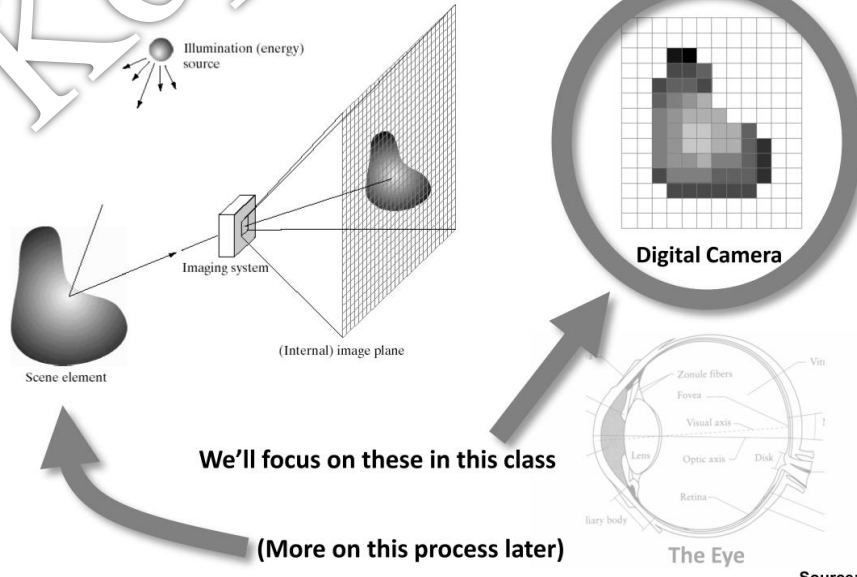


# Görüntü İşleme

What is an image?

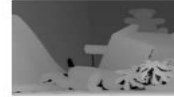
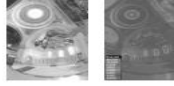


# Görüntü işlemek için neler yapılır?



- ♦ Gürültü” Temizlemek
- ♦ Parlaklığı Ayarlamak
- ♦ Koyuluk Ayarlamak
- ♦ Görüntü keskinleştirmek ve bulanıklaştırmak
- ♦ Doğru renk ayarlamak

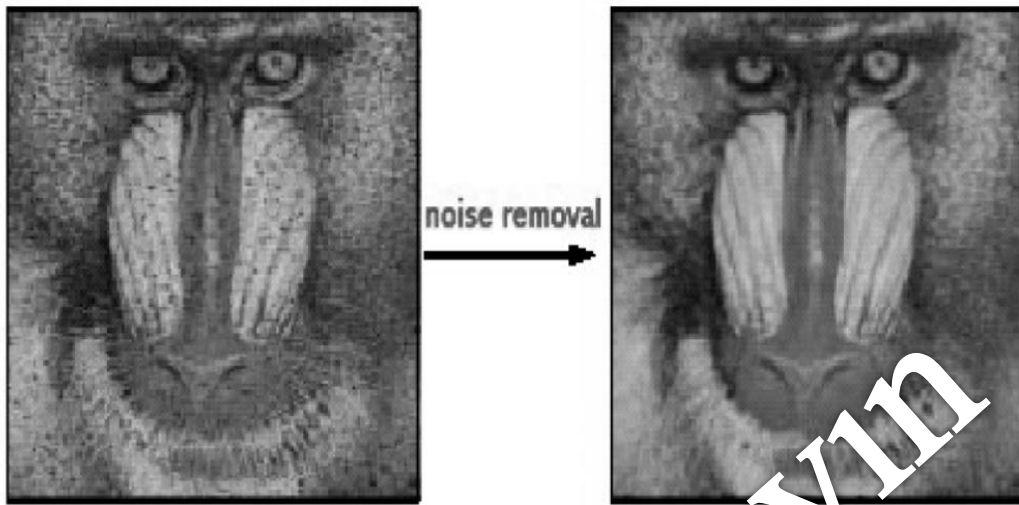
1 Image formation	29
Geometric primitives and transformations • Photometric image formation • The digital camera	
3 Image processing	99
Point operators • Linear filtering • More neighborhood operators • Fourier transforms • Pyramids and wavelets • Geometric transformations • Global optimization	
4 Feature detection and matching	205
Points and patches • Edges • Lines	
5 Segmentation	267
Active contours • Split and merge • Mean shift and mode finding • Normalized cuts • Graph cuts and energy-based methods	
6 Feature-based alignment	309
2D and 3D feature-based alignment • Pose estimation • Geometric intrinsic calibration	
7 Structure from motion	343
Triangulation • Two-frame structure from motion • Factorization • Bundle adjustment • Constrained structure and motion	



<b>8 Dense motion estimation</b>	<b>381</b>
Translational alignment • Parametric motion • Spline-based motion • Optical flow • Layered motion	
<b>9 Image stitching</b>	<b>427</b>
Motion models • Global alignment • Compositing	
<b>10 Computational photography</b>	<b>467</b>
Photometric calibration • High dynamic range imaging • Super-resolution and blur removal • Image matting and compositing • Texture analysis and synthesis	
<b>11 Stereo correspondence</b>	<b>533</b>
Epipolar geometry • Sparse correspondence • Dense correspondence • Local methods • Global optimization • Multi-view stereo	
<b>12 3D reconstruction</b>	
Shape from X • Active range finding • Surface representations • Point-based representations • Volumetric representations • Model-based reconstruction • Recovering texture maps and albedos	
<b>13 Image-based rendering</b>	<b>619</b>
View interpolation • Layered depth images • Light fields and Lumigraphs • Environment maps • Video-based rendering	
<b>14 Recognition</b>	<b>655</b>
Object detection • Face recognition • Instance recognition • Category recognition • Context-aware scene understanding	

## Görüntü İşlemenin Yararları- 1

- ♦ Resimler genellikle analog ortamlardan dijital ortamlara geçirildiği için bozukluk (noise) içerir. Görüntü işleme bu hataları düzeltmek için kullanılabilir.

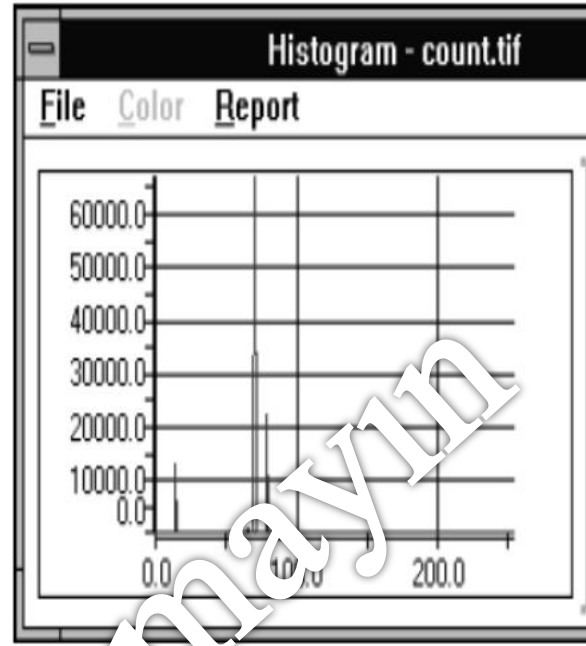
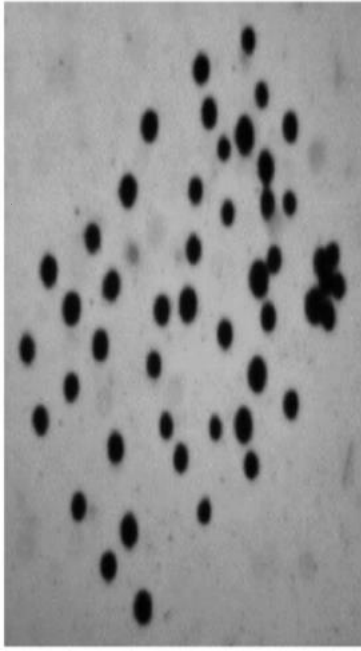
[illegible]

## Bir görüntünün Koordinatları

0,0 X

Y

	X-1,Y-1	X,Y-1	X+1,Y-1		
	X-1,Y	X,Y	X+1,Y		
	X-1,Y+1	X,Y+1	X+1,Y+1		



-Bir görüntüdeki kan sayısını ölçmek.

-Zarfların üzerindeki Zp-Kodun okunması

<b>R=10</b> <b>G=15</b> <b>B=5</b>	<b>R=1</b> <b>G=30</b> <b>B=5</b>	<b>R=10</b> <b>G=15</b> <b>B=5</b>	<b>R=20</b> <b>G=5</b> <b>B=10</b>
R=30 G=50 B=10	R=30 G=50 B=10	R=30 G=40 B=10	R=80 G=100 B=250
R=240 G=55 B=255	R=200 G=200 B=100	R=3 G=60 B=7	R=120 G=90 B=10
R=100 G=50 B=7	R=30 G=50 B=10	R=200 G=55 B=25	R=45 G=60 B=15

image

Slayt  
12

10	12	10	12
30	30	27	143
183	167	23	73
52	30	93	40

Grayscale image

Slayt  
13

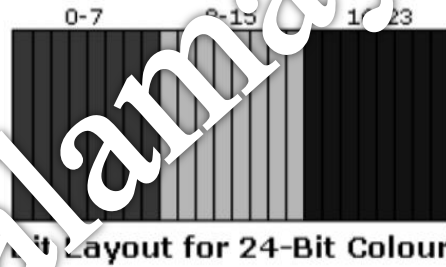






- Değişik renk paletleri kullanılarak aynı resim değişik şekillerde ifade edilir.

- Renkli resimlerde ayrı ayrı RGB değerlerinin tutulması yerine palet tutulması , bu bakımdan daha avantajlıdır.



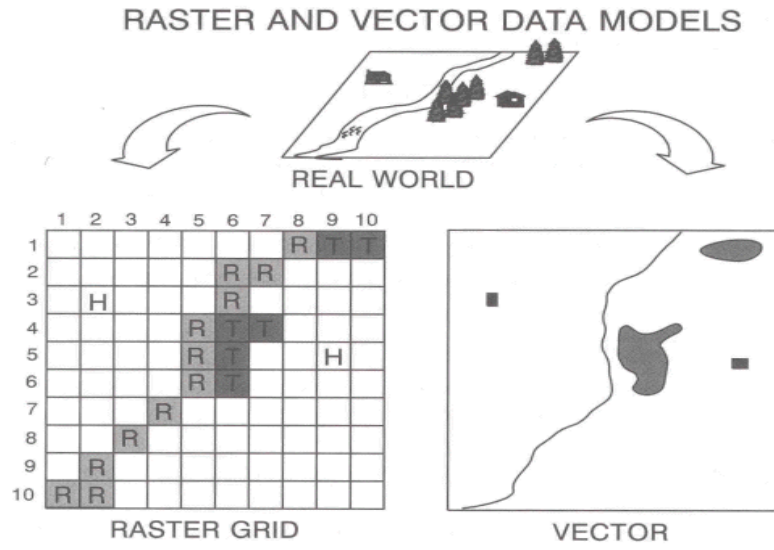


## Grafik nedir?

Gerçek hayattaki görüntülerin ve değerlerin bilgisayar ortamına aktarılmış haline grafik denir. Hareketsiz dir. Digital veri biçimindedir

### Grafik 2 ye ayrılır

- Bitmap tabanlı
- Vektör tabanlı



## Grafik üzerinde yapılan işlemler:

- Büyütme küçültme
- Yer değiştirme
- Döndürme
- Eğdirme
- Renk değiştirme

# GEOMETRİK DÖNÜŞÜMLER

--2 boyutlu düzlemde--

- Dönüşümler, nesnelerin **koordinat** tanımlarının değiştirilmesi için yapılan işlemlerdir.
- Nesnelerin **konum, boyut, şekil** gibi özelliklerinin değiştirilmesi gerektiğinde kullanılırlar.
- Temel geometrik dönüşümler kapsamına **yer değiştirme (translation)**, **döndürme (rotation)** ve **ölçeklendirme (scaling)** girer.

## 1. Ölçeklendirme

- Nesnelerin boyutunu değiştiren dönüşümdür. Noktaların koordinat değerlerinin, ölçeklendirme faktörleri ile çarpılması ile elde edilir.

- $x' = x \cdot s_x$
- $y' = y \cdot s_y$

## Lineer-Non Lineer Ölçeklendirme

Dönüşüm denklemleri,

$$\begin{matrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \\ \mathbf{P'} \end{matrix} = \begin{matrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{S} \\ \text{ölçeklendirme matrisi} \end{matrix} \cdot \begin{matrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ \mathbf{P} \end{matrix} = \begin{matrix} \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \\ 1 \end{bmatrix} \end{matrix}$$

$s_x$  ve  $s_y$ 'nin 1'den küçük değerleri nesnenin boyutunu küçültür (küçültme işlemi)

$s_x$  ve  $s_y$ 'nin 1'den büyük değerleri nesnenin boyutunu büyütür (büyütme işlemi)

$s_x$  ve  $s_y$ 'nin 1'e eşit olması nesnenin boyutunu değiştirmez

$s_x = s_y$  yani her ikisine de aynı değerin verilmesi ölçeklendirilmiş nesnin göreceli oranlarının aynı kalmasını sağlar ki buna düzgün ölçeklendirme denir

## Sorular

- W=200, H=300 piksel görüntü boyutu 0.5 büyütülürse ne olur?
- W=400, H=600 görüntü 3 kez büyütülürse son boyutu ne olur?
- W=1600, H=1200 görüntü 4 kez küçültülürse son boyutu ne olur?

## 2.Yer Değiştirme

- Bir nesnenin bir doğru üzerinde **bir noktadan bir noktaya** getirilmesidir.
- İki boyutlu **bir uzayda bir noktanın yeri**, **noktanın ilk konumuna  $P(x,y)$** , **yer değiştirme uzaklıkları eklenerek  $(t_x, t_y)$**  değiştirilir.
- Noktanın yeni konumu  **$P'(x',y')$**  olur.

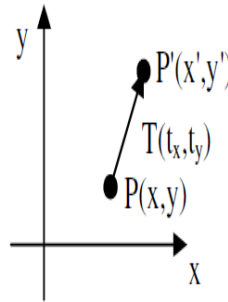
### Yer değiştirme geometrik gösterimi

$P(x,y)$  : Noktanın ilk konumu  
 $P'(x',y')$  : Noktanın yer değiştirme işleminden sonraki konumu  
 $T(t_x, t_y)$  : Yer değiştirme uzaklıkları

olmak üzere

$$\begin{aligned}x' &= x + t_x \\ y' &= y + t_y\end{aligned}$$

olarak hesaplanır.



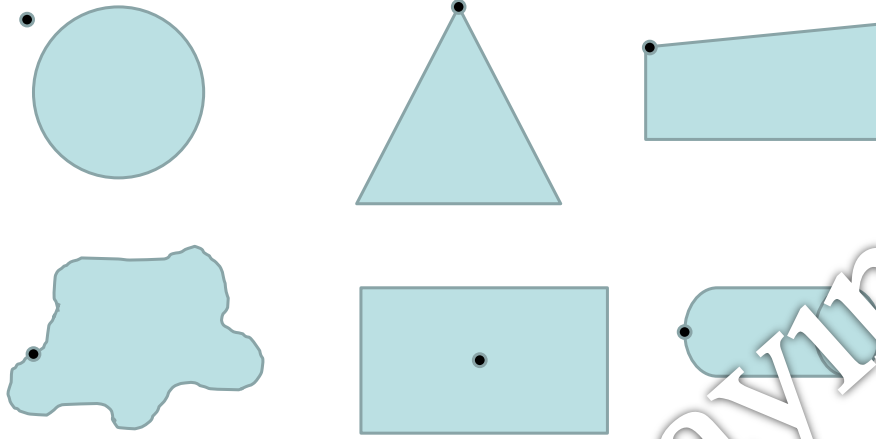
Şekil 2.1 : Bir noktanın P konumundan P' konumuna getirilmesi

$$P = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$P' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix}$$

$$T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

## Nesneler Merkez Noktaları



### Yer deęiřtirme DÖNÜŐ

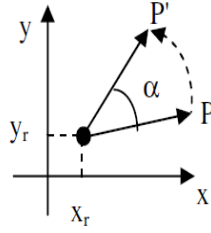
- $P' = T + P$  ekleinde ifade edilir.
- Yer deęiřtirme dönüőünde nesnelerin **her noktası eőit miktarlarda** hareket ettirildięinden nesnelerde bozulma olmaz.
- Çokgenlerin bir konumdan **dięer bir konuma tařınması** tüm köőelerine yer deęiřtirme dönüőünün uygulanması yeterli olur.
- Daire gibi eęrisel nesnelerin hareketinde **sadece merkez koordinatlarının yeri deęiřtirilerek** nesne yeniden çizdirilir.

### 3.Döndürme

- Döndürme bir noktanın **dairesel** bir yolda hareket ettirilmesi işlemidir.
- Döndürme işlemi için  **$q$  (döndürme açısı)** ve döndürme işleminin **yapılacağı  $(x_r, y_r)$  sabit (pivot) noktasının (döndürme noktası)** belirtilmesi gerekir.

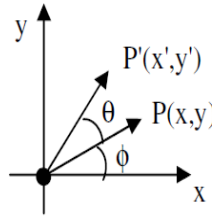


## Döndürmenin Geometrik Gösterimi



Şekil 2.2 : P noktasının sabit bir nokta etrafında döndürülmesi

Döndürme işleminden sonraki koordinatlar  $\theta$  ve  $\phi$  açıları cinsinden ifade edilebilir (Şekil 2.3).



Şekil 2.3 : P(x,y) noktasının orijin (koordinat başlangıç noktası) etrafında döndürülmesi

## Döndürmenin fonksiyonel gösterim

- P(x,y) noktasının orijin (koordinat başlangıç noktası) etrafında döndürülmesi sonucunda noktanın yeni konumu **P'(x',y')** bu şekilde hesaplanır.
- **$x' = r \cos(f+q) = r \cos f \cos q - r \sin f \sin q$**
- **$y' = r \sin(f+q) = r \cos f \sin q + r \sin f \cos q$**
- Noktanın kutupsal koordinatlarda ilk konumu
- $x = r \cos f, y = r \sin f$

## Döndürmenin matrissel gösterim.

x ve y yerine konulduğunda,

$$x' = x \cos\theta - y \sin\theta$$

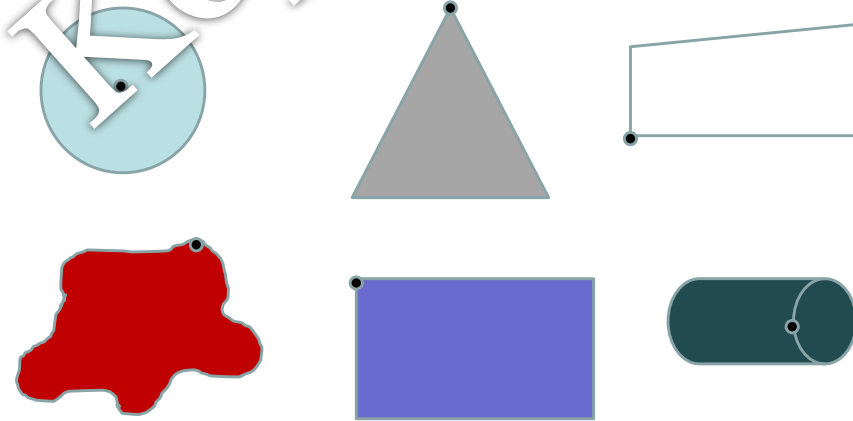
$$y' = x \sin\theta + y \cos\theta$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

R döndürme matrisi olmak üzere döndürme denklemlerinin matris şeklindeki ifadesi aşağıdaki gibidir :

$$P' = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \cos\theta - y \sin\theta \\ x \sin\theta + y \cos\theta \\ 1 \end{bmatrix}$$

## Nesnelerin Pivot Noktaları



## Pixel nedir.

**Pixel:** Ekranda kontrol edilebilen ve resimleri tanımlayabilen en küçük noktalara (ki bu noktalar kare şeklindedir) pixel denir.

**\*\*Bir pixel kırmızı, yeşil ve mavi renklerin karışımından oluşur.**

**\*\*Pixellerin sayısı ve renk derinliği resmin kalitesini belirler.**

**\*\*Pixel tek başına bir ölçüyü ifade etmez.**

**\*\*Verilen pixel sayısı belirtilen alan içerisindeki pixel sayısıdır.**

## Pixel üzerine hesaplamalar

- Soru: 10 cm de 10 piksel bulunursa 1 pikselin boyutu kaç mm'dir?
- Örneğin;
- 10 cm'deki pixel sayısı 10 denilirse;
- $10 \text{ cm} = 100 \text{ mm}$ ,
- Sonuç→
- pixel boyutu:  $100/10 = 10 \text{ mm'dir}$

## Piksel işlemleri

- Bir mesafede kaç Piksel bulunur.
- Bir alanda kaç Piksel bulunur.

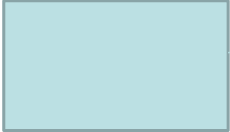
## Örnek Soru

- 1 pixelin bir kenarının boyutu 0,12 cm ise
- 20 adet pixelin yatayda oluşturabileceği mesafe kaç mm'dir?

1 cm      10 mm  
0,12 cm   x mm'dir

-----  
X = 1,2 mm -->

- → 1 pixel      1, 2 mm genişliğe sahipse  
20 pixel      X mm genişliğe sahiptir.
- -----
- $X = 20 \cdot 1,2 = 24$  mm'dir.
- Yukarıdaki soruyla ilgili...: eldeki değerlere göre
- 96 mm'lik mesafede kaç piksel bulunur?

-  → 36 mm\*12 mm de kaç piksel bulunur?

## Örnek Soru

- 1 pixelin yatay kenarı 0,6 mm dir. 48 mm'lik dikey mesafeye kaç adet pixel yerleştirilebilir? (varsayılan değerleri kullanın)
- 1 pixelin yatay kenarı 0,6 mm ise varsayılan değerlere göre kare biçimlidir.
- Yüksekliği de 0,6 mm dir.

- 0,6 mm de 1 pixel var ise
- 48 mm de X pixel vardır

-----  
 $X \cdot 0,6 = 48 \rightarrow x = 480 / 6 = 80$  piksel bulunur.

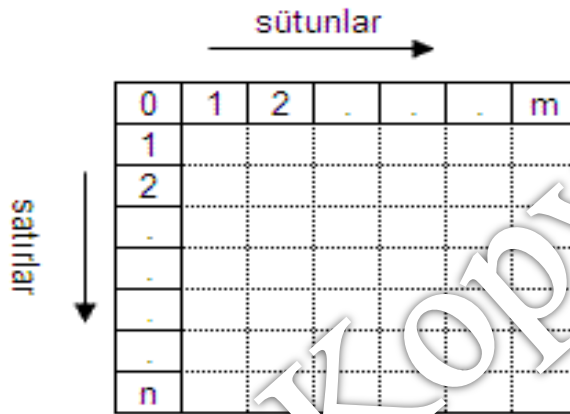
b-) Sorunun devamı olarak 72mm\*36mm lik görüntü üzerinde kaç piksel vardır?



## Örnek Sorular

- 240 mm dikey uzunlukta 480 adet pixel var ise bir pixelin genişliği kaç cm dir?
- Bir pixelin yüksekliği 0,125 cm ise 400 adet pixelin yatayda oluşturduğu mesafe kaç mm dir?
- 640 mm'lik dikey mesafede 0,320 cm bir kenarı olan kaç pixel yerleştirilebilir?

## Pixel Satır ve Sütunlar...





## Görüntü Ortalaması

RGB

50	40
40	80
100	20
70	80
60	200
240	200

Görüntü Ortalaması

60	60
95	95
140	140
60	60
95	95
140	140

$$\text{Red}_{\text{ort}} = 50 + 40 + 70 + 80 = 240 / 4 = 60$$

$$\text{Green}_{\text{ort}} = 40 + 80 + 60 + 200 = 380 / 4 = 95$$

$$\text{Blue}_{\text{ort}} = 100 + 20 + 240 + 200 = 560 / 4 = 140$$

Örnek Soru

RGB

100	20	30
30	20	30
80	20	90
60	180	40
60	200	40
120	100	100
80	150	210
90	180	120
40	120	150

Görüntü Ortalaması

?	?	?
?	?	?
?	?	?

## RGB -> Gri tonları

- ♦ RGB değerleri, gri tonlarına şu formülle dönüştürülür:

$$I_{grey}(p) = \frac{I_R(p) + I_G(p) + I_B(p)}{3}$$

## Gri Tonları -> İkili Resimler

- ♦ Gri tonları ise ikili resimlere şu formülle dönüştürülür:

$$I_{bin}(p) = \begin{cases} 1 & \text{if } I_{grey}(p) \geq d \\ 0 & \text{otherwise} \end{cases}$$

- ♦ Burada d belirli bir eşik değeridir ve bu değer, çevirim için ana noktadır. (**Threshold**)
- ♦ Eşik noktalarının kullanılması hesaplama işlemini kolaylaştırırken bilgiyi yeterli kullanmaması ve genellikle elle girilen bir değer oluşturması bir dezavantajdır.

RGB

100	20	30
30	20	30
80	20	90
60	180	40
60	200	40
120	100	100
80	150	210
90	180	120
40	120	150

Grayscale

70	20	50
80	160	60
70	150	160

Threshold ( $\geq 127$ )

0	0	0
0	1	0
0	1	1

RGB

100	20	30
30	20	30
80	20	90
60	180	40
60	200	40
120	100	100
80	150	210
90	180	120
40	120	150

Grayscale

70,70,70	20,20,20	50,50,50
80,80,80	160,160,160	60,60,60
70,70,70	150,150,150	160,160,160

Threshold ( $>127$ )-Bitmap

0,0,0	0,0,0	0,0,0
0,0,0	255,255,255	0,0,0
0,0,0	255,255,255	255,255,255

0,0,0	0,0,0	0,0,0
0,0,0	255,255,255	0,0,0
0,0,0	255,255,255	255,255,255

RGB

5	10	50
5	10	50
5	10	50
80	70	50
80	70	50
80	70	50
90	105	140
90	105	140
90	105	140

Grayscale

5	10	50
80	70	50
90	105	140

Threshold ( $\geq 80$ )

0	0	0
255	0	0
255	255	255

0	0	0
255	0	0
255	255	255

KOPYALAMAYIN

KOPYALAMAYIN