

# Dijital Görüntü İşlemede Temel Kavramlar

# Görüntü işleme

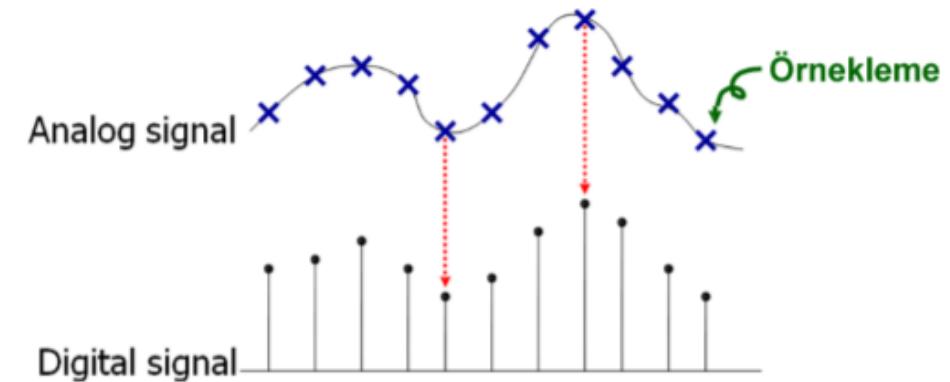
- Daha iyi bir görüntü elde etmek,
- Görüntüden faydalı bilgi çıkarmak amaçları ile görüntü üzerinde yapılan birçok farklı işlemi içerir.
- Girdi olarak bir görüntü, çıktı olarak yeni bir görüntü veya görüntü ile ilgili özellikler elde edilir.
- Savaş teknolojileri, güvenlik, tıp, fizik, sanayi, kişisel bilgisayara kadar birçok alanda kullanılmaktadır.
-

# Sinyal

- Sinyal, fiziksel değişkenlerin durumu hakkında bilgi taşıyan ve matematiksel olarak bir  $f$  fonksiyonu ile gösterilen kavramdır. 1, 2, ..., n boyutlu olabilir.
  - Dalgalar şeklinde yayılan ses
  - Görüntü bilgisinin mekânsal zamansal değişimi
  - Bir haberleşme uydusunun dünyaya gönderdiği elektromanyetik dalgalar,
  - Bir fotoğraf karesindeki parlaklığın  $x$  ve  $y$  değişkenleri biçiminde ifadesi örnektir.

| Sinyal    | Açıklama  | Şekil |
|-----------|---|-------|
| 1 boyutlu | <p>Örnek: Dalga formu<br/>Matematiksel olarak<br/><math>Genlik = f(t)</math><br/>ile ifade edilir. Burada, <math>t</math> bağımsız<br/>değişkendir, diğer bir ifade ile tek boyutlu<br/>bir sinyal olduğu için tek bir değişken vardır.</p>   |       |
| 2 boyutlu | <p>Örnek: Görüntü<br/>Matematiksel olarak<br/><math>Görüntü = f(x, y)</math><br/>ile ifade edilir. Burada, <math>x</math> ve <math>y</math>, bir pikselin<br/>konumunu ifade eden iki bağımsız<br/>değişkendir. Diğer bir ifade ile 2 boyutlu bir<br/>sinyal olduğu için iki değişken vardır.</p>   |       |
| 3 boyutlu | <p>Örnek: Uzay<br/>Matematiksel olarak<br/><math>Güneş sistemindeki = f(x, y, z)</math><br/><b>herhangi bir nokta</b><br/>ile ifade edilir. Güneş sistemindeki herhangi<br/>bir nokta güneş merkezli koordinat<br/>sisteminde <math>x</math>, <math>y</math>, <math>z</math> konumları ile belirlenebilir.</p>  |       |
| 4 boyutlu | <p>Örnek: Animasyon film<br/>Matematiksel olarak<br/><math>Animasyon film = f(x, y, z, t)</math><br/>ile ifade edilir. 3 boyuttan farklı olarak<br/>eklenen yeni bir boyut <math>t</math>, zamandır. <math>t</math> aynı<br/>zamanda değişimin ölçüldüğü zamansal<br/>boyut olarak da adlandırılır. Diğer bir ifade<br/>ile animasyonlu film, 3 boyutlu karakterlerin<br/>4. boyut olan zaman içinde hareketini<br/>gösterir.</p> |       |

# Analog ve Dijital Sinyal



Şekil. Analog sinyalinin, dijital sinyale dönüşümünde örneklerin alınması.

- **Analog sinyal:** Taşındıkları bilgiyi mekan ve zamana bağlı sürekli bir fonksiyon olarak ifade ederler.
  - Hava basıncı, herhangi bir t anında sürekli değer alabilen bir değişkendir.
  - Analog sinyal sonsuz sayıda değere sahip olabilir.
  - Genellikle sinüs dalgasıyla temsil edilir.
  - Mühendislikte karşılaşılan birçok sinyal analogdur.
- **Dijital sinyal:** Analog sinyalin bir alt kümesidir, sürekli formdaki analog sinyallerin belirli zaman aralıklarında örneklenmesinden oluşan sürekli olmayan ayrık zamanlı bir sinyaldir.
  - Genliği( bir dalga tepesi ile dalga çukuru arasındaki mesafenin yarısı) 0 veya 1 olan kare formundaki dalgalar şeklinde gösterilir.

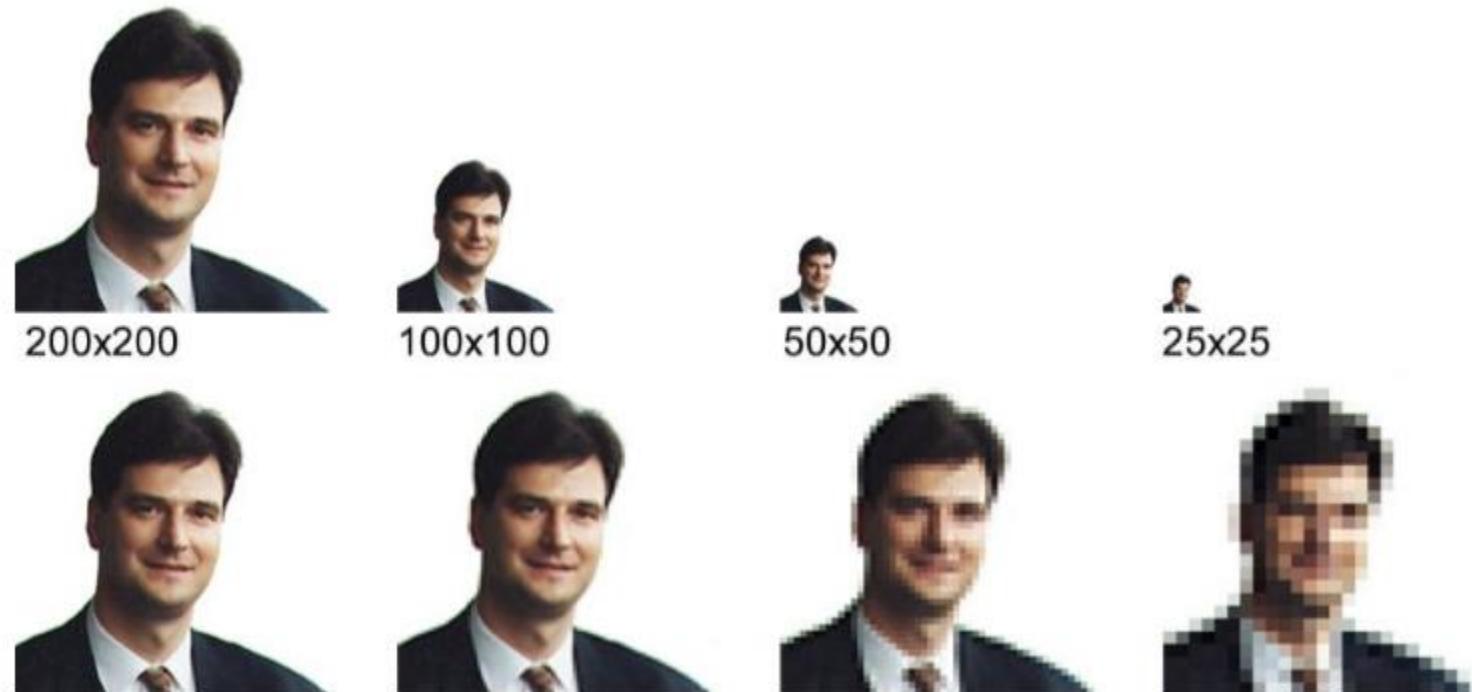
# Dijital Görüntü



| Karşılaştırma Ögesi | Analog Sinyal                                 | Dijital Sinyal  |
|---------------------|---|---|
| Analiz              | Zor   | Mümkün  |
| Gösterim            | Sürekli                                       | Ayrık   |
| Doğruluk            | Daha fazla                                    | Daha az   |
| Depolama            | Sonsuz hafıza ihtiyacı                        | Kolaylıkla depolanabilir.                               |
| Gürültü etkisi      | Var   | Yok   |
| Kayıt teknigi       | Orijinal sinyal korunur.                      | Sinyal örnekleri alınır ve korunur.                     |
| Örnekler            | İnsan sesi, Termometre, Analog telefonlar vb. | Bilgisayarlar, Dijital telefonlar, Dijital kalemler vb. |

# Piksel

- Dijital bir görüntünün 2-boyutlu dizi şeklindeki her bir elemanına bir piksel denir. Aşağıdaki şekilde aynı resmin farklı piksellerle gösterilmiştir.



Şekil. Aynı resmin farklı piksel boyutlarında gösterimi.

# Görüntü İşlemenin Kullanıldığı Uygulamalar

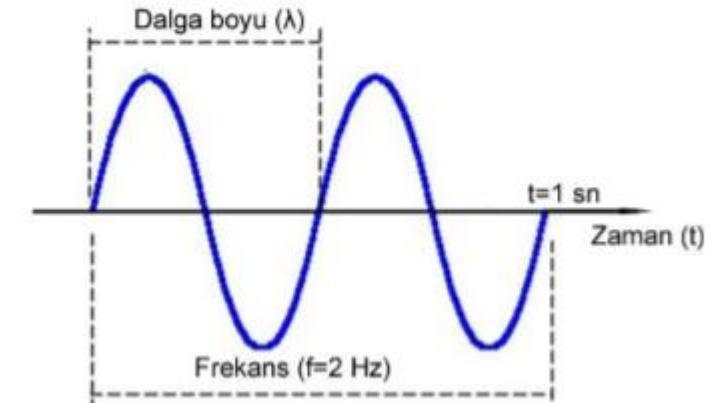
- Resim video çekme uygulamaları: Doğru diyafram ayarları ile uygun renk ve ışık kalitesini ayarlamak için kullanılır.
- Ekran görüntülemeleri ve Yazdırma işlemleri: Bu uygulamalarda görüntü boyutu, renk, tonlama ayarları için kullanılır.
- Resim saklama ve iletimini kolaylaştırmak: Görüntüyü verimli bir şekilde depolama ve aktarmak için kullanılır.
- Güvenlik uygulamaları: Güvenlik taramaları için renkleri zenginleştirme, netliği artırma vs uygulamaları için kullanılır.
- Bilgi çıkarma: Barkod okuma, plaka okuma, yazı karakterini metne dönüştürme gibi uygulamalar için kullanılır.
- Mosaik Resim oluşturma: Yüksek çözünürlüklü (GByte resimler) resimler, panoramik (geniş açılı) resimler yada 360 derece (heryerde resimler oluşturmak için bir çok görüntünün çakıştırılarak büyük resimler haline getirilmesi gereklidir).
- Yüz tanıma: Gerek güvenlik için gerekse insan sayma, otomatik yüz gizleme yada diyafram ayarı gibi hususlar için yüz tanıma ihtiyacı vardır.
- Otonom görüş yeteneği: Robotların ve otonom araçların etrafındaki dünyayı tanıyalabilmesi için görüntü işlemeye ihtiyaç vardır.
- Sanal gerçeklik uygulamaları: 3 boyutlu görüntülerin hazırlanması, gerçek dünya ile sanal dünya görüntülerinin birleştirilmesi gibi bir çok uygulama görüntü işleme ile yapılmaktadır.



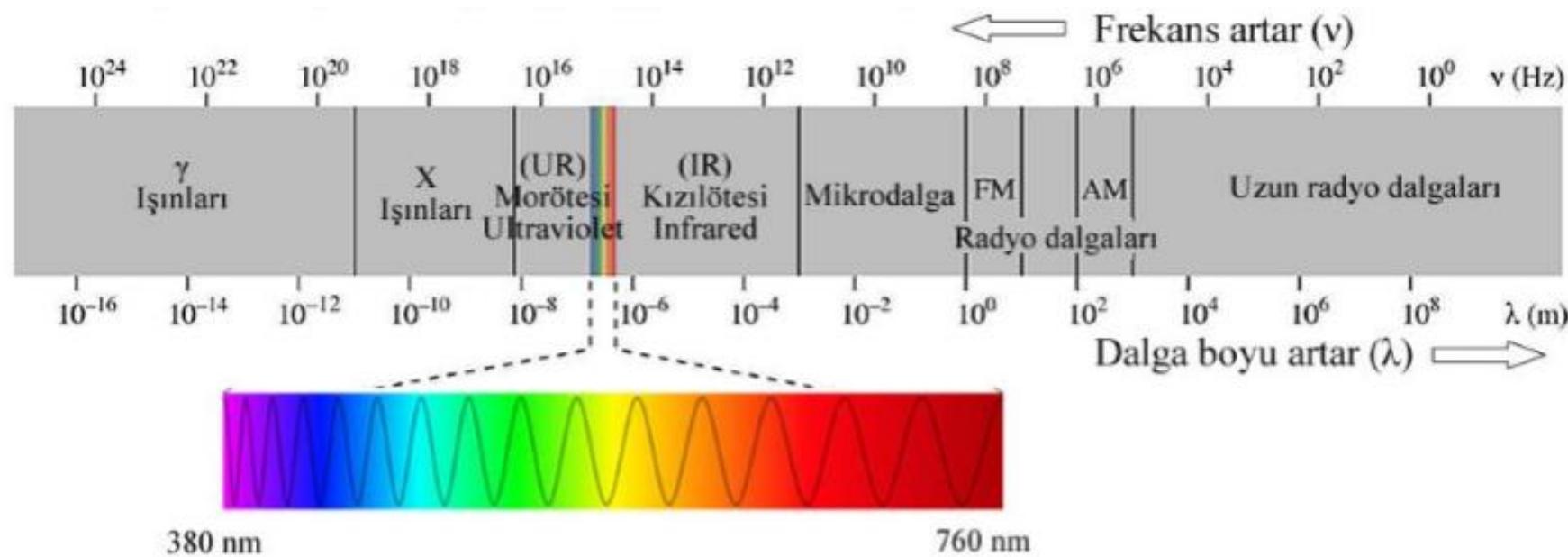
$$C_{(\text{Işık hızı})} = f_{(\text{frekans})} * \lambda_{(\text{dalga boyu})}$$

# Işık

- Işık kaynağı tarafından yayılan küçük dalgacıklardır.
- Bu dalgacıklar radyasyon enerjisi şeklindedir (elektromanyetik radyasyon).
- Görünen ışık elektro manyetik spektrumda çok küçük bir aralıkta bulunur (%2 civarı).
- Bu aralığın dışındaki ışık dalgalarını çiplak gözle göremeyiz
- Görünen ışık bölgesindeki her rengin bir dalga boyu vardır.
- Genel olarak gözlerimizle gördüğümüz ışığın dalga boyu 0,380-0,760 mikron (370 -760 nanometre) aralığındadır.
- ışığın hızını saniyede 300 bin km alırsak aşağıdaki bağlantıdan frekansını da bulabiliriz.
- Bu bağlantıya göre ışık, hem frekans ( $v$ ) hem de dalga boyu ( $\lambda$ ) ile ifade edilebilir.
- Işık hızı boşlukta 299.792.458 m/s dir (300.000 km/s). Dalga boyu ( $\lambda$ ) birimi metre (m) , frekans ( $f$ ) birimi ise Hertz (Hz) dir.

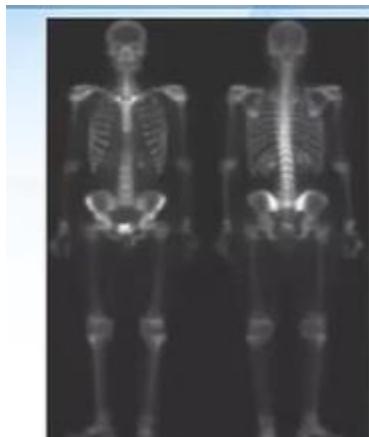


# Elektromanyetik Spektrum



Şekil. Elektromanyetik spektrum.

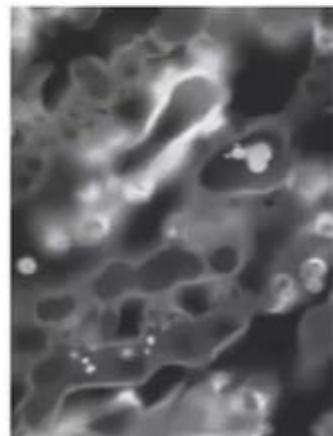
# Uygulama Alanları



(a)



(b)



(c)



(d)

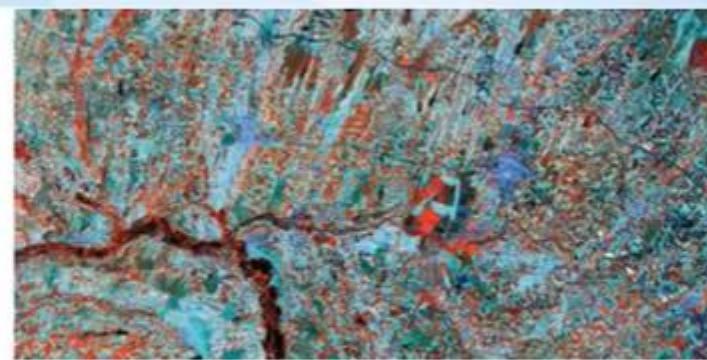
(a) Gama ( $\gamma$ ) ışınları. (b) X-ışınları.

(c) Morötesi ışınlar. (d) Radyo dalgaları

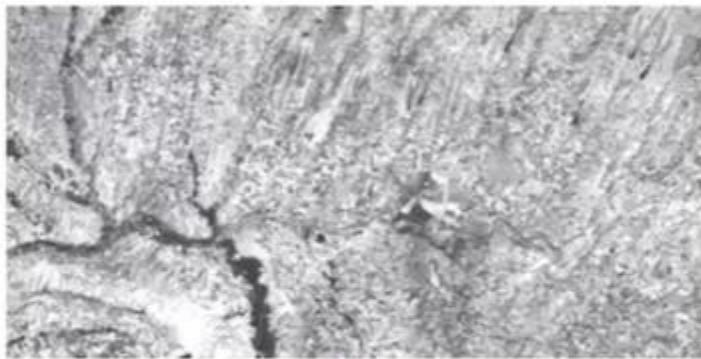
# Uygulama Alanları



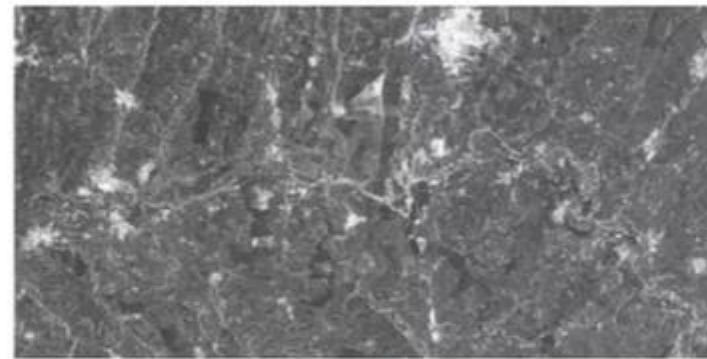
(a)



(b)



(c)



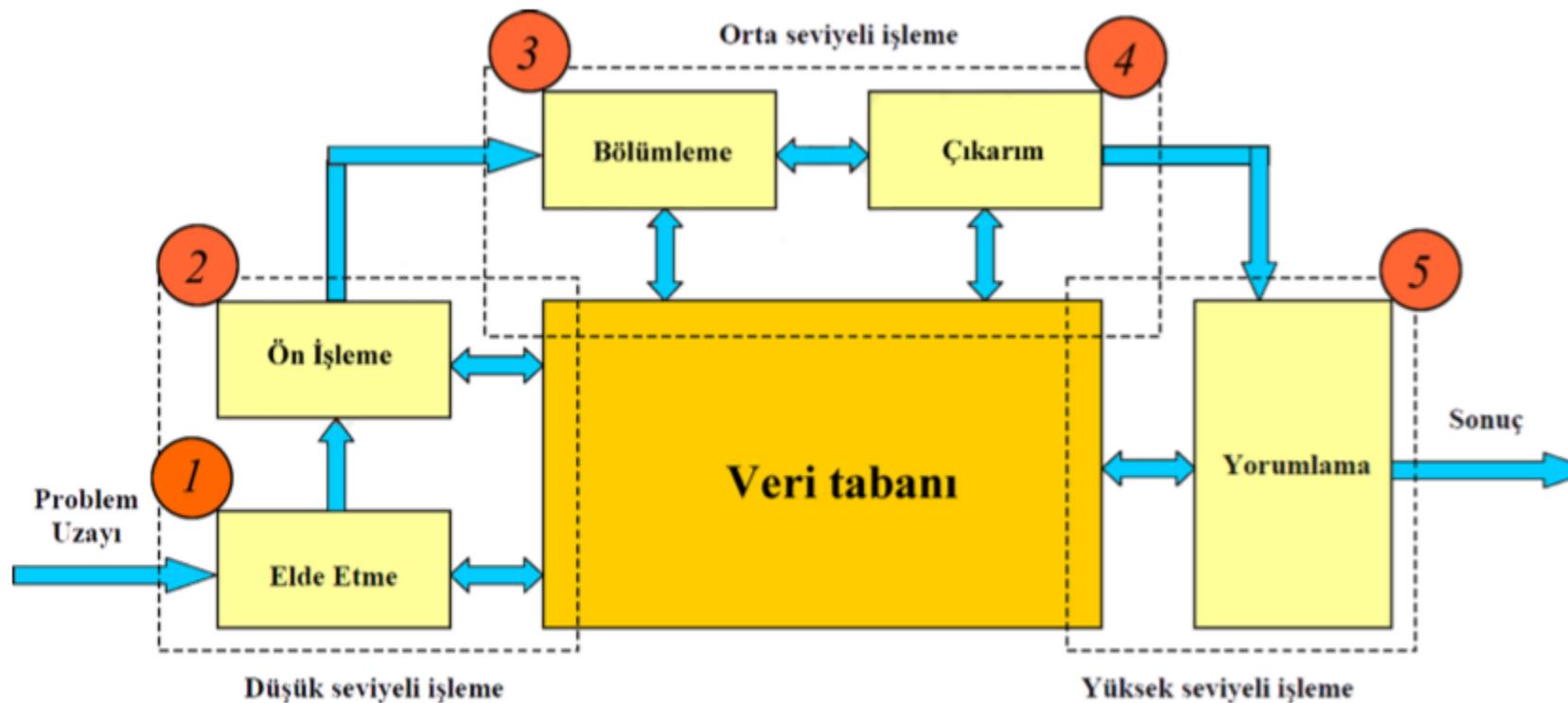
(d)



(a) Görünür bölge (b) Yansıtıcı kızılıötesi (c) Isıl kızılıötesi (d) Mikrodalga

# Sayısal Görüntü İşleme

- Sayısal görüntü işleme, analog bir görüntünün sayısal biçime dönüştürülmesi ve daha sonra çeşitli amaçlar için (iyileştirme, onarma, sınıflandırma, sıkıştırma, anlama ve yorumlama, v.b.) sayısal bilgisayarlarla işlenmesi ile ilgilenir.



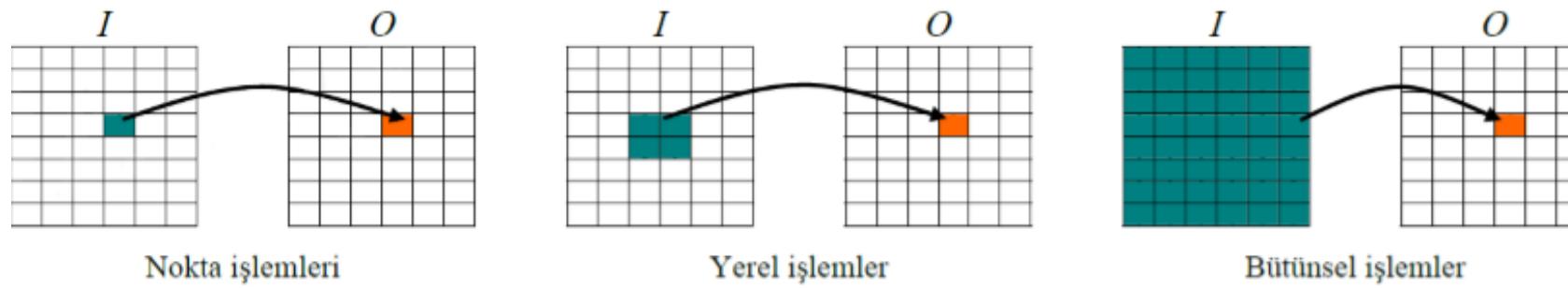
Şekil. Sayısal görüntü işlemenin temel basamakları.

# Sayısal Görüntü İşleme Adımları

- Elde Etme: Görüntü işlemenin ilk adım, sayısal görüntünün sayısal kamera ile elde etme işlemidir.
- Ön İşleme: Sayısal resim elde edildikten sonraki aşama ise ön-isleme'dir. Ön-isleme, elde edilen sayısal görüntüyü kullanmadan önce daha başarılı bir sonuç elde edebilmek amacıyla görüntünün bazı ön işlemlerden geçirilmesidir. Bu işlemler temel olarak;
  - görüntü iyileştirme (image enhancement),
  - görüntü onarma (image restoration) ve
  - görüntü sıkıştırma (image compression)
- Görüntünün elde edilmesi ve ön işlemeye tabi tutulması düşük seviyeli görüntü işleme olarak adlandırılır.
- Bölümleme: Ön-islemeden sonra bölümleme (böülütleme, parçalara ayırma, segmentasyon denir) aşamasına geçilir. Bölümleme, görüntü işlemenin en zor uygulaması olup bir görüntüdeki nesne ve arka planın veya görüntü içerisindeki ilgilenilen değişik özelliklere sahip bölgelerin birbirinden ayrıştırılması işlemidir. Bölümleme; bir görüntüdeki nesnenin sınırları ve alanlarını tespit ederek şekli üzerinde ham bilgiler üretir. Eğer nesnelerin şekilleriyle ilgileniyorsak, bölümleme bize o nesnenin kenarları, köşeleri ve sınırları hakkında bilgiler verir. Diğer taraftan görüntü içerisindeki nesnelerin yüzey kaplaması, alanı, renkleri, iskeleti gibi iç özellikleriyle ilgileniliyorsa bölgelik bölümleme yapılmalıdır. Karakter veya genel olarak örüntü (pattern) tanıma gibi oldukça karmaşık problemlerin çözümü için her iki bölümleme yönteminin (sınırlar ve alanlar) bir arada kullanılması gerekebilir.
- Çıkarım: Görüntüden elde edilen ham bilgilerin, ilgilenilen ayrıntıların ön plana çıkarılmasıdır. Yani aranan özellikli alanların arka plandan ve birbirinden ayrıştırılmasıdır.
- Yorumlama: Yüksek seviyeli görüntü işleme grubuna giren bu aşamada, çeşitli karar verme mekanizmaları (yapay zeka algoritmaları gibi) ile görüntüdeki arka plandan çıkarılmış nesnelerin veya bölgelerin etiketlendirilmesi, sınıflandırılması yapılır.

# Piksel üzerinde yapılan işlemler

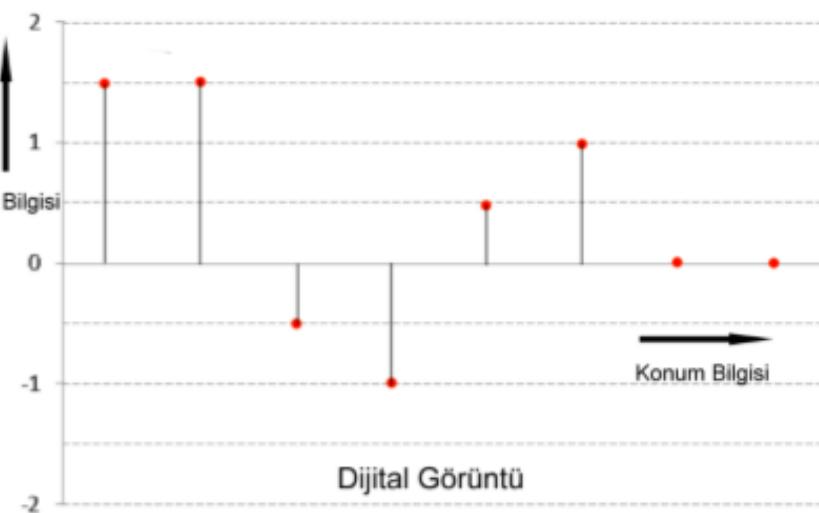
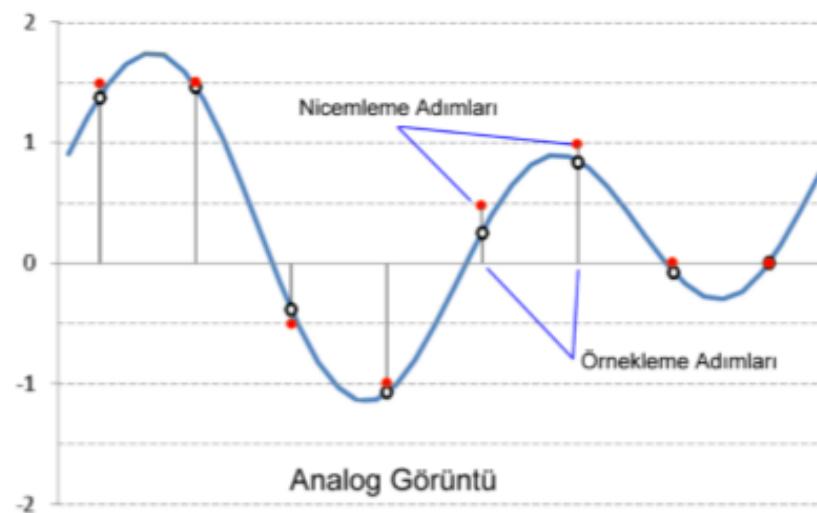
- Nokta İşlemleri: Çıkış resminin bir pikselini oluşturmak için, giriş resminin bir pikseli üzerinde yapılan işlemlerdir.
- Yerel (Local) İşlemler: Burada çıkış resmini oluşturacak bir noktanın rengi, giriş resminde bu noktanın komşularının renk özelliklerine de bağlıdır.
- Bütünsel (global) İşlemler: Çıktı görüntüsündeki bir pikselin değeri girdi görüntüsündeki tüm piksellerin değerlerine bağlıdır.



Şekil. Sayısal görüntü üzerinde yapılan işlemlerin gösterimi.

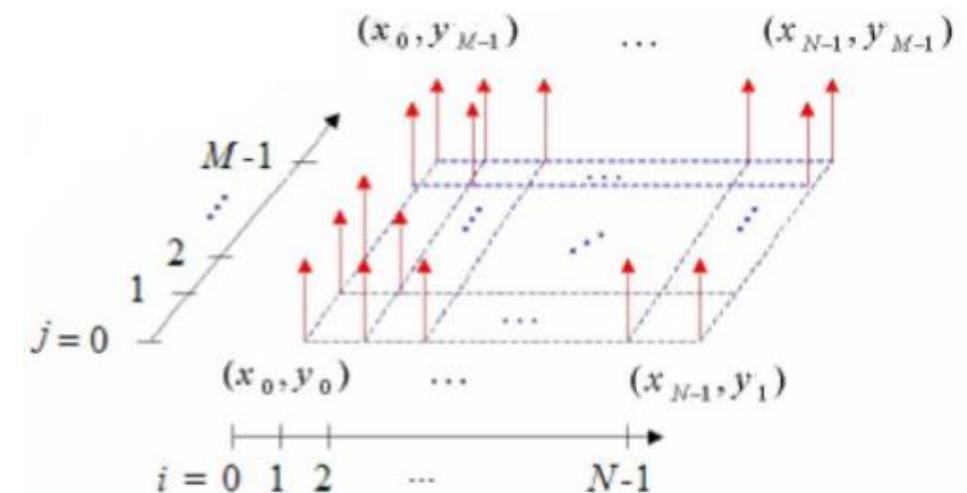
# Analog Görüntünün Sayısallaştırılması

- Analog görüntülerin bilgisayar ortamında işlenebilmesi için sayısallaştırılmaları gereklidir.
- Sayısallaştırmada resim üzerinde konumsal olarak örnekleme (sampling) ve renk derinliği için ise nicemleme (quantization/kuantalama) yapılır.
- Görüntü fonksiyonuna ilişkin koordinatlarının sayısallaştırılması, görüntü örnekleme olarak adlandırılırken; genlik değerlerinin sayısallaştırılmasına görüntü nicemleme adı verilir.



# Görüntü Örneklemme (Sampling)

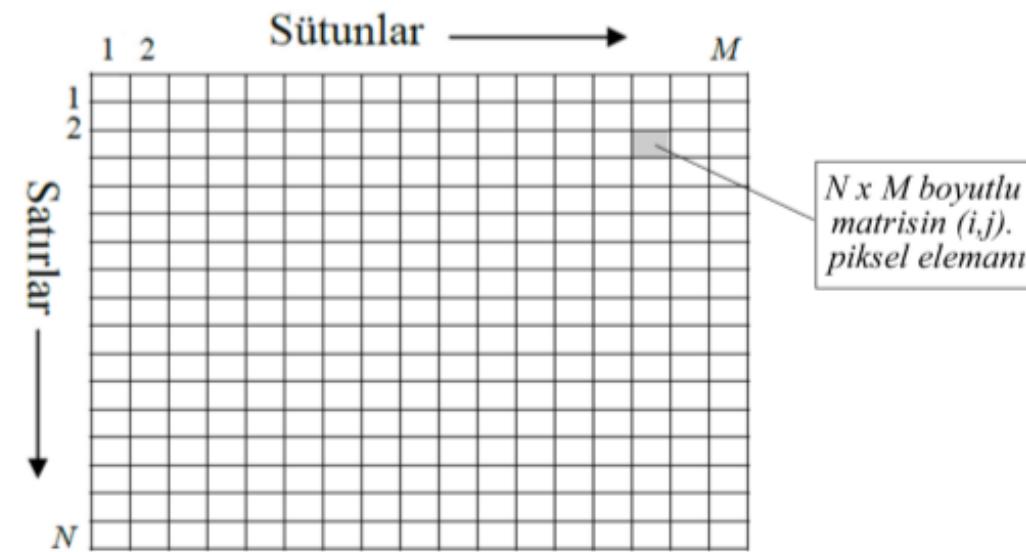
- Sayısal bir görüntü, sürekli bir görüntü fonksiyonu üzerinden eşit aralıklarla x-ekseni boyunca N adet örnek ve yekseni boyunca M adet örnek alınarak oluşturulabilir.
- Böylece, sürekli-zamanlı görüntü fonksiyonundan ayrık zamanlı görüntü fonksiyonuna geçiş gerçekleşmiş olur.
- 2-B ayrık-zamanda yatayda N ve düşeyde M örnekten oluşan toplam  $N \times M$  sonlu örnek değeri ile analog bir görüntü yaklaşık olarak ifade edilebilir.
- Bu işlemde, analog görüntü fonksiyonu düzgün örneklenmiş olur.
- Oluşan dijital (sayısal) görüntü aslında N satır ve M sütundan oluşan bir matrisdir.
- Bu işlemde bilgi kaybı vardır.



Şekil. Analog görüntünün düzgün örnekler alınarak sayısallaştırılması.

# Görüntü Nicemleme (Quantization)

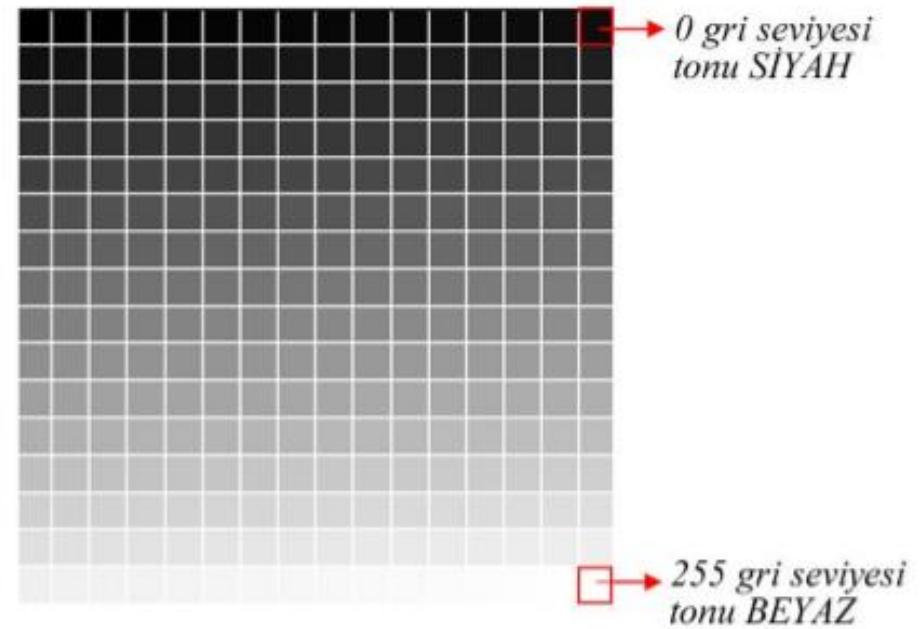
- Görüntünün her bir elemanın (pikselin) parlaklık şiddetini gösteren pozitif tamsayı değeri, nicemleme ile belirlenir.
- Görüntü elemanın en küçük ve en büyük genlik değerleri aralığı basamaklara ayrılarak, ilgili basamak değerine en yakın olan görüntü değerini almasıdır. Yani analog renk değerinin, geçerli en yakın dijital renk değerine dönüştürülmesidir.
- Örnekleme ve nicemleme işleminden sonra elde edilen sayısal görüntü, bileşenleri pozitif tamsayı değerlerinden oluşan iki-boyutlu matris yapısındadır.



Şekil.  $N \times M$  büyüğlüğüne sahip 2-B bir sayısal görüntünün temel yapısı.

# Görüntü Türleri

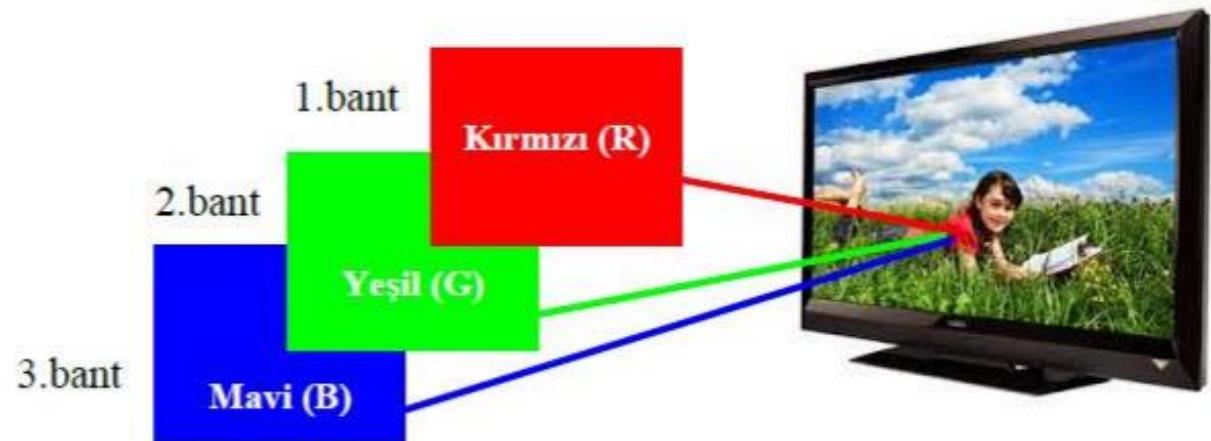
- Sayısallaştırma işleminde, görüntü boyutlarının ve her bir pikselin sahip olabileceği parlaklık değerinin belirlenmesi gereklidir. Sayısal görüntünün her bir pikselinin sahip olduğu parlaklık değeri gri seviyeler olarak adlandırılır. Her bir pikseldeki parlaklık değerinin kodlandığı bit sayısına göre gri seviye aralığı belirlenir.
- Gri seviye sınırlarında iki renk vardır, siyah ve beyaz. Bu ikisi arasında kodlanan görüntülere ise gri-ton (gray scale, monochromatic) görüntüler adı verilir.
- Uygulamada yaygın olarak kullanılan her bir piksel 8 bit ile kodlanmıştır. Bu tip görüntülerde her bir piksel  $2^8 = 256$  farklı gri ton karşılığı (parlaklık seviyesi) değerlerinden oluşur ve gri değer aralığı  $G = \{0, 1, 2, \dots, 255\}$  biçiminde ifade edilir. Kural olarak; 0 gri seviyesi siyah renge, 255 gri seviyesi ise beyaz renge ve bu değerler arasındaki gri seviyeler ise gri tonlara karşılık gelir. Şekil de  $N \times M = 16 \times 16$  'lık bir ızgara üzerinde 256 farklı gri seviyenin gösterimi verilmiştir.



Şekil.  $16 \times 16$  'lık bir ızgara üzerinde 256 farklı gri seviyenin gösterimi.

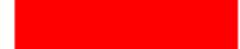
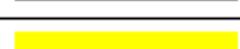
# Renkli Görüntü

- Renkli görüntüler, R(Kırmızı), G(Yeşil), B (Mavi) kodlanmış aynı cisim ait üç adet gri düzeyli görüntünün üst üste ekranada gösterilmesi ile oluşur. Renkli görüntüyü oluşturan bu üç renk bant olarak isimlendirilir.
- Renkli görüntülerin her bir pikseli bilgisayar ekranlarında 24 bit'lik veri olarak görüntülenir. Şöyle ki, her bir renk 8 bit ( $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^8 = 256$ ) ile kodlanacağına göre üç renk (RGB)  $3 \times 8 = 24$  bit ile kodlanacaktır. Bu durumda, RGB görüntülerin her bir pikseli  $28 \cdot 28 \cdot 28 = 224 = 16.777.216$  (yaklaşık 17 milyon) farklı renk değerine sahip olabilir ve bu üç rengin birleşiminin değer aralığı  $RGB = (0, 0, 0), \dots, (255, 255, 255)$  biçiminde gösterilir.



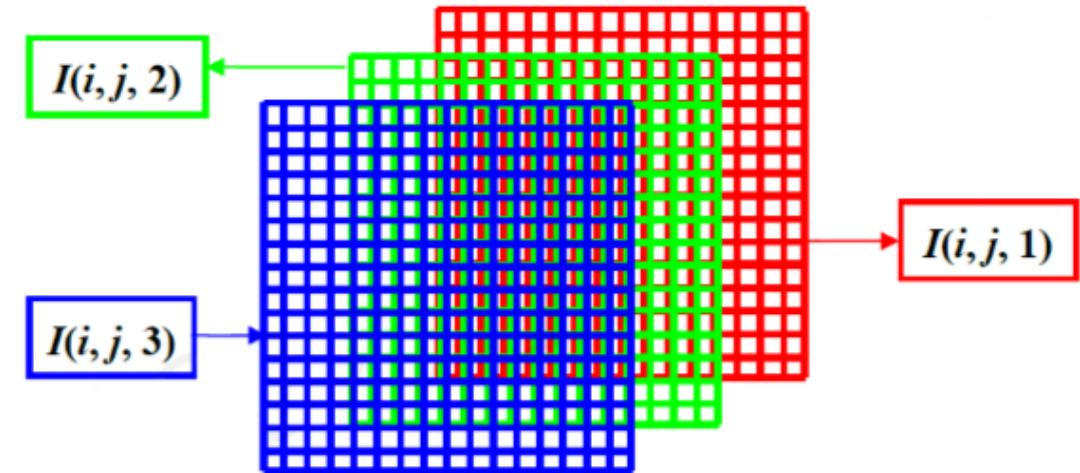
Şekil. Renkli görüntünün bilgisayar ekranındaki oluşumu.

Tablo. Bazı temel renklerin RGB değerleri

| Renk     | R   | G   | B   | Görünüm   |
|----------|-----|-----|-----|---|
| Kırmızı  | 255 | 0   | 0   |    |
| Yeşil    | 0   | 255 | 0   |    |
| Mavi     | 0   | 0   | 255 |    |
| Beyaz    | 255 | 255 | 255 |    |
| Siyah    | 0   | 0   | 0   |    |
| Açık Gri | 200 | 200 | 200 |    |
| Koyu Gri | 100 | 100 | 100 |    |
| Sarı     | 255 | 255 | 0   |   |
| Turkuaz  | 0   | 255 | 255 |  |
| Eflatun  | 255 | 0   | 255 |  |

# Renkli Görüntü

- Matris anlamında düşünüldüğünde doğal renkli 2 Boyutlu bir RGB görüntü, her biri  $(N \times M)$  büyüklüğünde üç matrisin (ızgaranın) bileşiminden oluşur ve matris gösteriminde
  - $\{I(i, j, k) | i = 1, 2, \dots, N; j = 1, 2, \dots, M; k = 1, 2, 3\}$  biçiminde temsil edilir.
- Genel olarak bu matrislerin her biri, pikselleri 256 seviyeden birine sahip görüntüyü temsil eder.

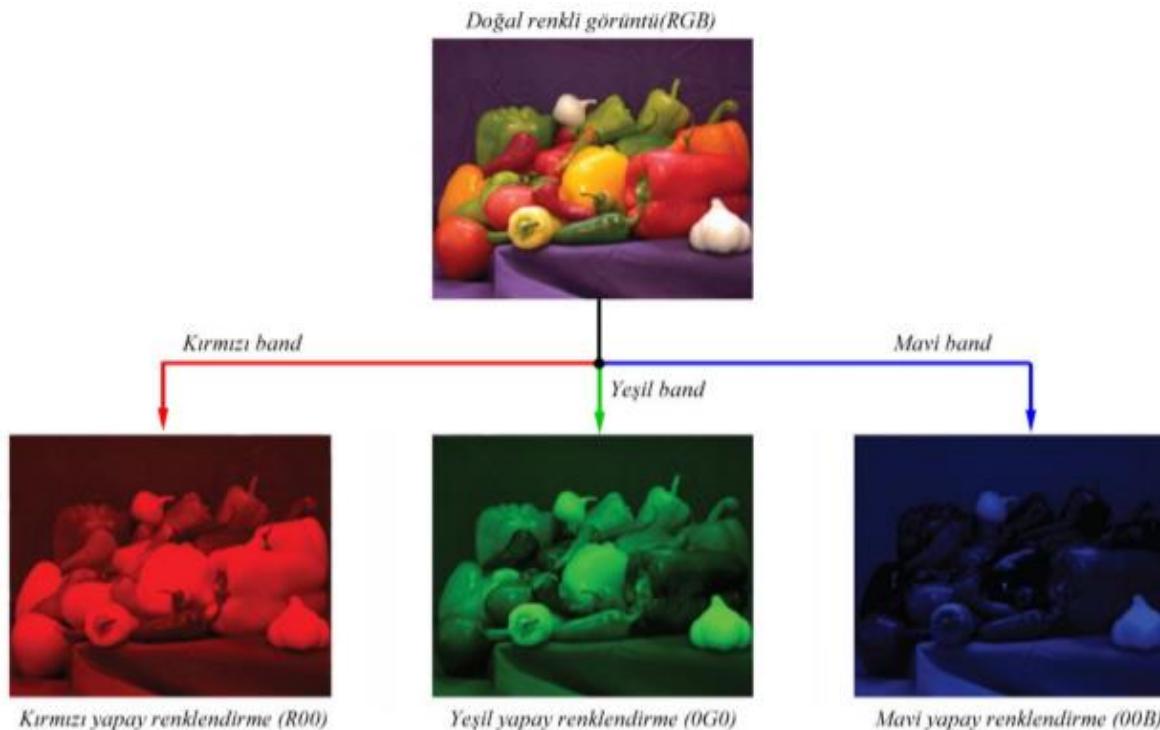


Şekil. RGB görüntüsünü oluşturan  $N \times M$  boyutlarındaki 3 bandın matris gösterimi

$I(i, j, 1); i = 1, 2, \dots, N, j = 1, 2, \dots, M \rightarrow$  Kırmızı banda ilişkin matris

$I(i, j, 2); i = 1, 2, \dots, N, j = 1, 2, \dots, M \rightarrow$  Yeşil banda ilişkin matris

$I(i, j, 3); i = 1, 2, \dots, N, j = 1, 2, \dots, M \rightarrow$  Mavi banda ilişkin matris



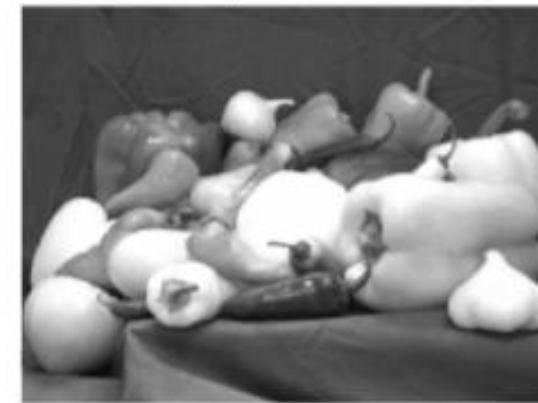
Şekil. Doğal renkli RGB bir görüntünün ana renklerle (Kırmızı-yeşil-mavi) yapay renklendirilmesi

- Renkli görüntünün yeşil ve mavi bantlarına ilişkin görüntülerin sıfır alınmasıyla sadece kırmızı bandın görüntüsü elde edilir. Böylece, RGB gösterimi kırmızı yapay renklendirilmiş görüntüsüne ulaşır. Bu durumda kodlama  $RGB = R00 = (0, 0, 0), \dots, (255, 0, 0)$  şeklinde olacaktır. Benzer şekilde, yeşil ve mavi yapay renklendirilmiş görüntüler de sırasıyla  $RGB = 0G0 = (0, 0, 0), \dots, (0, 255, 0)$  ve  $RGB = 00B = (0, 0, 0), \dots, (0, 0, 255)$  olacaktır. Bu şekilde oluşan yapay renklendirme görüntüleri aşağıdaki gibi olacaktır.
- RGB formatında doğal renklerden oluşmuş renkli bir görüntü için bantların doğru birleşimi 1-2-3 (Red, Green, Blue) sırasıyla olmalıdır.

*RGB görüntüsü (doğal renk)*



*Kırmızı (R) bandın (1. bant) gri-ton görüntüsü*



*Yeşil (G) bandın (2. bant) gri-ton görüntüsü*



*Mavi (B) bandın (3. bant) gri-ton görüntüsü*



Şekil. Doğal renkli bir görüntü ve bu görüntüye ait RGB bantlarının gri-ton görüntüler

- Üç rengin bir araya geldiği doğal rengin gri-ton görüntüsü ve her bir bandın ayrı ayrı gri-ton görüntüsü

*Doğal renkli görüntü;  
Bant birleşimi (1-2-3)*



*Yapay renkli görüntü;  
Bant birleşimi (2-1-3)*



*Yapay renkli görüntü;  
Bant birleşimi (1-3-2)*



*Yapay renkli görüntü;  
Bant birleşimi (2-3-1)*



Şekil. Doğal renkli RGB bir görüntü ile bu görüntülerden türetilen yapay renkli görüntüler

Eğer RGB gösterimde bant birleşimlerinin yerleri değiştirilecek olursa renkler de değişecektir. Bu şekilde oluşan görüntülere yapay renkli görüntüler adı verilir. Aşağıdaki resimler bununla ilgili örnekleri göstermektedir.

# Sayısal Görüntünün Kapladığı Bellek Alanı

- N satır ve M sütundan oluşan ve her bir pikseli m bit ile temsil edilen ( $N \times M$ ) büyüklükli sayısal bir görüntüyü depolamak için gerekli bit sayısı aşağıdaki gibi hesaplanır:

$$\text{Bit sayısı} = N \times M \times m$$

- 8 bit = 1byte, 1024 byte = 1Kbyte, 1024 Kbyte = 1Mbyte olarak hesaplanır.
- Örneğin her pikseli 8 bitle temsil edilen Gri bir resim NxM=1000x600 piksel boyutlarında ise  
 $b=1000x600x8=4.800.000 \text{ bit}/8 =600.000 \text{ byte} =600 \text{ Kbyte}=0,6 \text{ Mbyte}$  olur.
- Aynı resmi 24 bitlik bir kodlama ile (renkli resim) depolarsak (her bir pikseli RGB= $2^8+2^8+2^8=2^{24}=16.777.216$  milyon renkle gösterilir)

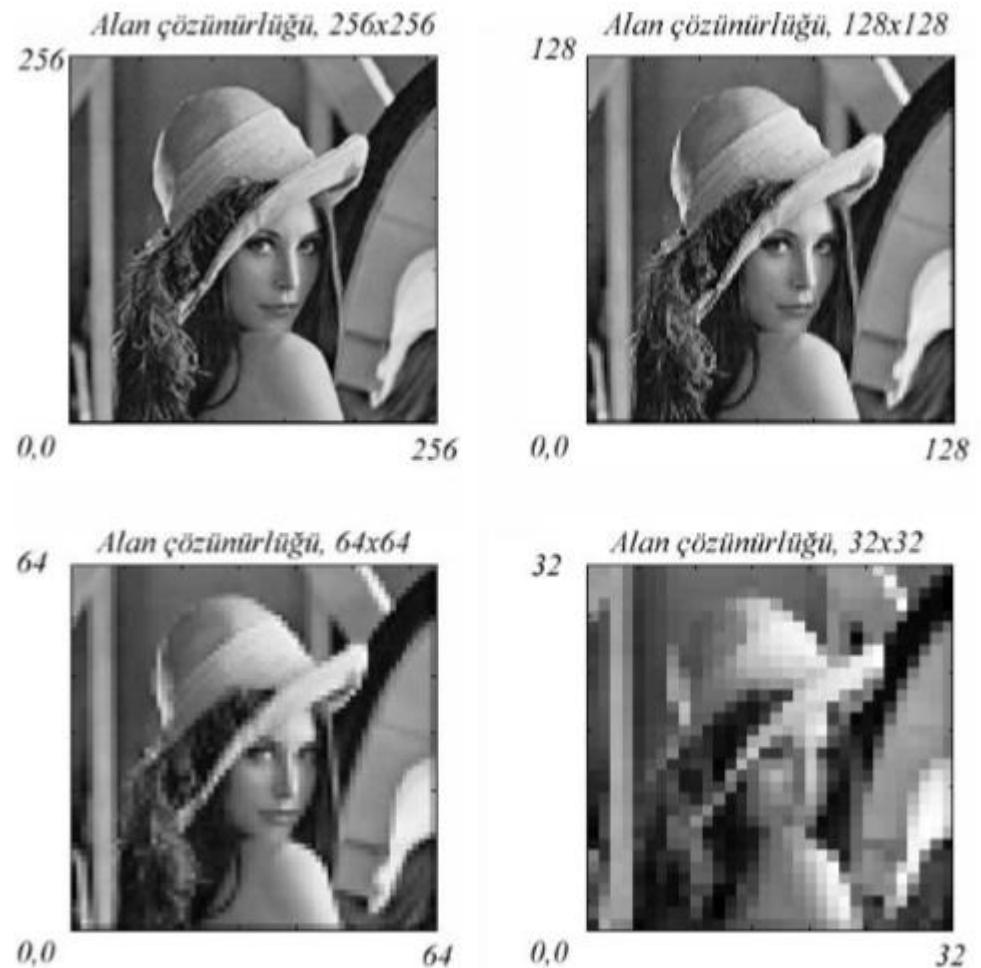
$$b=1000x600x24=14.400.000 \text{ bit}/8 =1.800.000 \text{ byte} =1.800 \text{ Kbyte}=1,8 \text{ Mbyte}$$
 olur.

# Çözünürlük

- Bir görüntünün çözünürlüğü, görüntü içerisindeki ayrıntıların fark edilebilme derecesidir.
- Bu kavram hem Alan çözünürlüğünü hem de Parlaklık çözünürlüğünü içinde barındırır.
- Alan çözünürlüğü yüzeyden alınan örnek sayısı ( $N \times M$ ) ile Parlaklık çözünürlüğü ise piksel üzerindeki rengin gri seviye karşılığı olarak parlaklığını belirtir.
- Bu parametrelerin değerlerindeki artış ne kadar fazla olursa sayısallaştırılmış görüntü orijinaline o oranda yaklaşır.
- Bununla birlikte, görüntünün kapladığı bellek alanı hızlı bir şekilde artar.

# Alan Hassaslığı (Piksel çözünürlüğü)

- Alan hassaslığı, sayısal görüntüyü elde etmek amacıyla analog görüntünün yatay/dikey taranması esnasında toplanan örnek sayısı ile ilgilidir.
- Sayısal görüntüyü oluşturan piksellerin sayısı ( $N \times M$ ) ne kadar fazla olursa orijinal (analog) görüntüye o kadar fazla yaklaşılır.
- Piksellerin sayısı azaltılırsa bu durumda görüntünün uzaysal çözünürlüğü düşer ve görüntü içerisindeki ayrıntılar kaybolmaya başlar.



Şekil. Her pikseli 8 bit le temsil edilen, farklı alan çözünürlüklerine sahip örnek resimler

# Parlaklık hassaslığı-renk derinliği (gri-seviye renk çözünürlüğü)

- Parlaklık hassaslığı, her bir pikselin parlaklığının karşılığı olan gri seviye skalarındaki sayısını gösterir.
- Gri seviyelerin sayısı azaldığında, görüntü içerisinde yapay çizgisel hatlar ortaya çıkmaya başlar
- Şekilde 256 renk (8 bit), 128 renk (7 bit), 64 renk (6 bit), 32 renk (5 bit), 16 renk (4 bit), 8 renk (3 bit), 4 renk (2 bit) ve 2 renk (1 bit) olan aynı alan çözünürlüğünə sahip ( $N \times M = 256 \times 256$ ) gri-ton görüntüler verilmiştir.



## 2-Boyutlu sayısal görüntülerin matris olarak gösterimi

- $(N \times M)$  piksele sahip 2-B sayısal bir görüntü, N satır ve M sütundan oluşan bir matris gibi düşünülebilir.
- Matrisin bileşenleri, satır ve sütunların kesiştiği her bir noktada –ki bu noktalar sayısal görüntünün en küçük parçası olan pikseli temsil eder – pozitif tamsayı değerlerine sahiptir.
- Örneğin, 256 adet gri seviye içeren görüntüler için bu değer 0 – 255 aralığındaki tamsayılardan oluşur.
- Sayısal bir görüntüyü temsil eden matrisin bileşenleri hiçbir zaman negatif ve tamsayı dışındaki değerlerden oluşamaz.
- Ancak, görüntü üzerinde işlemler yapıldıktan sonra elde edilen yeni görüntü içerisinde bu gibi sonuçlarla karşılaşmak olasıdır.
- Böyle durumlarda, yeni görüntüyü bilgisayar ekranında doğru bir biçimde görüntüleyebilmek için matris değerleri üzerinde uygun ölçeklendirme ve yuvarlama işlemleri yapılmalıdır.

$$A = \begin{bmatrix} 0 & 1 & \dots & 255 \\ 0 & 1 & \dots & 255 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 1 & \dots & 255 \end{bmatrix}$$

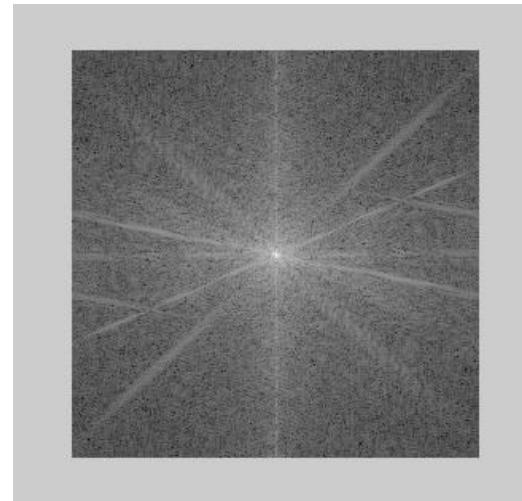
256 sütun →  
↓ 256 satır

# Bölüm 2 - Noktasal İşlemleri

- Herhangi bir görüntü işleme uygulaması piksel gri değerleri üzerinde dönüşüm yapar. Bu dönüşümü yapmak için gerektirdiği bilgiye bağlı olarak görüntü işleme uygulamaları 3 sınıfta incelenebilir:
- 1. Dönüşümler: Piksel değerleri başka bir domende ifade edilir. Fourier dönüşümü gibi. Tüm görüntü tek bir büyük blok gibi düşünülür.
- 2. Bölgesel İşlemler: Bir pikselin gri değerini değiştirmek için komşu piksellerden faydalananır.
- 3. Noktasal İşlemler: Piksel yeni gri değeri için sadece eski değerinden faydalananır. Basit işlemlerdir. Genellikle ön işleme için gereklidir.

# Tamamlayıcı Not

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



- **Frekans domain(Frequency domain ):** Görüntünün birçok farklı frekanslı bileşenden olduğu 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 (işlenmiş ) görüntüsüdür.

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

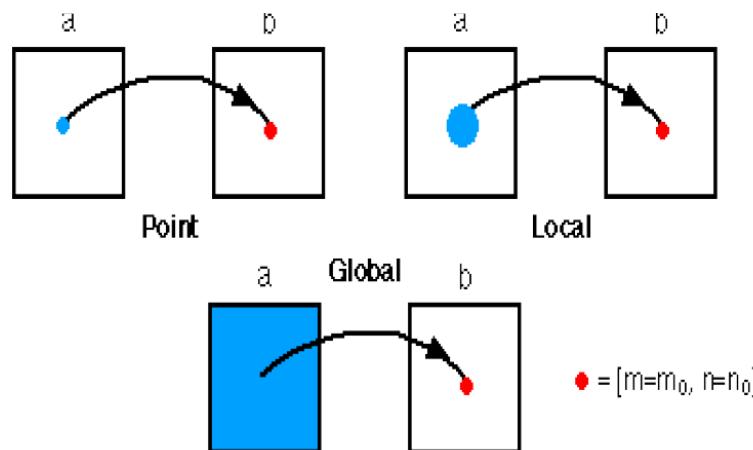
Örneğin T operatörü; K görüntülerinde gürültü azaltmak için, bir görüntü seti işlemi olarak ta çalışabilir.

## 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 “griton değişikliği” (modification) gibi isimler verilir. Nokta operasyonları genellikle “resim onarımı” (manipulation) için kullanılır. Mesela, bir resmin kontrastının yükseltilmesi gibi. **Nokta operasyonları sıfır hafıza operasyonlarıdır.**

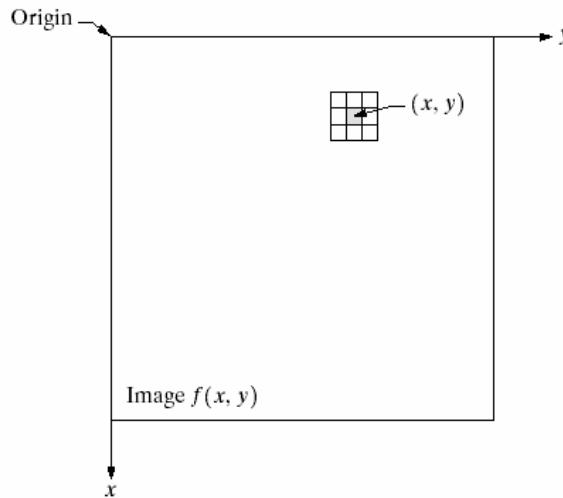
**Bölgesel (lokal-Komşuluk ilişkili) işlemlerde** merkez pikselin değeri komşu piksellerin değeri ile belirlenir. Filtreleme işlemlerinde çok kullanılır.

**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**  
 $3 \times 3$   
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 herbir lokasyona  $(x,y)$  uygulanarak lokasyonda işlenmiş  $g(x,y)$  çıkışı elde edilir.

En küçük lokasyon (komşuluk ilişkisi-bölgesel) resim içindeki  $1 \times 1$ 'lik bölgедir. 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.

# Kontrast Nedir?

- Kontrast görüntüdeki en parlak bölüm ile en karanlık bölüm arasındaki farka denir.
- Siyah ve beyaz arasındaki kontrast aralığı genişledikçe, iki uç arasındaki gri tonlar ya da ara seviyeler daha rahat görülür.
- Aradaki değer ne kadar fazla olursa görüntünün canlılığı o kadar iyi olur.

# Aritmetik İşlemler

- Her bir piksele sabit bir değer eklenebilir veya çıkarılabilir .
- Pikseller bir sabitle çarpılabilir.

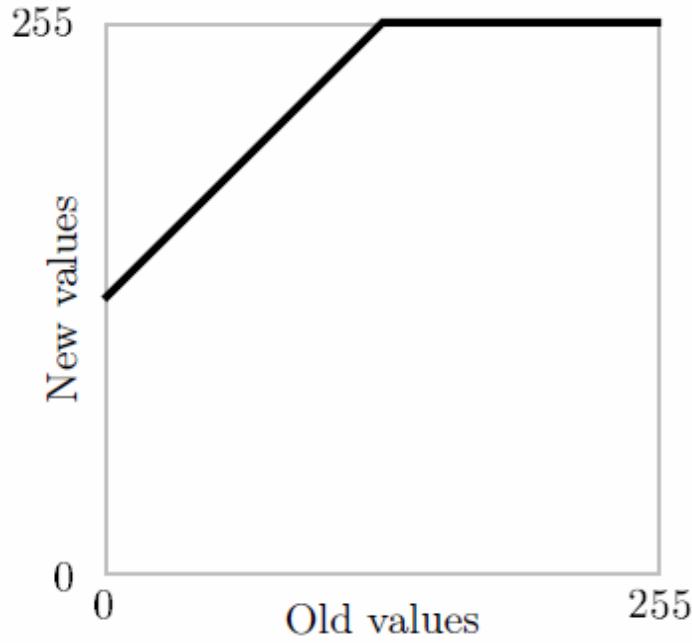
$$y = f(x)$$

$$y = x \pm C$$

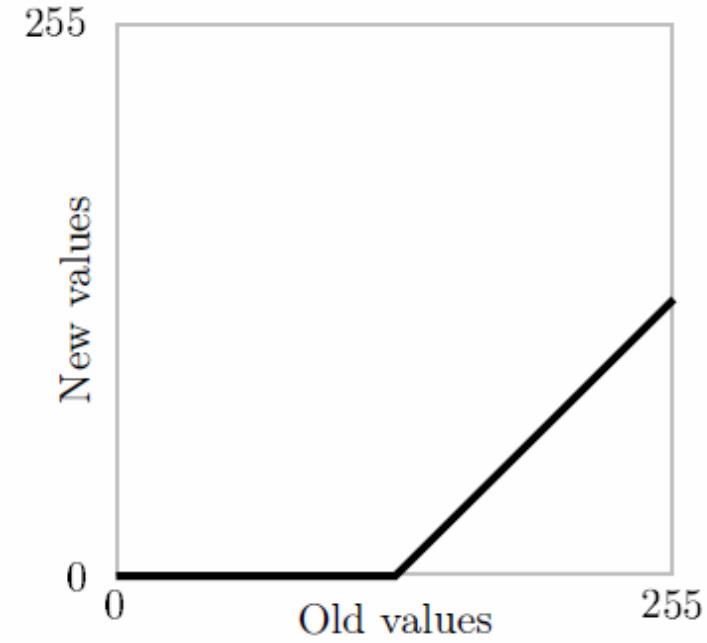
$$y = Cx.$$

$$y \leftarrow \begin{cases} 255 & \text{if } y > 255, \\ 0 & \text{if } y < 0. \end{cases}$$

# Aritmetik İşlemler



Adding 128 to each pixel



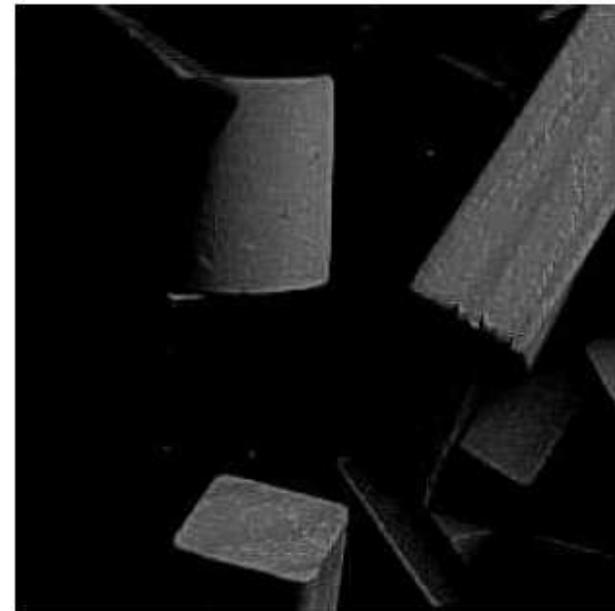
Subtracting 128 from each pixel

Figure 2.2: Adding and subtracting a constant

# Aritmetik İşlemler



b1: Adding 128



b2: Subtracting 128

Figure 2.3: Arithmetic operations on an image: adding or subtracting a constant

# Aritmetik İşlemler

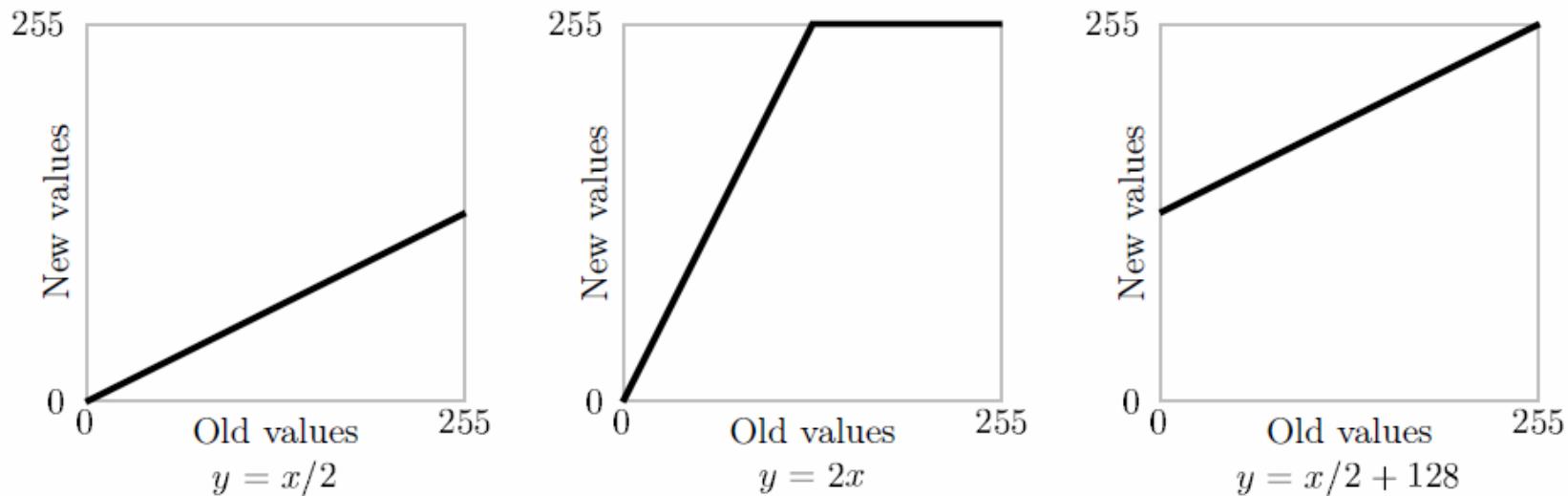
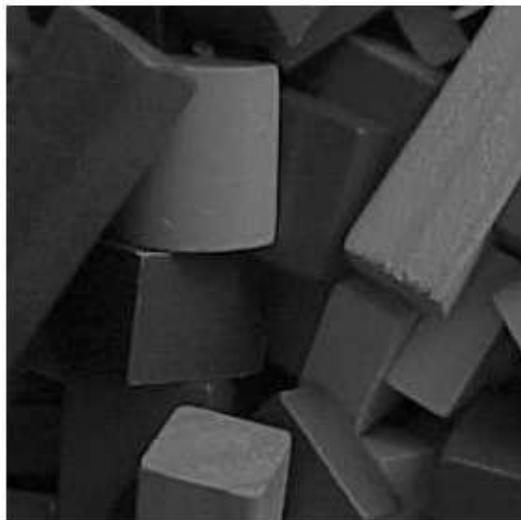


Figure 2.4: Using multiplication and division

# Aritmetik İşlemler



b3:  $y = x/2$



b4:  $y = 2x$



b5:  $y = x/2 + 128$

Figure 2.5: Arithmetic operations on an image: multiplication and division

# Negatif Alma

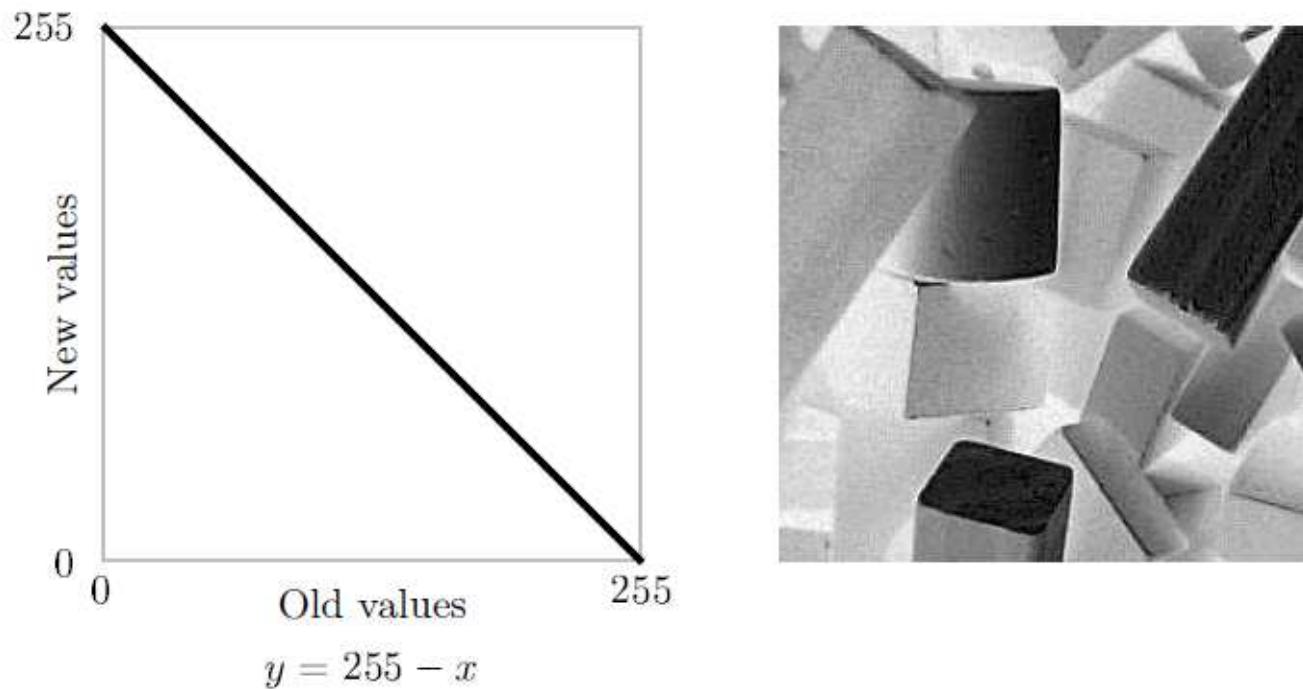


Figure 2.6: Image complementation

# Negatif Alma

Görüntünün sadece bir kısmının negatifi alınabilir. Örneğin gri değeri 128 ve üstü olanların veya tam tersi bir durumda olabilir.

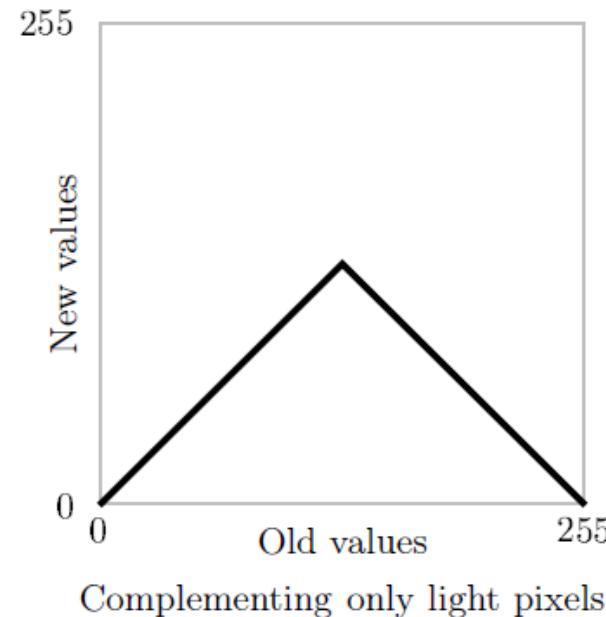
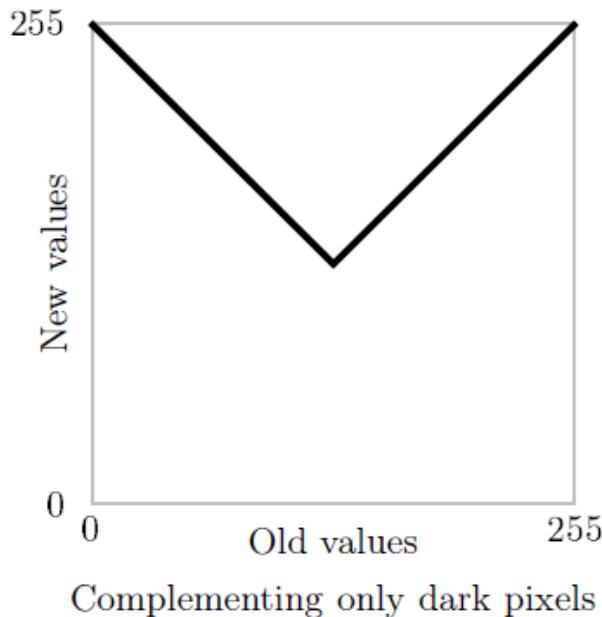


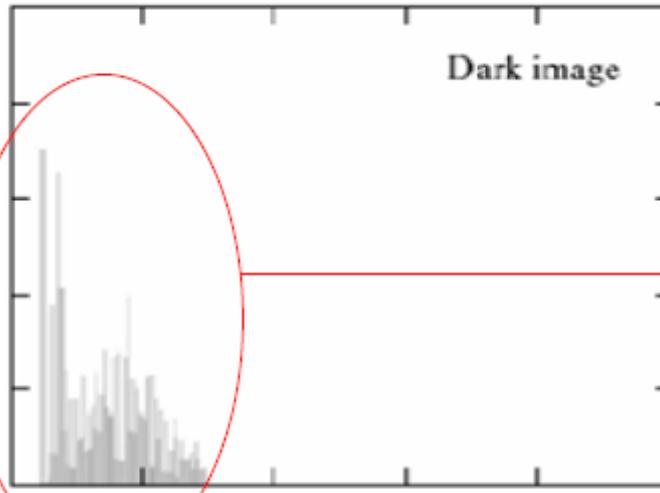
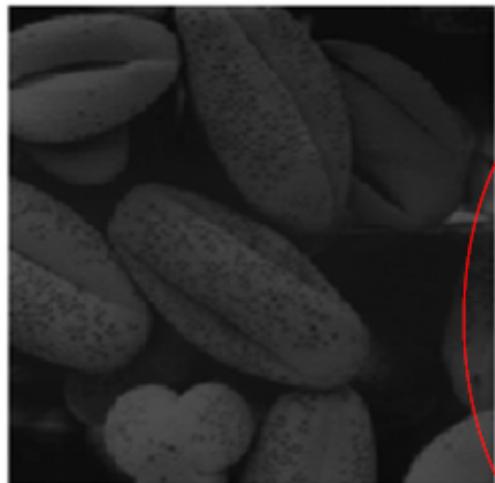
Figure 2.7: Part complementation

# Histogram

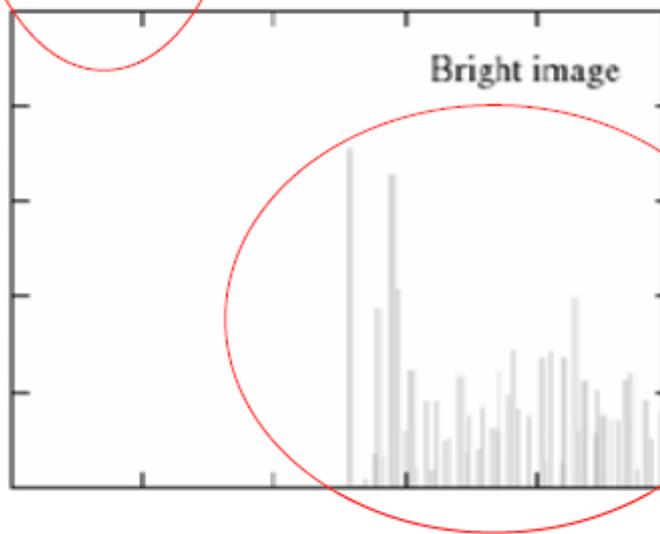
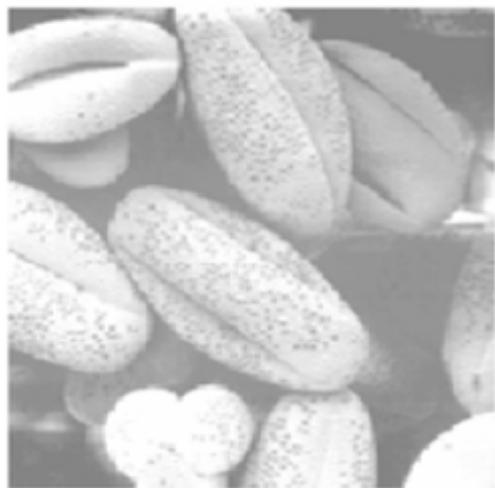
- Bir gri değerin görüntüde kaç tane olduğunu gösteren grafiktir. Görüntü hakkında bilgi vermesi açısından önemlidir.
- Karanlık bir görüntüde gri değerler histogramın alt ucunda,
- Aydınlık bir görüntüde gri değerler histogramın üst ucunda,
- İyi kontrastlı bir görüntüde ise, gri değerler aralıkta düzgün şekilde dağıılır.

# Histogramdan elde edilen bilgiler

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

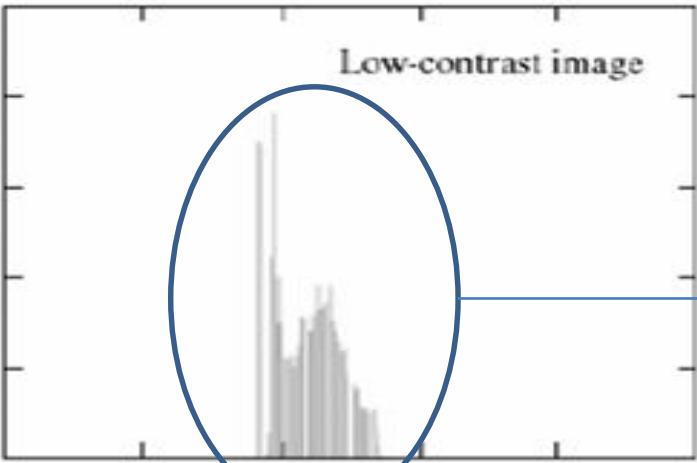
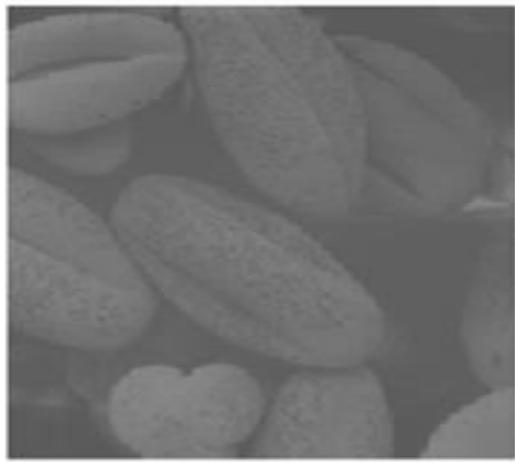


Karanlık imge

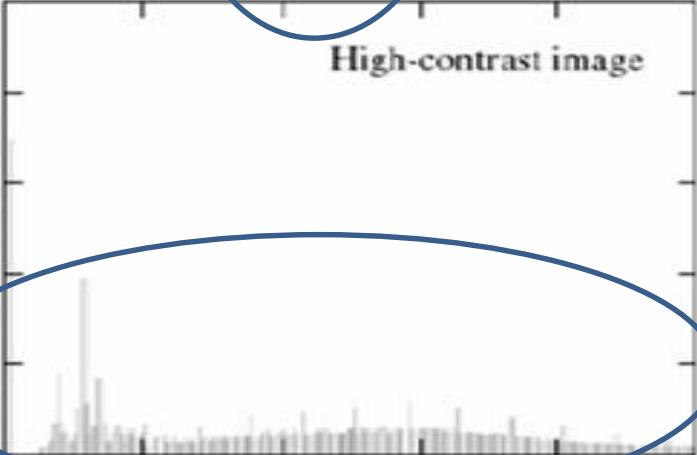


Parlak imge

- 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 olarak adlandırılır.



Karşılığı düşük



Karşılığı yüksek

# Histogram Germe

| 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  | 15 |    |

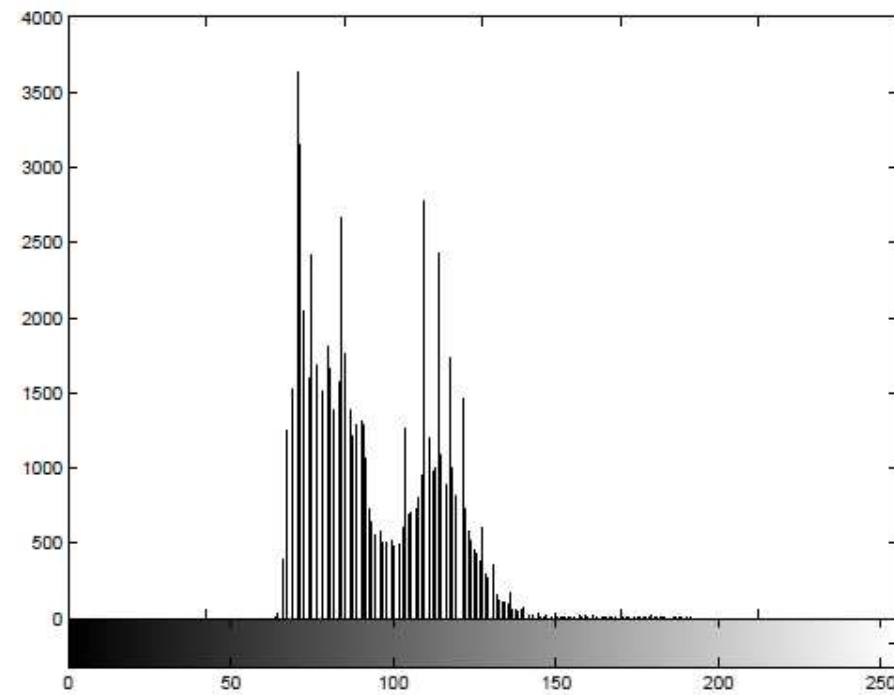


Figure 2.8: The image `pout.tif` and its histogram

# Histogram Germe

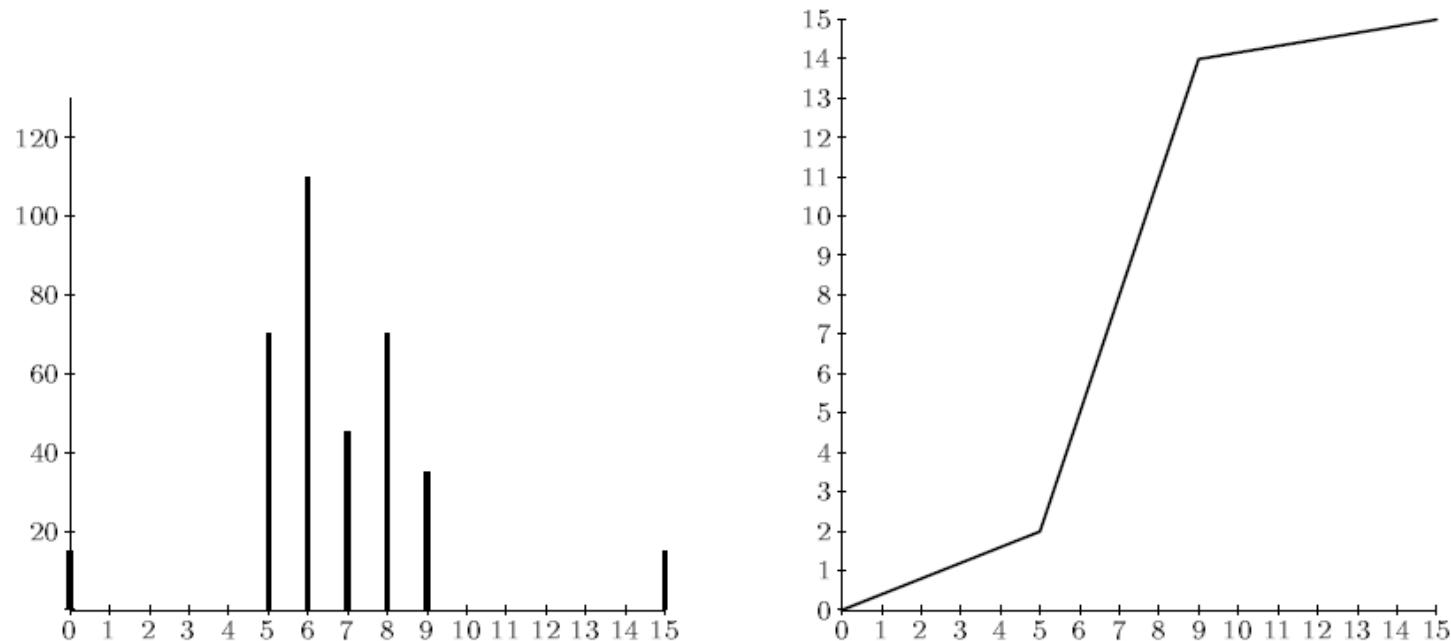


Figure 2.9: A histogram of a poorly contrasted image, and a stretching function

# Histogram Germe

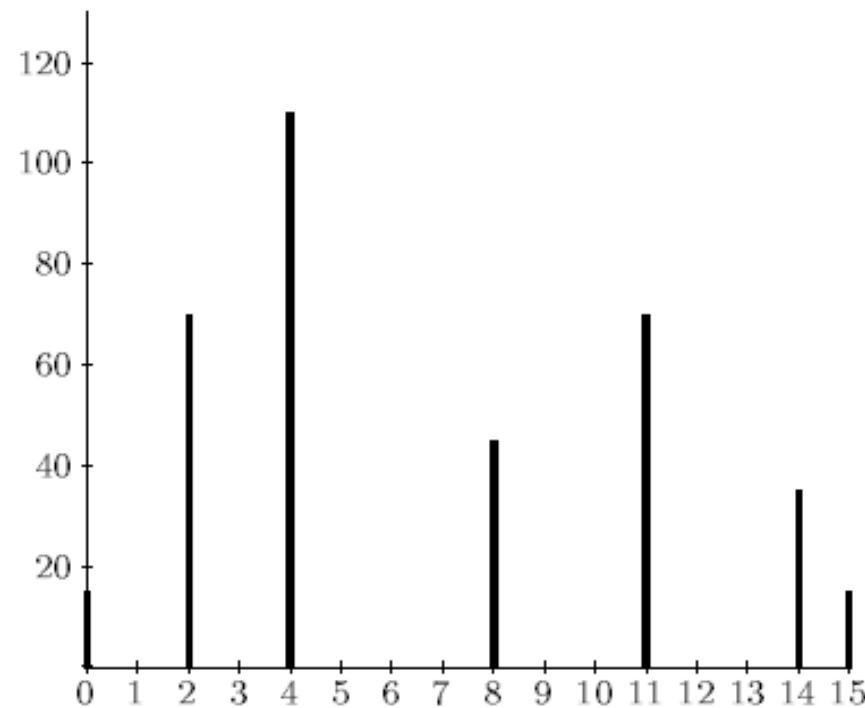
- 5-9 aralığındaki gri değerleri 2-14 aralığına çekmek için aşağıdaki lineer fonksiyon kullanılabilir.
- Burada  $i$  pikselin orijinal değeri  $j$  ise dönüşüm sonrasındaki değeridir.

$$j = \frac{YENI_{mak} - YENI_{min}}{ESKI_{mak} - ESKI_{min}} (i - ESKI_{min}) + YENI_{min}$$

$$j = \frac{14 - 2}{9 - 5} (i - 5) + 2 \quad \begin{array}{ccccccc} i & 5 & 6 & 7 & 8 & 9 \\ j & 2 & 5 & 8 & 11 & 14 \end{array}$$

# Histogram Germe

İşlem Sonucu Yeni Histogram



# Histogram Eşitleme

- Historam germanin; bir kullanıcı girişine ihtiyaç duyma gibi bir dezvantajı 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.)
- Ideal 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 bölgесine de uygulanabilir. Tüm resme uygulanırsa global histogram eşitleme, resmin belli bir bölgесine uygulandığında ise lokal histogram eşitleme adını alır.

## Histogram eşitlemenin özeti

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

- **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 büyülüktür.

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 biribirine 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,

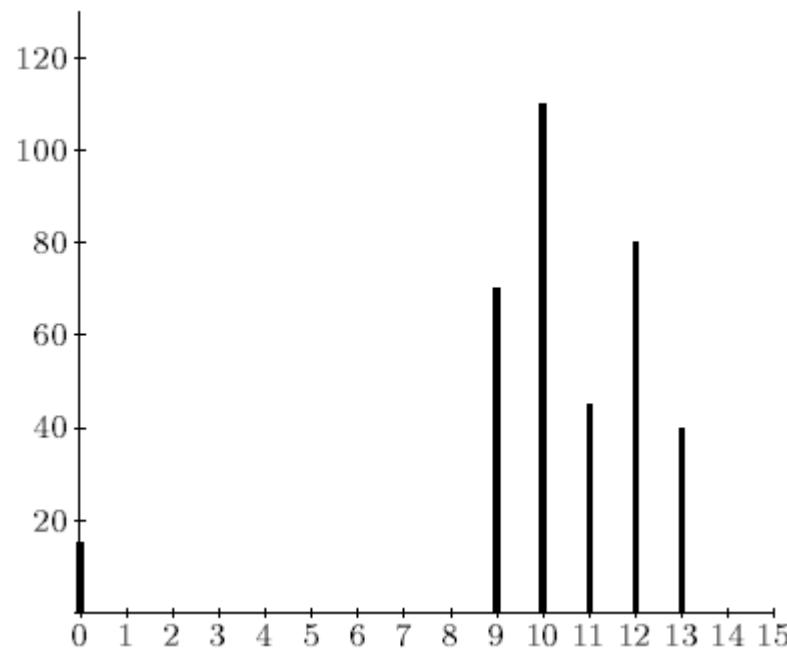
**L:** mümkün olan (veya istenilen) toplam gri seviye sayısı( 8 bit renk derinliğinde 255 v.b)

**$s_k$ :** Daha iyi kontraslı 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) $N_i$ | : | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 70 | 110 | 45 | 80 | 40 | 0  | 0  |

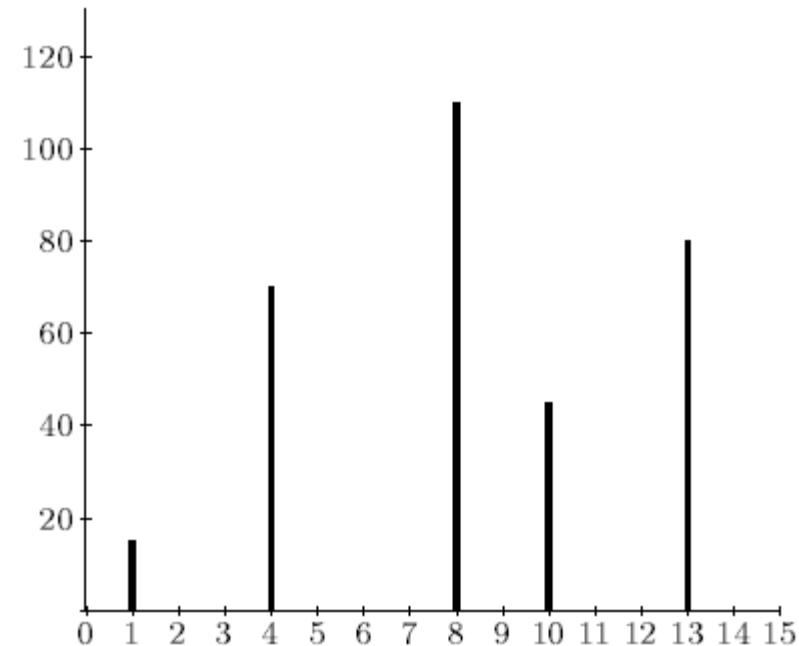
Yukarıda, renk derinliği 4 bit olan gri seviyeli bir görüntünün histogram tablosu görülmektedir. Bunun histogram grafiği ise aşağıdaki şekilde dir. (Toplam piksel sayısı  $n=360$ , istenen gri seviye  $L=16$ )



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 $i$ | $n_i$ | $\Sigma n_i$ | $(1/24)\Sigma n_i$ | 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 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.

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



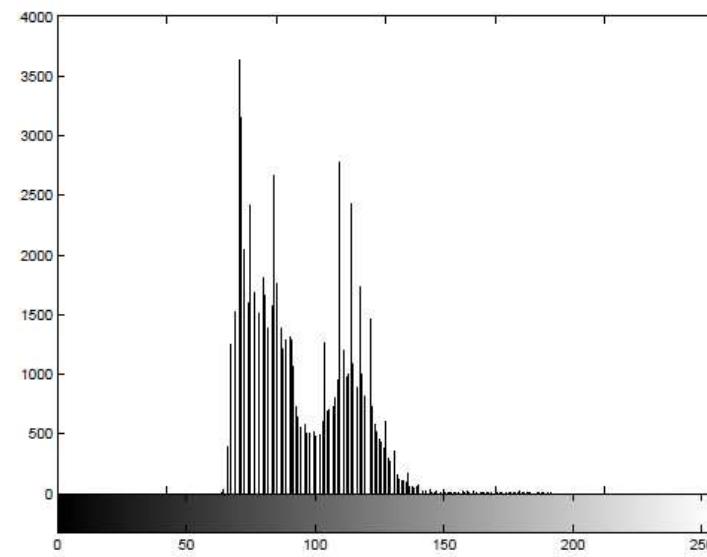
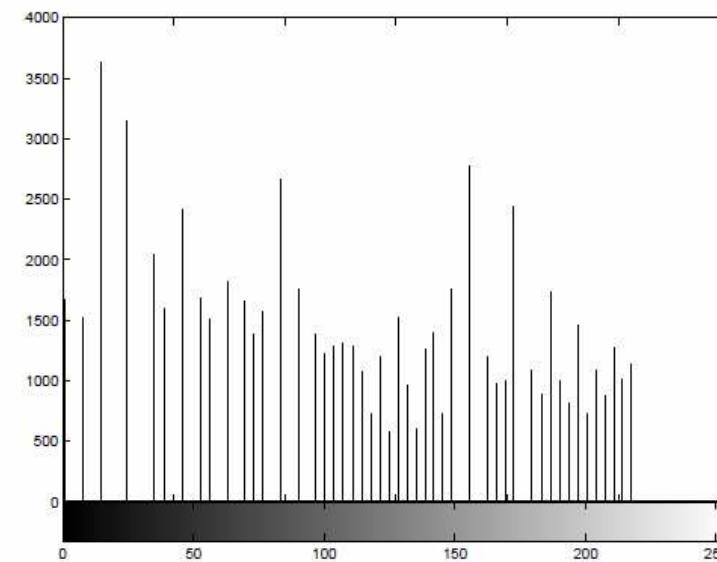


Figure 2.8: The image `pout.tif` and its histogram



02.03.2015

Figure 2.19: The histogram of figure 2.8 after equalization

**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 |

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

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

| 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     |       |       |

histogram

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

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

| 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 |

# Sayısal İşaret ve Görüntü İşleme

## 4. Hafta

### Matlab Uygulama- Eşikleme (thresholding) ve Bit Düzlem Dilimleme

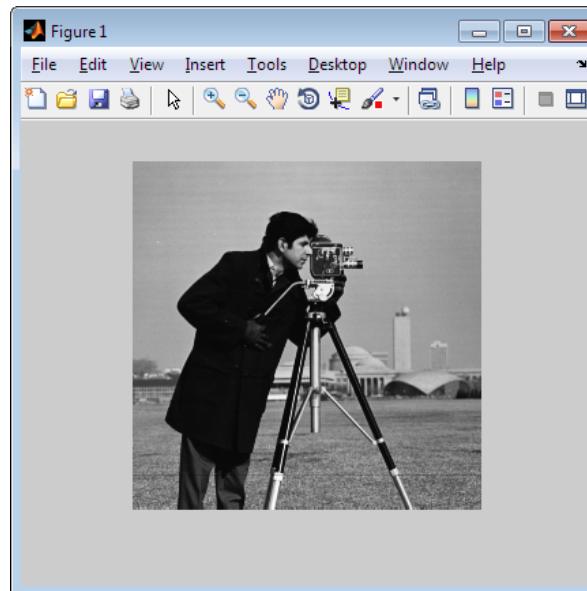
Yrd. Doç. Dr. Tuba KURBAN

# Görüntü Eşikleme

- Bu fonksiyon genellikle gri tonlu bir görüntüden, ikili görüntü oluşturmak için kullanılır.
- Renkli görüntülerle de kullanılabilir.
- Kaynak görüntünün piksel değerlerini çok büyük ya da çok küçük değerlere filtreler.

# Gri Görüntüler Üzerinde Eşikleme

- Test için gri seviye görüntü alınmaktadır.
- `imge = imread('cameraman.tif');`
- `imshow(imge);`

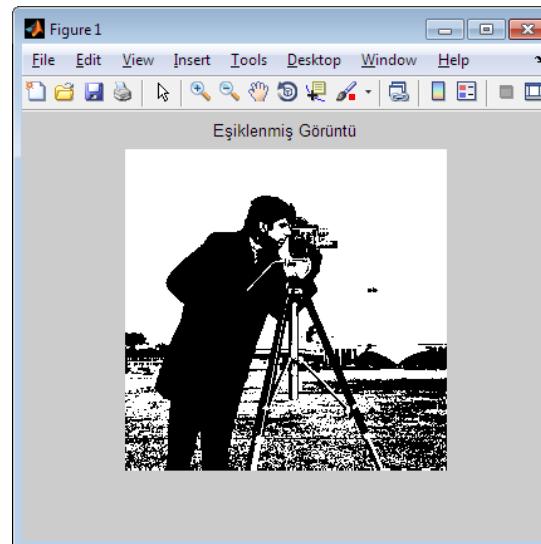


# im2bw Fonksiyonu ile Eşikleme

- im2bw fonksiyonu parametre olarak bir eşikleme sınır değeri alır.
- Sınır değerini ister deneme yanılma ile, istersek histogram incelemeye ile bulunabilir.
- Optimum bir eşik değeri için görüntünün histogramını incelemek daha faydalı olacaktır.

# im2bw ile Eşikleme

- Önce elimizdeki gri görüntüye 0.5 sınır değeri ile eşikleme yapalım.
- `imge2 = im2bw(imge,0.5);  
imshow(imge2),title('Eşiklenmiş Görüntü')`



# Graythresh Fonksiyonu ile Eşik Değerinin Alınması

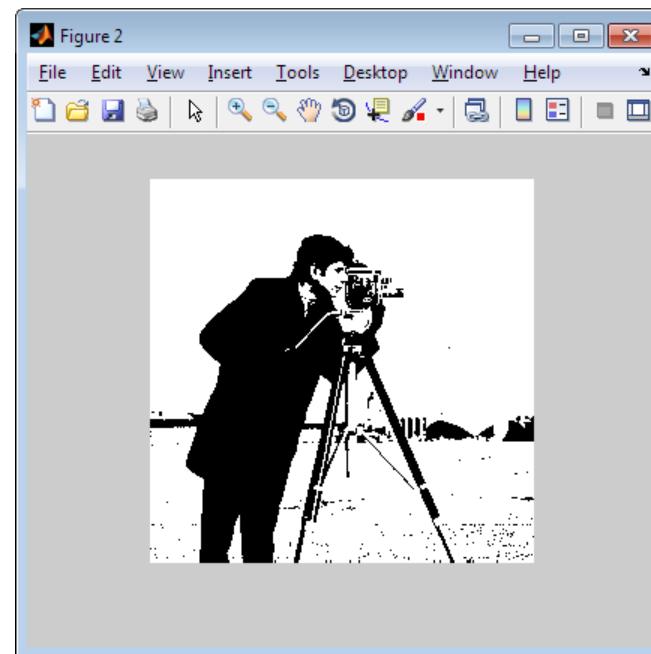
- Şimdi Matlabda tanımlı graythresh fonksiyonunu çağrıarak elimizdeki görüntü için otomatik olarak ideal bir eşik sınırı elde etmeye çalışalım.
- Graytrash histogram bazlı bir yöntemdir. Bu yöntemde algoritma önce görüntünün histogramını hesaplar, ardından otomatik bir threshold değeri döner.

# Graythresh Fonksiyonu ile Eşik Değerinin Alınması

- Esikdegeri=graythresh(imge)
- Esikdegeri=0.3451 belirlenir. (Daha önce alınan ‘cameraman.tif’ görüntüsü için)
- Buradan çıkar değer bizim belirlediğimiz yaklaşık 0.29 değer uzaklığında.
- Otomatik bulunan bu değerle eşikleme yapılırsa.

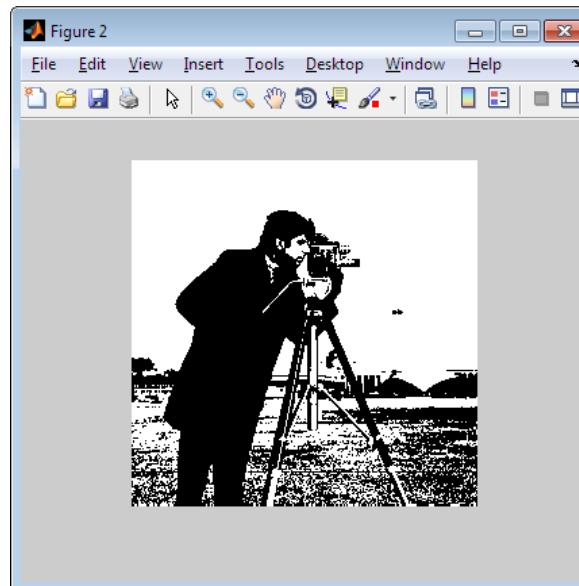
# im2bw ile Eşikleme

- `imge2 = im2bw(imge,esikdegeri);`
- `Figure, imshow(imge2)`



# İlişkisel Operatörler ile Manuel Eşikleme

- `imge = im2double(imge);`
- `imshow(imge>0.5)`
- % Ekrana sadece 0.5 değerinden yüksek pikselleri getirir, diğerlerini 0' a eşitler.



# Döngü ile Matlab de Eşikleme

```
I=imread('cameraman.tif');
for i=1:size(I,1)
    for j=1:size(I,2)
        if I(i,j)>120
            I2(i,j)=255;
        else I2(i,j)=0;
        end
    end
end

imshow(I), figure(), imshow(I2)
```

# Bit Düzlem Dilimleme

- 8 bitlik bir gri r
- esmi alarak bu resmin her değerli bitinin etkisini görmek amacıyla kullanılır.
- Eğer seçilen değerdeki bit 1 ise 255 renk değeri 0 (sıfır) ise 0 (sıfır) renk değeri verilerek her bit için yeni bir Siyah-Beyaz resim elde edilecektir.
- Böylece her bit değerinin resme yaptığı etki belirlenmiş olur.

# Bit Düzlem Dilimleme

- Burada amaç gri seviye görüntütüyü değil, her bir bitin görüntüye olan katkısını vurgulamaktır.
- 8 bitlik bir görüntütüyü düşünecek olursak bit düzlemleri 0 (en az öneme sahip bit- LSB (least significant bit))-7 (en fazla öneme sahip bit- MSB) arasında değişir.
- 0. bit düzlemi görüntütudeki tüm LSB bitlerini içerir.
- 7. bit düzlemi ile görüntütudeki tüm MSB bitlerini içerir.

# Bit Düzlem Dilimleme

- Bir sayısal görüntüyü bit düzlemlerine ayırma işlemi her bir bitin önemini analiz etmek için önemlidir.
- LSB gibi görüntüye daha az etkisi olan bitler görüntü sıkıştırma, veri gizleme gibi uygulamalar için kullanılabilir.

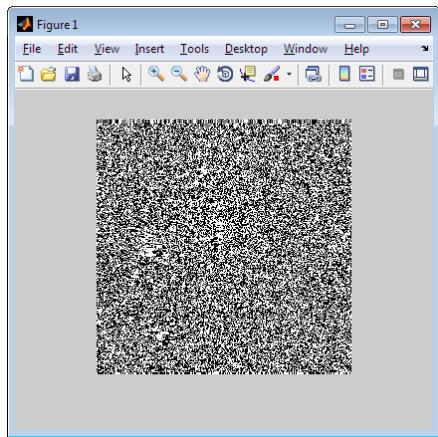
# Bit Düzlem Dilimleme

```
c=imread('cameraman.tif'); %görüntüyü oku
imshow(c);
cd=double(cd);

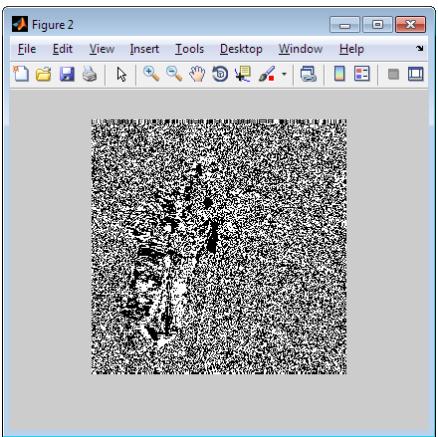
c0=mod(cd,2); %lsb bit    %bit düzlemleri hesapla
c1=mod(floor(cd/2),2);
c2=mod(floor(cd/4),2);
c3=mod(floor(cd/8),2);
c4=mod(floor(cd/16),2);
c5=mod(floor(cd/32),2);
c6=mod(floor(cd/64),2);
c7=mod(floor(cd/128),2);

figure, imshow(c0); % bit düzlemleri göster
figure, imshow(c1);
figure, imshow(c2);
figure, imshow(c3);
figure, imshow(c4);
figure, imshow(c5);
figure, imshow(c6);
figure, imshow(c7);

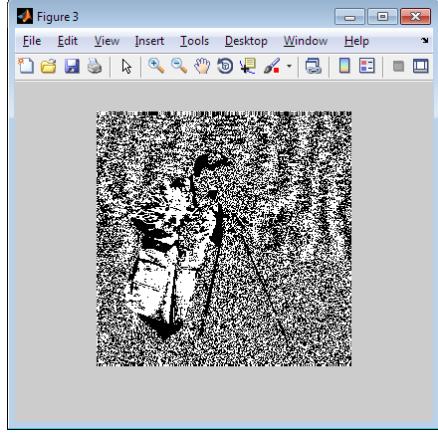
cc=2*(2*(2*(2*(2*(2*c7+c6)+c5)+c4)+c3)+c2)+c1)+c0; %orjinal görüntü bit düzlemlerden elde edilebilir.
imshow(uint8(cc));
```



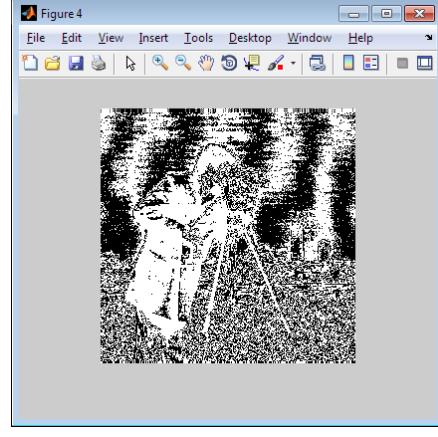
0. bit düzleme



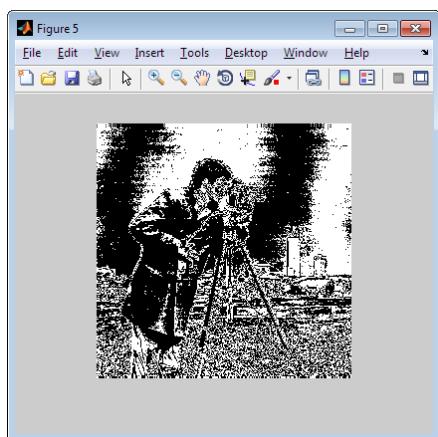
1. bit düzleme



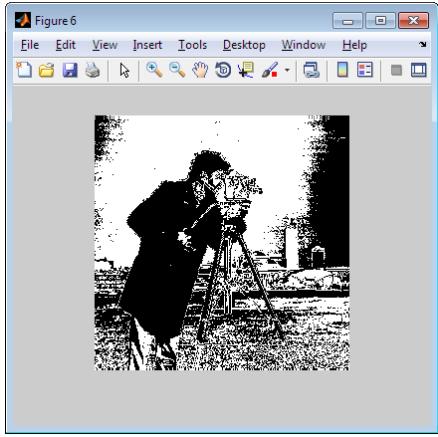
2. bit düzleme



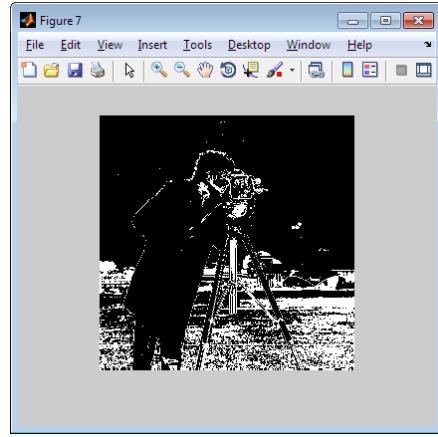
3. bit düzleme



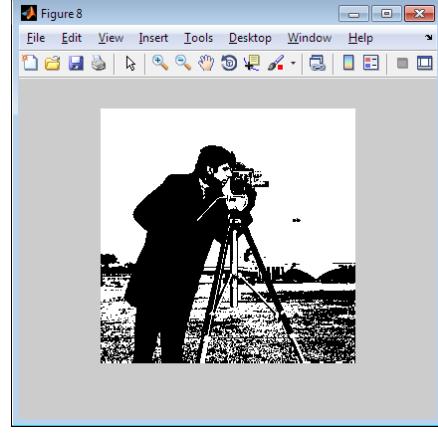
4. bit düzleme



5. bit düzleme



6. bit düzleme



7. bit düzleme

# Bit düzlemlerden elde edilen görüntü



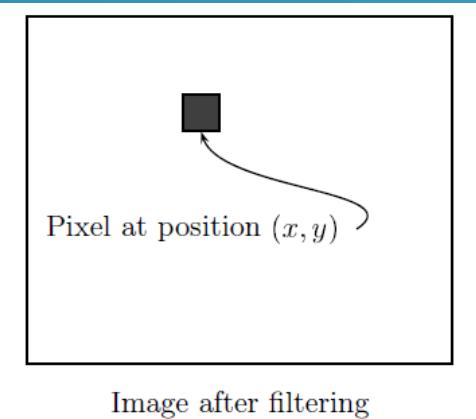
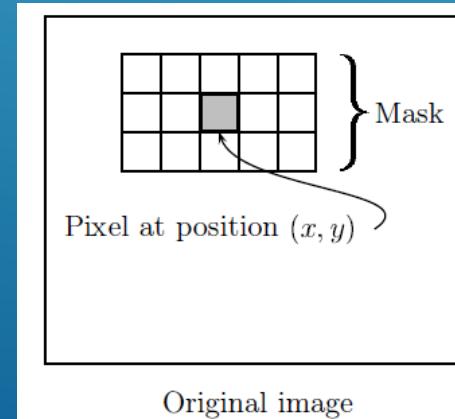
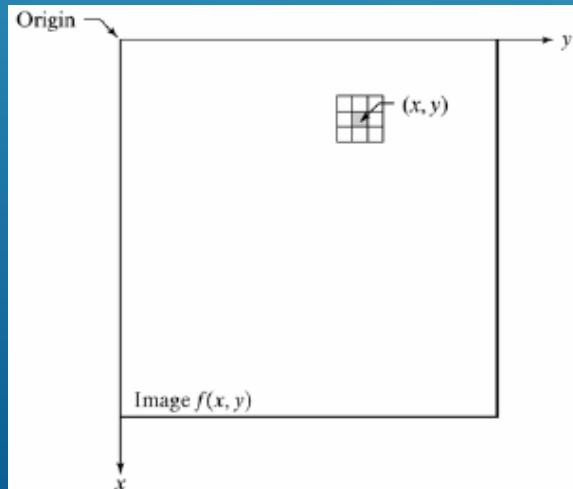
# KOMŞULUK İLİŞKİSİ İŞLEMLER (UZAYSAL FILİTRELER)

Noktasal işlemler imgedeki her piksele, diğer piksellerden bağımsız olarak  $y=f(x)$  işleminin uygulanışıydı.

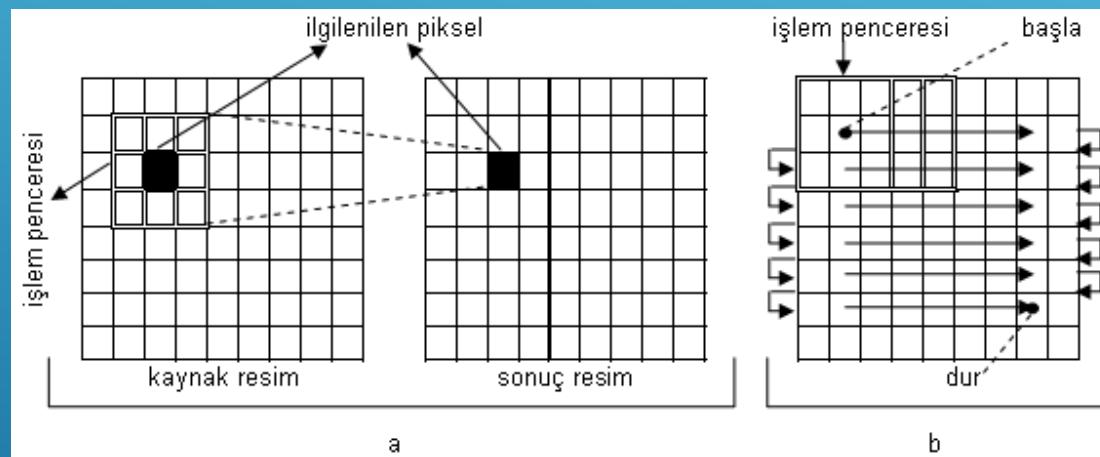
Komşuluk ilişkili işlemler (Bölgesel İşlemler) ise; Noktasal işlemlerin genişletilerek  $y=f(x)$  fonksiyonunu her bir pikselin komşuluk ilişkilerine göre merkez piksele uygulamaktır. Noktasal işlemlere göre aradaki fark; her bir piksele bağımsız olarak işlem yapılması yerine, merkez pikselin yeni değeri hesaplanırken komşu piksellerinde göz önüne alınmasıdır;

Komşuluk ilişkili işlemler aşağıda tanımlanan 4 adımdan oluşur.

- Bir merkez noktası tanımlamak ( $x, y$ )
- Bu merkez noktasının yeni değerini elde etmek için, sadece önceden tanımlanmış komşuların piksellerini de içeren bir operasyon yapmak.
- O noktadaki (Merkez noktadaki) işlemin o operasyonun cevabı olmasına izin vermek;
- İmgedeki her nokta için bu işlemi tekrarlamak (Merkez noktayı kaydırma).



- Bölgesel işlemlerin amacı komşu piksellerin gri tonlarını vurgulamak veya “görünmesine engel olmaktadır. Bu işlem şekilde görülmektedir.
- **Merkezi piksel etrafında tanımlanan komşuluğa “maske” veya “pencere”** denir.
- Bütün resmin işlenmesi için, maskenin adım adım kaydırılarak resmin tamamını taraması gereklidir ( **Bu işlem Merkez noktayı kaydırma işlemidir. Yani bir görüntüdeki her bir piksel için bir tane olacak şekilde yeni komşuluklar oluşturma işlemidir.**). Bu işlem resmin sol üst köşesinden başlar . Yeni griton hesaplandıktan sonra maske (pencere) bir piksel sağ tarafa kaldırılır. Yeniden hesaplanır ve aynı işlemlere devam edilir. Satırın sonuna gelince, işlemleri bir sonraki satırın başından devam edilir.



- a) Belirli bir algoritma ile pencere içerisindeki gritonlar işleme tabi tutulur ve sonuç resimde aynı pozisyon'a atanır.
- b) pencere adım adım kaydırılarak resmin tamamı taranır.

- İlgilenilen pikselin resmin sınırlarına ulaşmadığı açıkça gözükmektedir. Bu yüzden resim bölgesel işlemlerden dolayı küçülür. Genellikle bu küçülme önemli değildir. Fakat kenar piksellere griton verilmemişinden emin olunması gereklidir. Böyle problemlere engel olmak için başlangıçta sonuç resmin bütün piksellerine 0 değeri atanmalıdır.

# KOMŞULUK İLİŞKİSİ İŞLEMLER (2)

Bölgesel resim işlemenin iki önemli kuralı;

1-Sonuç resim, kaynak resimden ayırt edilmeli.

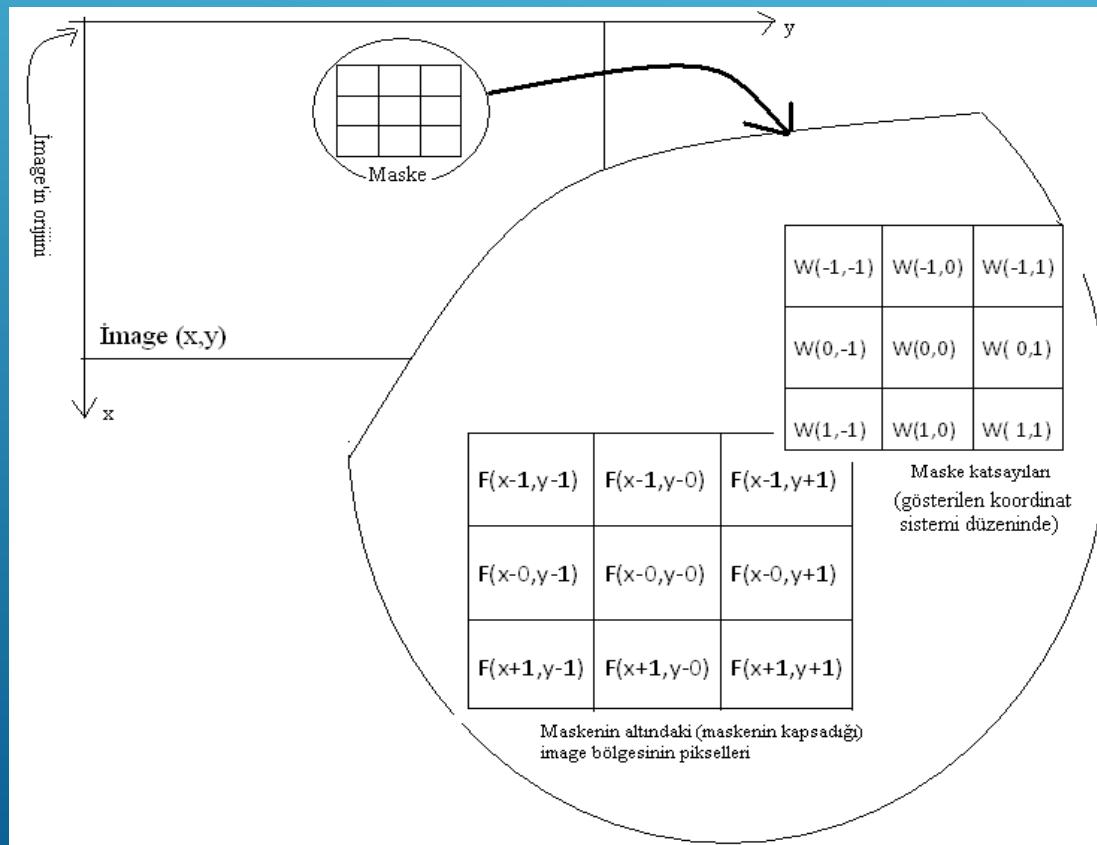
2-Bir işleme başlamadan önce sonuç resmin bütün değerlerine 0 atanmalı.

- ▶ Komşuluk ilişkili işlemler için en uygun yol; uygun bir maske (filtre maskesi-pencere ) oluşturup bu maskeyi tüm görüntü üzerinde hareket ettirerek  $y=f(x)$  fonksiyonunu uygulamaktır. (Bu maske genellikle, kenarları tek sayıda piksellerden oluşturulmuş dikdörtgen şeklidir.)
- ▶ Bu maske ve fonksiyonun kombinasyonuna **Filtreleme** denir. (Filtreleme işlemi çoğu literatürde komşuluk ilişkili İşlemlerin yerine kullanılır)
- ▶ Eğer pikselde yeni gri seviye oluşturma fonksiyonu lineer ise ve maskedeki tüm gri seviye değerleri için lineer uygulanıyorsa bu filtre , **Lineer Uzaysal Filtre (Linear Spatial Filter)** 'dir.
- ▶ Üç klasik bölgeselfiltreleme vardır.
  - 1- **Griton (Gri seviye) düzgünleştirme** (grey level smoothing - Yumuşatma)
  - 2- **Griton farklılıklarını vurgulamak** (emphasizing grey level differences)
  - 3- **Griton geçişlerini keskinleştirmek** (sharpening grey level steps - Keskinleştirme).

# LINEER UZAYSAL FILTRENİN YAPISI

3x3'lük bir maske ve onunla ilgili komşuluk ilişkili bölge altındaki pikseller resimde görülmektedir. Lineer uzaysal filtrelemenin daha iyi açıklanabilmesi için iki önemli kavram vardır. **Korelasyon ve Konvolüsyon.**

- Korelasyon; maske  $w$ 'nin  $f$  görüntü dizisinden geçirilme işlemidir.
- Konvolusyon ; maskenin  $180^\circ$  döndürülüp aynı işlemin yapılması sürecidir.



## İMAGE FILTRELEME

Filtreleme resmin üzerinde bir filtre varmış gibi düşünüp her piksel değerinin yeniden hesaplanmasıdır. Filtreler sayesinde girdi resminden yeni resim değişik efektler verilerek elde edilir.

Filtreleme işlemi aşağıdaki formülle elde edilir. Burada  $m(s,t)$  **filtre matrisidir** (ağırlıklar matrisi-Kernel - Maske) .  $P(i,j)$  filtrelenecek görüntünün matrisidir.  $Y(i,j)$  elde edilecek yeni görüntünün matrisidir ( Bu formül 2 boyutlu korelasyon formülü gibi yorumlanabilir ).

$$y(i,j) = \sum_{s=-1}^1 \sum_{t=-2}^2 m(s,t)p(i+s, j+t).$$

Burada filtre matrisimzi  $3 \times 5$  boyutunda bir matris olup, merkez pikseli, yeni değeri hesaplanacak pikselin üzerine gelecek şekilde konumlandırılıp, denkleme göre yeni değerler hesaplanır. Maske  $p(i,j)$  matrisinin tüm elemanlarını kapsayacak şekilde gezdirilerek, yeni piksel değeri kendisinin ve komşu elemanların oluşturacağı ağırlıklar toplamı şeklinde bulunur.

|            |            |           |           |           |
|------------|------------|-----------|-----------|-----------|
| $m(-1,-2)$ | $m(-1,-1)$ | $m(-1,0)$ | $m(-1,1)$ | $m(-1,2)$ |
| $m(0,-2)$  | $m(0,-1)$  | $m(0,0)$  | $m(0,1)$  | $m(0,2)$  |
| $m(1,-2)$  | $m(1,-1)$  | $m(1,0)$  | $m(1,1)$  | $m(1,2)$  |

$3 \times 5$  Ağırlıklar (Maske - kernel-Filtre) matrisi

|              |              |            |              |              |
|--------------|--------------|------------|--------------|--------------|
| $p(i-1,j-2)$ | $p(i-1,j-1)$ | $p(i-1,j)$ | $p(i-1,j+1)$ | $p(i-1,j+2)$ |
| $p(i,j-2)$   | $p(i,j-1)$   | $p(i,j)$   | $p(i,j+1)$   | $p(i,j+2)$   |
| $p(i+1,j-2)$ | $p(i+1,j-1)$ | $p(i+1,j)$ | $p(i+1,j+1)$ | $p(i+1,j+2)$ |

Filtrelenecek görüntü matrisinin bir kısmı

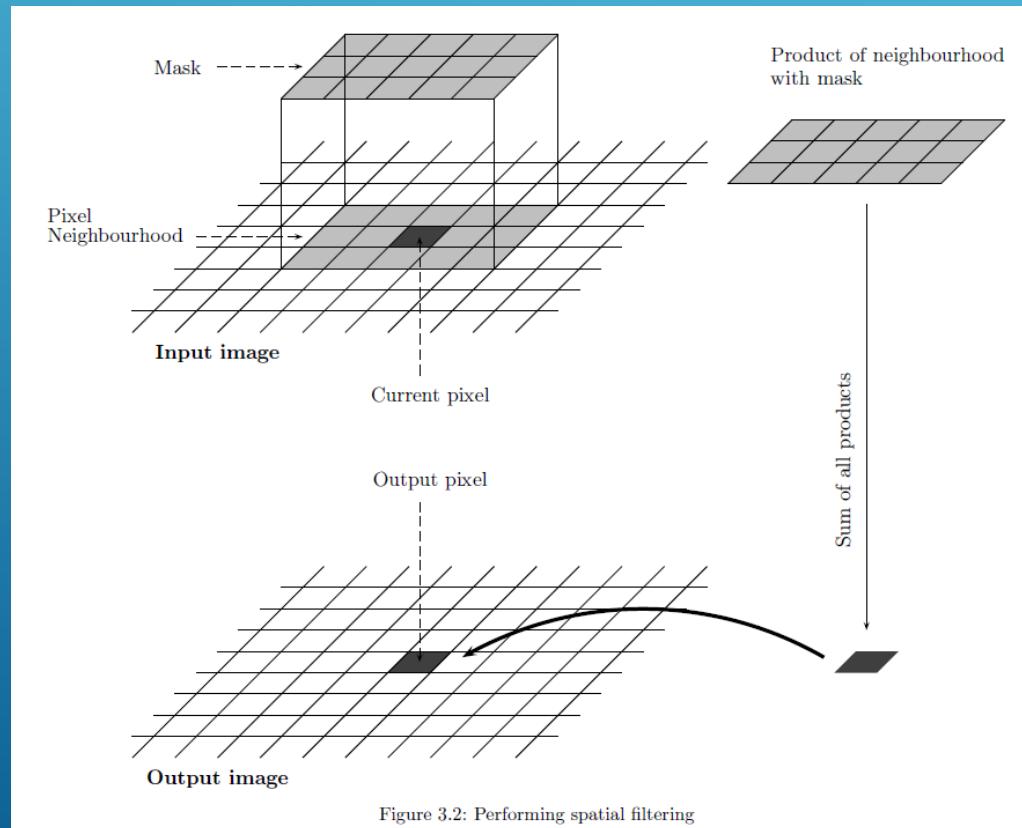
Uzaysal Filtre işlemini gerçekleştirmek için 3 adım gerekir.

1-Maskeyi (Filtreyi) , çalışılacak piksel merkeze gelecek şekilde konumlamak.

2- Maskenin tüm elemanları ile ilgili pikseller komşuluk ilişkisi olan piksellerin çarpılması

3-çarpımların toplanması işlemi.

Bu işlem tüm imgedeki tüm pikseller için gerçekleştirilir.



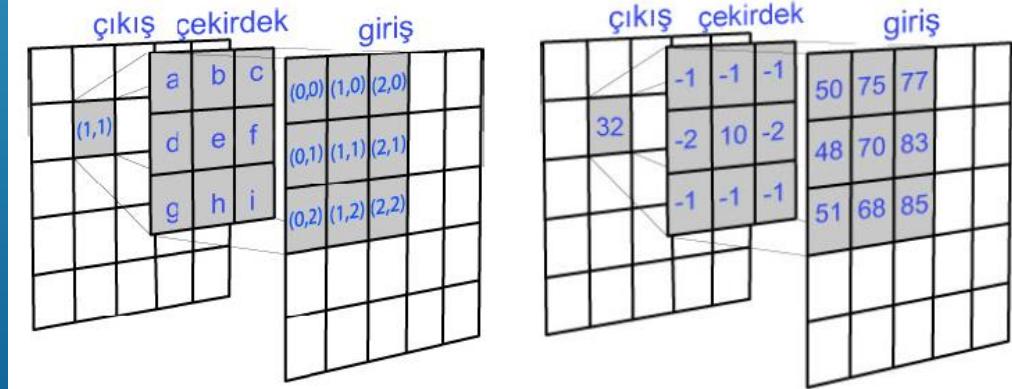
Aşağıdaki denklem iki boyutlu konvolüsyonu ifade ederler. Buna göre  $f(x,y)$  orijinal image,  $h(i,j)$  ağırlık (filtre-Maske-Impulse yanıtı) matrisidir.  $f'$  ( $x,y$ ) ise elde edilen yeni imagedir.

$$f'(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i, j) \times f(x - i, y - j)$$

- ▶ Konvolüsyon; yumuşatma, keskinleştirme, kenar belirleme gibi görüntü işleme fonksiyonlarını gerçekleştirmede çok sık kullanılmaktadır.
- ▶ Konvolüsyonda bir pikselin yeni değeri kendisinin ve çevresindeki piksellerin ağırlıklı ortalaması ile bulunmaktadır. Konvolüsyon şablonu (Filtre matrisi-kernel) uygulamaya göre farklı boyutlarda olabilmekle beraber genelde 3x3 lük bir matristir.

$$imge = \begin{bmatrix} i(0,0) & i(0,1) & i(0,2) \\ i(1,0) & i(1,1) & i(1,2) \\ i(2,0) & i(2,1) & i(2,2) \end{bmatrix} = \begin{bmatrix} 55 & 75 & 77 \\ 48 & 70 & 83 \\ 51 & 68 & 85 \end{bmatrix} \quad \text{çekirdek} = \begin{bmatrix} -1 & -1 & -1 \\ -2 & 10 & -2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$i(1,1) = (-1 \times 55) + (-1 \times 75) + (-1 \times 77) + (-2 \times 48) + (10 \times 70) + (-2 \times 83) + (-1 \times 51) + (-1 \times 68) + (-1 \times 85) = 32 \quad (3.4)$$



Örnek: A giriş görüntüsünün h kerneli ile konvolüsyonu sonucu oluşan çıkış görüntüsünün (2,4) pikselinin değerini hesaplayalım:

Çözüm:

1. h kernelini yatay ve düşey eksende 180 derece döndür.
2. h'ın merkez elemanı A(2,4) noktasına çakışacak şekilde kerneli kaydır.
3. A ve h'daki karşılıklı elemanları çarp ve hepsini toplayarak (2,4) pikselindeki çıkış değerini bul.

$$A = \begin{vmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{vmatrix}$$

$$h = \begin{vmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{vmatrix}$$

# ÖRNEK DEVAM

$$A = \begin{vmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{vmatrix} \quad h = \begin{vmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{vmatrix}$$

Döndürülmüş kernel değerleri

Giriş  
görüntüsünün  
değerleri

|    |    |        |        |        |
|----|----|--------|--------|--------|
| 17 | 24 | $1^2$  | $8^9$  | $15^4$ |
| 23 | 5  | $7^7$  | $14^5$ | $16^3$ |
| 4  | 6  | $13^6$ | $20^1$ | $22^8$ |
| 10 | 12 | 19     | 21     | 3      |
| 11 | 18 | 25     | 2      | 9      |

Cıktısını bulmak istediğimiz piksel

(2,4) pikselinin çıkışı

$$1.2 + 8.9 + 15.4 + 7.7 + 14.5 + 16.3 + 13.6 + 20.1 + 22.8 = 575$$

# BAZI ÖNEMLİ LINEER UZAYSAL FILTRELER

Gaussian Blur, bulanıklaştırma derecesini ayarlamamıza olanak vererek görüntünün hafifçe yumuşatılmasından tüm görüntüyü kalın bir sisle kaplamaya kadar değişen efektler yaratır. Adını, renk değerlerinin değişimini gaussian çanı denilen eğriyle eşlestirmesinden alır.

|   |  |
|---|--|
| Basic 3x3 blurring filter   | $\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |
| $\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |  |
| Gaussian 3x3 blurring filter  |  |
| $\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$   | Gaussian 5x5 blurring filter   |
| $\frac{1}{112} \times \begin{bmatrix} 1 & 2 & 4 & 2 & 1 \\ 2 & 6 & 9 & 6 & 2 \\ 4 & 9 & 16 & 9 & 4 \\ 2 & 6 & 9 & 6 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{bmatrix}$                   |  |

|   |  |  |
|---|--|--|
| <b>Horizontal</b>   | <b>Vertical</b>  | <b>Diagonal</b>  |
| $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$                          | $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$ |
| Prewitt filters   |  |  |
| $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$                          | $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ | $\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$ |
| Sobel filters   |  |  |
| $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ |  |  |
| Roberts filters   |  |  |

# GRİTON DÜZGÜNLEŞTİRME (GREY LEVEL SMOOTHING) LINEER FİLTRELERİ NORMAL ORTALAMA (TEMEL BLURRING (BULANIKLAŞTIRMA)) FİLTRE

Aşağıdaki kaynak resim iki bölgeden oluşmaktadır: Karanlık kısım (griton 1) ve aydınlatık kısım (griton 10). Diğer gritonlar gürültü olarak yorumlanırsa yapılacak iş ya onları yok etmek veya diğer bir deyişle iki bölgeyi elde etmektir. En basit düzgünleştirme metodu ortalama işlemidir. Kaynak resme ortalama işlemi uygulanarak elde edilen sonuç resmi yanda görülmektedir.

Uygulanan maskenin boyutları 3x3'tür. Maske içerisindeki piksellerin gritonları toplanıp 9'a bölünür. Açıkça görüleceği üzere gürültü pikselleri gritonları arzulanan griton değerlerine yaklaşırıldı. Buna rağmen kaynak resimdeki bölgeler arasındaki dik olan geçiş yassılaşmıştır (bulanıklaşmıştır). Pozitif veya negatif etki olarak bunun kıymeti uygulamaya bağlıdır.

1x1+1x1+1x9+1x1+2x1+8x1+1x1+1x1+10x1 = 35 /9 = 3.8 = 4

|   |   |   |   |    |    |    |    |
|---|---|---|---|----|----|----|----|
| 1 | 1 | 1 | 1 | 10 | 10 | 10 | 10 |
| 1 | 1 | 6 | 1 | 8  | 10 | 2  | 10 |
| 1 | 3 | 1 | 1 | 1  | 9  | 10 | 7  |
| 1 | 1 | 1 | 1 | 2  | 1  | 8  | 9  |
| 1 | 1 | 1 | 1 | 1  | 1  | 10 | 10 |
| 1 | 4 | 1 | 2 | 9  | 10 | 2  | 10 |
| 1 | 2 | 1 | 8 | 10 | 10 | 10 | 10 |
| 1 | 1 | 1 | 1 | 10 | 10 | 10 | 10 |

Kaynak resim

|   |   |   |   |   |   |    |   |   |
|---|---|---|---|---|---|----|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 |
| 0 | 2 | 2 | 4 | 7 | 8 | 9  | 0 | 0 |
| 0 | 2 | 2 | 4 | 6 | 8 | 9  | 0 | 0 |
| 0 | 1 | 1 | 4 | 7 | 9 | 10 | 0 | 0 |
| 0 | 1 | 2 | 4 | 7 | 9 | 9  | 0 | 0 |
| 0 | 1 | 2 | 5 | 8 | 9 | 9  | 0 | 0 |
| 0 | 1 | 2 | 5 | 8 | 9 | 9  | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 |

kaynak resme 3x3 ortalama maskesi  
kullanılarak elde edilen sonuç resim

## GAUSSIAN BLURRING FILTRE (AĞIRLIKLI ORTALAMA)

Bu durumda, maske içerisindeki gri tonlar belirli ağırlıklarla (coefficients (kat sayısı) ) çarpılır. Şekilde “Gaussian alçak geçiren滤re” ve normal ortalamanın ağırlıklarını gösterilmektedir. Normal ortalama operatörü (3x3 maske) durumunda, maske içerisindeki gritonlar eşit ağırlıklıdır (yani ağırlıklar 1 değerine eşittir). Maskenin şeklinden dolayı bu maskeyi kullanan filtreye kutu filtre denir.

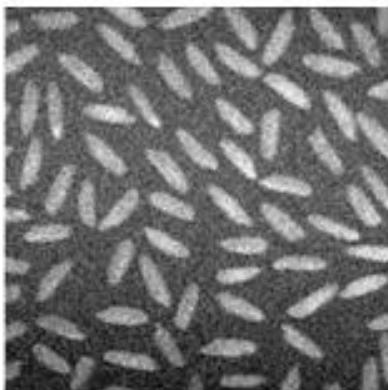
Gaussian alçak geçirenin düzgünleştirme etkisi kutu filtreninkinden biraz daha iyidir. Fakat griton basamaklarının yassılaştırma problemi bunda da mevcuttur.

|     |   |   |   |   |   |   |   |   |   |   |
|-----|---|---|---|---|---|---|---|---|---|---|
| 1/9 | <table border="1"><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1   | 1   | 1 |   |   |   |   |   |   |   |   |
| 1   | 1   | 1 |   |   |   |   |   |   |   |   |
| 1   | 1   | 1 |   |   |   |   |   |   |   |   |

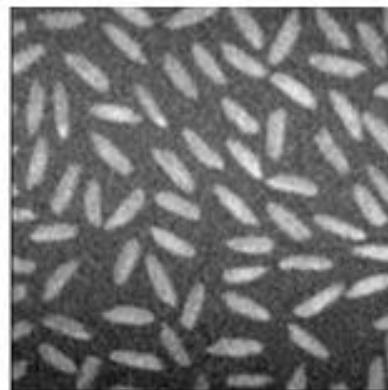
Normal Filtre (kutu filtre)

|      |   |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|---|
| 1/16 | <table border="1"><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>2</td><td>4</td><td>2</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table> | 1 | 2 | 1 | 2 | 4 | 2 | 1 | 2 | 1 |
| 1    | 2   | 1 |   |   |   |   |   |   |   |   |
| 2    | 4   | 2 |   |   |   |   |   |   |   |   |
| 1    | 2   | 1 |   |   |   |   |   |   |   |   |

Gaussian alçak geçiren filtre



Gürültülü resim,



Gaussian alçak geçiren filtre uygulanmış sonuç resim

# LINEER UZAY SAL FILTRELER İÇİN MATLAB FONKSİYONLARI

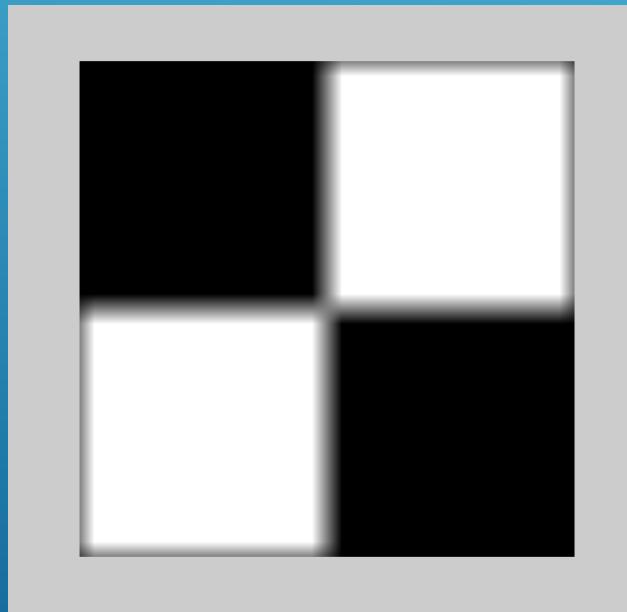
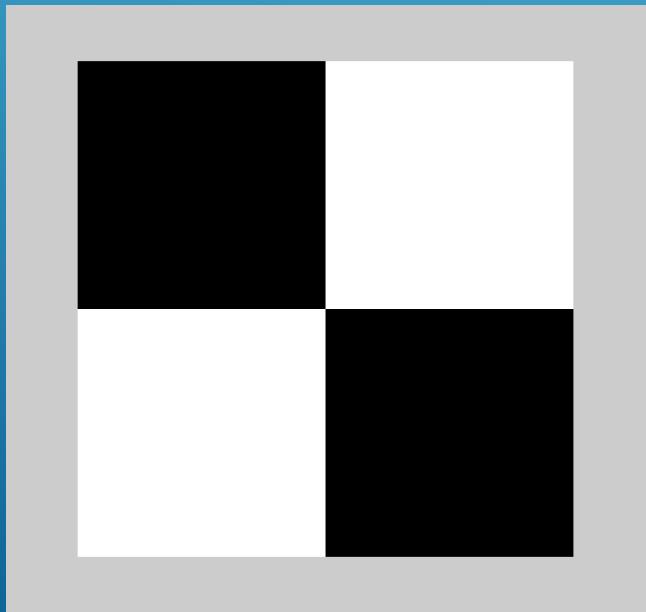
Burada f giriş görüntüsü, w filtre maskesi(kernel), g filtrelenmiş sonuçtur. Diğer parametreler ise tabloda gösterilmektedir.

```
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.   |

# ÖRNEK: ORTALAMA FILTRE

```
>> X=[ZEROS(256,256) ONES(256,256);ONES(256,256) ZEROS(256,256)];  
>> Z=DOUBLE(X);  
>> IMSHOW(Z)  
  
>> X=[ZEROS(256,256) ONES(256,256);ONES(256,256) ZEROS(256,256)];  
>> Z=DOUBLE(X);  
>> W=ONES(31);  
>> GD=IMFILTER(Z,W);  
>> IMSHOW(GD,[])
```



# ÖNEMLİ NOT-1

- ▶ 2 boyutlu görüntülerin filtrelenmesi için diğer bir fonksiyon ise `filter2` fonksiyonudur. Bu fonksiyon aşağıdaki gibi kullanılır. Sonuç matrisinin veri tipi `double`'dır.

>> c = filter2(filter, image, shape)

- ▶ Filter parametresi; istenen filtrelemeyi (filtre tipini),
- ▶ Image parametresi;滤relenecek image'i,
- ▶ Shape parametresi;(same, full,valid olabilir.)
- ▶ same (filtrelenmiş görüntü orijinal görüntü boyutundadır)
- ▶ Full (Filtrelenmiş görüntü paddingden dolayı orijinal imajdan daha büyük boyutludur.)
- ▶ valid (Filtrelenmiş görüntü orijinal görüntülerden küçüklür. Çünkü işlemlerden kenar pikseller etkilenmez).

# DOĞRUSAL OLMAYAN UZAY SAL FILTRELER (NONLINEER SPATİAL FILTERS)

- ▶ Komşuluk ilişkili Doğrusal olmayan filtreleme operasyonları  $m \times n$  boyutlu filtre matrisinin merkez noktasının kaydırılması işlemidir. Bu haliyle komşuluk ilişkili doğrusal uzaysal filtreleme ile aynıdır.
- ▶ Doğrusal olmamanın anlamı her merkez nokta pikseline aynı işlemin uygulanmamasından kaynaklanır. Örneğin; herhangi bir merkez noktası pikselini komşu piksellerdeki en büyük piksel değerine eşitleme işlemi bir nonlineer filtreleme işlemidir.
- ▶ Daha az kullanılan bir tarif ise; maskenin nonlineer olmasıdır.
- ▶ Esas olan merkez piksellerinyeni değerlerinin bulunması için nonlineer işlemlerin yapılmasıdır.

## MİNİMİZE OPERATÖRÜ (MİN OPERATÖRÜ) (NON LINEER FİLTRELEMEDİR)

Griton basamaklarını koruyan çok basit düzgünleştirme operatörü "min operatörü" dür. min operatörü maske içerisindeki minimum gritonu yeni griton olarak verir.

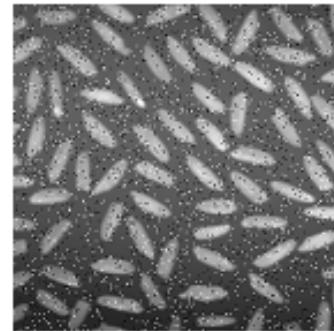
Şekillerden de görüleceği üzere resmin karanlık bölgesi (griton 1) temizlenir. fakat önceki aydınlık bölgede bozulmalar meydana gelir.

orijinal resim

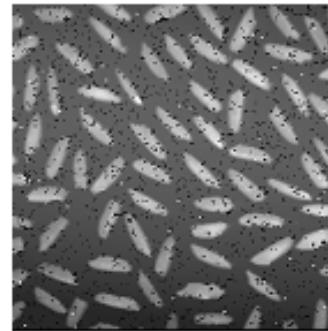
|   |   |   |   |    |    |    |    |
|---|---|---|---|----|----|----|----|
| 1 | 1 | 1 | 1 | 10 | 10 | 10 | 10 |
| 1 | 1 | 6 | 1 | 8  | 10 | 2  | 10 |
| 1 | 3 | 1 | 1 | 9  | 10 | 7  | 10 |
| 1 | 1 | 1 | 2 | 8  | 9  | 10 | 10 |
| 1 | 1 | 1 | 1 | 10 | 10 | 10 | 10 |
| 1 | 4 | 1 | 2 | 9  | 10 | 2  | 10 |
| 1 | 2 | 1 | 8 | 10 | 10 | 10 | 10 |
| 1 | 1 | 1 | 1 | 10 | 10 | 10 | 10 |

min operatörü ile elde edilen resim

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 2 | 2 | 0 |
| 0 | 1 | 1 | 1 | 1 | 2 | 2 | 0 |
| 0 | 1 | 1 | 1 | 1 | 8 | 7 | 0 |
| 0 | 1 | 1 | 1 | 1 | 2 | 2 | 0 |
| 0 | 1 | 1 | 1 | 1 | 2 | 2 | 0 |
| 0 | 1 | 1 | 1 | 1 | 2 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



Kaynak resim

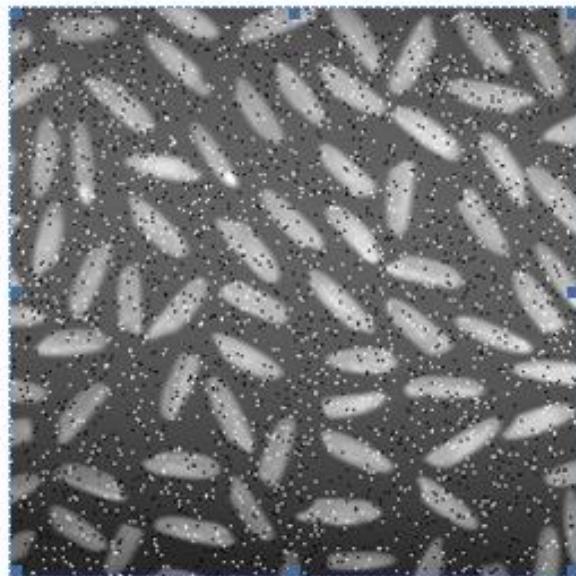


Min operatörü uygulanmış sonuç resim

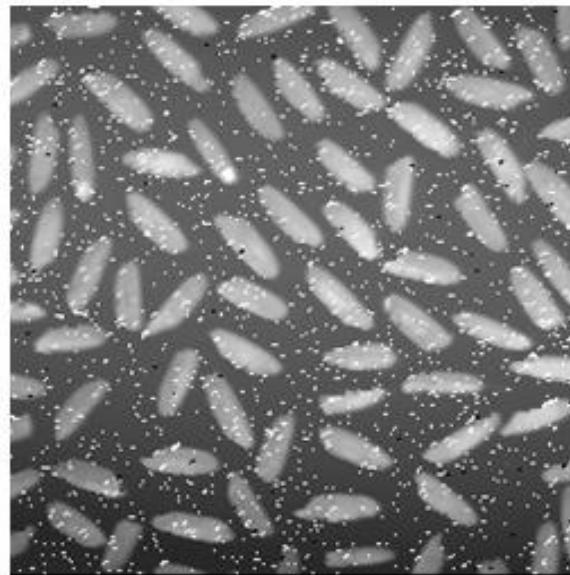
## MAX OPERATÖRÜ: (BASIT BİR NONLİNEER FİLTRELEMEDİR)

Min operatörünü tamamlayıcı “max operatörü”dır; yani maske içerisindeki en yüksek griton değeri, yeni griton değeri olarak sonuç resimde ilgilenilen noktaya atanır.

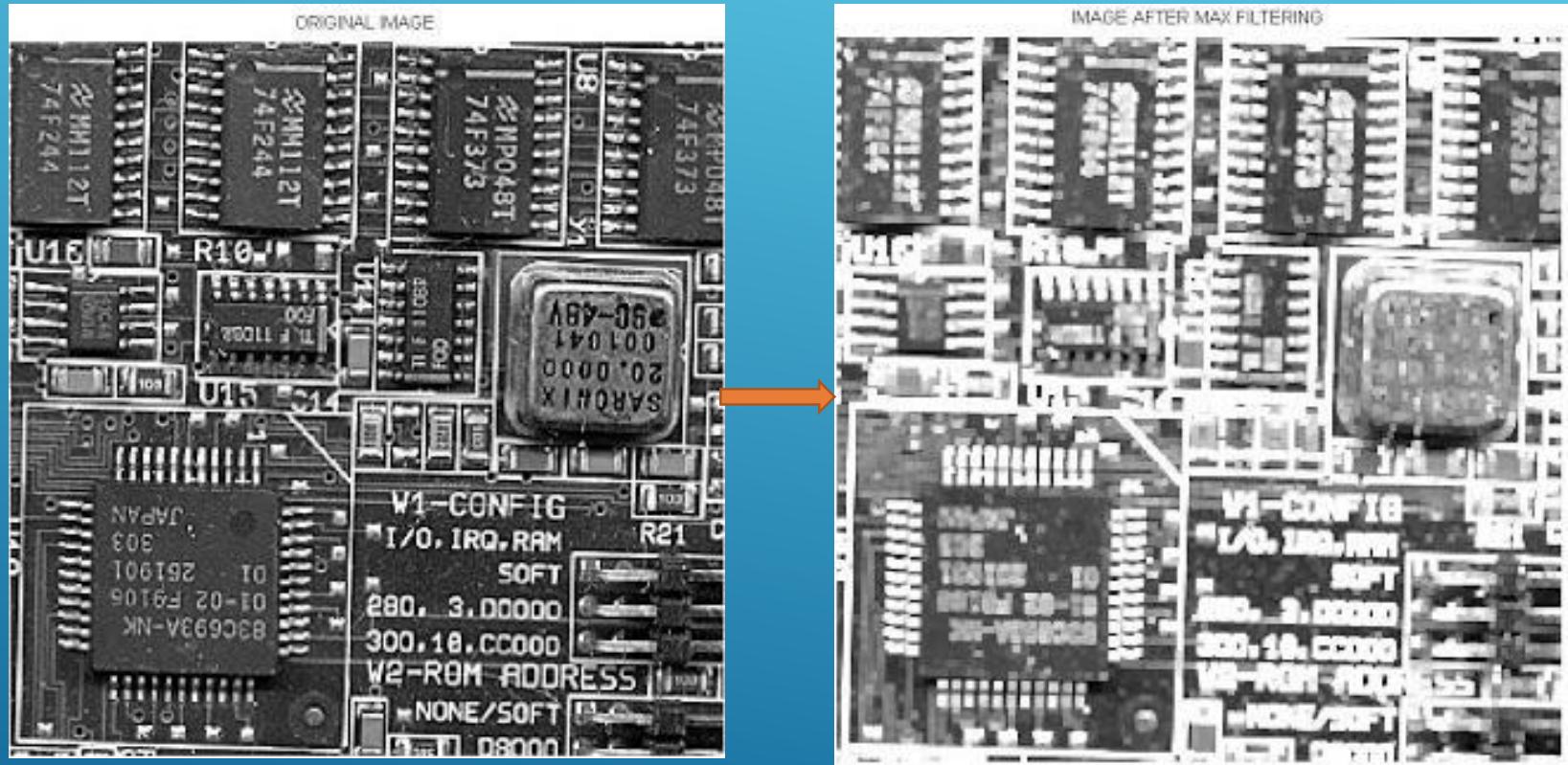
Dolayısıyla sonuç resimde aydınlık bölge temizlenir. Fakat karanlık bölgede bozulmalar meydana gelir. Şekilde Max operatörünün resim üzerindeki etkisini göstermektedir.



Kaynak resim.

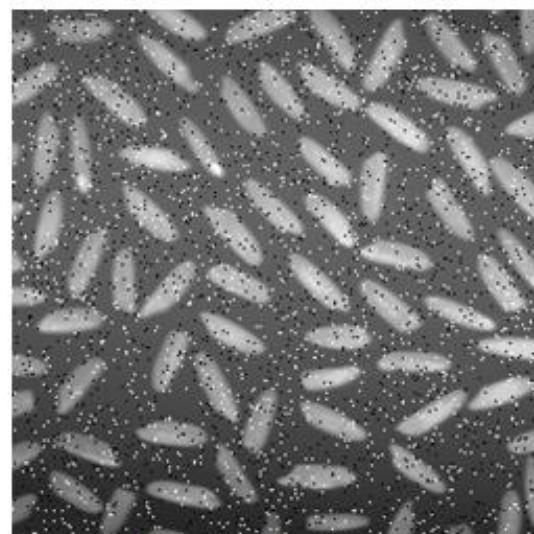


Max operatörü uygulanmış sonuç resim

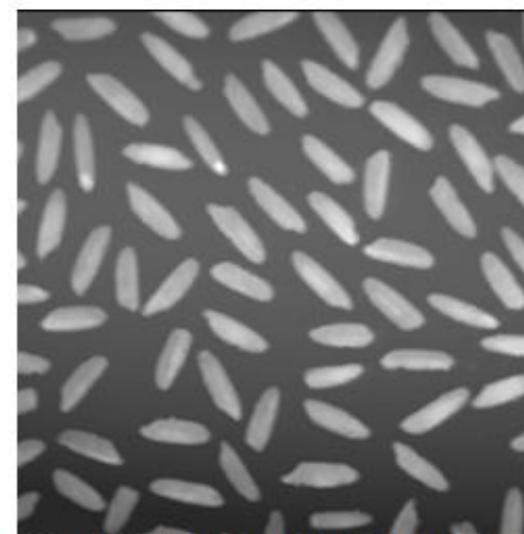


## ORTA-DEĞER (MEDİAN) OPERATÖRÜ (NONLİNEER FILTRELEMEDİR)

- min ve max operatörlerinin fonksiyonlarını birleştiren ve onların yukarıda bahsedilen dezavantajlarına sahip olmayan bir operatöre ihtiyaç vardır. Şekil bu çözümü göstermektedir.
- Orta-değer operatöründeki fikir, maske içerisindeki gritonları değerlerine göre sıralamaktır. Bu sıralamada ortada bulunan değer sonuç resmin ilgilenilen pikseli için kullanılır. Bu strateji yüksek ve alçak gritonların pikelerini, griton bölgeleri ayıran griton basamaklarını yassılaştırmadan yok eder. Orta-değer operatörünün dezavantajı; komşu piksel gritonlarının sıralamasından dolayı hesaplama zamanın yüksek olmasıdır.
- Orta-değer operatörünün resim üzerindeki etkisini göstermektedir. Resimden de görüleceği üzere, gürültülü resim tamamen onarılmıştır.



Kaynak resim



Orta-değer operatörü uygulanmış sonuç resim

## K-ENYAKIN-KOMŞU (NONLİNEER FİLTRELEMEDİR:)

Diğer bir kenar korumalı düzgünleştirme metodu “k-enyakin-komşu” metodudur. Bu, maskenin bütün pikselleri üzerinde kullanılmayan bir normal ortalama operatörür (kutu filtre).

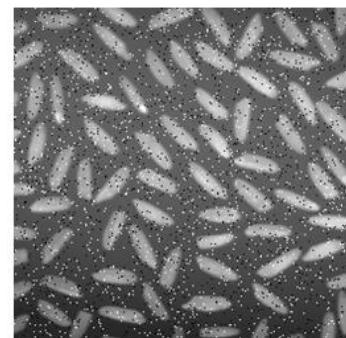
Sadece gritonları ilgilenilen pikselin gritonuna en yakın k tane piksel üzerine uygulanır. Şekilde  $3 \times 3$  enyakin-komşu operatörünün  $k = 3$  ile (ilgilenilen piksel dahil) kaynak resme uygulanması ile elde edilen sonucu göstermektedir. Ortalamanın sadece 3 griton kullanılarak hesaplanmasıdan dolayı, düzgünleştirme etkisi “orta-değer” operatöründen azdır.

Genellikle  $k$ , maskedeki piksel sayısının yarısından fazla olmalıdır. Şekil, k-enyakin-komşu operatörünün resim üzerindeki uygulamasını göstermektedir.

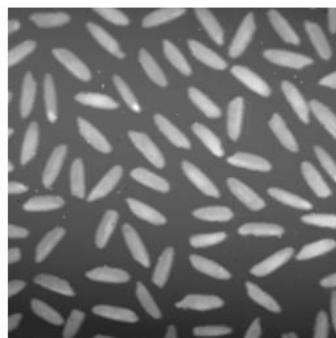
| orijinal resim |   |   |   |    |    |    |    |  |
|----------------|---|---|---|----|----|----|----|--|
| 1              | 1 | 1 | 1 | 10 | 10 | 10 | 10 |  |
| 1              | 1 | 6 | 1 | 8  | 10 | 2  | 10 |  |
| 1              | 3 | 1 | 1 | 9  | 10 | 7  | 10 |  |
| 1              | 1 | 1 | 2 | 8  | 9  | 10 | 10 |  |
| 1              | 1 | 1 | 1 | 10 | 10 | 10 | 10 |  |
| 1              | 4 | 1 | 2 | 9  | 10 | 2  | 10 |  |
| 1              | 2 | 1 | 8 | 10 | 10 | 10 | 10 |  |
| 1              | 1 | 1 | 1 | 10 | 10 | 10 | 10 |  |

| 3x3 en yakın komşu operatörü uygulananmiş sonuç resim ( $k=3$ ) |   |   |   |    |    |    |   |   |
|---|---|---|---|----|----|----|---|---|
| 0   | 0 | 0 | 0 | 0  | 0  | 0  | 0 | 0 |
| 0   | 1 | 3 | 1 | 9  | 10 | 6  | 0 |   |
| 0   | 2 | 1 | 1 | 9  | 10 | 9  | 0 |   |
| 0   | 1 | 1 | 1 | 9  | 9  | 10 | 0 |   |
| 0   | 1 | 1 | 1 | 10 | 10 | 10 | 0 |   |
| 0   | 2 | 1 | 1 | 10 | 10 | 7  | 0 |   |
| 0   | 1 | 1 | 9 | 10 | 10 | 10 | 0 |   |
| 0   | 0 | 0 | 0 | 0  | 0  | 0  | 0 |   |



Kaynak resim



k-enyakin-komşu operatörü uygulananmış sonuç resim

# NONLİNEER UZAY SAL FILTRE FONKSİYONLARI

- ▶ Matlab image processing toolbox'ta(IMT) nonlinear filtreleme için iki önemli fonksiyon vardır:
  - nlfilter
  - colfilt

Her ikiside doğrudan 2 boyutta çalışır.

colfilt; datayı sütun formunda organize eder. Daha fazla hafızaya ihtiyaç gösterir. nfilter fonksiyonundan daha hızlı çalışır.

Birçok görüntü işleme uygulamalarında hız önemli bir faktördür. Nonlineer uzaysal filtreleme işlemlerinde daha çok colfilt kullanılır.

```
g=colfilt(f, [m n], 'sliding', @fun, parameters)
```

m ve n filtre boyutudur.

'sliding', f giriş görüntüsünde pikselden piksele mxn'lik filtrenin kaydırma işlemini belirtir.

@fun; references a function, which we denote arbitrarily as fun,

parameters indicates parameters (separated by commas) that may be required by function fun.

The symbol @ is called a *function handle*, a MATLAB *data type* that contains information used in referencing a function.

**fp=padarray(f, [r c], method, direction)**

- Lineer filtrelerde söylendiği gibi, görüntünün kenarlarında filtrenin uygulanabilmesi için görüntü datasına genellikle 0'lar eklenmeliydi.
- Colfilt kullanılacaksa, filtrelemeden önce muhakkak giriş datasına yeterli miktarda ekleme yapılmalıdır. Bunun için **padarray** fonksiyonu kullanılır. Burada f giriş görüntüsü, fp paddlenmiş(eklenmiş) görüntü,[r c], f'ye eklenecek pad'in boyutunu verir.  
method ve direction yeni tablodaki yerleşimi bildirir.

| Options          | Description   |
|------------------|---|
| <i>Method</i>    |   |
| 'symmetric'      | The size of the image is extended by mirror-reflecting it across its border.                      |
| 'replicate'      | The size of the image is extended by replicating the values in its outer border.                  |
| 'circular'       | The size of the image is extended by treating the image as one period of a 2-D periodic function. |
| <i>Direction</i> |   |
| 'pre'            | Pad before the first element of each dimension.   |
| 'post'           | Pad after the last element of each dimension.   |
| 'both'           | Pad before the first element and after the last element of each dimension. This is the default.   |

```
>> A=[1 2 3; 4 5 6]
```

A =

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

```
>> B=padarray(A,[2 3],0,'post')
```

B =

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

```
>> B=padarray(A,[2 3],0,'post')
```

B =

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

```
>> B=padarray(A,[2 3],0,'pre')
```

B =

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

```
A = [ 1 2; 3 4];  
B = [ 5 6; 7 8];  
C = cat(3,A,B)  
C(:,:,1) =
```

```
1 2  
3 4
```

```
C(:,:,2) =  
5 6  
7 8
```

```
D = padarray(C,[3 3])
```

```
D(:,:,1) ==  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 1 2 0 0 0  
0 0 0 3 4 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0  
  
D(:,:,2) ==  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 5 6 0 0 0  
0 0 0 7 8 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0
```

```
a = [ 1 2 3 4 ];  
b = padarray(a,[0  
3],'symmetric','pre')  
b =
```

```
3 2 1 1 2 3 4
```

```
>>A = [ 1 2; 3 4];  
>>B = padarray(A,[3  
2],'replicate','post')
```

```
B =  
1 2 2 2  
3 4 4 4  
3 4 4 4  
3 4 4 4  
3 4 4 4
```

# ÖNEMLİ NOT: FREKANS BİLEŞENİ KAVRAMI

- ▶ Bir görüntüdeki filtrenin etkisini standart bir formda ifade edebilmek için en önemli kriterlerden birisi görüntünün frekans bileşenleridir.
- ▶ Kabaca bir görüntünün frekans bileşenleri; mesafeye göre gri seviye değişiminin miktarı olarak ifade edilir.
- ▶ Görüntünün Yüksek frekanslı bileşenleri; küçük mesafelerde piksellerin gri değerlerin büyük miktarda değişikliklerini karakterize eder. Yüksek frekans bileşenlerine örnek olarak; resmin kenarları (en büyük gri seviye değişimleri kenarlarda olur) ve gürültüler verilebilir.
- ▶ Görüntünün Düşük frekanslı bileşenleri: Resimdeki piksellerin gri değerlerinin mesafeye göre pek az değiştiği görüntü parçaları ile karakterize edilir. Bunlara örnek arka planlar (gri seviyeleri çok az değişen yüzeyler), cilt dokuları verilebilir.
- ▶ Bu tanımlamalar göre filtreleri;
- ▶ Yüksek geçiren Filtreler: Yüksek frekanslı bileşenleri geçirir. Düşük frekanslı bileşenleri yok eder. Örnek kenar çıkarma işlemleri.
- ▶ Alçak geçiren Filtreler: Görüntüdeki Alçak frekans bileşenlerini geçirir veya kuvvetlendirir. Yüksek frekans bileşenlerin yok eder. Örneğin 3x3Lük averagefiltresi bir alçak geçiren filtredir. Çünkü resmin kısa mesafede değişen (Özellikle kenarlar - Gürültüler) piksellerini yok eder. Belirli mesafede az değişen piksellerini muhafaza eder. Dolayısıyla resim bulanıklaşır (Blurring).

## FSPECIAL FONKSİYONU

- Lineer uzaysal filtrelemenin diğer bir yolu da özel lineer filtrelerin yaratılabilmesidir. Buna göre, oluşturulacak lineer filtre maskesi için aşağıdaki deyim kullanılır.

```
>> F = fspecial ('type', parameters)
```

Burada 'type', özel filtre tipini belirtir. Parameters ise filtreyi tanımlayan değerlerdir. Bu şekilde kullanabileceğimiz hazır özel filtreler vardır. Bunlar tabloda görülmektedir.

# ÇOK KULLANILAN BAZI ÖZEL LİNEER FILTRELERİN FSPECIAL İLE TANIMI

**Average:** Ağırlıklı ortalama filtredir. Default boyutu 3x3 boyutlu elemanlar 1 olan bir matristir. Eğer r,c yerine tek bir sayı yazılırsa kare matris filtre olur.

**Disk:** Bir dairesel ortalama filtredir. r dairenin yarıçapıdır. R'nin default değeri 5'tir.

**Gaussian:** Alçak geçiren bir filtredir. Standart sapması pozitif olan ve boyutu rxc olan bir filtredir. Rxc 'eğer tek bir değer ise kare matris filtredir.

**Laplacian ve log**, 2.türev işlemeye dayanan yüksek filtre tipleridir.

| Type        | Syntax and Parameters   |
|-------------|---|
| 'average'   | <code>fspecial('average', [r c])</code> . A rectangular averaging filter of size $r \times c$ . The default is $3 \times 3$ . A single number instead of <code>[r c]</code> 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 $3 \times 3$ and 0.5. A single number instead of <code>[r c]</code> specifies a square filter.       |
| 'laplacian' | <code>fspecial('laplacian', alpha)</code> . A $3 \times 3$ 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 $5 \times 5$ and 0.5. A single number instead of <code>[r c]</code> specifies a square filter. |

## ÇOK KULLANILAN BAZI ÖZEL LİNEER FILTRELERİN FSPECIAL İLE TANIMI (2)

Prewit ve sobel Filtreleri yüksek geçiren basit tip filtrelerdir. Görüntüde kenar çıkarma, kenar iyileştirme işlemlerinde kullanılabilirler.

| Type      | Syntax and Parameters   |
|-----------|---|
| '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 <code>len</code> pixels. The direction of motion is <code>theta</code> , 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 $3 \times 3$ Prewitt mask, <code>wv</code> , that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: <code>wh=wv'</code> .  |
| 'sobel'   | <code>fspecial('sobel')</code> . Outputs a $3 \times 3$ Sobel mask, <code>sv</code> , that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: <code>sh=sv'</code> .  |
| 'unsharp' | <code>fspecial('unsharp', alpha)</code> . Outputs a $3 \times 3$ unsharp filter. Parameter <code>alpha</code> controls the shape; it must be greater than or equal to 0 and less than or equal to 1.0; the default is 0.2.  |

```
>> fspecial('average',[3 3])
```

```
ans =
```

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

```
>> a=fspecial('disk',3)
```

```
a =
```

```
0 0.0003 0.0110 0.0172 0.0110 0.0003 0  
0.0003 0.0245 0.0354 0.0354 0.0354 0.0245 0.0003  
0.0110 0.0354 0.0354 0.0354 0.0354 0.0354 0.0110  
0.0172 0.0354 0.0354 0.0354 0.0354 0.0354 0.0172  
0.0110 0.0354 0.0354 0.0354 0.0354 0.0354 0.0110  
0.0003 0.0245 0.0354 0.0354 0.0354 0.0245 0.0003  
0 0.0003 0.0110 0.0172 0.0110 0.0003 0
```

```
>> fspecial('gaussian',[3,3],0.5)
```

```
ans =
```

```
0.0113 0.0838 0.0113  
0.0838 0.6193 0.0838  
0.0113 0.0838 0.0113
```

```
>> a=fspecial('prewitt')
```

```
a =
```

```
1 1 1  
0 0 0  
-1 -1 -1
```

```
>> a=fspecial('sobel')
```

```
a =
```

```
1 2 1  
0 0 0  
-1 -2 -1
```

```
>> a=fspecial('motion',9,90)
```

```
a =
```

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

# ÇOK KULLANILAN ÖZEL NONLİNEER UZAY SAL FILTRELER

```
g=ordfilt2(f,order, domain)
```

Matlabda; özel Nonlineer uzaysal filtre işlemi genel olarak `ordfilt2` fonksiyonuyla gerçekleştirilir. Bu filtre dereceli (sıralı)- istatistiksel filtrelere (rank -order- sayı sıralı-Filtre) işlemidir.

Bu fonksiyon 3 parametreye ihtiyaç gösterir.

**F:** filtrelenecek görüntü matrisi,

**order;** rakamsal bir değerdir. Maske içindeki en küçük veya en büyük n.değere göre işlem yapmayı ifade eder.

**Domain;** maske boyutunun ve elemanlarının tanımı'dır.

En çok bilinenleri, min filtre, Maks.Filtre ve Median Filtredir.

```
g=ordfilt2(f, 1, ones(m,n))
```

İşlemi, mxn boyutlu ve elemanları 1 olan bir maske kullanılarak, f görüntüsünü **min filtreye** tabi tutmaktadır. g'nin ilgili elemanını değeri maskenin kapsadığı komşu elemanlar içerisinde en küçük değerde olmalıdır(order=1 olduğu için).

# MİN.FİLTRELEME ÖRNEĞİ

```
>> 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 |

```
>> d=uint8(B)
```

d =

|     |     |    |     |     |
|-----|-----|----|-----|-----|
| 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,1,ones(3,3))
```

c =

|   |    |    |    |   |
|---|----|----|----|---|
| 0 | 0  | 0  | 0  | 0 |
| 0 | 2  | 14 | 26 | 0 |
| 0 | 10 | 56 | 35 | 0 |
| 0 | 0  | 0  | 0  | 0 |
| 0 | 0  | 0  | 0  | 0 |

```
>> c=ordfilt2(d,2,ones(3,3))
```

c =

|   |    |    |    |   |
|---|----|----|----|---|
| 0 | 0  | 0  | 0  | 0 |
| 0 | 14 | 26 | 37 | 0 |
| 0 | 34 | 56 | 45 | 0 |
| 0 | 10 | 29 | 35 | 0 |
| 0 | 0  | 0  | 0  | 0 |

```
>> c=ordfilt2(d,4,ones(3,3))
```

c =

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 2  | 14 | 26 | 0  |
| 2  | 34 | 56 | 56 | 37 |
| 10 | 56 | 67 | 56 | 35 |
| 10 | 56 | 56 | 56 | 35 |
| 0  | 0  | 0  | 0  | 0  |

## MAKS.FİLTRELEME ÖRNEĞİ G=ORDFİLT2(F,M\*N,DOMAİN)

Maks.filtre maskenin kapsadığı elemanlar içindeki en büyük değerli elemanı yeni elemanla değiştirmek olduğuna göre, minfiltrede ( $m \times n$ ) değerini order kısmına yazmak yeterlidir.

```
>> 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

>> d=uint8(B)
d =
      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,9,ones(3,3))
c =
      90      90      90     100     100
     200     200     255     255     255
     200     200     255     255     255
     200     200     255     255     255
     100     100    178     200     200
```

## MEDİAN FİLTRELEMEYE ÖRNEK

G=ORDFILTER2(F, MEDİAN (1:M\*N), ONES(M,N))

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

```
>> B=[2 14 26 37 99;34 90 67 56 100;100 200 200 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
```

# MEDİAN 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 mxn 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-yansımalı simetriklilik 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.

MEDİAN FİLTRE İÇİN BAŞKA BİR FONKSİYON

G=MEDFILT2(F, [M N], PADOPT)

## ÖRNEK:medfilt2

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



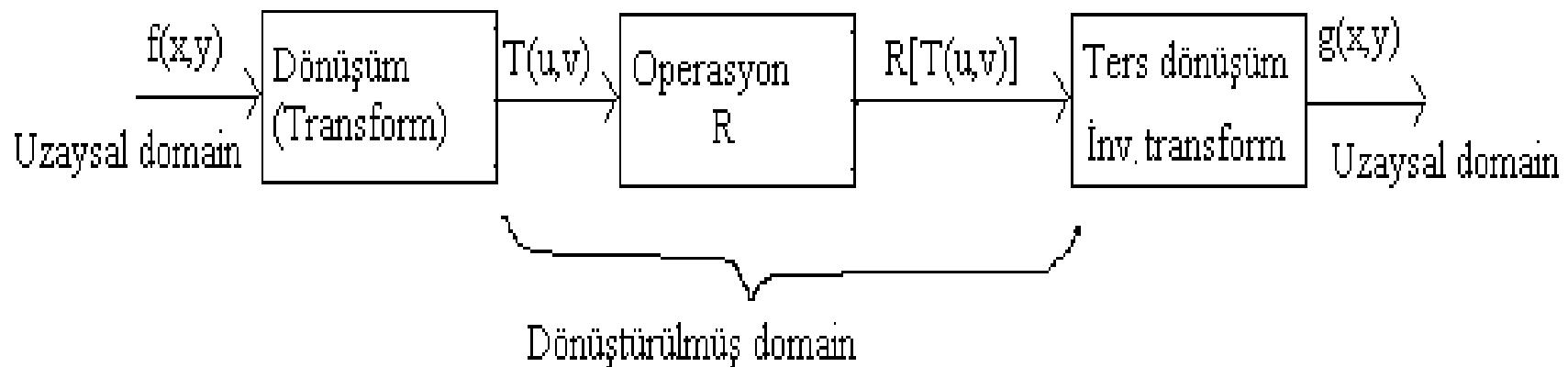
# Frekans domain’inde İşlemler

# Domain Dönüşümü

Dönüşüm, bir sinyalin, başka parametrelerle ifade edilmesi şeklinde düşünülebilir.

Ters dönüşüm ise , sinyalin ilk halindeki parametrelerle ifade edilebilir şekilde geri döndürülme işlemidir.

Fourier transformasyonları (dönüştürücüler) bir sinyalin frekans domainine dönüşümünü sağlar.



# Frekans uzayı

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

# Fourier Dönüşümü

- Bu dönüşüm, görüntü işlemenin çok önemli konularından biridir. Uzaysal domainde başarılması zor işlemleri, frekans domain’inde başaracak yapıda olan bu dönüşüm,  
*“görüntüyü oluşturan frekans bileşenlerini biribirinden ayırt edebildiği için değişik derecelerde alçak ve yüksek geçiren filtreleme işlemleri”*  
kolaylıkla başarılabilir.
- Önce tek boyutlu sonra iki boyutlu Fourier dönüşümünü kısaca hatırlayalım;

# Temel Bilgi notu(Fourier analizi)

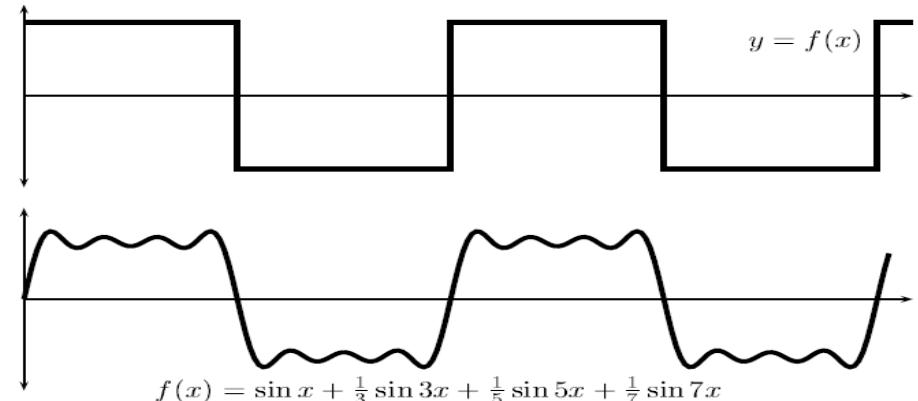
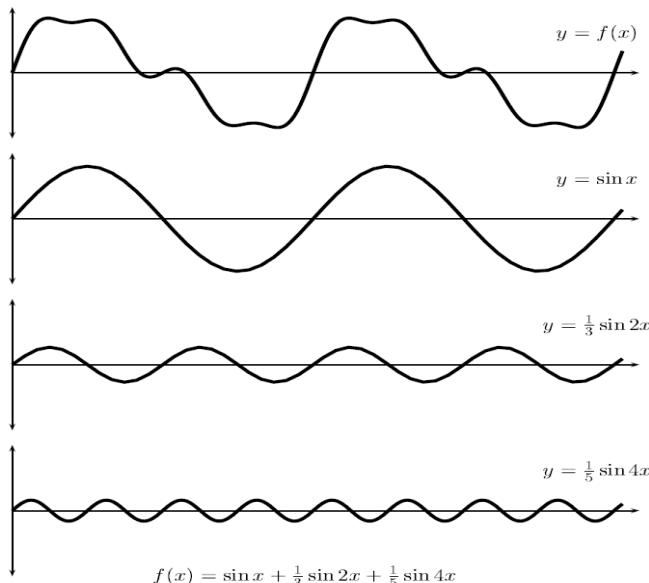
Periodik ve sonlu değer alabilen her fonksiyon, değişik frekanslarda titreşen sinüs veya cosinüslü bileşenlerin toplamından oluşur.

$$f(x) = f(x + T) \quad (-\infty < x < \infty)$$

$$f(x) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty} a_k \cos(\omega kx) + b_k \sin(\omega kx) \quad (\omega = \frac{2\pi}{T})$$

$$a_k = \frac{2}{T} \int_0^T f(x) \cos(\omega kx) dx$$

$$b_k = \frac{2}{T} \int_0^T f(x) \sin(\omega kx) dx$$



# Tek boyutlu ayrık Fourier dönüşümü (DFT)

Herhangi bir fonksiyon ayrıklaştırıldığında sonlu sayıda elemanlı bir dizi şeklinde ifade edilir. Örneğin bir kare dalganın ayrık zamanlı dizi şeklinde ifadesi; 1 1 1 1 -1 -1 -1 -1 şeklinde olabilir.

$$\mathbf{f} = [f_0, f_1, f_2, \dots, f_{N-1}]$$

Eleman sayısı N olan bir ayrık  $\mathbf{f}$  fonksiyonun, ayrık Fourier katsayıları dizisi aşağıda tanımlansın.

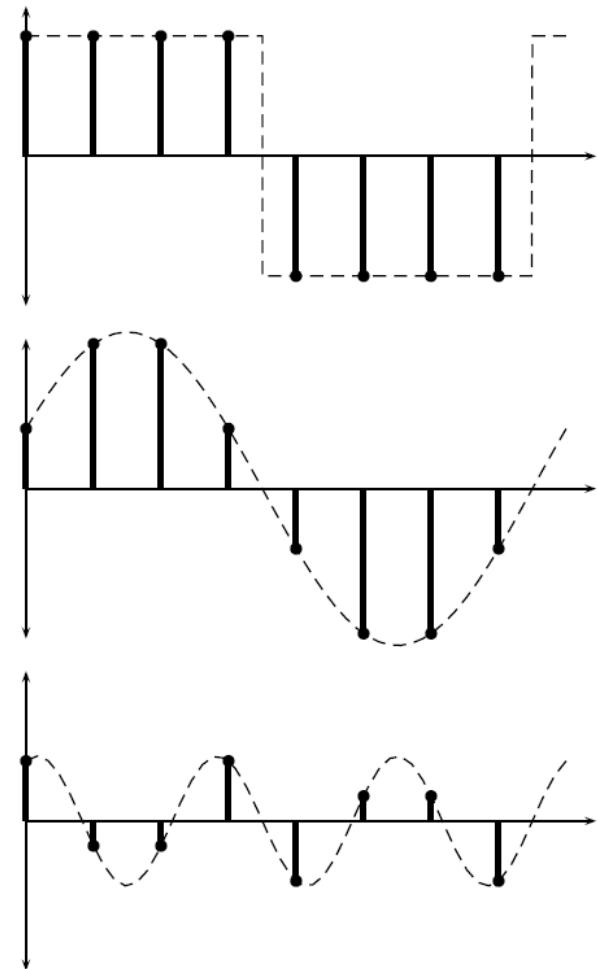
$$\mathbf{F} = [F_0, F_1, F_2, \dots, F_{N-1}]$$

Burada;  $u$ .ayırık Fourier bileşeninin katsayısı aşağıdaki bağıntıyla hesaplanır (DFT).

$$F_u = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cdot \exp \left[ -2\pi i \frac{xu}{N} \right]$$

Aynı şekilde, Fourier katsayılarından dizi elemanını elde etmek için ters Fourier dönüşümü yapılır (IDFT).

$$f(x) = \sum_{u=0}^{N-1} F(u) \cdot \exp \left[ 2\pi i \frac{xu}{N} \right]$$



## Frekans Uzayı – Ayrık Fourier Dönüşümü

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

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

## Frekans Uzayı – Ayrık Fourier Dönüşümü

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

- $F(u)$  yu bulmak için:

- Bu işlem için  $M^2$  çarpma ve toplama gereklidir.

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

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

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

## Frekans Uzayı – Ayrık Fourier Dönüşümü

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

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

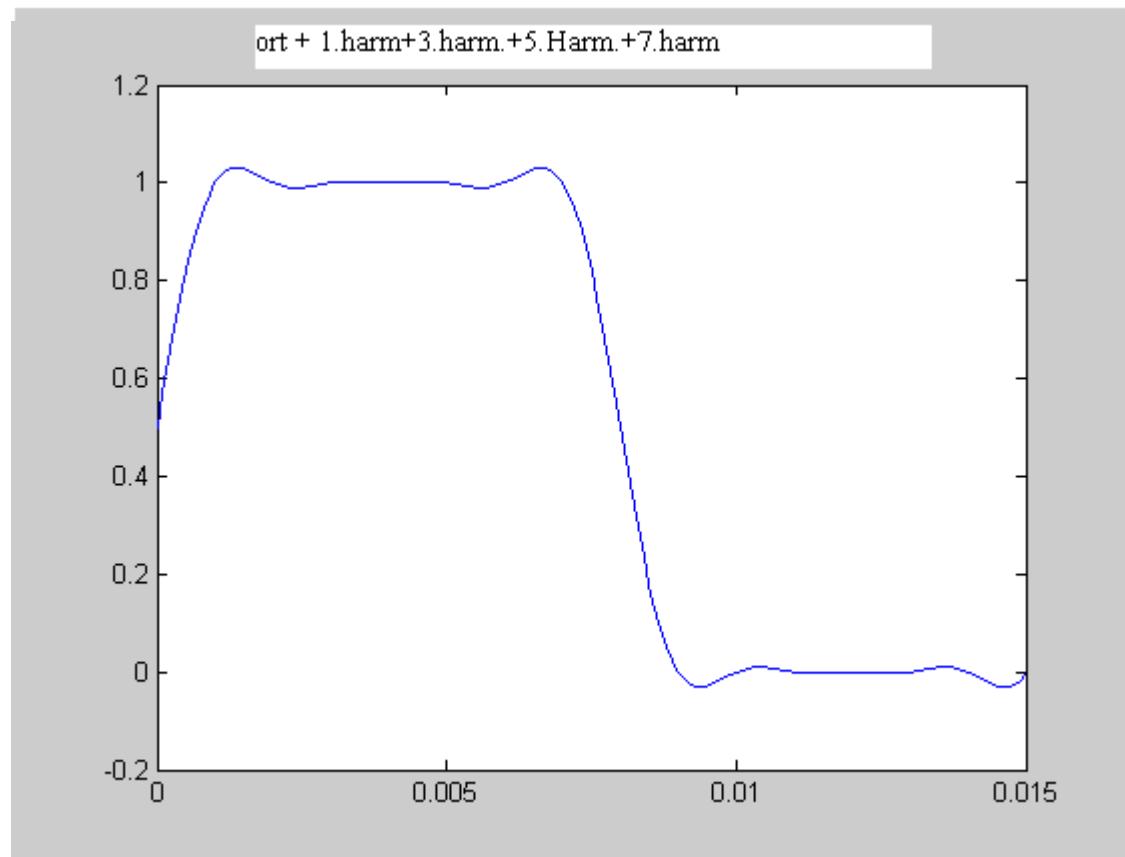
$$\left. \begin{array}{c} F(0) \\ F(1) \\ \vdots \\ \vdots \\ F(M) \end{array} \right\}$$

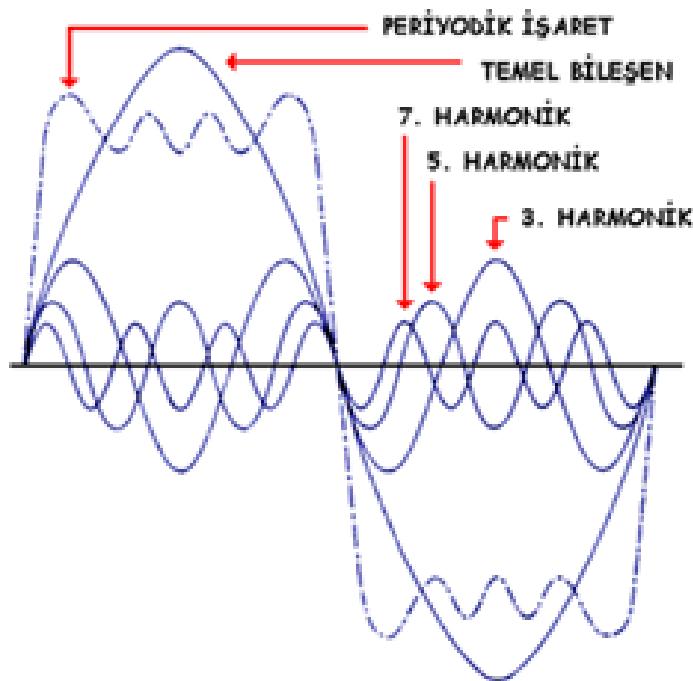
Dönüşümün  
frekans  
bileşenleri

$$F(u) = |F(u)| e^{-j\varphi(u)}$$
$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \quad \text{Genlik}$$
$$\varphi(u) = \tan^{-1} \left[ \frac{I(u)}{R(u)} \right] \quad \text{Faz açısı}$$

# Tek boyutlu DFT işlemiyle ilgili....

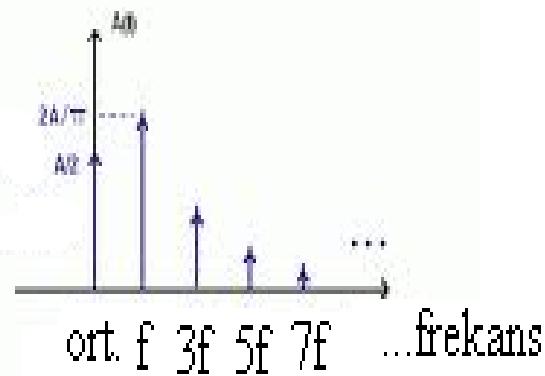
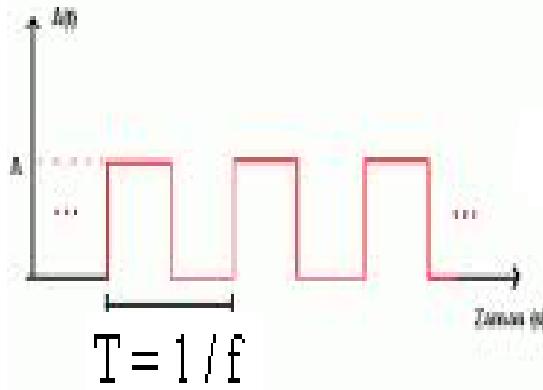
$$F_{(n)} = (1, 1, 1, 1, 0, 0, 0, 0)$$





Zaman üzerinde kare şeklinde değişen sinyal

Bu kare dalganın Harmonik bileşen katsayıları  
(Max.genlik ) değerleri)



# Fast Fourier Dönüşümü

- DFT, ayrık fonksiyonların fourier dönüşümünü yapmak için çok önemli bir yapı olmakla beraber, dizi eleman sayısının çok fazla olduğu dönüşümler için uzun zaman harcamaktadır. Özellikle resim işleme gibi data sayısı fazla olan işlemler için yavaştır.
- Bunun yerine FFT( Hızlı fourier dönüşümü) ile bileşen katsayıları bulunur. Burada dikkat edilecek konunun dizinin  $2^{\text{n}}$  nin kuvvetleri sayısında elemandan oluşturulmasıdır.

# 2-D (2 Boyutlu) DFT

2 boyutlu DFT; Matris şeklinde bir diziyi giriş kabul edip, yine matris şeklinde bir çıkış elde eder. Farz edelim ki;

$$f(x, y), \quad x = 0, 1, 2, \dots, M - 1 \text{ ve } y = 0, 1, 2, \dots, N - 1,$$

$M \times N$  boyutunda bir görüntü olsun. Bu  $f$  görüntüsünün 2-D DFT'sindeki  $F(u, v)$  katsayıları aşağıdaki formülden hesaplanır.  $F$ 'nin Fourier dönüşümü;

$$F = \mathcal{F}(f)$$

Ters fourier dönüşümü ise;

$$f = \mathcal{F}^{-1}(F)$$

2-D DFT

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

for  $u = 0, 1, 2, \dots, M - 1$  and  $v = 0, 1, 2, \dots, N - 1$ .

2-D IDFT

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

for  $x = 0, 1, 2, \dots, M - 1$  and  $y = 0, 1, 2, \dots, N - 1$ .

## 2-D DFT

Eğer;  $f(x, y)$  dizisi reel ise, bunun fourier dönüşümü genellikle kompleks sayılardır. Dönüşürtülmüş 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 imaginer 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

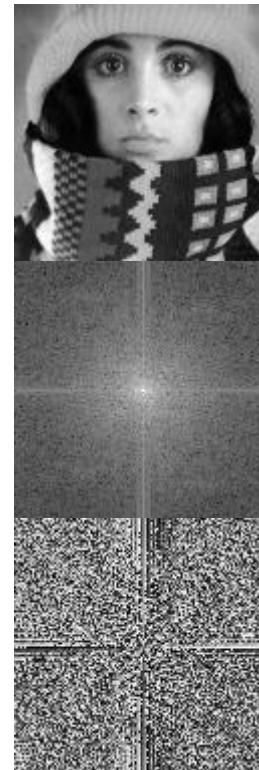
$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)}$$

$F(u, v)$  nin faz açısı

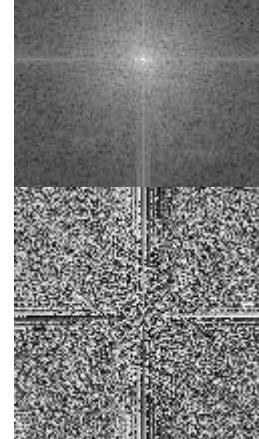
$$\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$$

Güç Spektrumu

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



Orijinal

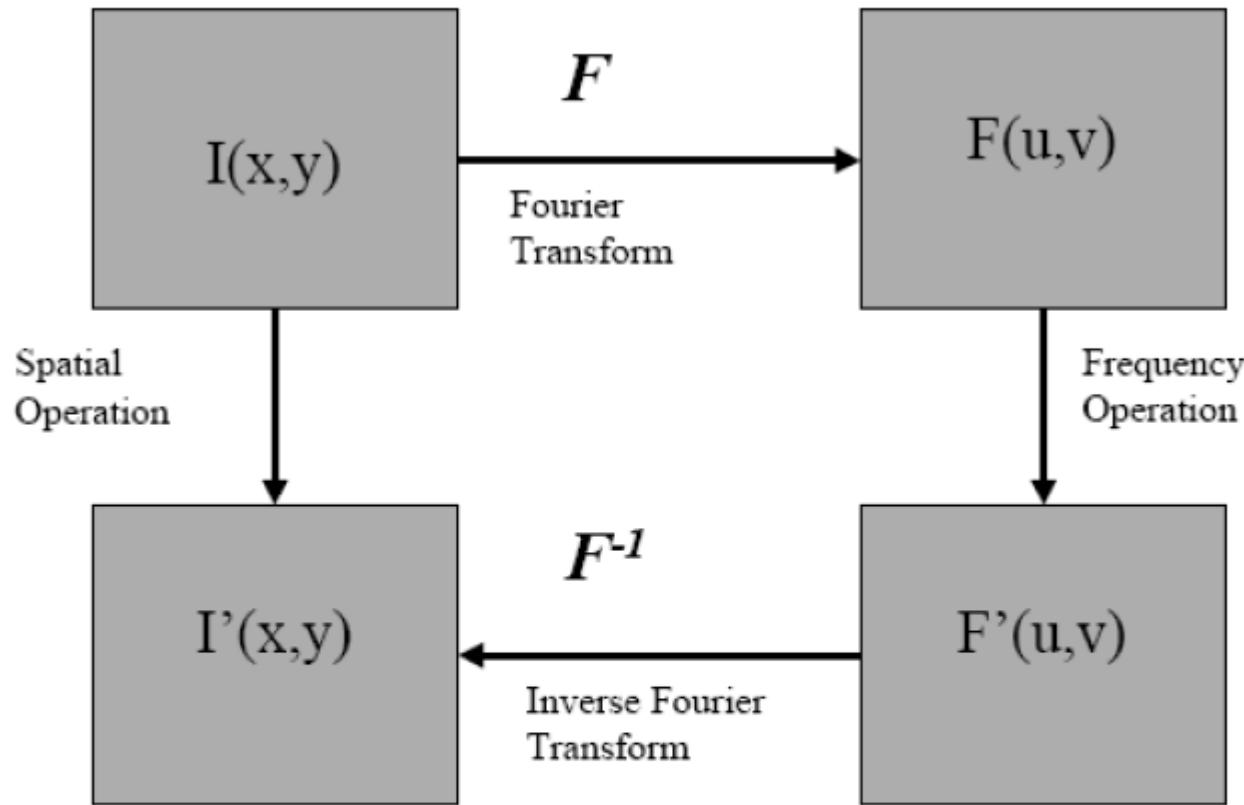


Genlik spk.

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

Faz spk.

## Frekans Uzayı – İmge İşleme Aşamaları



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

$H$ : Süzgeç

## DC katsayısı ve kaydırma (shifting)

DFT'deki DC katsayısı  $F(0,0)$  değeridir. Aşağıdaki denklemde  $v=u=0$  konulduğunda bulunur. Buna göre DC katsayısının değeri orijinal görüntü matrisindeki tüm eleman değerlerinin toplamı olarak ortaya çıkar. Bu değer yeni matrisin en üst sol köşesindedir.

**Kaydırma:** Görüntüleme amacı için; DC katsayısının matrisin orta nokta elemanı olarak olması uygundur. Bunun için kaydırma yapılabilir.

Kaydırma işlemi, dönüşümden önce  $f(x,y)$  matrisinin tüm elemanlarının  $(-1)^{x+y}$  ile çarpılacağı anlamındadır.

$$F(0,0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp(0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y).$$

|     |     |
|-----|-----|
| $A$ | $B$ |
| $C$ | $D$ |

An FFT

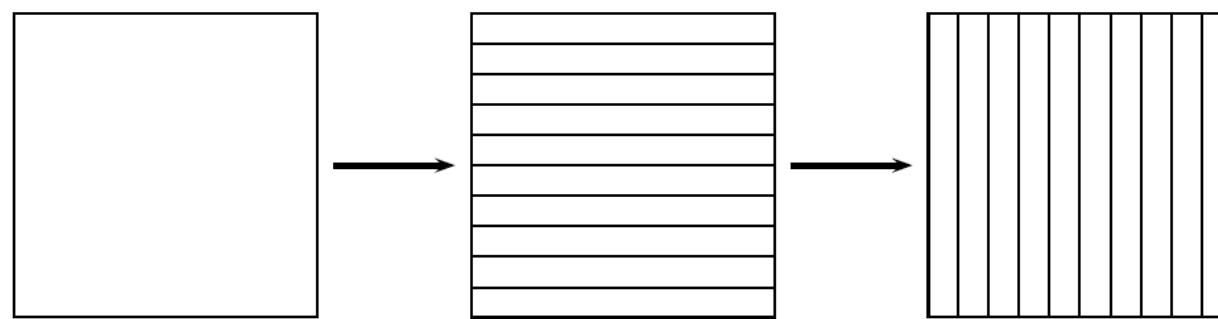
|     |     |
|-----|-----|
| $D$ | $C$ |
| $B$ | $A$ |

After shifting

## 2-D DFT'nin bazı özellikleri

$F$  ve  $f'$ 'yi hesaplayan denklemlerdeki üstel ifadeleri yandaki gibi ayırtırabiliriz. Bunun anlamı  $x$  ve  $u$ 'ya ve  $y$  ve  $v$ 'ye göre ayrı ayrı işlem yapabiliz. Yani iki boyutlu DFT tek boyutlu 2 adet DFT işleminden oluşmuştur. Bunuda resimdeki satırların ve sutunların ayrı ayrı DFT'sinin alınıp biribirileriyle ilişkilendirilmesi şeklinde düşünebiliriz.

$$\exp \left[ 2\pi j \left( \frac{xu}{M} + \frac{yv}{N} \right) \right] = \exp \left[ 2\pi j \frac{xu}{M} \right] \exp \left[ 2\pi j \frac{yv}{N} \right]$$



(a) Original image

(b) DFT of each row of (a)

(c) DFT of each column of (b)

# Dönüşümün gösterilmesi

- $F(x,y)$  görüntüsünden elde edilen  $F(u,v)$  fourier katsayılar matrisinin elemanları kompleks sayılardır. Kompleks sayılar direkt olarak görüntülenemez.
- Onların magnitüdleri (genlikleri)  $|F(u,v)|$  alınır . Bu bir takım double sınıfı büyük sayılardır. Bu büyük sahada uğraşmak için;
- 1-)  $|F(u,v)|$  'deki en büyük değer  $m$  bulunur (**Bu DC değerdir**). Ve  $|F(u,v)|/m$  işlemi yapılır. Böylece imshow ile görüntülenebilir.
- 2-)  $|F(u,v)|$  'yi görmek için mat2gray fonksiyonunu doğrudan kullanabiliriz.
- Genelde DC değerler çok büyük değerler olacağından; Bunun oluşturacağı görüntüde bu değer baskın çıkar. Bunun önüne geçmek için  $|F(u,v)|$  'nın logaritmasının alınması daha uygundur.  
 $\log(1+ |F(u,v)|)$
- Fourier transformasyonun magnitude'unun (genliğinin) görüntülenmesi, transformasyonun spektrumu diye isimlendirilir.

# MATLAB'da Fourier Dönüşümü

**fft** :Tek boyutlu DFT yapar(çıkışı vektördür)

**ifft** : DFT vektörünün tersini alır.

**fft2**: 2 boyutlu DFT yapar. Çıkışı Matristir.

**ifft2**: DFT matrixinin tersini alır.

**fftshift**: Bir transformasyonu kaydırır. (örnek DC değerini merkeze alma gibi)

# Frekans Uzayı – Matlab'da Frekans Uzayı İşlemleri

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

## Frekans Domain'de FILTRELEME

Fourier dönüşümü bir domain dönüşüm işlemiydi. Filtreleme de bir filtre matrisinin görüntü matrisi ile konvolusyonu olduğundan Frekans domenindefiltreleme için fourier dönüşümü çok uygundur.

Filtre matrisi yaratmanın en çok kullanılan yolu ise; Matlab kodu aşağıda verilen yapıdır. Burada 1.komut ile 256x256lik matris oluşturulur ( Bunların eleman değerleri -128 den başlayıp 1'er artarak 127 ye kadar değer alır.

Burda z herbir x,y koordinatının genlik değeridir.

```
>> [x,y]=meshgrid(-128:217,-128:127);  
>> z=sqrt(x.^2+y.^2);  
>> c=(z<15) ; Alçak geçen Filtreye örnek  
>> c =(z >15) ; Yüksek geçen Filtreye örnek
```

# İdeal Alçak Geçiren Filtre

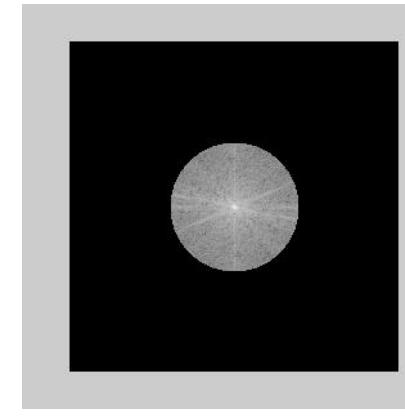
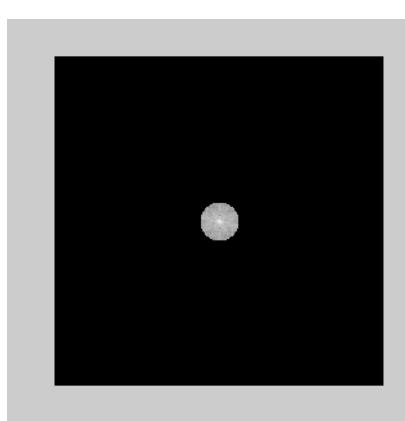
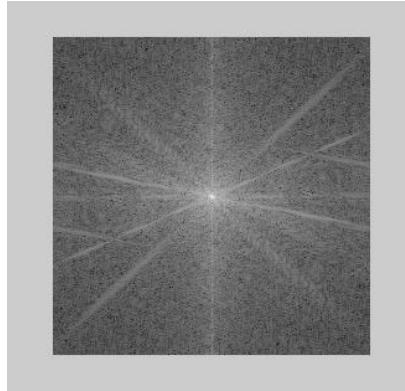
Varsayılmı̄ ki bir görüntünün Fourier dönüşüm matrisi  $F$  elimizde mevcut olsun. Ve bunun DC katsayısını merkeze kaydırduğumuzda, dolayısıyla düşük frekanslı bileşen katsayıları da merkeze kaymış olacaktır.

Bu durumda bir alçak geçiren滤re matrisi ( $c$ ) ile bu  $F$  dönüşüm matrisini uygun bir şekilde işleme tabi tutarak (elementer çarpım işlemi  $\cdot *$  ( $F \cdot * c$ )), merkezdeki ve yakınındaki (alçak frekanslı bileşenler) değerler korunur ve merkezden uzak değerler (yüksek frekanslı bileşenler) ya küçülür veya yok olur. Böylece alçak geçiren滤re işlemi gerçekleşmiş olur.

$$c(x, y) = \begin{cases} 1 & \text{Eğer } (x, y), \text{ merkeze } D \text{ den daha yakın ise} \\ 0 & \text{Eğer } (x, y), \text{ merkeze } D \text{ den daha uzakta ise} \end{cases}$$

Bu filtrelenmiş FFT (DFT) matrisin inversi  $F^{-1}(F.c)$  alınarak uzaysal domaine dönülmüş olur.

**Filtrenin yarıçapı ne kadar küçükse elenen yüksek frekanslı bileşenler o kadar büyük olur. Resim bulanıklaşır**



## Frekans Domeninde ALÇAK GEÇİREN FİLTRE

```
>> cm=imread('cameraman.tif');
>> imshow(cm)
>> cf=fftshift(fft2(cm));
>> imshow(mat2gray(log(1+abs(cf))))
```

Buradan merkeze kaydırılmış transform matrisini (cf) bir daire matrisi ile çarparak (\* simbolu MATLAB'da iki matrisin element-wise çarpımına eşittir)

```
>> [x,y]=meshgrid(-128:127,-128:127);
>> z=sqrt(x.^2+y.^2);
>> c=(z<15);
>> cf1=cf.*c;
>> imshow(mat2gray(log(1+abs(cf1))));
```

```
>> s=ifft2(cf1);
>> imshow(mat2gray((1+abs(s))));

> c=(z<50);
>> cf1=cf.*c;
>> imshow(mat2gray(log(1+abs(cf1))));
```

```
>> s=ifft2(cf1);
>> imshow(mat2gray((abs(s))));
```

## Frekans domaininde yüksek geçiren filtre

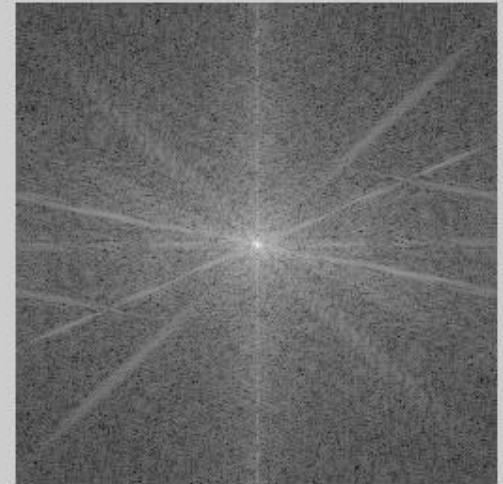
- Yüksek geçiren filtre ise, DC değeri merkeze kaydırılmış  $F$  matrisinin merkez veya merkeze yakın eleman değerlerini (Alçak bileşenleri) yok etmek veya zayıflatmak, merkezden uzak eleman değerlerini ise korumak şeklindedir. Filtre matrisi oluşturma işlemi alçak geçiren filtrededeki gibidir. Tek fark aşağıdaki deyimdedir. Buradaki 15 örnek olarak verilmiştir.

$$c = (z > 15)$$

- Oluşturulan  $c$  filtre matrisi ile  $F$  fft matrisi elemanter çarpma işlemine tabii tutulur. Filtreleme işlemi başarılı olmuştur.
- Bunun ters FFT’si ifft2 alınarak,filtrelenmiş görüntü uzaysal domaine dönüştürülür.
  - Yüksek geçiren滤releme görüntüdeki nesne kenarlarının belirginleşmesine, gövdenin ise siyahlaşmasına neden olur.

## Frekans domaininde yüksek geçen filtre

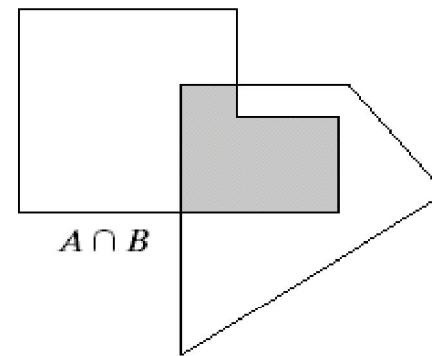
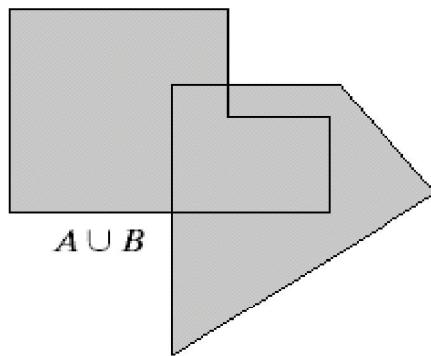
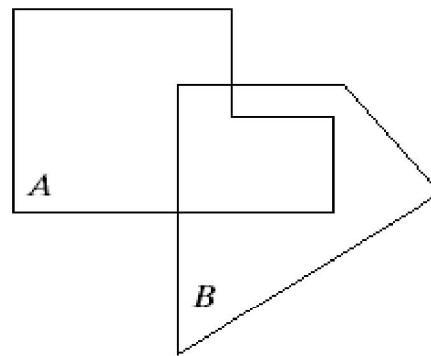
```
>> cm=imread ('cameraman.tif');
>> [x,y]=meshgrid (-128:127,-128:127);
>> z=sqrt(x.^2+y.^2);
>> c=(z>15);
>> cf= fftshift(fft2(cm));
>> cfh=cf.*c;
>> imshow(mat2gray(log(1+abs(cfh))));
>> cfhi=ifft2(cfh);
>> imshow(uint8(abs(cfhi)))
```



# Morfolojik Görüntü İşleme

# Morfoloji nedir?

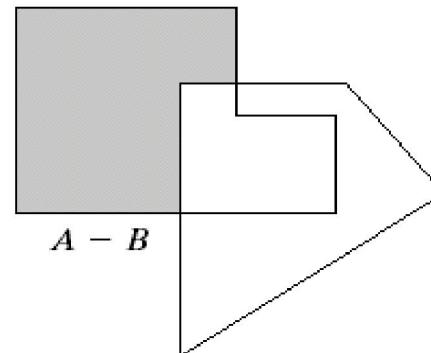
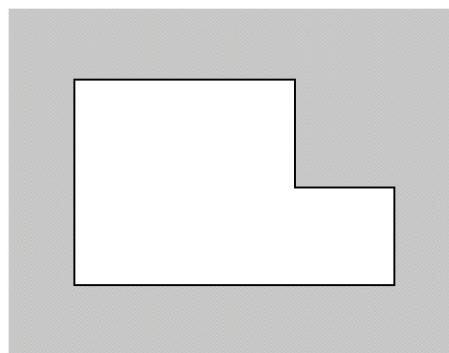
- Morfoloji, görüntü biçimi ve yapısı olarak tarif edilebilir (form and structure).
- Matematiksel morfoloji, görüntü üzerinde objeleri gösteren küme teorisine dayanır.
- Görüntü işlemede morfoloji, ikili görüntüler üzerinde 2B sayı kümelerinin tespitidır.
- Görüntüdek sınırlar, iskelet gibi yapıların tanınması ve çıkarılması, gürültü giderimi, bölütleme gibi uygulamalar için gerekli bir araçtır.
- İkili görüntüler üzerindeki işlemler gri seviye görüntüler için genişletilebilir.



|   |   |   |
|---|---|---|
| a | b | c |
| d | e |   |

**FIGURE 9.1**

- (a) Two sets  $A$  and  $B$ .
- (b) The union of  $A$  and  $B$ .
- (c) The intersection of  $A$  and  $B$ .
- (d) The complement of  $A$ .
- (e) The difference between  $A$  and  $B$ .



# Yansıma ve Öteleme

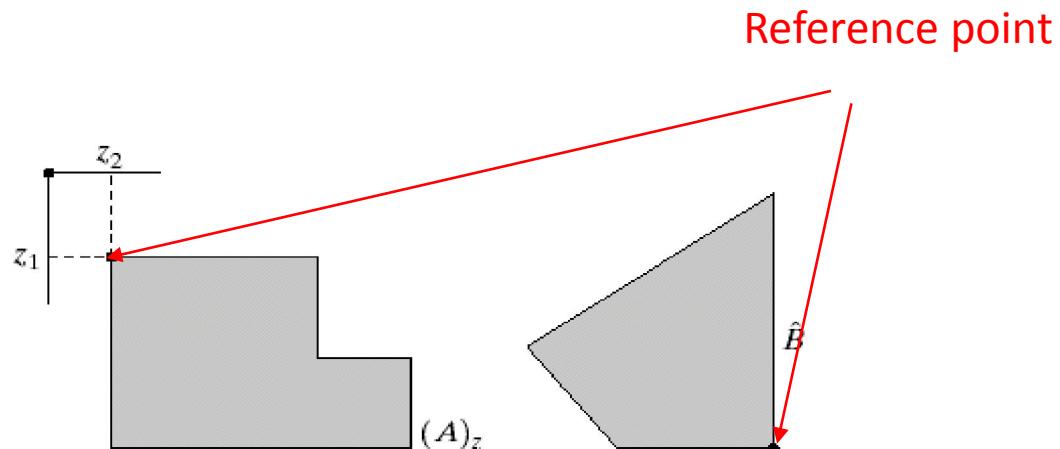
Bu işlemleri için bir referans nokta gerekmektedir.

- ❖ **Yansıma:** B noktası için:

$$= \{x \mid x = -b, \text{ for } b \in B\}$$

- ❖ **Translation:** A noktasının  $x=(x_1, x_2)$  kadar ötelenmesi sonucu:

$$(A)_x = \{c \mid c = a + x, \text{ for } a \in A\}$$



a b

**FIGURE 9.2**  
(a) Translation of  
 $A$  by  $z$ .  
(b) Reflection of  
 $B$ . The sets  $A$  and  
 $B$  are from  
Fig. 9.1.

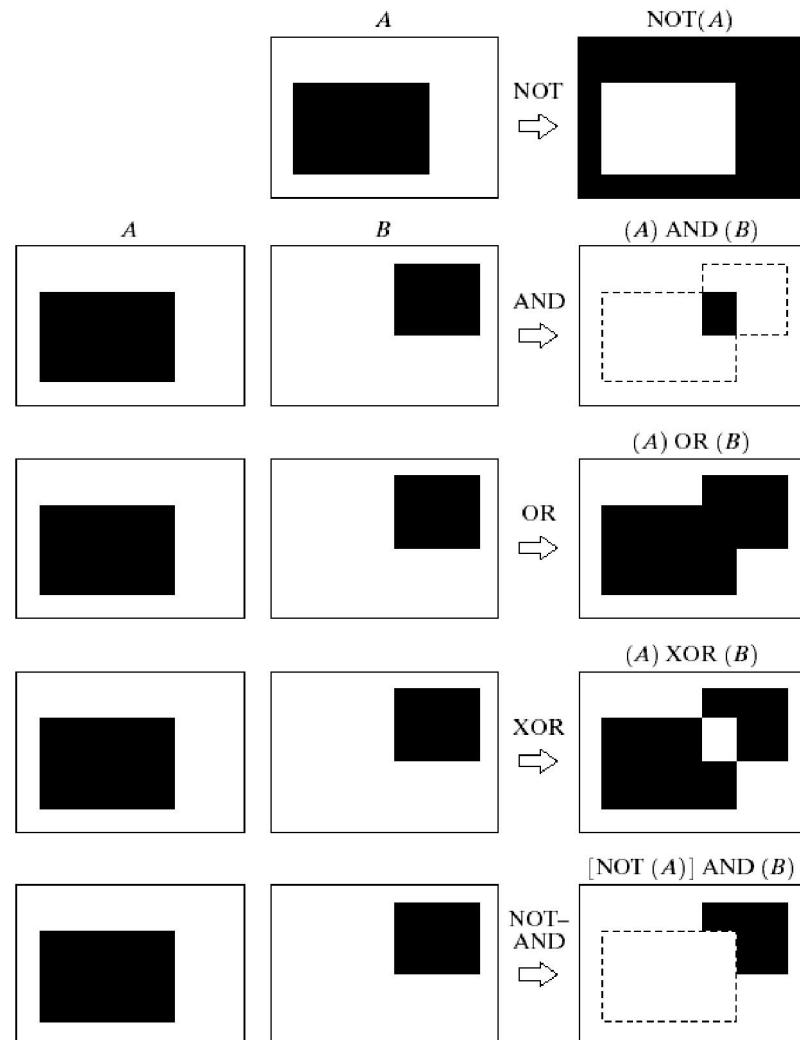
# Temel Mantıksal İşlemler

**TABLE 9.1**

The three basic logical operations.

| $p$ | $q$ | $p \text{ AND } q$ (also $p \cdot q$ ) | $p \text{ OR } q$ (also $p + q$ ) | $\text{NOT } (p)$ (also $\bar{p}$ ) |
|-----|-----|--|-----------------------------------|-------------------------------------|
| 0   | 0   | 0                                      | 0                                 | 1                                   |
| 0   | 1   | 0                                      | 1                                 | 1                                   |
| 1   | 0   | 0                                      | 1                                 | 0                                   |
| 1   | 1   | 1                                      | 1                                 | 0                                   |

# Binary Görüntülerde Mantıksal İşlemler



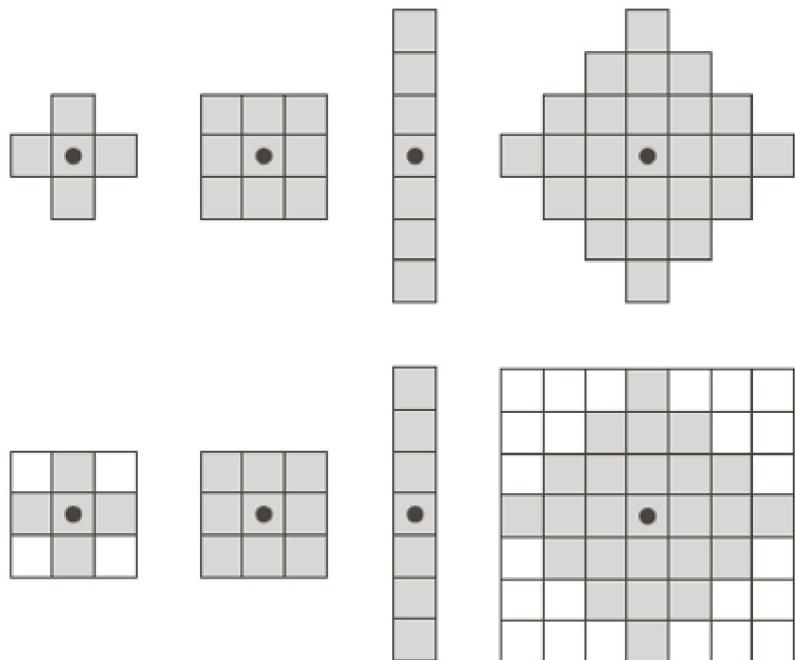
**FIGURE 9.3** Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

# Matematiksel Morfoloji

- Matematiksel morfoloji iki temel operasyonu içerir:
  - dilation
  - erosion
- ayrıca birkaç komposit ilişkisi içerir:
- Opening
- closing
- conditional dilation
- ...
- Morfolojik operatörlerin iki girişi vardır:
  - Görüntü
  - Yapısal eleman (structure element).

# Yapısal Eleman Örnekleri

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

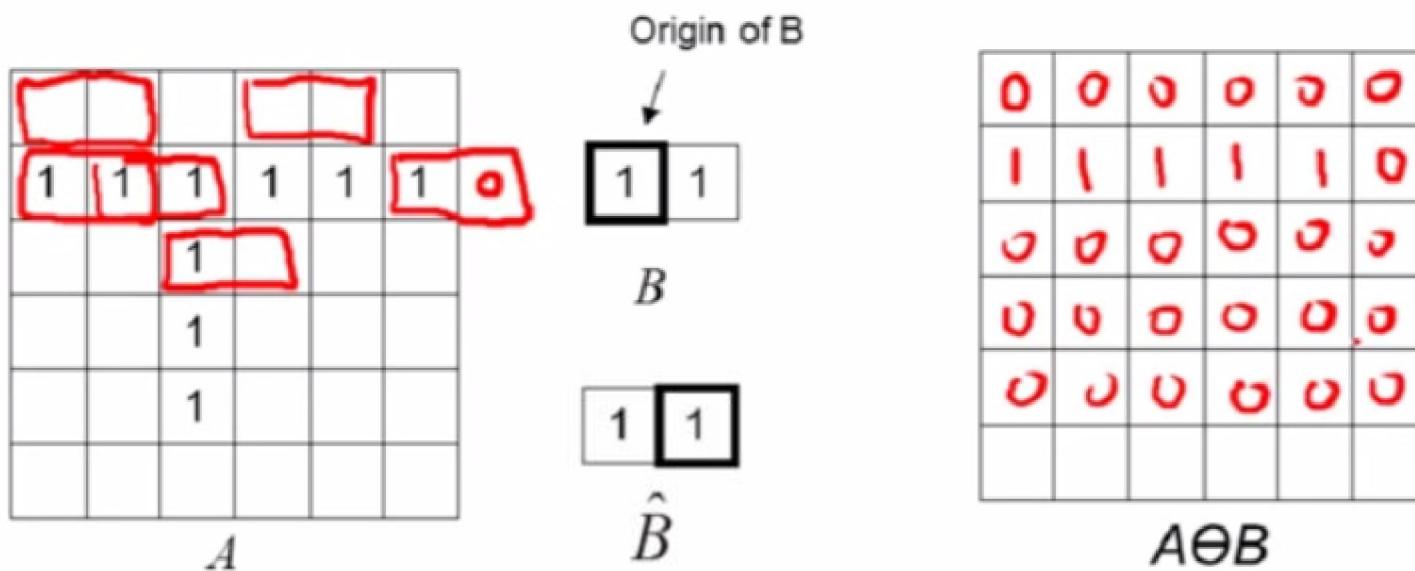


**FIGURE 9.2** First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

# Erosion (Aşınma)

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

- A kümesinin B yapısal elemanın z noktasına göre kaydırılmış hali ile erosion işlemi sonucunda, eğer yapısal elemanın tamamı kümeye içerisinde ise sonuç 1 olur. Aksi takdirde 0 olur.



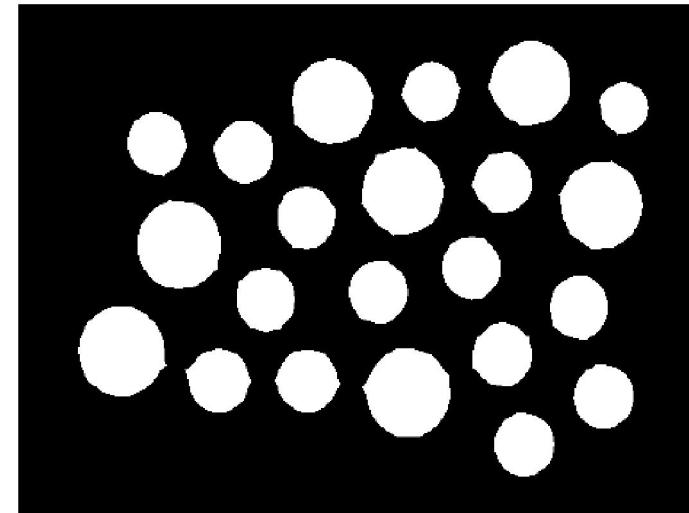
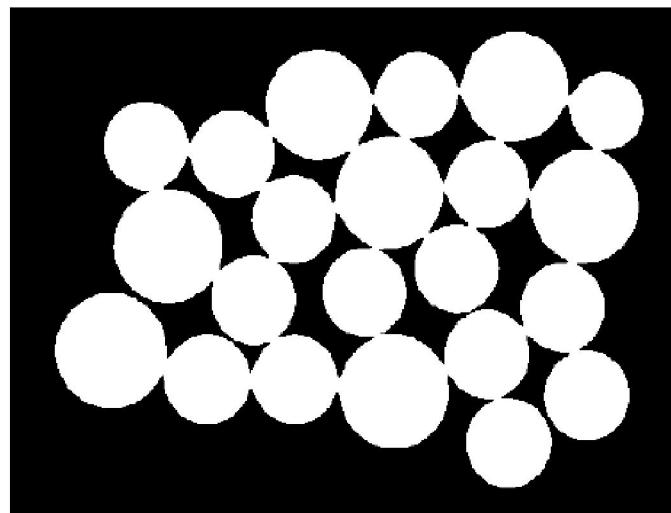
# Erosion

- ikili imgedeki nesneyi küçültmeye ya da inceltmeye yarayan morfolojik işlemidir.
- İmge içerisindeki nesneler ufalır, delik varsa genişler, bağlı nesneler ayrılma eğilimi gösterir

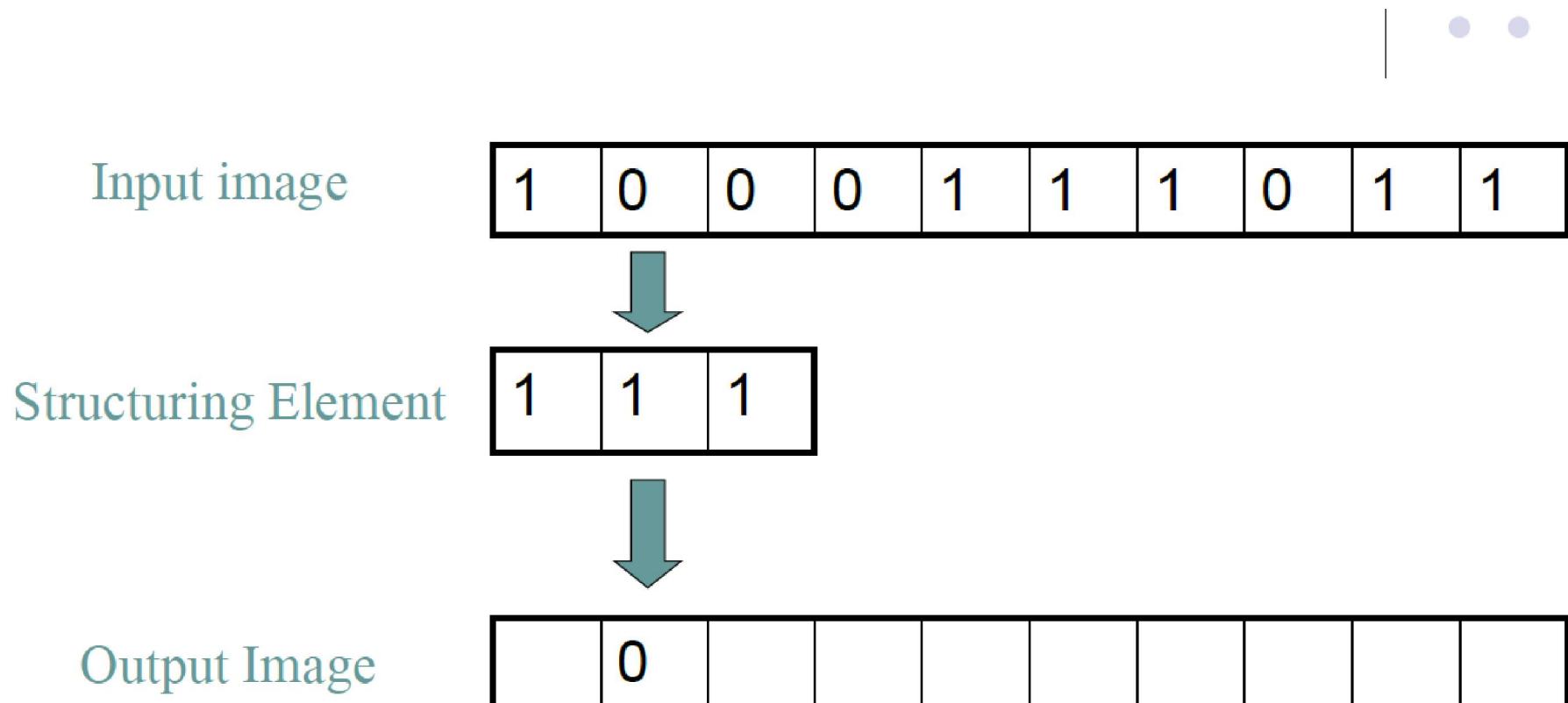
# Erosion-Aşınma

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

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



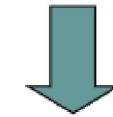
# Erosion-Aşınma



# Erosion-Aşınma

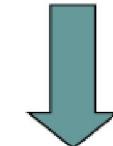
Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



Structuring Element

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



Output Image

|  |   |   |  |  |  |  |  |  |  |
|--|---|---|--|--|--|--|--|--|--|
|  | 0 | 0 |  |  |  |  |  |  |  |
|--|---|---|--|--|--|--|--|--|--|

# Erosion-Aşınma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|

Structuring Element



Output Image

|  |   |   |   |  |  |  |  |  |  |
|--|---|---|---|--|--|--|--|--|--|
|  | 0 | 0 | 0 |  |  |  |  |  |  |
|--|---|---|---|--|--|--|--|--|--|

# Erosion-Aşınma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



Structuring Element

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



Output Image

|  |   |   |   |   |  |  |  |  |  |
|--|---|---|---|---|--|--|--|--|--|
|  | 0 | 0 | 0 | 0 |  |  |  |  |  |
|--|---|---|---|---|--|--|--|--|--|

# Erosion-Aşınma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



Output Image

|  |   |   |   |   |   |  |  |  |  |
|--|---|---|---|---|---|--|--|--|--|
|  | 0 | 0 | 0 | 0 | 1 |  |  |  |  |
|--|---|---|---|---|---|--|--|--|--|

# Erosion-Aşınma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



Output Image

|  |   |   |   |   |   |   |   |  |  |
|--|---|---|---|---|---|---|---|--|--|
|  | 0 | 0 | 0 | 0 | 0 | 1 | 0 |  |  |
|--|---|---|---|---|---|---|---|--|--|

# Erosion-Aşınma

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 |  |  |
|--|---|---|---|---|---|---|---|--|--|

# Erosion-Aşınma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



Structuring Element

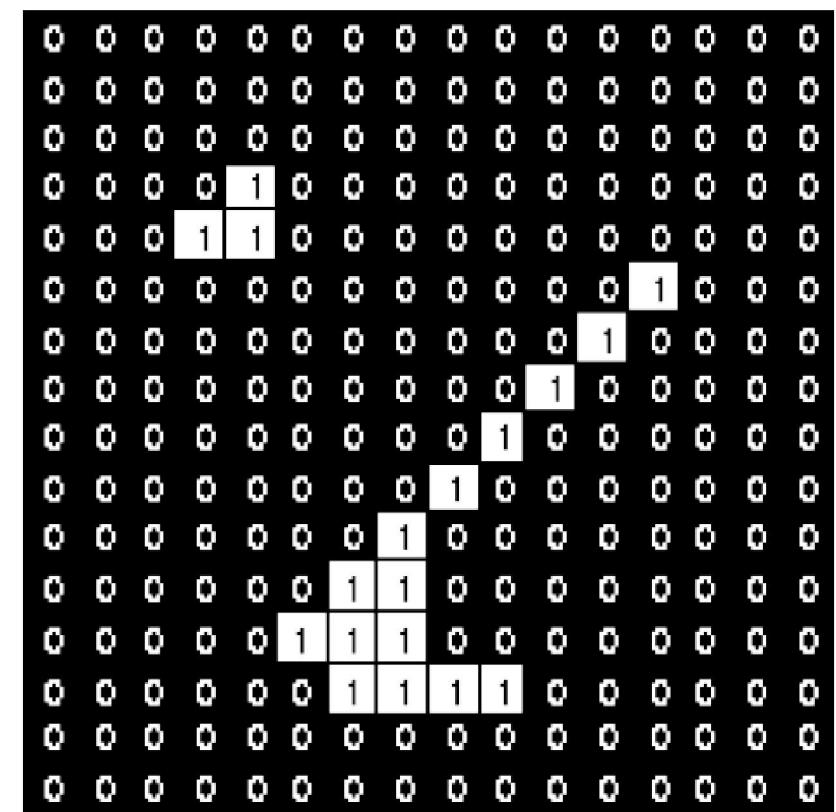
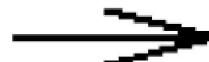
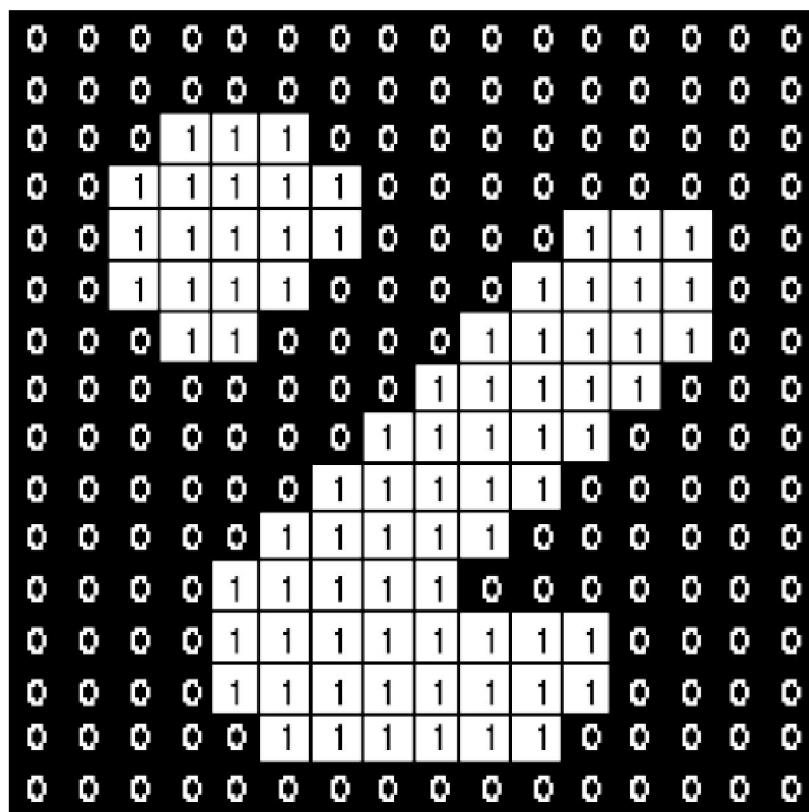
|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



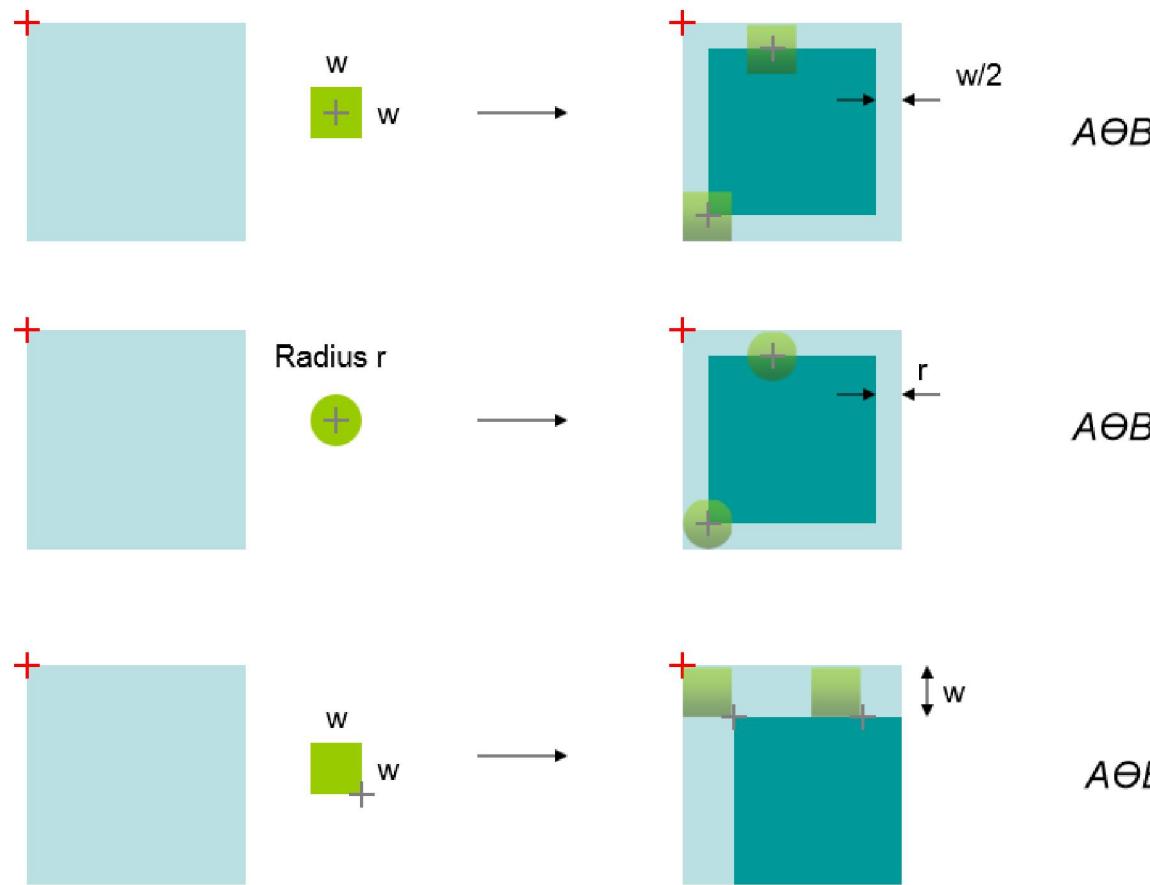
Output Image

|  |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|
|  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|--|---|---|---|---|---|---|---|---|---|

# Erosion-Aşınma

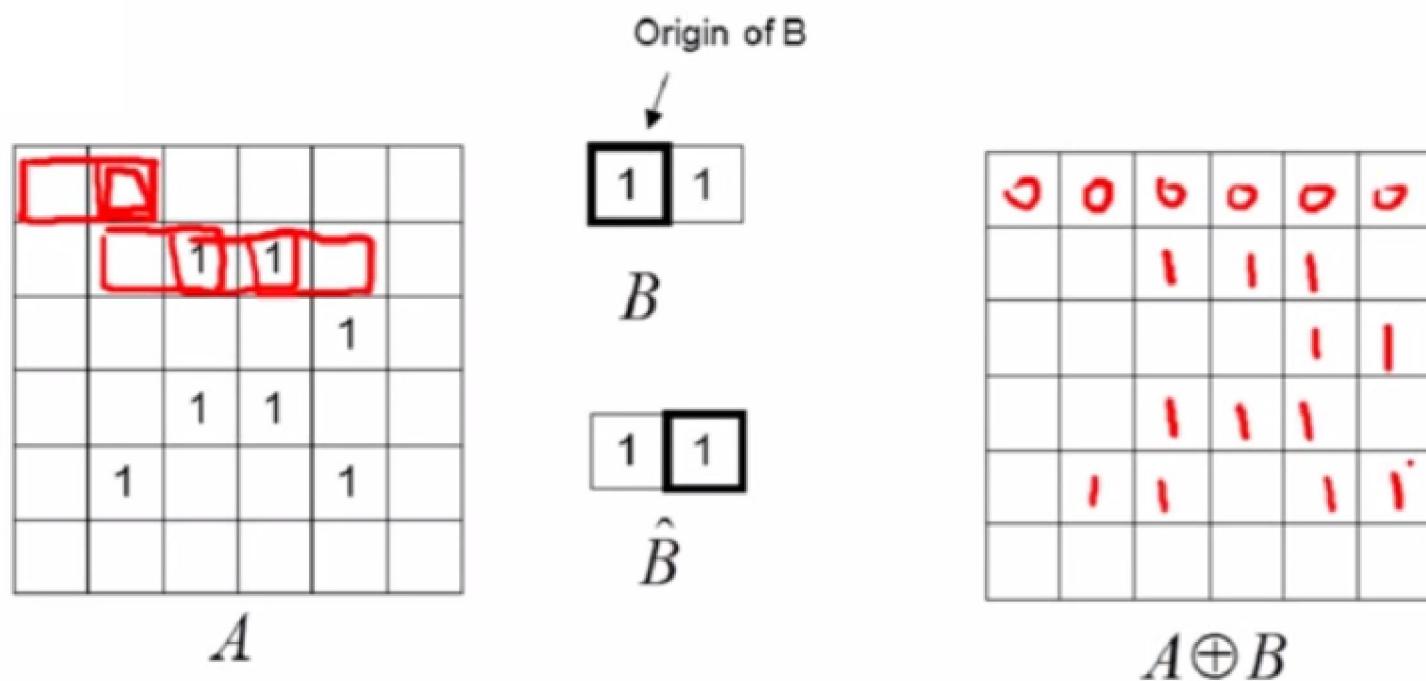


# Aşınma Örnek



# Dilation (Yayma)

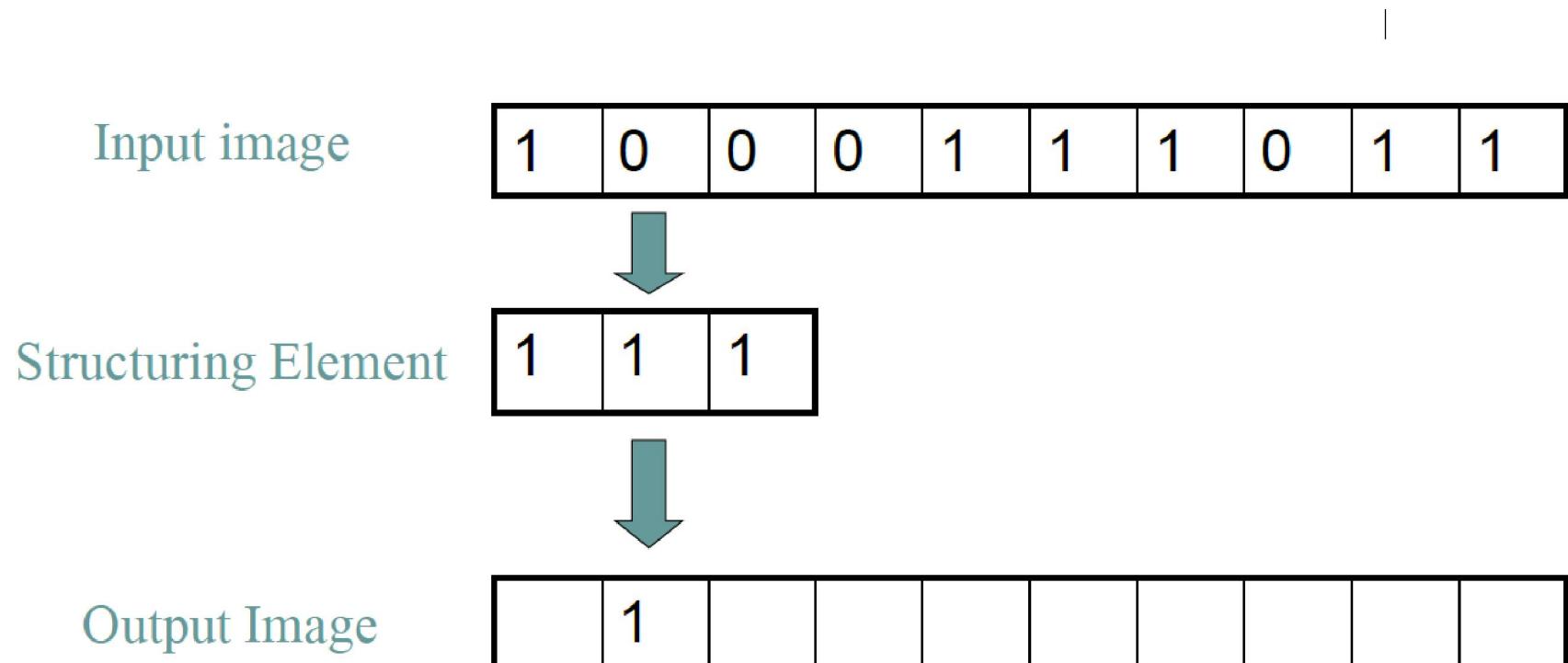
- Z noktasına göre kaydırılmış B yapısal elemanı ile A kümesinin keşisen ortak bir elemanı varsa merkez noktanın değeri 1 olur.



# Dilation

- ikili imgedeki nesneyi büyütmeye ya da kalınlaştırmaya yarayan morfolojik işlemidir.
- Sayısal bir imgeyi genişletmek imgeyi yapısal elemanla kesiştiği bölümler kadar büyütmektir.
- İşlenecek imgenin her bir pikseli, yapısal elemanın merkez noktasına oturtularak genleşme işlemi yapılmaktadır.
- Kalınlaştırma işleminin nasıl yapılacağını yapısal eleman belirler.
- Yayma işlemi uygulanmış bir imgede, imge içerisindeki deliklerin ve boşlukların doldurulması ve köşe noktasının yumuşaması gözlenir

# Dilation - Yayma



# Dilation - Yayma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



Output Image

|  |   |   |  |  |  |  |  |  |  |
|--|---|---|--|--|--|--|--|--|--|
|  | 1 | 0 |  |  |  |  |  |  |  |
|--|---|---|--|--|--|--|--|--|--|

# Dilation -Yayma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|

Structuring Element



Output Image

|  |   |   |   |  |  |  |  |  |  |
|--|---|---|---|--|--|--|--|--|--|
|  | 1 | 0 | 1 |  |  |  |  |  |  |
|--|---|---|---|--|--|--|--|--|--|

# Dilation -Yayma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



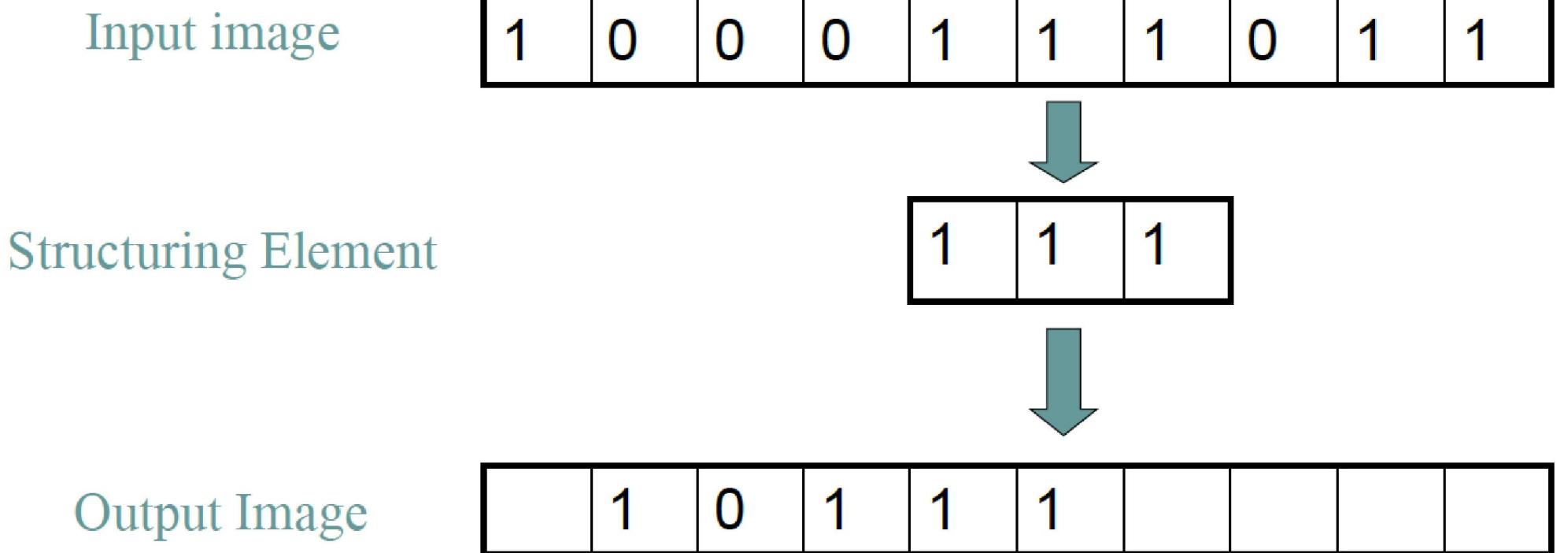
|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



Output Image

|  |   |   |   |   |  |  |  |  |  |
|--|---|---|---|---|--|--|--|--|--|
|  | 1 | 0 | 1 | 1 |  |  |  |  |  |
|--|---|---|---|---|--|--|--|--|--|

# Dilation -Yayma



# Yayma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|

Structuring Element

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|

Output Image

|  |   |   |   |   |   |   |  |  |  |
|--|---|---|---|---|---|---|--|--|--|
|  | 1 | 0 | 1 | 1 | 1 | 1 |  |  |  |
|--|---|---|---|---|---|---|--|--|--|

# Yayma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



Structuring Element

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



Output Image

|  |   |   |   |   |   |   |   |  |  |
|--|---|---|---|---|---|---|---|--|--|
|  | 1 | 0 | 1 | 1 | 1 | 1 | 1 |  |  |
|--|---|---|---|---|---|---|---|--|--|

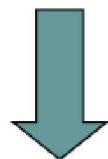
# Yayma

Input image

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|



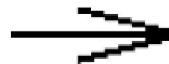
|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|



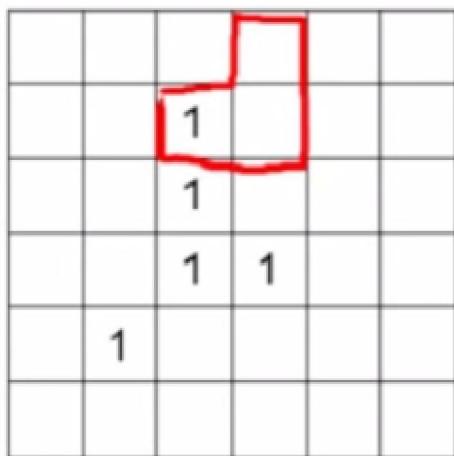
Output Image

|  |   |   |   |   |   |   |   |   |  |
|--|---|---|---|---|---|---|---|---|--|
|  | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |  |
|--|---|---|---|---|---|---|---|---|--|

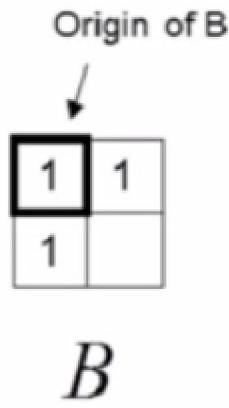
# Dilation -Yayma



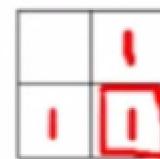
# Dilation Örnek



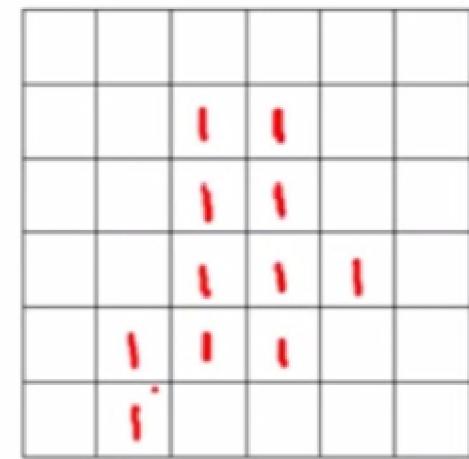
$A$



$B$

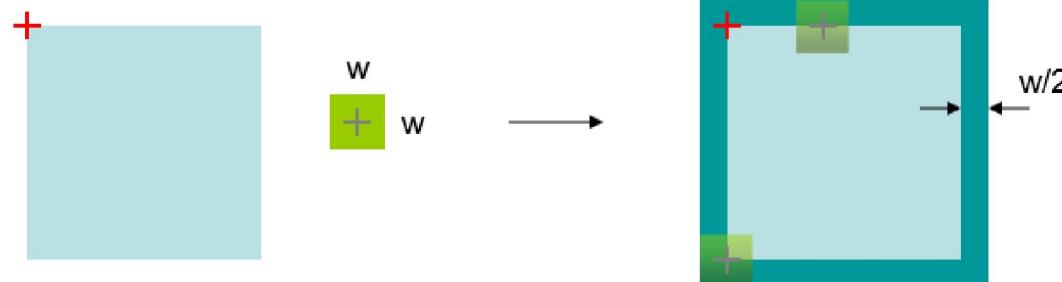


$\hat{B}$

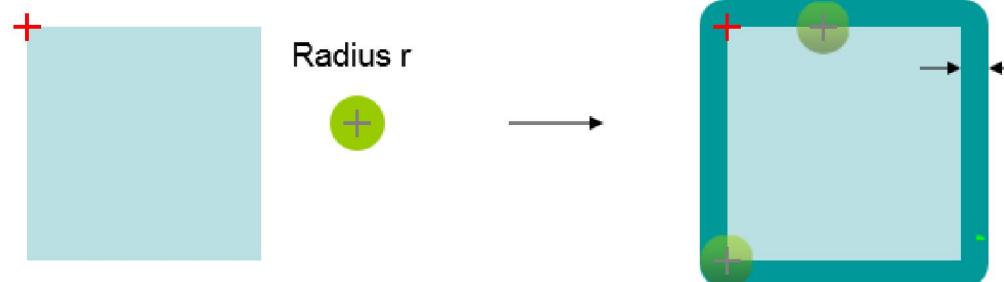


$A \oplus B$

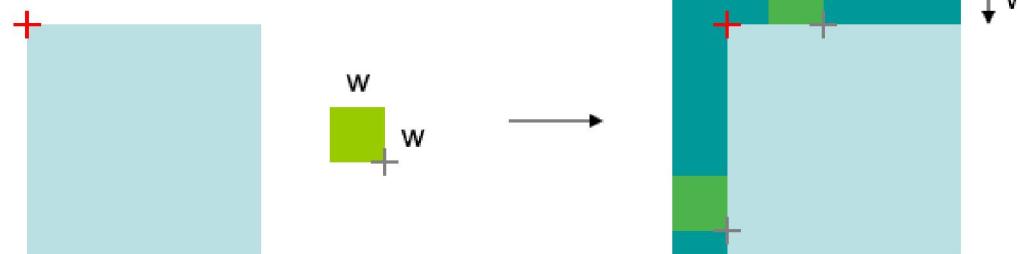
# Dilation Örnek



$A \oplus B$



$A \oplus B$



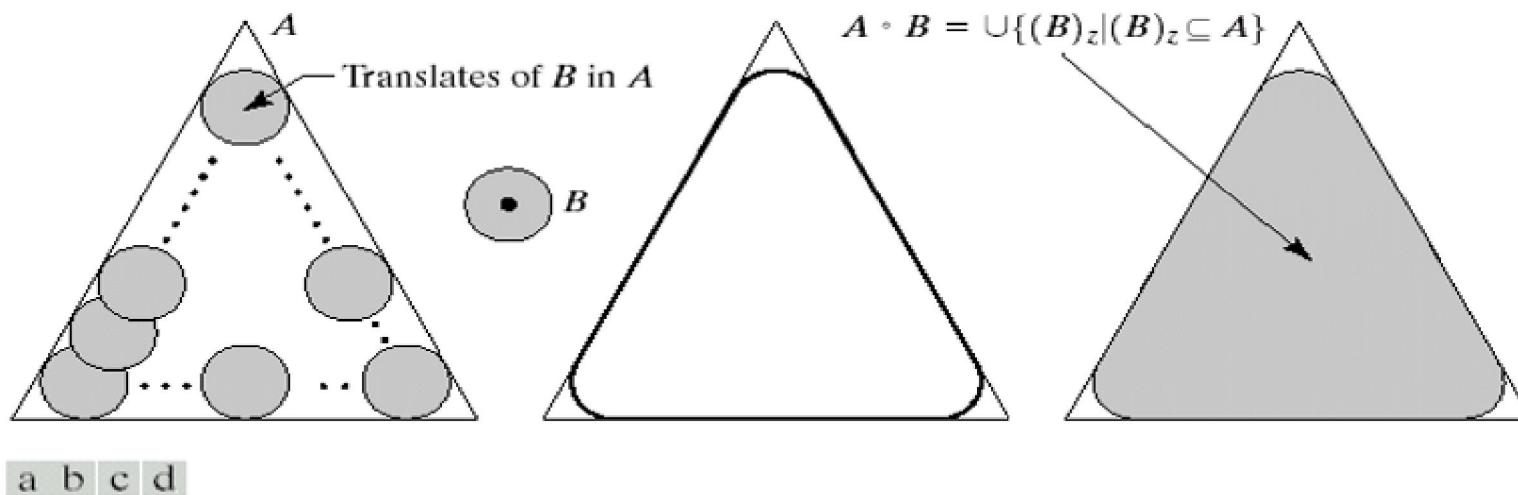
$A \oplus B$

# Matlab Examples

- Make a structuring element
  - `SE = strel('square', W)` or
  - `SE = strel('disk', R )`
- Dilation and erosion
  - `imdilate(IM, SE)`
  - `imerode(IM, SE)`

# Opening

- Opening – Bir erosion işlemi ardından bir dilation uygulanmasıdır.



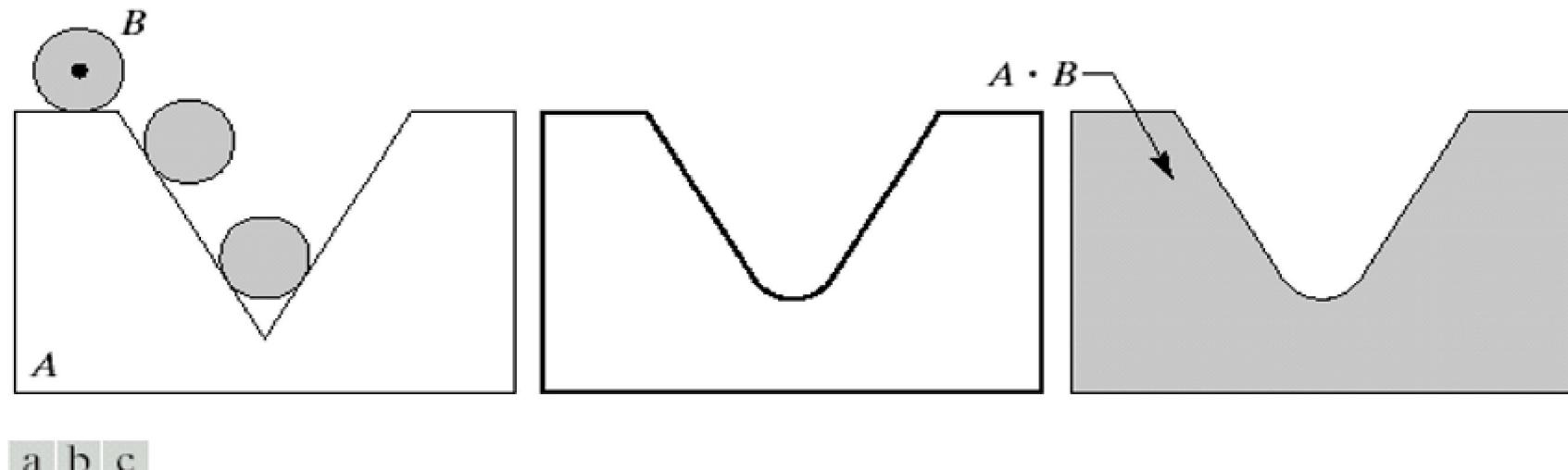
**FIGURE 9.8** (a) Structuring element  $B$  “rolling” along the inner boundary of  $A$  (the dot indicates the origin of  $B$ ). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

# Opening

- İmge üzerinde aşınma işleminin hemen ardından genleşme işlenmesi uygulanması sonucu açma işlemi elde edilir.
- İmge içerisindeki nesneler ve nesneler arasındaki boşluklar yapısal elemanın büyüğünü göre temizlenir.
- İmge üzerinde kalan nesneler orijinal imgedeki şekillerinden biraz daha küçük hale gelir.
- Açıma işlemi ile birbirine yakın iki nesne imgede fazla değişime sebebiyet vermeden ayrılmış olurlar

# Closing

Bir dilation işlemi ve ardından bir erosion işlemini tanımlar. İki işlemde aynı yapısal elemanı kullanır.



a b c

**FIGURE 9.9** (a) Structuring element  $B$  “rolling” on the outer boundary of set  $A$ . (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

# Closing

- imge üzerinde genleşme işleminin hemen ardından aşınma işleminin uygulanması sonucu kapama işlemi elde edilir.
- Dolayısıyla birbirine yakın iki nesne imgede fazla değişiklik yapılmadan birbirine bağlanmış olur.
- Kapama işlemi sonunda imge içerisindeki noktalar birbirlerini kapatırlar, imgedeki ana hatlar daha da dolgunlaşır.
- Genleşme işlemine benzer bir şekilde kapama işleminde de birbirine yakın olan noktalar arasındaki boşluklar dolar ve noktalar birleşir

# Matlab Morfolojik İşlemler 1

- **strel:** strel, morfolojik işlemlerde kullanılan yapısal filtre elemanıdır. Morfolojik işlemleri hangi şekil ve parametrelerle uygulayacağımızı strel ile belirleriz.

- ▶ Kullanım şekli; **SE = strel (filtreleme şekli, parametre)**
- ▶ Örnek; SE =strel('disk', R) R; yarıçap  
SE = strel('square', L) L; karenin bir kenarının uzunluğu

**imdilate:** açma işlemini yapar.

**imerode:** aşındırma işlemi yapar.

# Matlab Morfolojik İşlemler 2

- **imclose**, **imopen**: imdilate ve imerode işlemlerinin birlikte kullanılması ile oluşan filtrelerdir. Uygulama mantıkları benzerdir.  
**imfill**: gri seviyede veya binary modda açıklıkları doldurur.
- **bwarea**: binary modda alan nesnelerin alanını hesaplar.  
**bwareaopen**: binary modda küçük parçaları(bağılı olmayan) yok eder. aşındırma işleminin benzeridir.  
**bwconncomp**: bağlı komponentleri buldurur.
- **bwmorph**: binary modda morfolojik işlemler yapılır. Birçok maskeleme çeşidi vardır.

# 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.

## Örnek 2

```
W = imread('circles.png');
```

```
mshow(BW);
```

```
W2 = bwmorph(BW,'remove');
```

```
gure
```

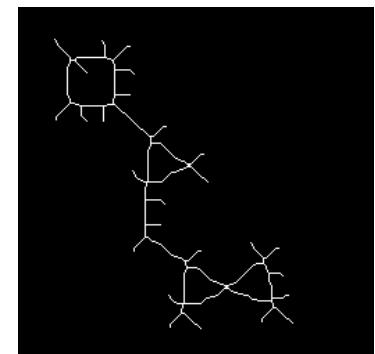
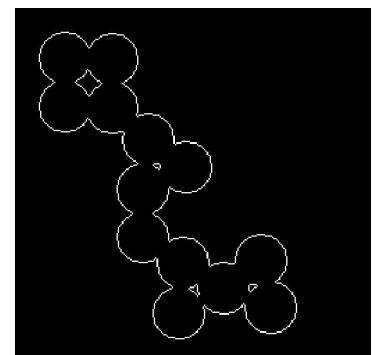
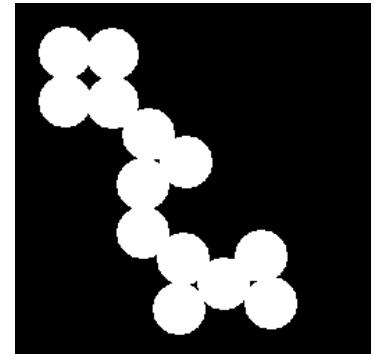
```
W3 = bwmorph(BW,'skel',Inf);
```

```
gure
```

```
mshow(BW3)
```

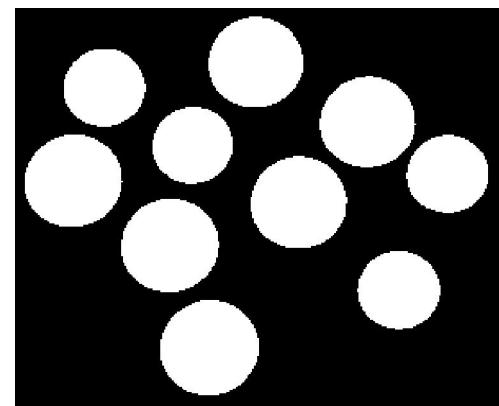
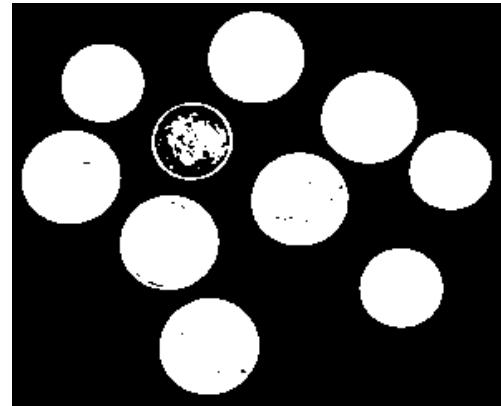
```
gure
```

```
mshow(BW2)
```



# Örnek 3

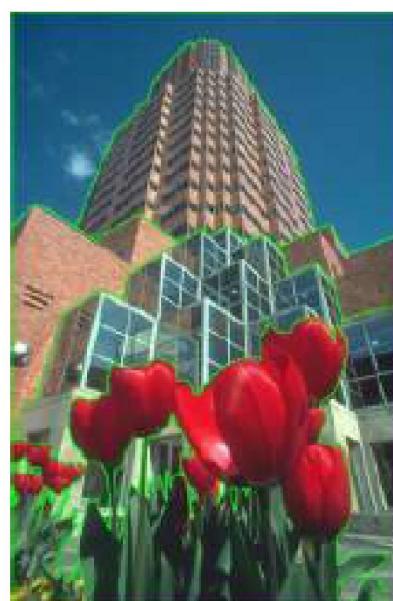
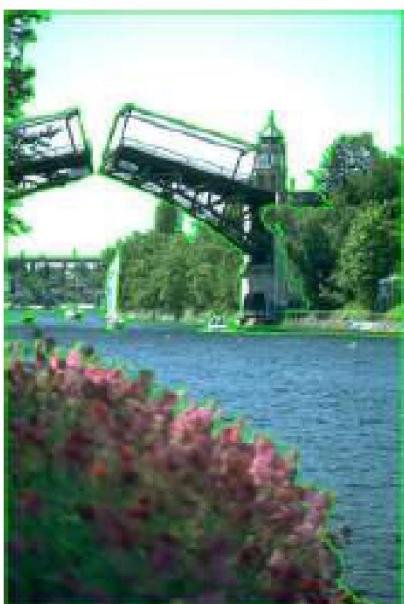
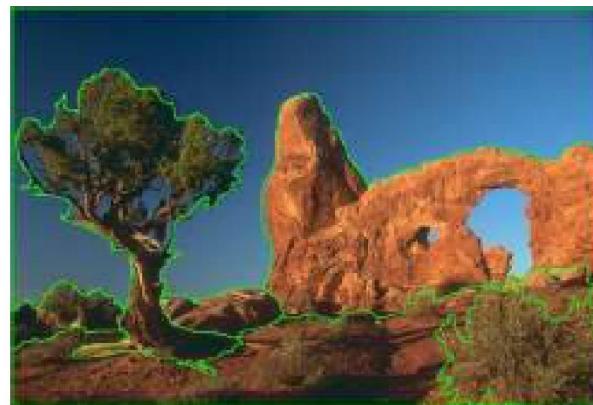
- `BW4 = im2bw(imread('coins.png'));`
- `BW5 = imfill(BW4,'holes');`
- `imshow(BW4), figure, imshow(BW5)`



# Görüntü Bölümü

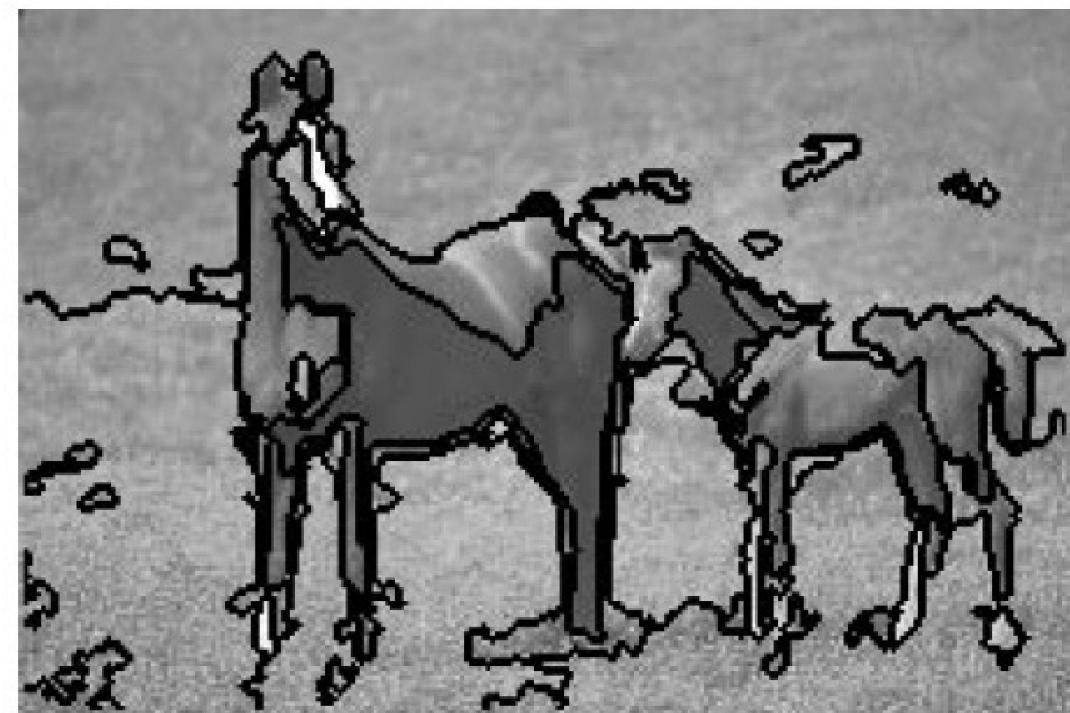
# Bölütleme nedir?

- Bölütleme, bir görüntüyü anlamlı ve türdeş bölümlerden oluşan kümelere parçalamayı içerir
- Parçalanmış her bölümdeki pikseller, nitelikleri veya özellikleri aynı olan türdeş bir kümeye sahiptirler.
- Görüntü özelliklerinin bu kümeleri; piksel grilik seviyesi, piksel RGB rengi, pikselin kameradaki sırası, pikselin yeri, bölgesel es-değişirlik matrisi, gri seviyeler, zıtlık, spektral değerler veya niteliksel özellikleri içerebilir.
- Bölütlemenin sonucu, her biri tek etikete sahip olan türdes bölümlerin bir sayısıdır.
- Böylece bir görüntü, birbirine bağlı ve üst üste binmeyen bölümlerin birkümesi olarak tanımlanır, bu durumda görüntüdeki her piksel ait olduğu bölümü gösteren tek bir bölüm etiketine sahip olur. Görüntü bölütlemenin sonuçları genellikle daha yüksek seviyeli sayısal görüntü işleme işlemlerinin ilk parametreleri olarak kullanılırlar.



**Segmentasyon,  
görüntüyü aynı cinsten  
obje ve bölgelere  
ayırmaktır.**





Orijinal Görüntü ve Bölütlenmiş Görüntü

# Görüntü Bölümleme

- Görüntü üzerindeki elemanların/parçaların bir grup olup olmamasına dair bir çok faktör etkili olmaktadır.
  - Yakınlık (Proximity)
  - Benzerlik (Similarity)
  - Ortak/genel/yaygın kader (Common fate)
  - Ortak bölge (Common region)
  - Paralellik (Parallelism)
  - Kapalılık/yakınlık(Closure)
  - Simetri (Symmetry)
  - Sürekllilik (Continuity)
  - Bilinen yapılanma/konfigürasyon (Familiar configuration)



Not grouped



Proximity



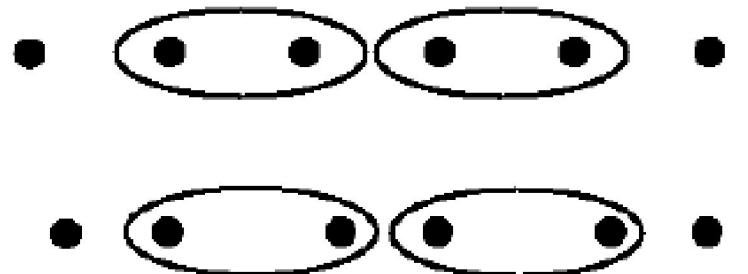
Similarity



Similarity



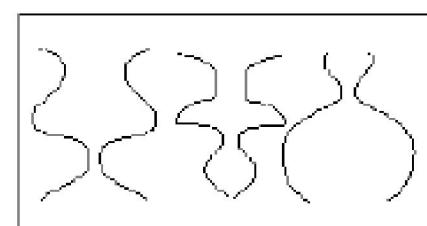
Common Fate



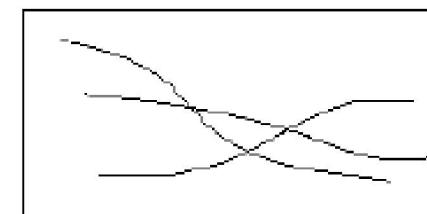
Common Region



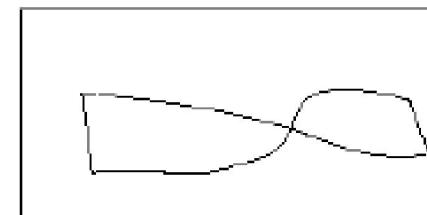
Parallelism



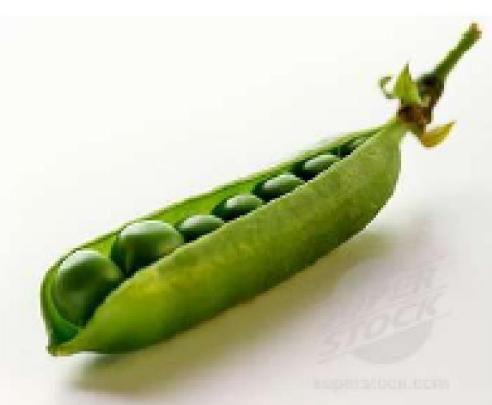
Symmetry



Continuity



Closure

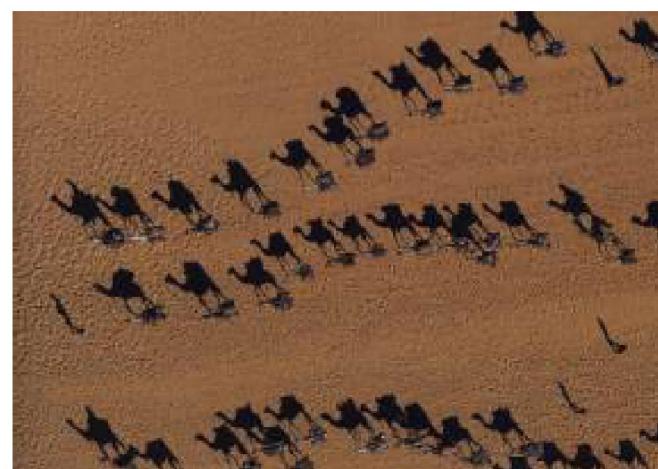


Benzerlik

Simetri



Yakınlık



Ortak kader

# Görüntü Segmentasyonu

- Yapılması gereken bölgeler arasındaki benzerliğin tanımlanması ve uygulanmasıdır.
- Örnek yaklaşımlar:
  - Histogram tabanlı (Histogram-based)
  - Kümeleme tabanlı (Clustering-based)
  - Bölge büyütme (Region growing)
  - Ayırma ve birleştirme (Split-and-merge)
  - Morfolojik (Morphological)

# Görüntü Bölümleme Kullanım Alanları

- Medikal görüntülerde;
  - Tümörlerin yerlerinin tespitinde ve diğer patolojik bulgularda,
  - Doku hacimlerinin ölçülmesinde,
  - Bilgisayar güdümlü cerrahi operasyonlarda,
  - Teşhis koyulmasında,
  - Tedavi planlamasında,
  - Anatomik yapı çalışmalarında;
- Uydu görüntüleri aracılığıyla nesnelerin belirlenmesi (yollar, ormanlar vb.);
- Yüz tanımlama sistemleri;
- Parmak izi tanımlama sistemleri;
- Trafik kontrol sistemleri vb....

# K-means ile segmentasyon

- K-means algoritması bir kümeleme algoritmasıdır.
- Kümeleme algoritmaları otomatik olarak verileri daha küçük kümelere yada alt kümelere ayırmaya yarayan algoritmalarıdır.
- Algoritma istatistiksel olarak benzer nitelikteki kayıtları aynı gruba sokar.
- Bir elemanın yalnızca bir kümeye ait olmasına izin verilir.
- Küme merkezi kümeyi temsil eden değerdir.

# K-means ile segmentasyon

- Alogritmanın isminde yer alan “k” harfi, küme sayısını belirtir
- Algoritma, hata hesaplamada yaygın olarak kullanılan Karasel Hata Fonksiyonunu en aza indirgiyecek “k” küme sayısını da arar.
- Verilen “n” sayıdaki veri seti “k” tane kümeye bu hata fonksiyonunu en aza indirgeyecek şekilde yerleştirilir.
- Bu nedenle küme benzerliği kümedeki değerlerin ortalamaya yakınlıkları ile ölçülür. Bu da kümenin ağırlık merkezidir. Kümenin merkezinde yer alan değer kümenin temsilci değeridir ve medoid olarak adlandırılır.

# K means algoritması

- Burada en önemli iki amaç şudur:
- 1- Küme içindeki değerler birbirlerine en çok benzemeli,
- 2- Kümeler birbirine mümkün olduğunca benzememeli

# K means algoritması adımları

- Bu istekleri gerçekleştirmek için algoritma tarafından sırasıyla şu adımlar gerçekleştirilmelidir:
- 1- Sınıf merkezlerinin belirlenmesi
- 2- Örneklerin mesafelere göre sınıflandırılması
- 3- Yapılan sınıflandırma sonrasında yeni merkezlerin belirlenmesi
- 4- İstenilen hale gelinceye kadar 2. ve 3. adımların algoritmik olarak tekrarlanması.

```
X = [randn(100,2)*0.75+ones(100,2);  
randn(100,2)*0.5-ones(100,2)];  
  
figure;  
plot(X(:,1),X(:,2),'.');  
title 'Randomly Generated Data';  
opts = statset('Display','final');  
[idx,C] = kmeans(X,2,'Distance','cityblock',...  
'Replicates',5,'Options',opts);  
  
figure;  
plot(X(idx==1,1),X(idx==1,2),'r','MarkerSize',12)  
hold on  
plot(X(idx==2,1),X(idx==2,2),'b','MarkerSize',12)  
plot(C(:,1),C(:,2),'kx', 'MarkerSize',15,'LineWidth',3)  
legend('Cluster 1','Cluster 2','Centroids','Location','NW')  
title 'Cluster Assignments and Centroids'  
hold off
```

# K means örnek

