



BSM 409

GÖRÜNTÜ İŞLEME

Hafta - 4

Geometrik Dönüşümler

Doç. Dr. Eftal ŞEHİRLİ

Bilgisayar Mühendisliği Bölümü

İçerik

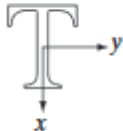





- Geometrik Dönüşümler
 - Ölçekleme (Scaling)
 - Döndürme (Rotation)
 - Aynalama (Mirroring, Flipping)
 - Taşıma - Öteleme (Move, Translation)
 - Eğme-Bükme (Shear)

Geometrik Dönüşümler

Koordinatların Dönüşümü

- Geometrik dönüşümler resim üzerindeki her pikselin bir konumdan (x_1, y_1) başka bir konuma (x_2, y_2) haritalanmasıdır.
- Temel İşlemler
 - Ölçekleme (Scaling)
 - Döndürme (Rotation)
 - Aynalama (Mirroring - Flipping)
 - Taşıma - Öteleme (Moving - Translation)
 - Eğme-Bükme (Shear)
- $$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = [A] \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + [B]$$
- A matrisi Scaling, Rotation, Mirroring ve Shear için kullanılır.
- B matrisi Translation işlemi için kullanılır.

Özet – Koordinatların Dönüşümü

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v \cos \theta - w \sin \theta$ $y = v \sin \theta + w \cos \theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Interpolasyon Yöntemleri

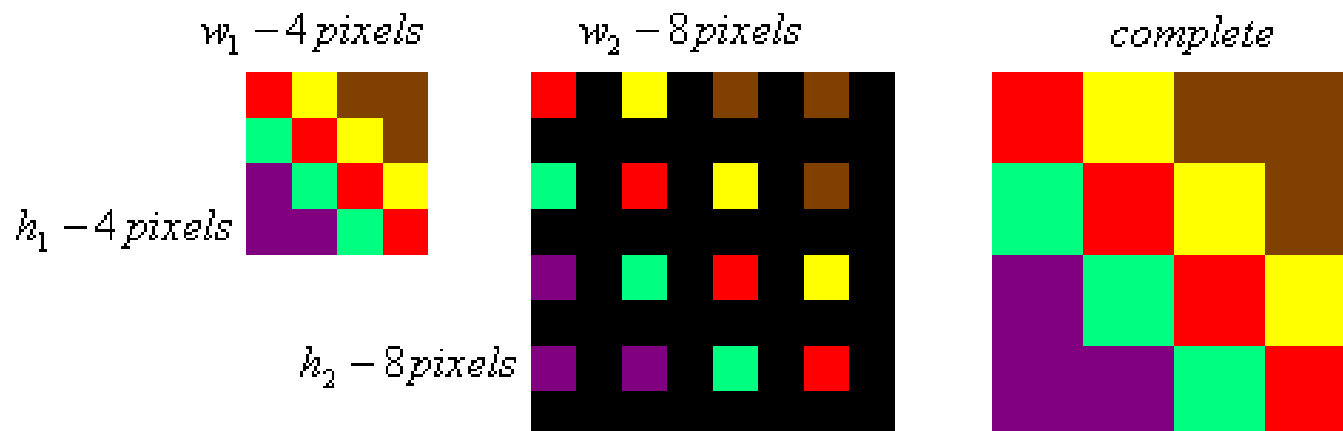
- *Bilinmeyeni tahmin etmek için bilineni kullanmak*
- Yöntemler:
 - En yakın komşu interpolasyon (Nearest neighbor interpolation)
 - Çift lineer interpolasyon (Bilinear interpolation)
 - Çift kübik interpolasyon (Bicubic interpolation)

En Yakın Komşu

- Eksikleri tamamlamak için en yakın komşu kullanılır.
- Hızlı ön izlemelerde, gerçek zamanlı uygulamalarda tercih edilir.

✓ Avantaj: Hızlı, kolay.

✗ Dezavantaj: Köşeler **bloklı, pikseli** görünür.



En Yakın Komşu

a	b
c	d

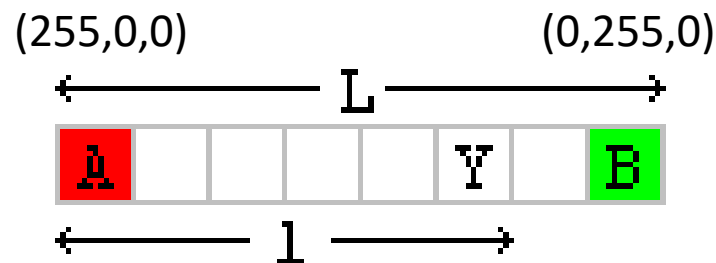
En Yakın Komşu

a	b
c	d

a	a	b	b
a	a	b	b
c	c	d	d
c	c	d	d

Linear Interpolasyon

- Bilinen iki nokta arasındaki her hangi bir noktanın değerini bulmak.



$$\frac{Y - A}{l} = \frac{B - A}{L}$$
$$Y = A + \frac{l(B - A)}{L}$$

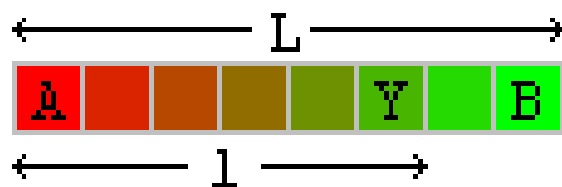
Y pikselinin renk değeri nedir?

Linear Interpolasyon

- Bilinen iki nokta arasındaki her hangi bir noktanın değerini bulmak.

$$\frac{Y - A}{l} = \frac{B - A}{L}$$

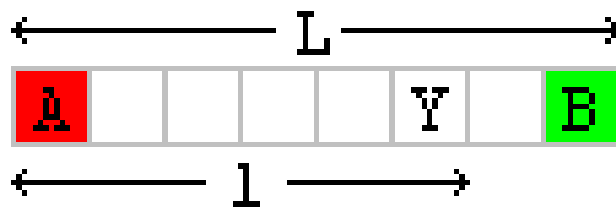
$$Y = A + \frac{l(B - A)}{L}$$



1	red	green	blue
0	255	0	0
1	218	36	0
2	182	72	0
3	145	109	0
4	109	145	0
5 (Y)	72	182	0
6	36	218	0
7	0	255	0

Linear Interpolasyon

- Bilinen iki nokta arasındaki her hangi bir noktanın değerini bulmak.





$$\frac{Y - A}{l} = \frac{B - A}{L}$$
$$Y = A + \frac{l(B - A)}{L}$$

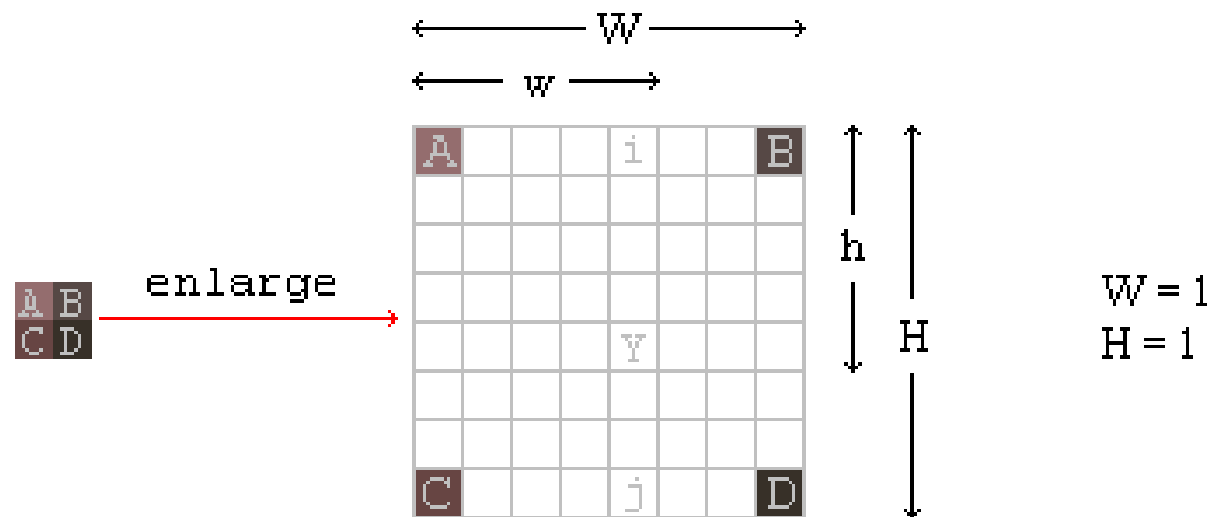
R=72 G=182 B=0



Bilinear Interpolasyon

- Çevreleyen pikseller kullanılır (en yakın 4 komşu). İki yönlü(x ve y) doğrusal (linear) interpolation birleşimi yapılır.
- Dengeli kullanımlar için önerilir.
-  Avantaj: Daha **yumuşak** geçişler. Daha **fazla** detay.
-  Dezavantaj: Biraz **bulanık** olabilir. Daha **fazla** süre.

Bilinear Interpolasyon



$$\frac{i - A}{w} = \frac{B - A}{W}$$

$$i = A + \frac{w(B - A)}{W}$$

$$i = A + w(B - A) \rightarrow 1$$

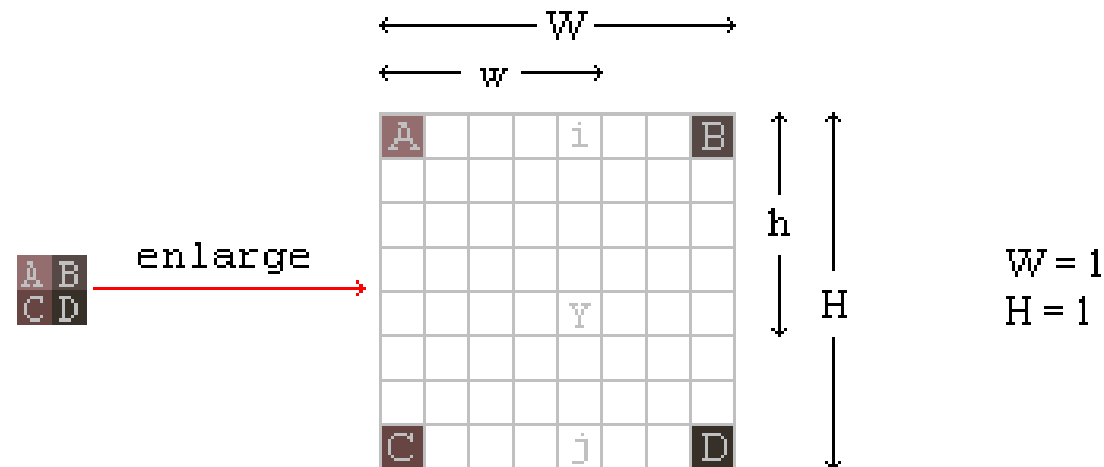
$$\frac{j - C}{w} = \frac{D - C}{W}$$

$$j = C + w(D - C) \rightarrow 2$$

$$\frac{Y - i}{h} = \frac{j - i}{H}$$

$$Y = i + h(j - i) \rightarrow 3$$



Bilinear Interpolasyon

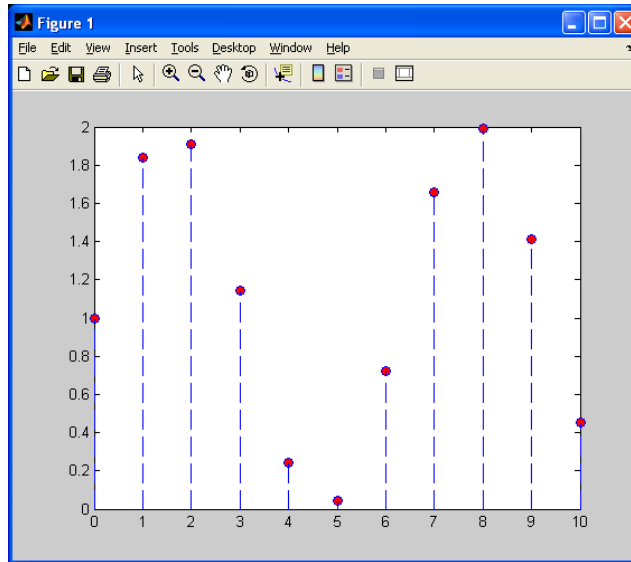


$$Y = A + w(B - A) + h(C + w(D - C) - (A + w(B - A)))$$

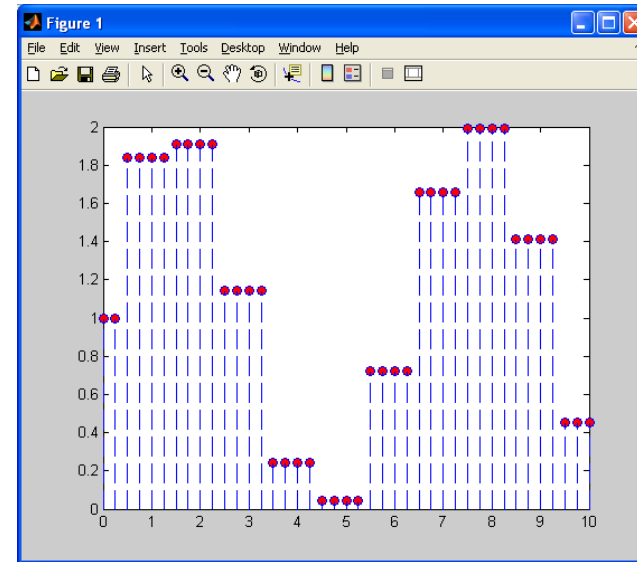
$$Y = A(1 - w)(1 - h) + B(w)(1 - h) + C(h)(1 - w) + D(wh)$$

Bikübik Interpolasyon

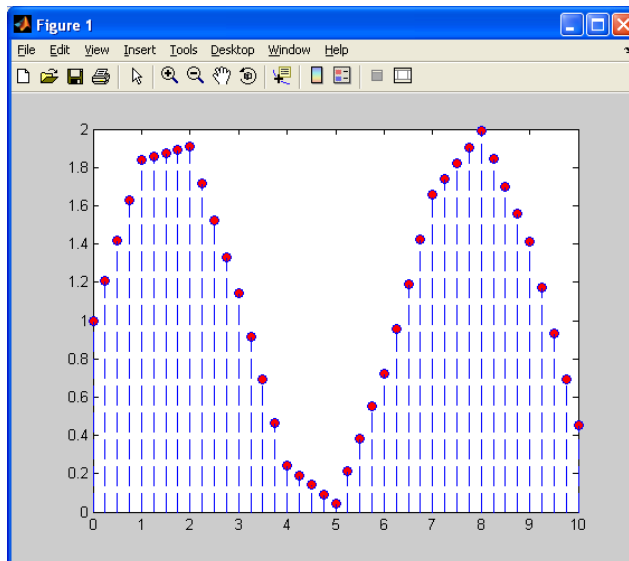
- Yeni piksel, **çevresindeki 16 pikselin (4×4 komşuluk)** kübik polinomlarla hesaplanan ağırlıklı ortalaması alınarak bulunur.
- Daha karmaşık matematik → kübik spline fonksiyonları.
- Sonuç: çok daha **pürüzsüz** geçişler.
- Detayları daha iyi korur.
- En iyi örnekleme için kullanılır. Photoshop standardı.
- Örneğin, 100–150 geçişinde yeni piksel değeri **125 yerine ~122–128 arası** çıkabilir (çünkü kübik eğri düz ortalama yerine eğimli geçiş yapar).
-  Avantaj: Çok kaliteli, doğal görünümlü.
-  Dezavantaj: Hesaplama maliyeti yüksek.



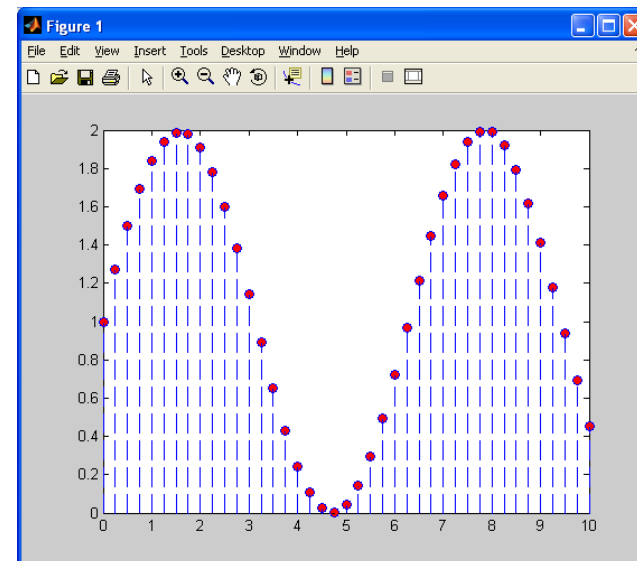
Original



En Yakın Komşu



Bilinear



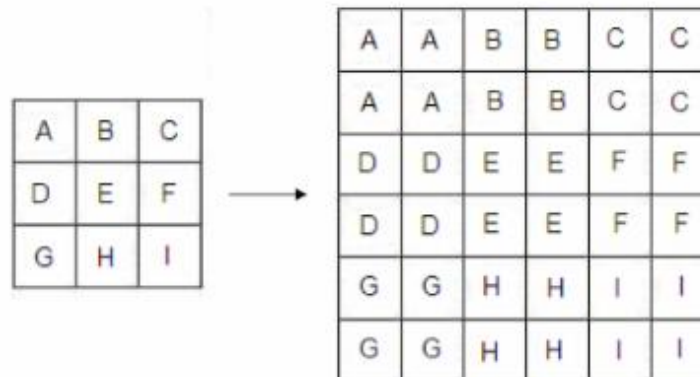
Bicubic

Orijinal – En Yakın Komşu – Bilinear - Bicubic



Ölçekleme – Scaling – Zoom in

- Görüntünün belirli kurallar dahilinde boyutunun değiştirilmesidir.
- Yeni pikseller üretmek gerekir.
- Yöntemler: En yakın komşu (nearest neighbor), bilinear, bicubic, ...
- Örnek 2x büyütme – 2x zoom in (nearest neighbor)



Ölçekleme – Scaling – Zoom out

- Yöntemler: Nearest, bilinear, bicubic, ...

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

$\frac{A+B+E+F}{4}$	$\frac{C+D+G+H}{4}$
$\frac{I+J+M+N}{4}$	$\frac{K+L+O+P}{4}$

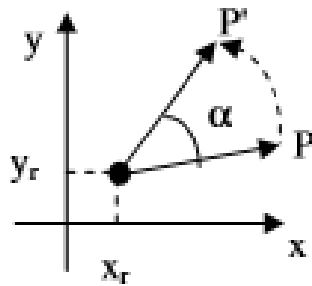


Soru?

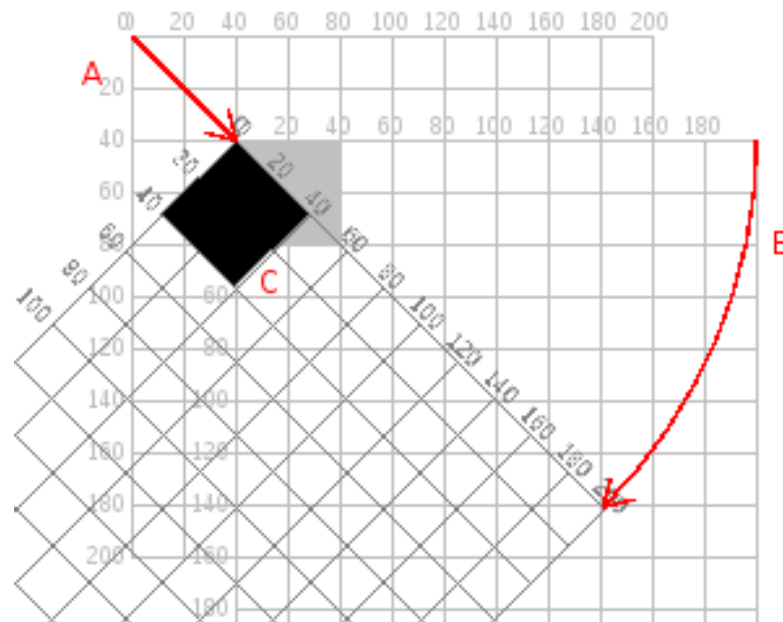
- Optik Zoom(Analog Zoom) vs Digital Zoom
- Hangisi daha iyi?

Rotation

- Pikselin dairesel bir yörünge üzerinde hareketidir.



$$x' = x \cos \alpha - y \sin \alpha$$
$$y' = x \sin \alpha + y \cos \alpha$$



Rotation (x) – Aynalama



Orjinal imge



Ters çevrilmiş imge

Rotation (γ) – Aynalama



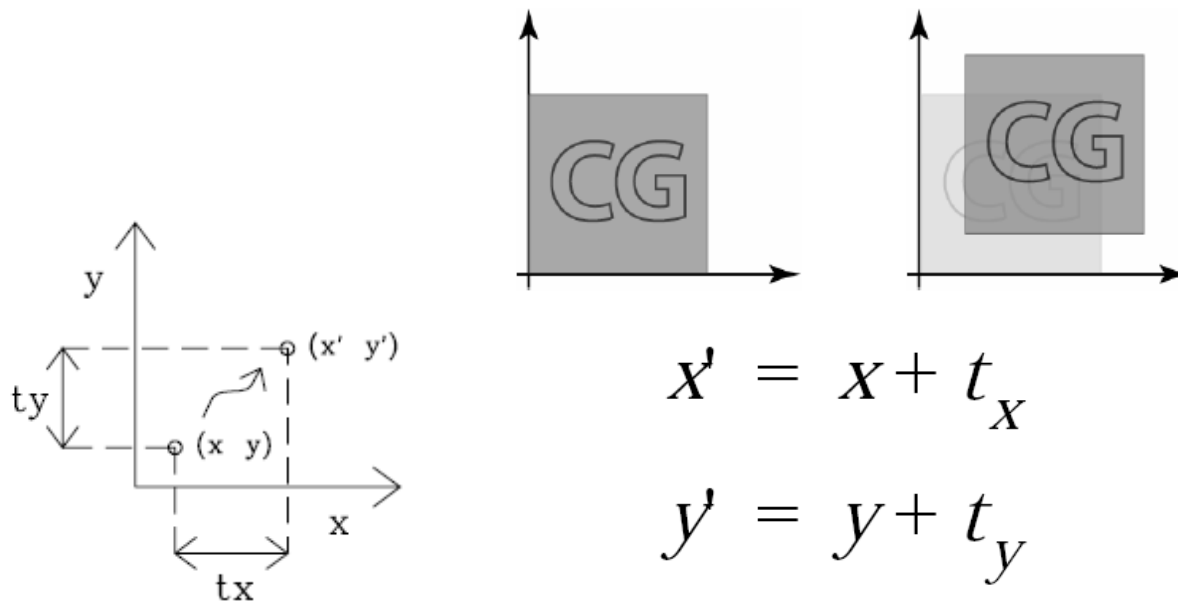
Orjinal imge



Aynalanmış imge

Translation – Taşıma - Öteleme

- Görüntünün belirli kurallar dahilinde konumunun değiştirilmesidir.



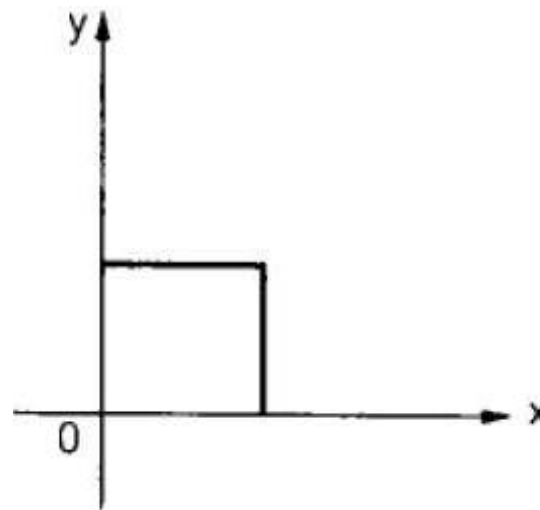
Vertical Shear

- Görüntünün y eksenin taşınması ile oluşturulur.

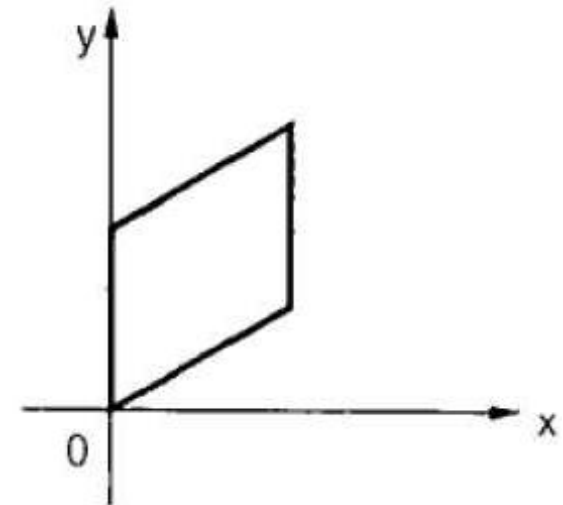
$$Y_{sh} \begin{bmatrix} 1 & sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Y' = Y + Sh_y \cdot X$$

$$X' = X$$



(a) Original object



(b) Object after y shear

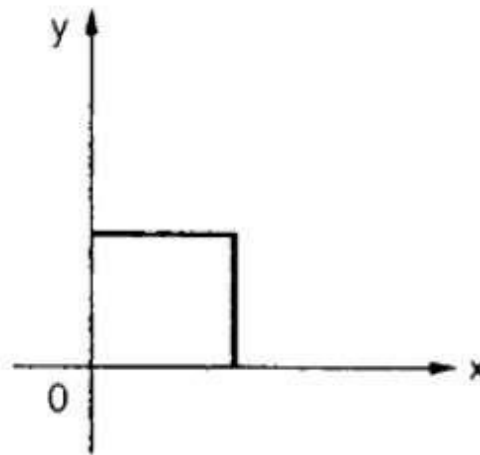
Horizontal Shear

- Görüntünün x eksenin taşınması ile oluşturulur.

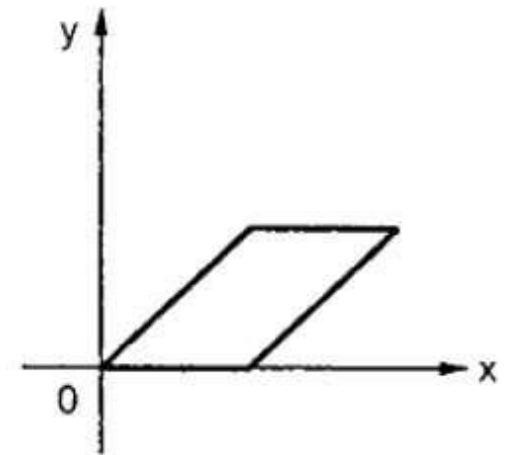
$$X_{sh} = \begin{bmatrix} 1 & 0 & 0 \\ shx & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X' = X + Sh_x \cdot Y$$

$$Y' = Y$$



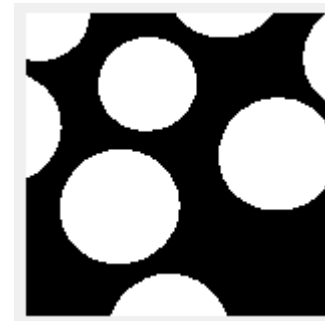
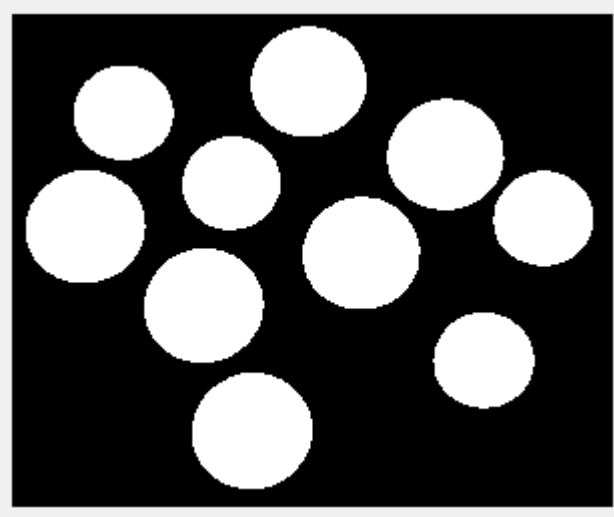
(a) Original object



(b) Object after x shear

Kırpma

- Görüntü üzerinde belli bir bölgenin elde edilmesidir.
- Diğer geri kalan bölgenin görüntüden atılmasıdır.



Uygulama

- Kırpma işlemi
- Rotate işlemi

Ödev

- Ölçekleme
- Taşıma
- Aynı görüntü üzerinde, hem kendi geliştireceğiniz programı hem de MATLAB'in hazır (built-in) fonksiyonlarını kullanarak geometrik dönüşüm işlemlerini gerçekleştiriniz.



- Ders Sonu