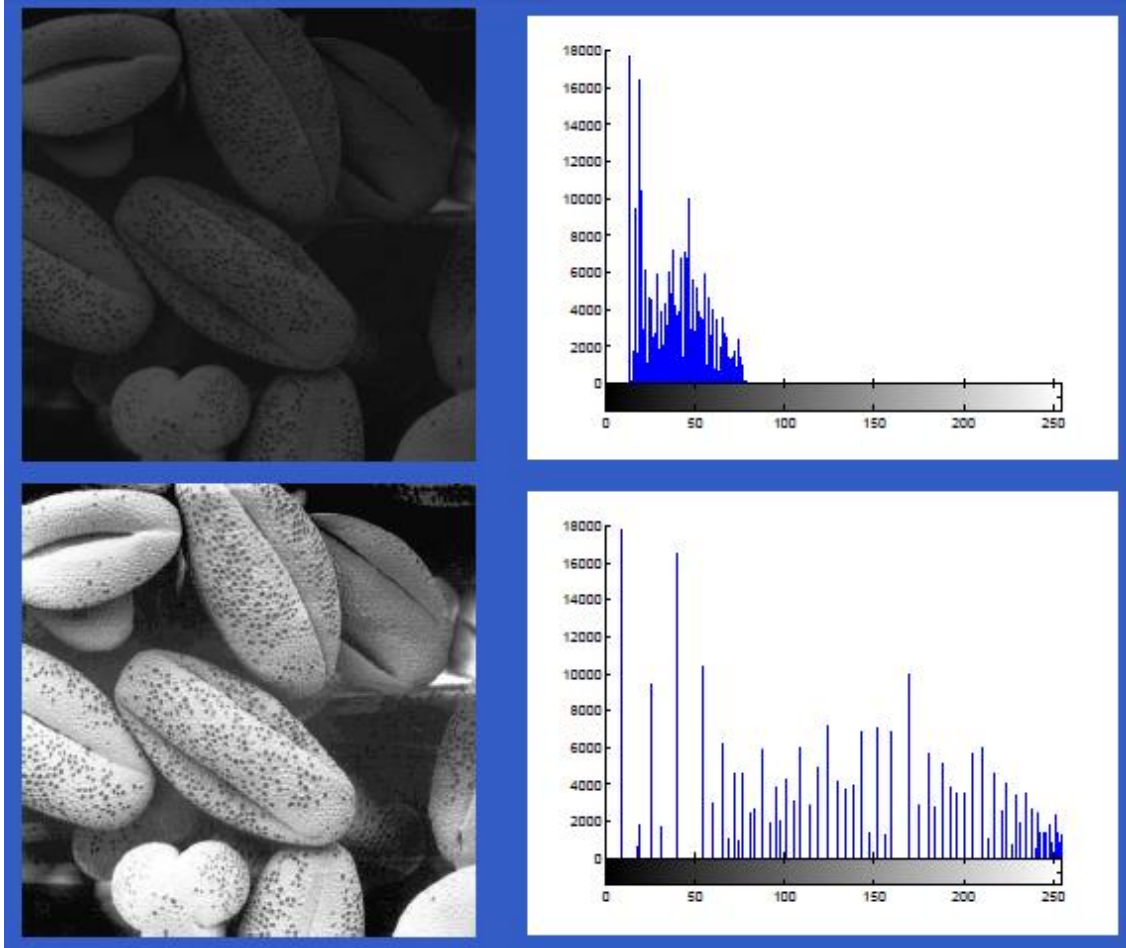
```
g=histeq(f,nlev)
```

- f : giriş görüntüsüdür.
- $nlev$: Çıkış görüntüsü için özelleştirilmiş(belirtilmiş) gri seviye sayısıdır.
- $nlev$ L (giriş görüntüdeki mümkün seviyelerin toplam sayısı) 'ye eşit ise, o zaman `histeq` doğrudan dönüşüm fonksiyonu (önceki slaytta tanımlanan) uygular.
- Eğer $nlev$ L 'den küçük ise; `histeq` gri seviyeleri düz bir histogram olacak şekilde dağıtmaya çalışır.
- `imhist`'in aksine `histeq`'in default değeri 64'dür.

Histogram eşitlemeye örnek

```
>> f=imread('pollen.tif');  
>> imshow(f)  
>> figure, imhist(f)  
>> ylim('auto')  
>> g=histeq(f,256);  
>> figure, imshow(g)  
>> figure, imhist(g)  
>> ylim('auto')
```

Bir polen'in elektron mikroskopta 700 defa büyütülmüş resmi



Histogram Karşılaştırma (Histogram Matching)

- Bir görüntünün histogramının, bir diğer görüntü ile uyumlu bir hale gelmesini veya belirli bir fonksiyona uymasını isteyebiliriz.

```
g=histeq(f,hspec)
```

- Burada f giriş görüntüsüdür.
- $hspec$ belirtilen (istenen) histogramdır. (İstenen değerlerden oluşturulmuş satır vektörü v.b)
- g çıkış görüntüsüdür. Bunun histogramı istenen histograma yaklaştırılmıştır.

Aritmetiksel İşlemler

Noktasal operasyonlardan olan aritmetiksel işlemler;

$$y=f(x)$$

gibi bir basit fonksiyonun, görüntüdeki herbir gri seviye değerine uygulanmasından ibarettir. Gri seviye resimde genellikle 0,...255 seviye değerleri arasında çalışıldığından, yukarıdaki fonksiyon bu değerler arasında geçerli olmalıdır. Bu basit fonksiyon ;

Bir sabit değeri (c) herbir piksele ekler veya çıkartır ($Y=x \pm c$)

Veya herbir pikseli bir sabit değer (c) ile çarpar ($y= x.c$)

İşlem sonuçlarının tamsayıya yuvarlatılması yapılır.

Ayrıca 0...255 arasında çalışıldığında kırpma yapılır.

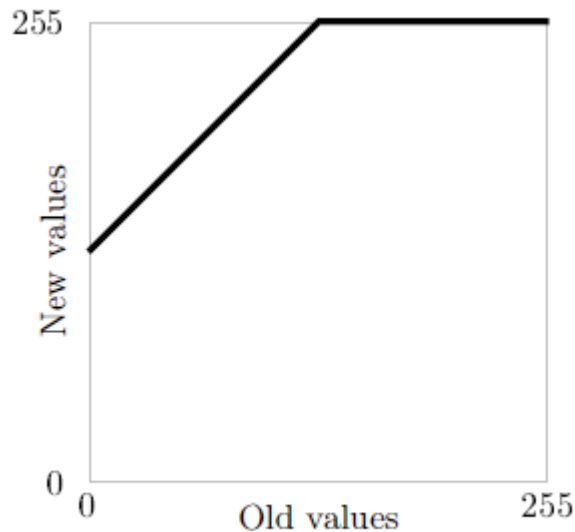
$$y > 255 \quad \text{ise} \quad y=255$$

$$y < 0 \quad \text{ise} \quad y=0 \quad \text{yapılır.}$$

$Y=f(x)$ fonksiyonunun etkisini incelemek için aşağıdaki grafikleri inceleyelim.

Resim piksellerine 128 eklendiğinde, resim piksellerindeki 127 ve daha büyük değerler 255 ile ifade edilecektir.

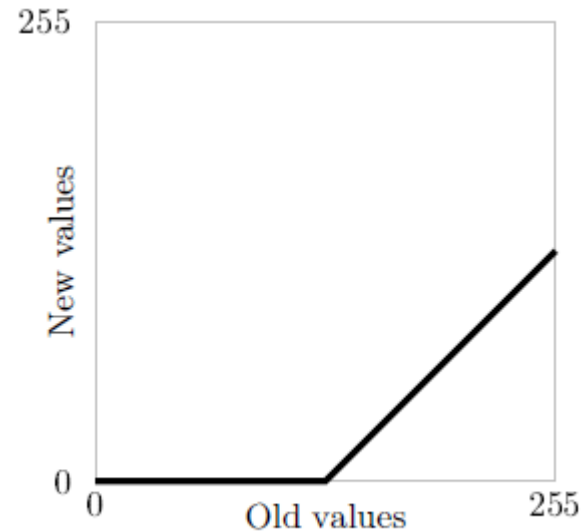
Not: Resim piksellerine ilave edilen her değer resmin daha açık renkte görünmesine neden olur.



Adding 128 to each pixel

Resim piksellerinden 128 çıkarıldığında, değerleri 128'den küçük olan piksellerin seviye değerleri 0 ile ifade edilecektir.

Not: Resim piksellerinden çıkarılan her değer resmin daha koyu renkte görünmesine neden olur.



Subtracting 128 from each pixel

ÖRNEK

```
>> b=imread('blocks.tiff');
```

```
>> whos b
```

Name	Size	Bytes	Class
b	256x256	65536	uint8 array

Grand total is 65536 elements using 65536 bytes

```
>> b1=b+128
```

```
??? Error using ==> +
```

```
Function '+' is not defined for values of class 'uint8'.
```

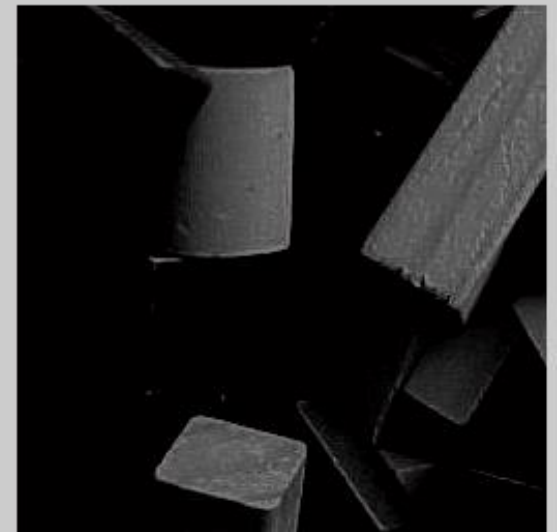
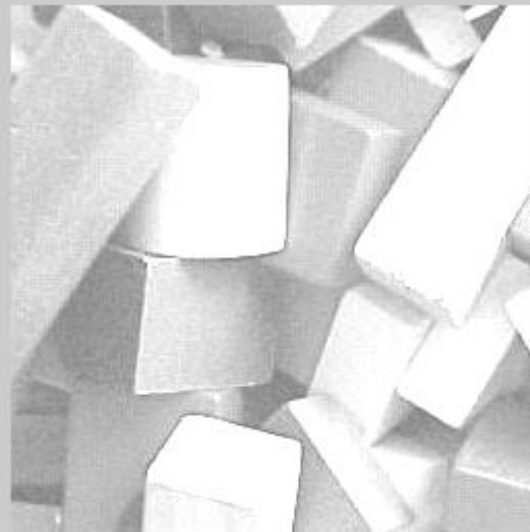
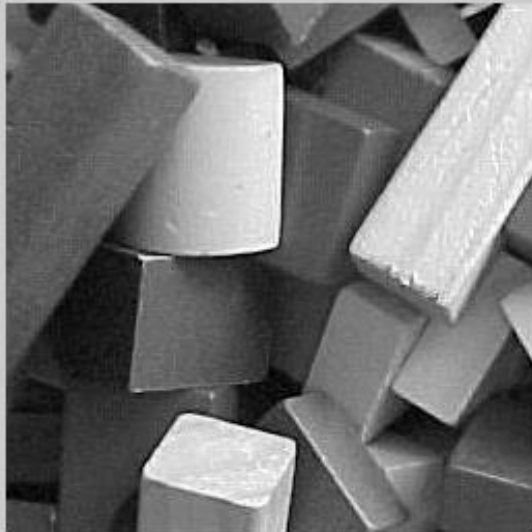
```
>> b1=uint8(double(b)+128); 1.yöntem
```

```
>> b1=imadd(b,128);
```

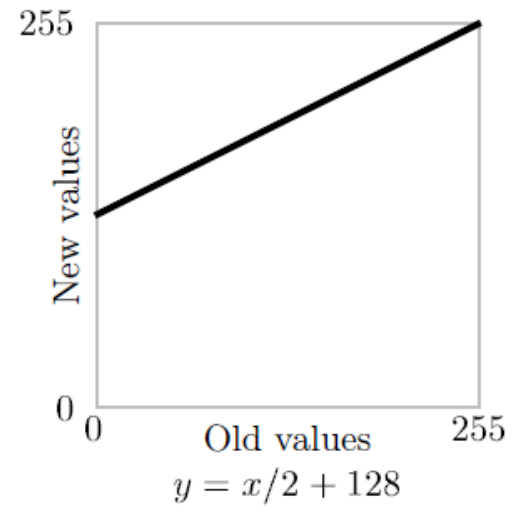
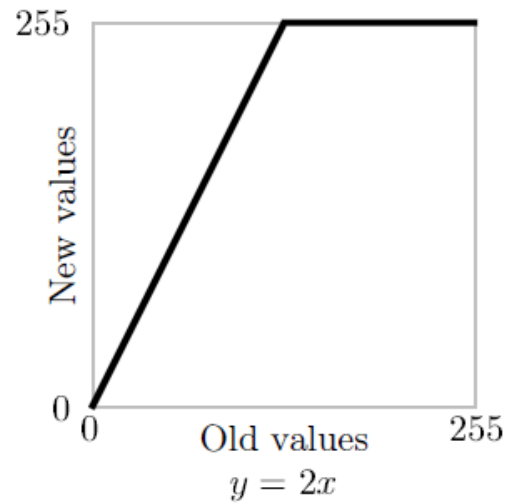
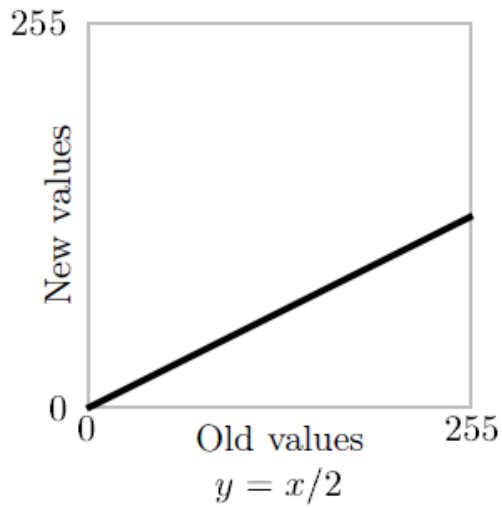
2.yöntem: Bu yöntem daha elegant bir yöntemdir.

```
>> b2=imsubtract(b,128);
```

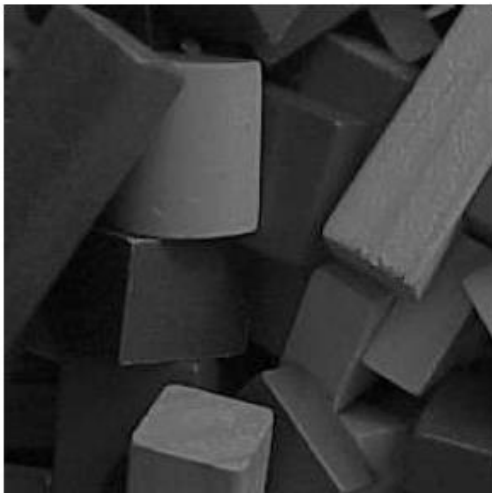
```
>> imshow(b),figure,imshow(b1),figure,imshow(b2)
```



Çarpma ve bölme uygulamaları



$y = x/2$ `b3=immultiply(b,0.5); or b3=imdivide(b,2)`
 $y = 2x$ `b4=immultiply(b,2);`
 $y = x/2 + 128$ `b5=imadd(immultiply(b,0.5),128); or b5=imadd(imdivide(b,2),128);`



b3: $y = x/2$



b4: $y = 2x$



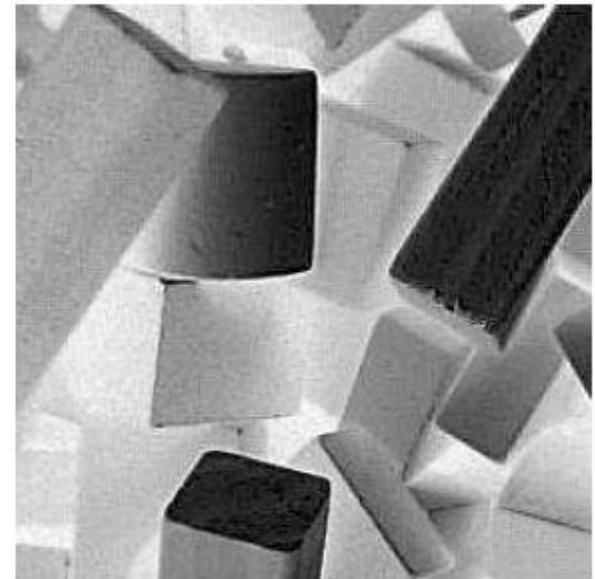
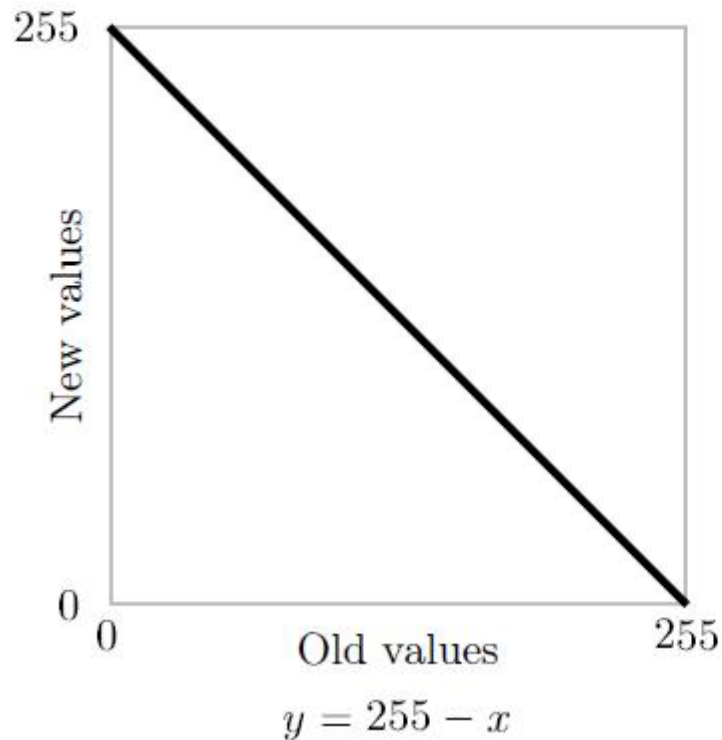
b5: $y = x/2 + 128$

Aritmetiksel İşlemler (Ters Alma)

- Gri skala görüntüde tersleme, görüntünün negatifi anlamına gelir.
- Bir görüntü matrisi m 'nin elemanları;
- double class ise; gray scala değeri 0-1 arasında değişir. Biz negatifi elde etmek için;
$$>> 1-m$$
- Eğer image Binary ise:
$$>> \sim m$$
- Eğer uint8 class ise; hazır fonksiyon olan
- `imcomplement` kullanılır.

- $Y=255 - x'$ e göre oluşturulan fonksiyon ve onun işlenişi aşağıdadır.

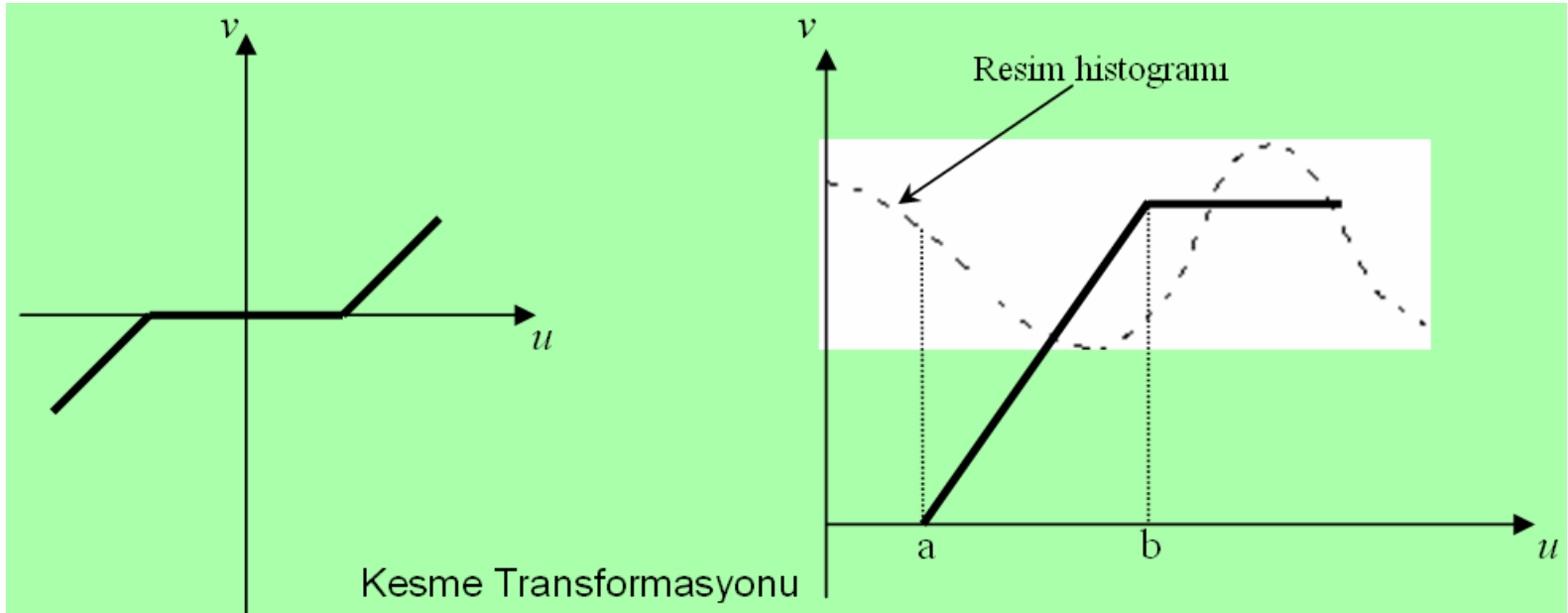
```
>> bc=imcomplement(b);  
>> imshow(bc)
```



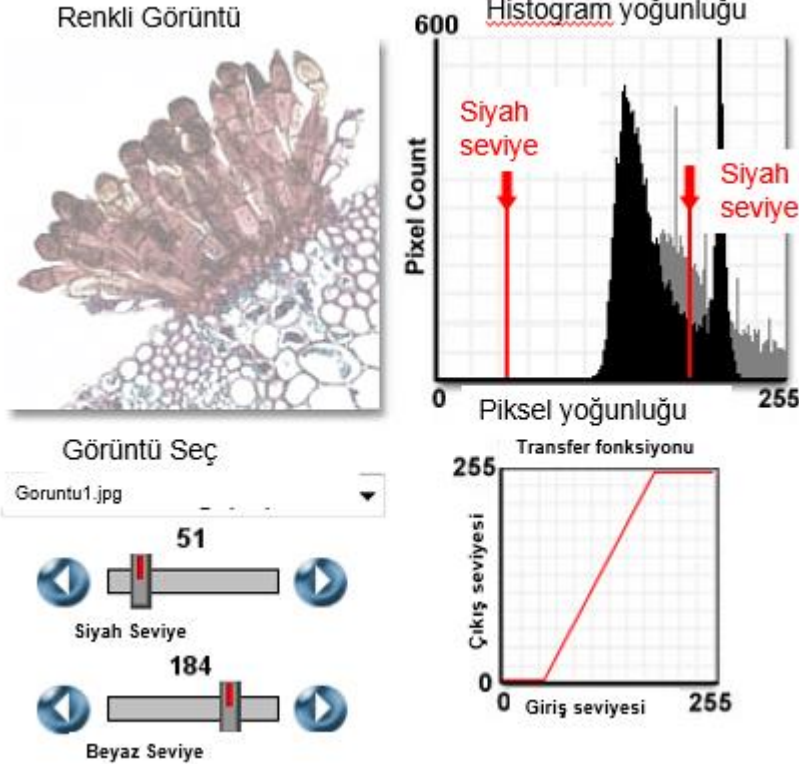
Diğer Nokta operasyonu İşlemleri

KESME (CLIPPING)

- Kontrast genişletmenin özel bir durumu 'kesme'dir. Bu işlem giriş sinyalinin bulunduğu aralık $[a,b]$ bilindiğinde gürültü azaltmak için oldukça kullanışlıdır.



Ödev:



- Bu ödevde sizden bir RGB görüntüsüne istenilen transfer fonksiyonunu uygulamanız isteniyor. Listedен bir görüntü seçilecek bu görüntüye yanda verilen transfer fonksiyonu uygulanacaktır.

Örneğin Beyaz seviyeyi gösteren kaydırma çubuğu 184 olduğu için görüntüde 184 üzerinde olanlara 255 değeri atılmaktadır. Siyah kaydırma çubuğu 51'de olduğu için 51'den düşük değere sahip piksellere 0 değeri atanmaktadır.

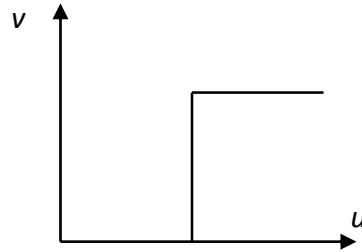
Beyaz ile siyah kaydırma çubuğu arası mainimum 3 piksel değeri olabilir. Yani beyaz seviye 130 ise siyah seviye maksimum 127 olabilir.

- Ödev Teslim tarihi:**
26.11.2013

Diğer Nokta operasyonu İşlemleri

EŞİKLEME (THRESHOLDING)

- Gritonlu resimleri siyah-beyaz gösterme metodu (binaryleştirme) resim bölümlenmenin (image segmentation) en popüler metodudur. Bu, özellikle endüstriyel resim işlemede kullanılır. Binaryleştirme aynı zamanda bir nokta operasyonudur.



20	20	20	20	20	20	20	40
160	60	60	60	60	60	60	40
160	60	70	70	70	70	60	40
160	60	70	80	80	70	60	40
160	60	70	80	80	70	60	40
160	60	70	70	70	70	60	40
160	60	60	60	60	60	60	40
160	120	120	120	120	120	120	120

Orijinal resim

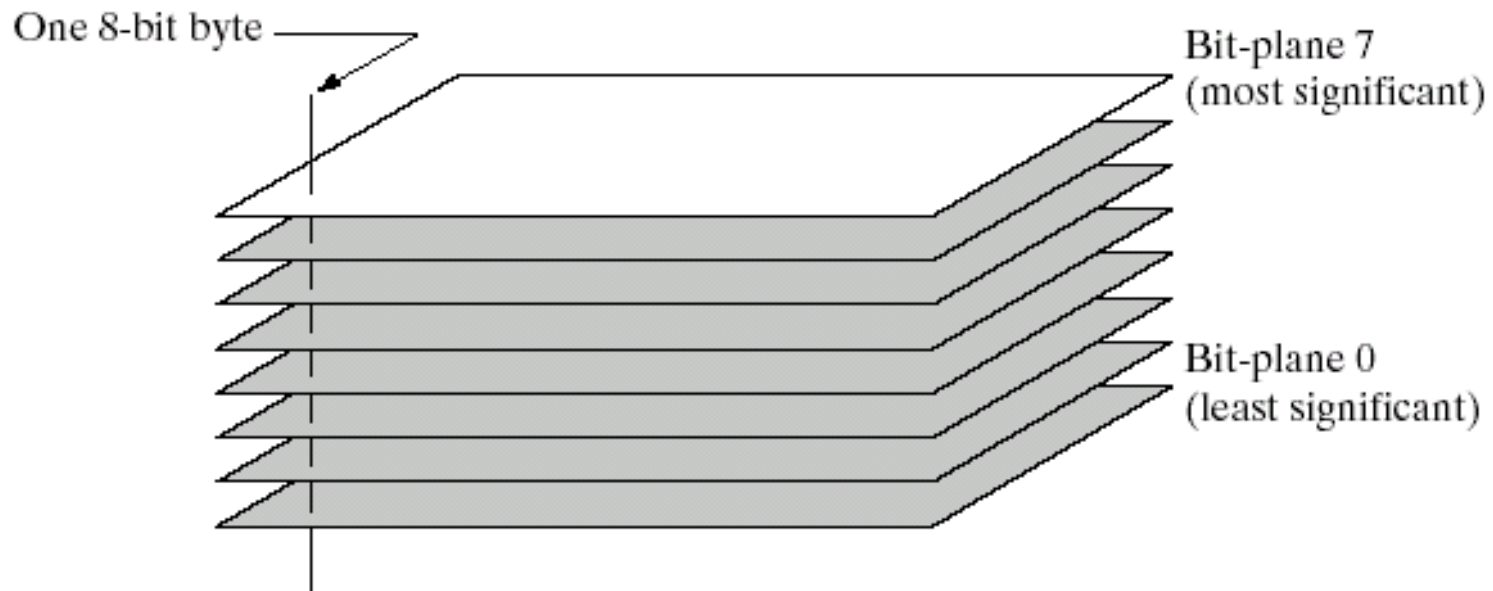
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	1	0	0
1	0	1	1	1	1	0	0
1	0	1	1	1	1	0	0
1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

65 eşiği uygulanmış resim

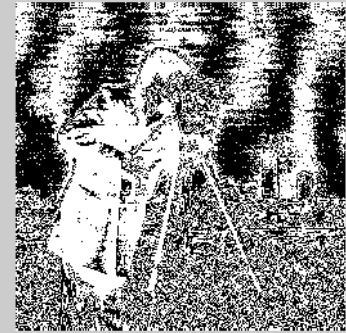
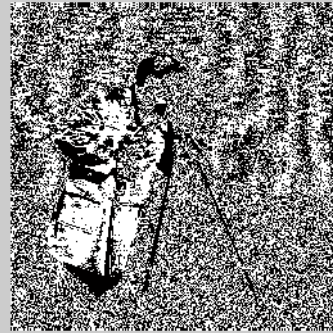
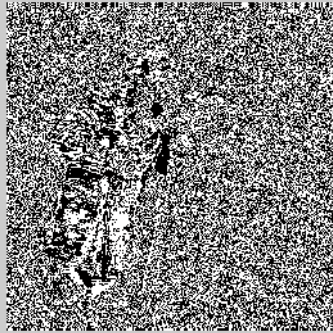
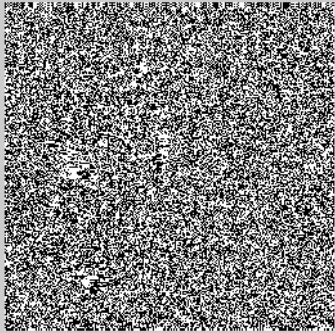
Diğer Nokta operasyonu İşlemleri

Bit plane

- Gri skala görüntüler, bir dizi binary görüntü şeklinde ifade edilebilirler. Her bir pikselinin koyuluk seviyesinin 8 bitle ifade edildiği bir görüntüyü gözönüne alırsak; 0. bit planı görüntünün en önemsiz bit planıdır. 7. bit planı ise görüntü üzerinde etken olan en önemli bit planıdır.

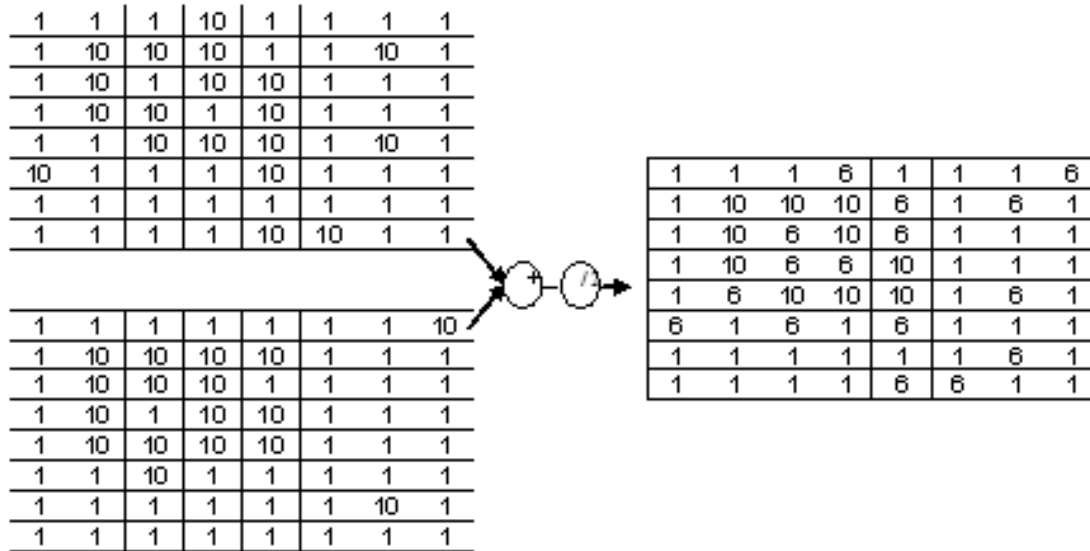


Bit planes



İki Resim Arasındaki Aritmetiksel İşlemler (Toplama)

- İki veya daha fazla resimlerin piksel piksel birleştirilmesidir. Şekilde iki bölgeden müteşekkil iki resmi (sol) göstermektedir. Bir kaç bozulmuş veya gürültülü piksellerin dışında bölgelerin gritonları hemen hemen aynıdır (gritonlar 1 ve 10). Karşılıklı piksellerin ortalamalarının alınması bozulmanın sertliğini yok eder. Buradaki işlem birbirine uygun fakat gürültülü resimlerin gürültüsünü azaltmaktır. Resim sayısı arttıkça toplamının gürültüyü azaltma tesiri artar. Şekil tekniğin gerçek resimlere uygulanmasını göstermektedir.



Sağdaki her iki resmin ortalaması alınarak gürültüsü daha az olan bir sonuç resim elde edilmiştir. (+)'ın manası iki gritonun toplamıdır. (/2)'nin manası toplamın 2'ye bölünmesidir.



a



b



c

Şekil 2.26 a) Kaynak resim b) gaussian gürültü eklenmiş kaynak resim c) sonuç resim $((a+b)/2)$