

Bilgisayar Görmesi

Ders 3: İSTATİSTİKSEL İŞLEMLER

Dr. Öğr. Üyesi Serap ÇAKAR



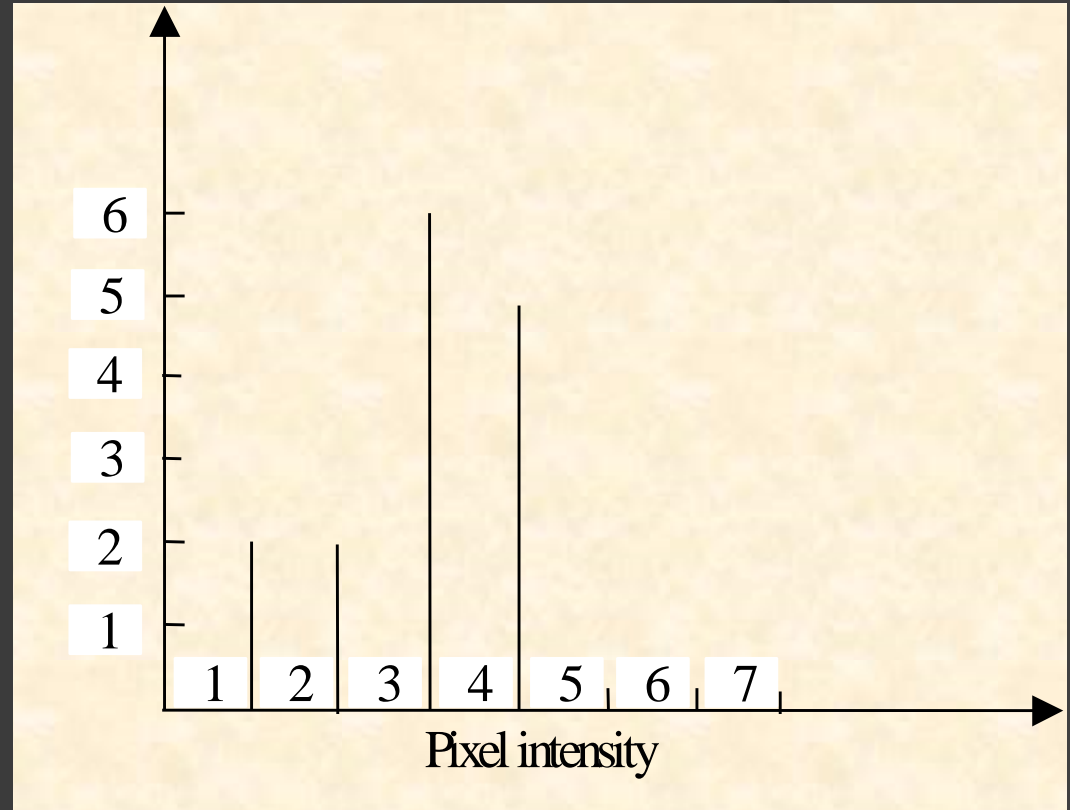
Histogram

Görüntü Histogramı bir görüntünün yoğunluk profiline bakmak için kullanılan değerli bir araçtır. Histogram, **zıtlık** ve görüntünün **yoğunluk** dağılımı ile ilgili bilgi sağlar. Görüntü histogramı piksel yoğunluğunun basit bir çubuk grafiğidir. Piksel yoğunluğu x eksenini boyunca ve her bir yoğunluktan kaç tane olduğu da y eksenini boyunca gösterilir.

Koyu görüntüler **sol** tarafa doğru yığılmış bir histograma sahiptir. **Parlak** görüntüler ise **sağ** tarafa doğru yığılmış bir histograma sahiptir. İdeal bir görüntüde histogram boyunca düzenli bir dağılım vardır.

4	4	3	3
4	4	3	3
4	1	2	3
0	1	2	3

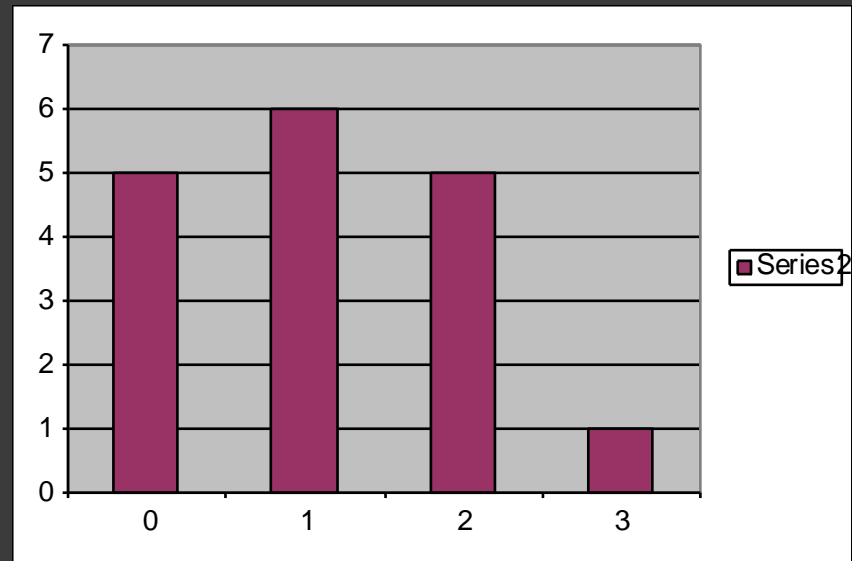
Görüntü



Histogram

- Bir görüntü histogramı her bir parlaklıktaki piksel sayısını sayar.

1 1 1 2 2 3
0 0 1 1 2 2
0 0 0 1 2 2



Yoğunluk Dönüşümü

Yoğunluk dönüşümü daha önce tanımlanan fonksiyonlara dayanarak eski piksel değerinin yeni piksel değerine dönüştüren bir işlemdir. Bu dönüşümler look-up table kullanılarak kolayca uygulanabilir. Bu look-up table'ların giriş çıkış ilişkisi grafiksel olarak gösterilebilir. Orijinal piksel değerleri yatay eksen boyunca gösterilir ve çıkış pikseli eski piksel değeri ile aynıdır.

Look-up table teknikleri

Nokta işleme algoritmaları look-up table (LUT) tekniği kullanılarak en etkili şekilde işletilebilir. LUT'ler dizi indeksi olarak piksel değerlerini kullanan basit diziler kullanırlar. Yeni değer bu indeks tarafından işaretlenen dizi elemanıdır. Yeni görüntü her piksel için işlemin tekrarlanması ile oluşur. LUT'ler kullanılarak gereksiz tekrarlanan hesaplamalardan kaçınılmış olur. Örneğin, 8-bitlik görüntülerle çalışıldığında görüntünün büyüklüğüne bakmaksızın sadece 256 değer hesaplanması gerekir.

7	7	5	4	2
7	6	4	3	0
7	6	4	3	1
6	6	4	2	0
5	5	3	1	0

0	0
0	1
1	2
1	3
2	4
3	5
4	6
5	7

5	5	3	2	1

3-bitlik look-up-table işleyişi

Gamma doğrultma fonksiyonu

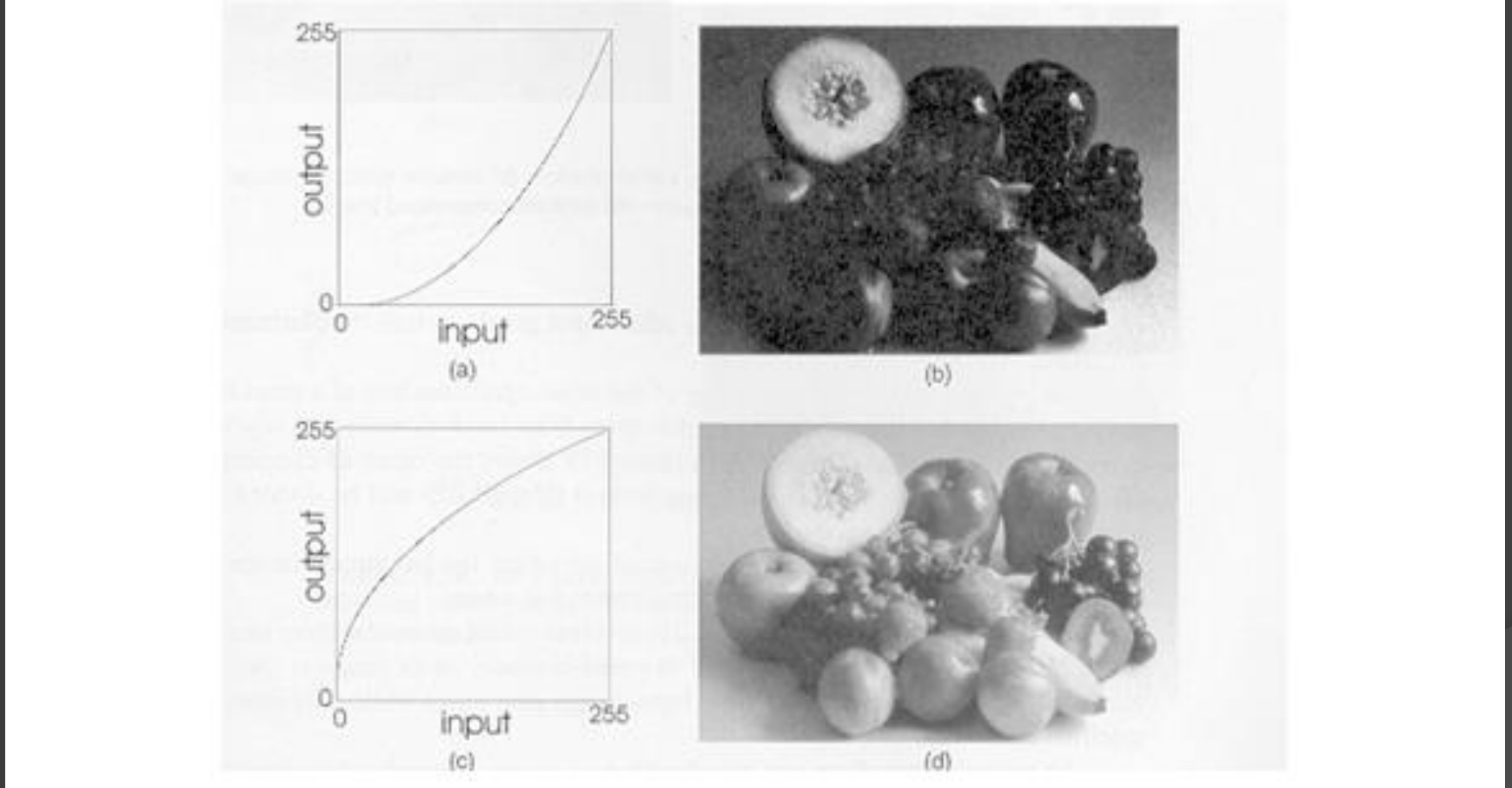
Dönüşüm makrosu gamma doğrultma fonksiyonunu kullanır. Bir görüntünün parlaklığı gamma doğrultma dönüşümü ile ayarlanabilir. Bu, CRT'de parlaklık kontrolü yapan lineer olmayan bir dönüşümdür. Gamma doğrultma fonksiyonları görüntü sensörleri, displayler ve filmlerde lineer olmayan cevapları dengelemek için görüntü işlemede kullanılır.

$$\text{çıkış} = \text{giriş}^{1/g}$$

Eğer $g=1.0$ ise sonuç null'dur. Eğer $0 < g < 1.0$ ise g görüntüyü loş yapan üstel bir eğri oluşturur. Eğer $g > 1.0$ ise sonuç görüntüyü parlaklaştıran logaritmik bir eğridir. RGB monitörleri 1.4 ve 2.8 arasındaki gamma değerlerine sahiptir.

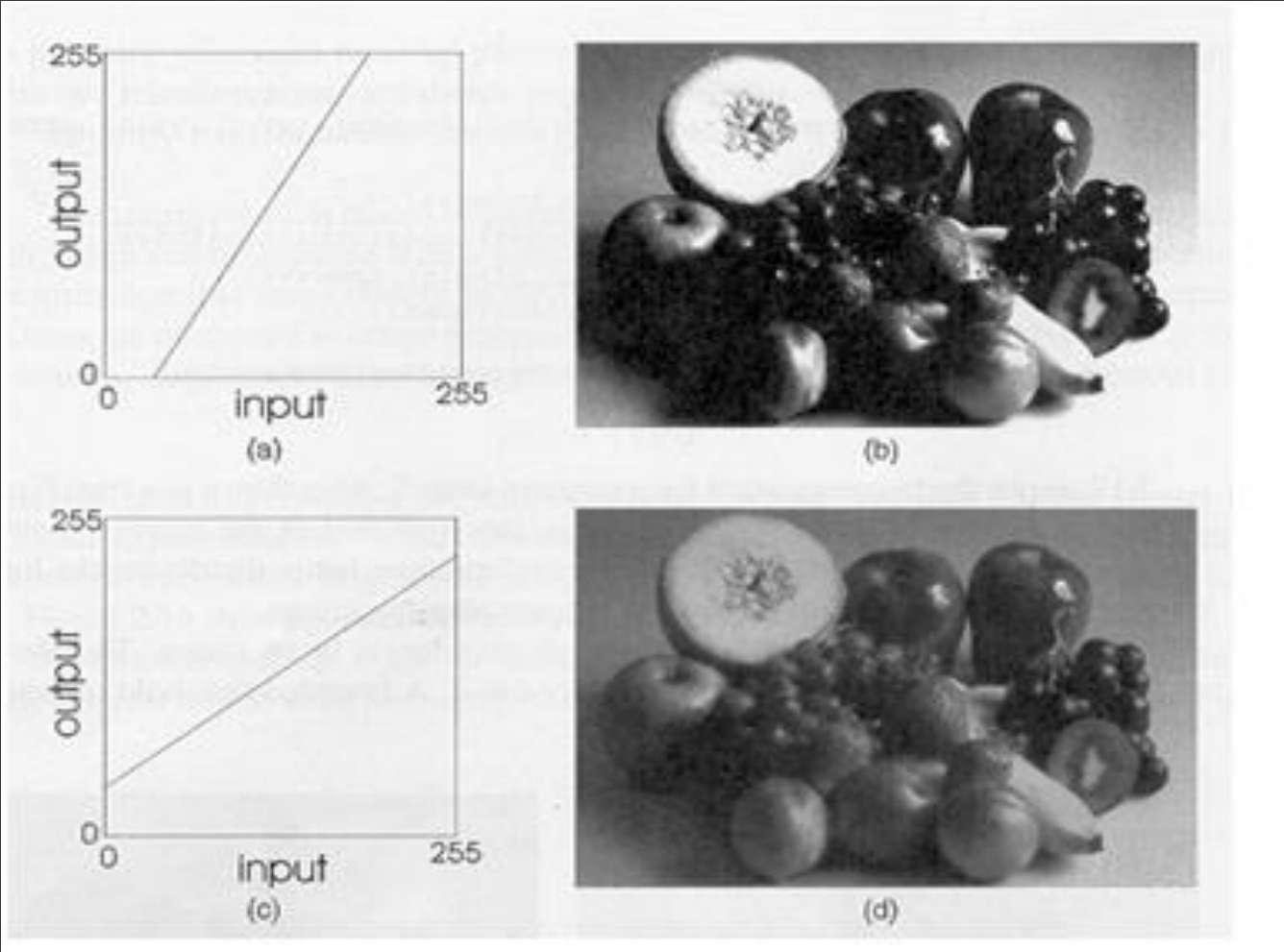
Zıtlık Germe (Contrast stretching)

Zıtlık germe bir yoğunluk dönüşümüdür. Yoğunluk dönüşümü yapıldıkça zıtlıklar daha iyi bir dağılım için gerilir, sıkıştırılır ve değiştirilir.



Şekil (a) $\gamma = 0.45$ için gamma doğrultma fonksiyonu; (b) doğrultulmuş görüntü; (c) $\gamma = 2.2$ için gamma doğrultma fonksiyonu; (d) doğrultulmuş görüntü.

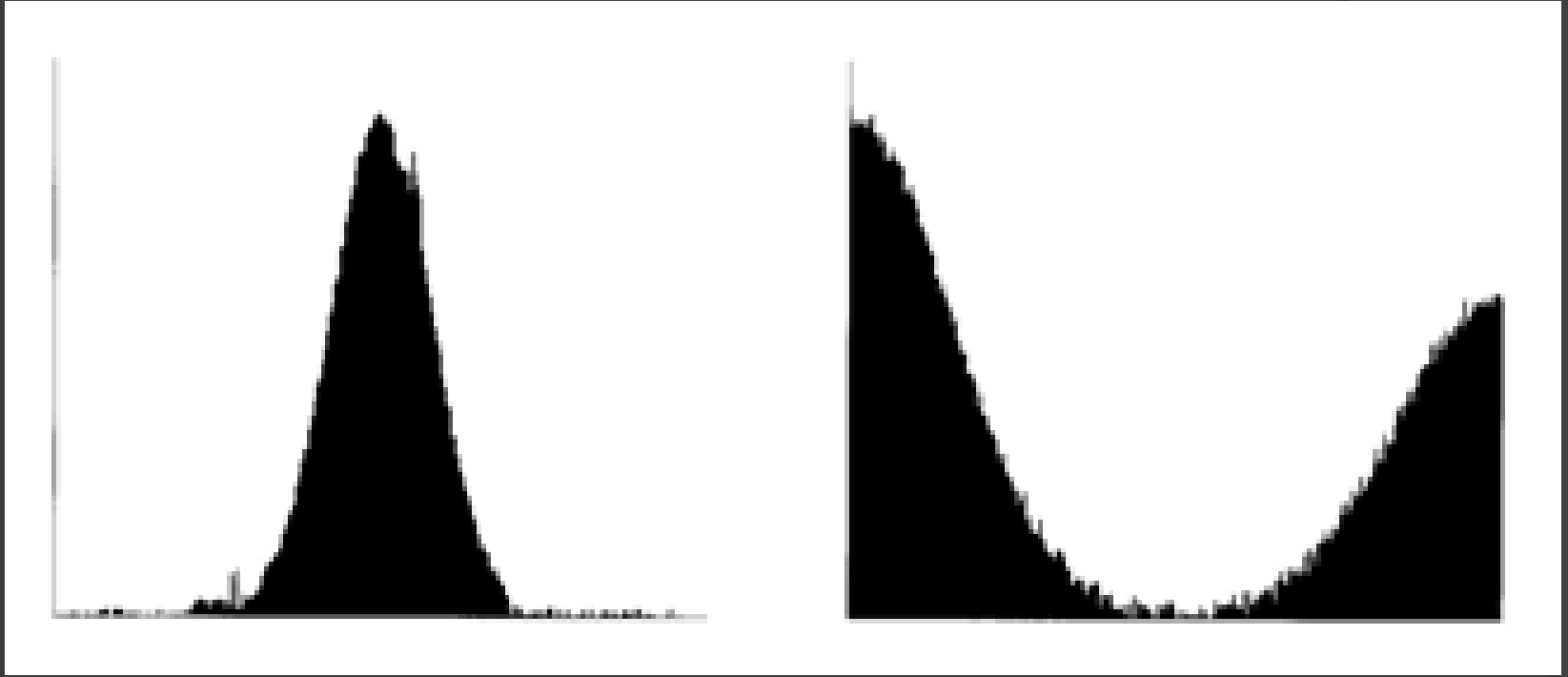
Zıtlık Germe (Contrast stretching)



Şekil (a) Zıtlık germe dönüşümü; (b) gerilmiş görüntü; (c) zıtlık sıkıştırma dönüşümü; (d) sıkıştırılmış görüntü.

Bir görüntünün kontrastı parlak ve koyu piksellerin dağılımıdır. Düşük zıtlıklı gri seviyeli görüntüler daha koyu, daha parlak veya daha gridir. Düşük zıtlıklı bir görüntünün histogramında pikseller sağa, sola veya ortaya dağılmıştır. Ayrıca histogram çubukları birlikte sıkıca kümelenmiştir ve bütün olası piksel değerlerinin küçük bir kısmını kullanır.

Yüksek zıtlıklı görüntüler hem koyu, hem de açık bölgelere sahiptir. Yüksek zıtlıklı görüntü problemi görüntünün geniş koyu bölgeye ve geniş açık bölgelere sahip olmasıdır. Güneşli bir günde bir pencerenin önünde duran birinin resmi yüksek zıtlıklıdır. Yüksek zıtlıklı görüntülerin histogramları iki büyük tepeye sahiptir. Bir tepe düşük bölgenin ortasında, diğeri yüksek bölgenin ortasında yer alır.



Şekil. Düşük ve yüksek zıtlıklı histogramlar.

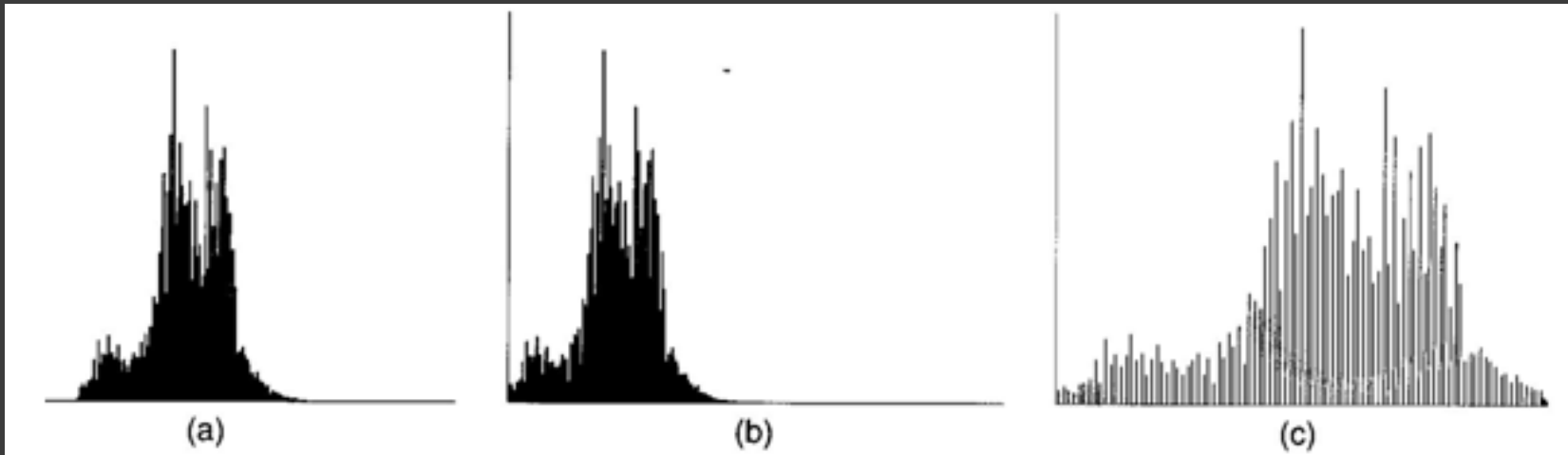
Zıtlık germe görüntünün bütün dinamiklerini doldurmak için histogramı germek için görüntüye uygulanır. Düşük zıtlıklı görüntüleri düzeltmek için faydalı bir tekniktir. Gauss veya Gauss'a yakın bir dağılıma sahip olan görüntülerde iyi çalışır.

Zıtlık germenin iki popüler tipi vardır. Bunlar temel zıtlık germe ve end-in-search yöntemidir. Temel zıtlık germe en iyi bütün piksellerin histogramının bir parçasında, örneğin, ortasında yoğunlaştığı görüntülerde çalışır. Zıtlık germe görüntü histogramını bütün piksel seviyelerine dağıtır.

Temel zıtlık germe

En yüksek ve en düşük değerli pikseller dönüşümde kullanılır.

$$new\ pixel = \frac{old\ pixel - low}{high - low} \times 255$$

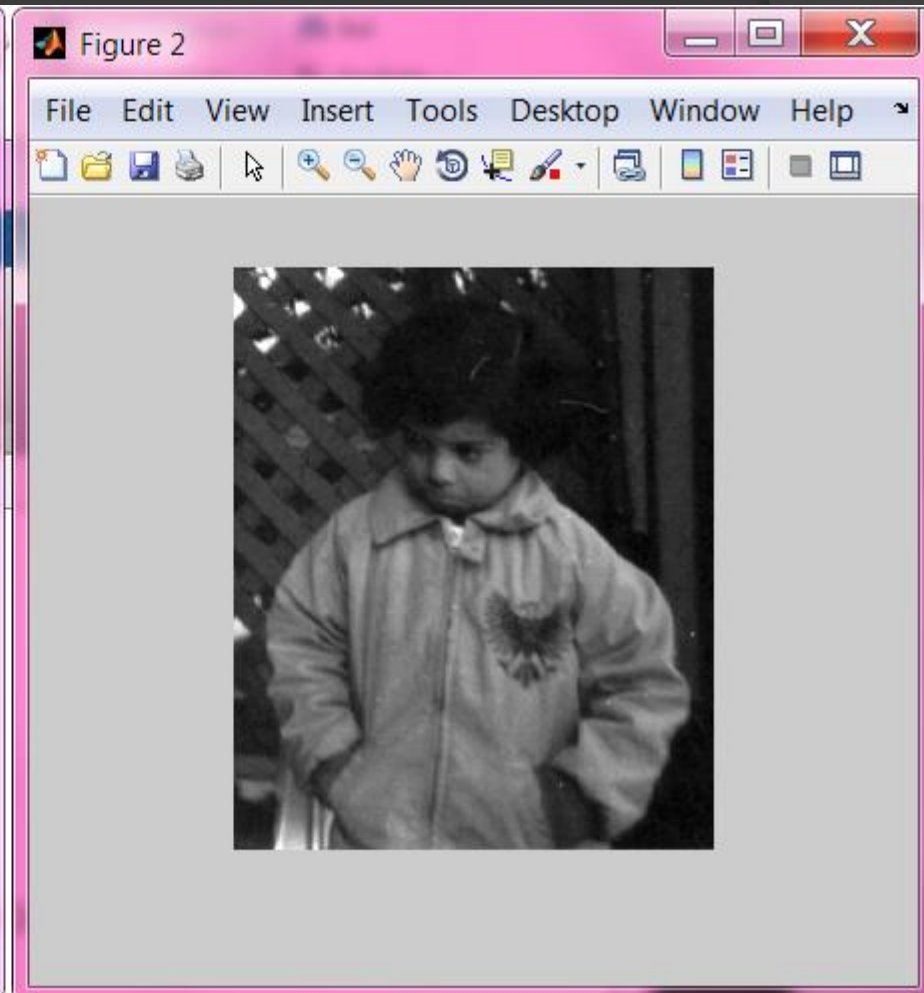
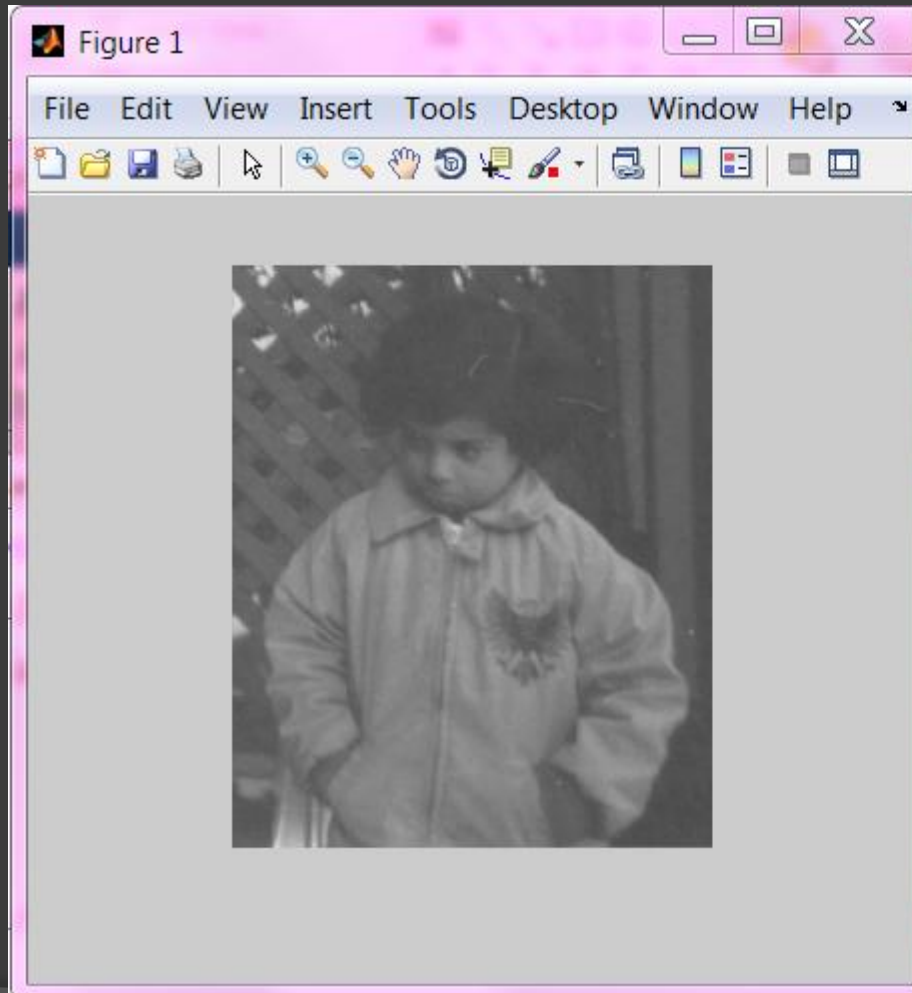


Şekil. (a) Orijinal histogram; (b) histogram-düşük; (c) $(old - low) * 255 / (high - low)$.

Temel zıtlık germe

```
1 - a=imread('c:\pout.tif');
2 - dusuk=min(min(a));
3 - yuksek=max(max(a));
4 - [m, n]=size(a);
5 - for i=1:m
6 -     for j=1:n
7 -         b(i,j)=(a(i,j)-dusuk)*(255/(yuksek-dusuk));
8 -     end
9 - end
10 - imshow(a);
11 - figure;imshow(b);
```

Temel zıtlık germe



Posterizing bir görüntüdeki grilik seviyesinin sayısını azaltmaya denir. Grilik seviyesinin 2'ye düşürülmesi ile **eşikleme** oluşur.

Bit-kesme (bit-clipping) bir pikselin belirli sayıda en anlamlı bitlerinin 0'a ayarlanmasıdır. Bu dönüşüm görüntüdeki aynı yoğunluklu bölgeleri siyah ve beyaz olarak ayırır.

Son birkaç dönüşüm **radyometrik analizi** gibi görüntü işlemenin özel alanlarında kullanılır. Aşağıdaki iki dönüşüm **dijital ressamlıkta** kullanılır.

İlki **solarizing** işlemi olarak adlandırılır. Bu, görüntüyü aşağıdaki formüle göre dönüştürür.

$$output(x) = \begin{cases} x & \text{for } x \leq \text{threshold} \\ 255 - x & \text{for } x > \text{threshold} \end{cases}$$



Sonuncusu **parabola dönüşümüdür** ve aşağıdaki formüller kullanılır.

$$output(x) = 255 - 255(x/128)$$

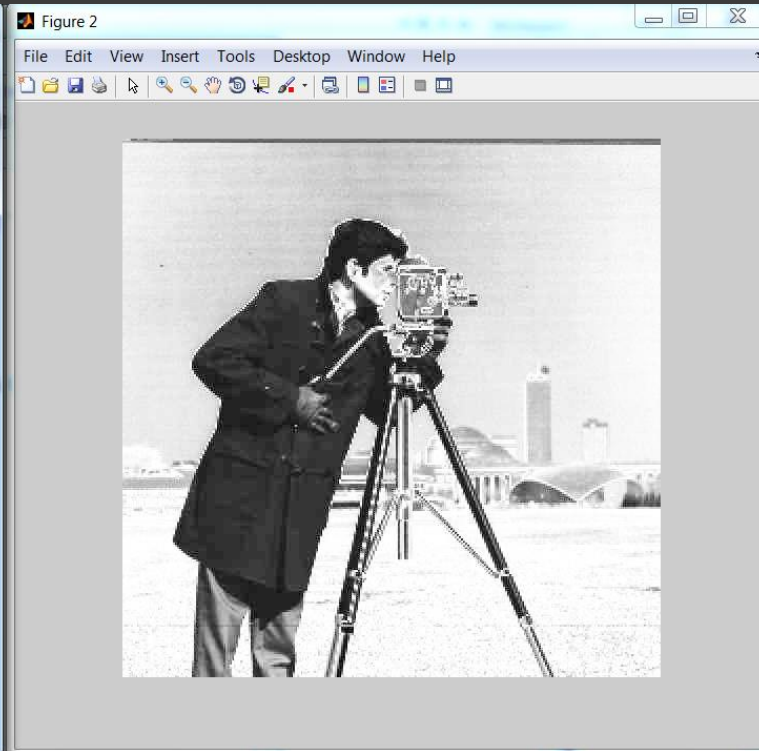
ve

$$output(x) = 255(x/128)$$

Parabola Dönüşümü

```
1 - clear all;
2 - A=imread('cameraman512.jpg');
3 - [m,n]=size(A);
4 - A=double(A);
5 - for i=1:m
6 -     for j=1:n
7 -         B(i,j)=255-(255*(A(i,j)/128-1)^2);
8 -     end
9 - end
10 - A=uint8(A);
11 - B=uint8(B);
12 - imshow(A);
13 - figure;imshow(B);
```

Parabola Dönüşümü



End-in-search

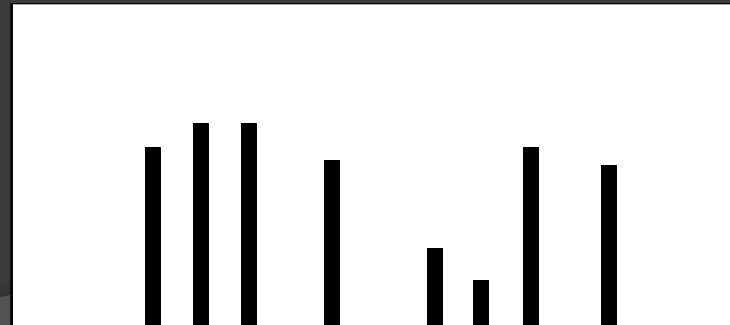
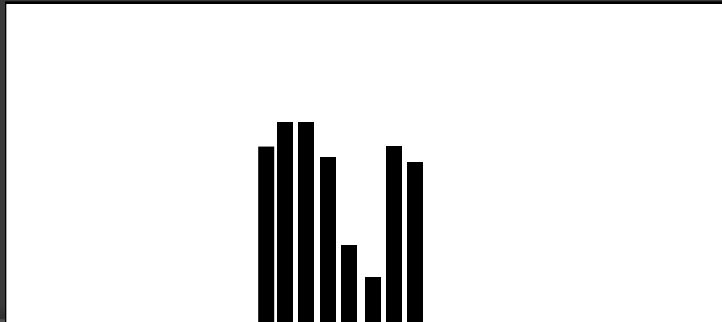
Zıtlık germedeki ikinci yöntem end-in-search yöntemi olarak adlandırılır. En iyi bütün yoğunluklarda piksel değerlerine sahip olan görüntülerde çalışır, fakat histogramın bir tarafında piksel konsantrasyonu olan görüntülerde çalışmaz. Görüntü işlemciler bu tekniği daha çok kullanır. Piksellerin belli bir yüzdesini tam beyaz veya tam siyah ile doygun hale getirmek gereklidir. Algoritma en düşük eşik değerini bulmak için histograma bakar. En düşük eşik değeri histogramda en düşük yüzdelik nerede yakalanırsa o değerdir. Histograma bakarak en yüksek eşik değeri bulunur. Daha sonra LUT başlatılır.

$$output(x) = \begin{cases} 0 & \text{for } x \leq low \\ 255 \times (x - low) / (high - low) & \text{for } low \leq x \leq high \\ 255 & \text{for } x > high \end{cases}$$

End-in search yöntemi yüksek ve düşük değerlerin kodlanması ile otomatikleştirilebilir. Bu değerler histogram analizinin farklı metodları ile de belirlenebilir.

Histogramdaki Bilgi

- ⦿ Zıtlık (parlaklıktaki değişim)
 - Çubuklar bütün seviyelere dağılmış mı?
- ⦿ Önplan ve arka plan rengi
 - Arasında vadi olan iki tepe var mı?



Histogram Uygulamaları

- Eşikleme; gri seviyeli görüntüden ikili görüntüye dönüşümdür.
 - Eşik değerinin üstündeki pikseller beyaza, altındaki pikseller siyaha dönüştürülür.
 - Genellikle ön plan ile arka plan birbirinden ayrılır.
 - Histogram, eşikleme için gerekli değildir fakat daha iyi bir eşikleme için kullanışlıdır.
 - Örneğin: Histogramda iki tepe bulunur ve ikisinin ortası eşik değeri olarak seçilir.
 - Daha karmaşık yöntemler de vardır. Bunlar histogram modeline veya görüntüdeki şekillerin modeline bağlı olarak değişebilir.

Histogramdan Eşikleme

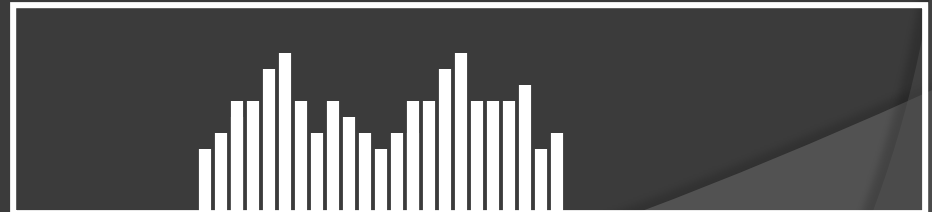
Sadece iki yoğunluk



Bimodal (iki tepe)



Yakın tepeli Bimodal, daha çok gürültü



Eşikleme Fikirleri

- ⊙ En yüksek sayıya sahip olan piksel değerini bul
 - Eğer iki tane ise eşik değerini ikisinin ortası olarak belirle
- ⊙ Bir vadi bul
 - En düşük sayıya sahip olan piksel değeri veya iki tane varsa ortasını bul.
- ⊙ Histogram için bir model kullan. Örneğin iki tane çakışan normal eğrisi
 - Arka plan için varyansı maksimize et (otsu yöntemi)
- ⊙ İteratif yöntem (2-means sınıflandırma)
 - Eşik değerini tahmin et. Eğer yeterince iyi değilse daha iyi bir tahmin yap

İteratif Yöntem

- İsteğe bağlı bir eşik değeri seç.
Örneğin sınıra yakın olan ve olmayan piksellerin ortalamasının orta noktası.
- Eşik değerini değiştir.
 - Arka plan ve ön plan ortalamalarını hesapla
 - Yeni eşik değeri belirle
 $(\text{arka plan ort.} + \text{ön plan ort})/2$

Bu neden çalışır?

- Eğer eşik değeri çok düşük ayarlanmışsa düşük taraftaki bütün pikseller arka plan olarak ayarlanmış olur ve yüksek taraf ve bazı arka plan pikselleri nesneyi oluşturur. Eşik değeri yukarıya çekilir ve bazı arka plan pikselleri diğer tarafa sıçrar.
- Eğer eşik değeri doğru ise düşük taraftaki bütün pikseller arka plan ve yüksek taraftaki bütün pikseller ön plan olur. Bir iterasyon sonrasında hiçbir piksel diğer tarafa geçmez ve eşik değeri sabit olur.

Eşiklemeyi Geliştirme

- ⦿ Eşiklemeden önce kenar piksellerini histogramdan çıkar.
 - Kenar pikselleri ortalamaya yakındır. İki tepe arasına düşer.
- ⦿ Eşik değerinde komşuluk bilgisini sakla (dinamik eşikleme)
 - Gölgedeki pikseller aydınlıktaki pikseller gibi ortalamadan farklıdır.

Histogram bulma

```
1 - |clc;clear all;close all;
2 - |h=imread('C:\lena512.jpg');
3 - | [M,N]=size(h);
4 - | t=1:256;
5 - | n=0:255;
6 - | count=0;
7 - | for z=1:256
8 - |     for i=1:M
9 - |         for j=1:N
10 - |             if h(i,j)==z-1
11 - |                 count=count+1;
12 - |             end
13 - |         end
14 - |     end
15 - |     t(z)=count;
16 - |     count=0;
17 - | end
18 - | disp(t');
19 - | stem(n,t);
20 - | grid on;
21 - | ylabel('yogunluk seviyelerinden kac tane oldugu');
22 - | xlabel('yogunluk seviyeleri');
23 - | title('GORUNTUNUN HISTOGRAMI');
```

