

# Computer Graphics

(SE Computer Engineering 2019-course)

## Unit III- 2D, 3D Transformations and Projections

*Happiness*  
is not the  
absence of  
problems, it's  
the ability to  
deal with them.

# Course Objectives:

The Computer Graphics course prepares students for activities involving the design, development and testing of modeling, rendering, and animation solutions to a broad variety of problems found in entertainment, sciences, and engineering.

- Remembering: To acquaint the learner with the basic concepts of Computer Graphics.
- Understanding: To learn the various algorithms for generating and rendering graphical figures.
- Applying: To get familiar with mathematics behind the graphical transformations.
- Understanding: To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting.
- Creating: To generate Interactive graphics using OpenGL.

# Course Outcomes:

On completion of the course, learner will be able to–

**CO1:** Identify the basic terminologies of Computer Graphics and interpret the mathematical foundation of the concepts of computer graphics.

**CO2:** Apply mathematics to develop Computer programs for elementary graphic operations.

**CO3:** Describe the concepts of windowing and clipping and apply various algorithms to fill and clip polygons.

**CO4:** Understand and apply the core concepts of computer graphics, including transformation in two and three dimensions, viewing and projection.

**CO5:** Understand the concepts of color models, lighting, shading models and hidden surface elimination.

**CO6:** Describe the fundamentals of and implement curves, fractals, animation and gaming.

# Learning Resources

## Text Books:

1. S. Harrington, “Computer Graphics”, 2<sup>nd</sup> Edition, McGraw-Hill Publications, 1987, ISBN 0 – 07 – 100472 – 6.
2. Donald D. Hearn and Baker, “Computer Graphics with OpenGL”, 4<sup>th</sup> Edition, ISBN-13: 9780136053583.
3. D. Rogers, “Procedural Elements for Computer Graphics”, 2<sup>nd</sup> Edition, Tata McGraw-Hill Publication, 2001, ISBN 0 – 07 – 047371 – 4

## Reference Books:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, “Computer Graphics Principles and Practice”, 2<sup>nd</sup> Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Rogers, J. Adams, “Mathematical Elements for Computer Graphics”, 2<sup>nd</sup> Edition, Tata McGraw Hill Publication, 2002, ISBN 0 – 07 – 048677 – 8.

# Learning Resources contd...

## e-Books:

- <https://open.umn.edu/opentextbooks/textbooks/introduction-to-computer-graphics>
- <http://www2.cs.uidaho.edu/~jeffery/courses/324/lecture.html>

## MOOC/ Video Lectures available at:

- <https://nptel.ac.in/courses/106/106/106106090/>
- <https://nptel.ac.in/courses/106/102/106102065/>

<b>Unit</b>	<b>Lect Content details as per syllabus</b>	
<b>III 2D, 3D Transformations and Projections</b>	<b>1</b>	<b>2-D transformations: introduction, homogeneous coordinates,</b>
	<b>2</b>	<b>2-D transformations : Translation, scaling, rotation and shear</b>
	<b>3</b>	<b>rotation about an arbitrary point. 3-D transformations: introduction,</b>
	<b>4</b>	<b>3-D transformations - Translation, scaling, rotation</b>
	<b>5</b>	<b>Shear transformation, rotation about an arbitrary axis.</b>
	<b>6</b>	<b>Projections : Parallel (Oblique: Cavalier, Cabinet and orthographic: isometric, diametric, trimetric)</b>
	<b>7</b>	<b>Perspective ( Vanishing Points – 1 point, 2 point and 3 point)</b>
<b>Exemplar/Case Studies</b>	<b>Study of transformations and Projections using different software like VRML, blender</b>	
<b>Course Outcomes</b>	<b>CO4</b>	

# Overview

## *2D Transformations*

- Basic 2D transformations

## *3D Transformations*

- Basic 3D transformations
- Matrix representation
- Matrix composition

# Transformation

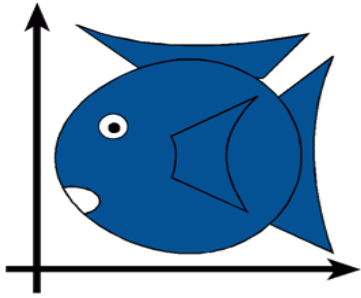
Maps points  $(x, y)$  in one coordinate system to points  $(x', y')$  in another coordinate system

$$x' = ax + by + c$$

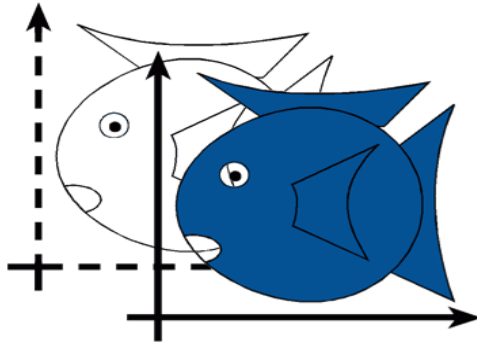
$$y' = dx + ey + f$$



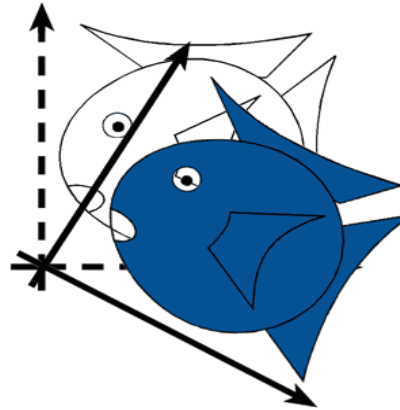
# Simple Transformations



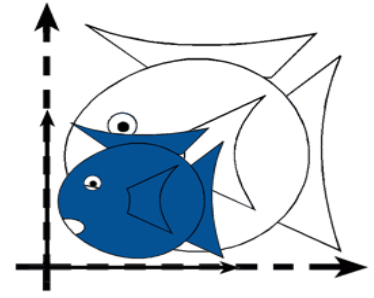
Identity



Translation



Rotation



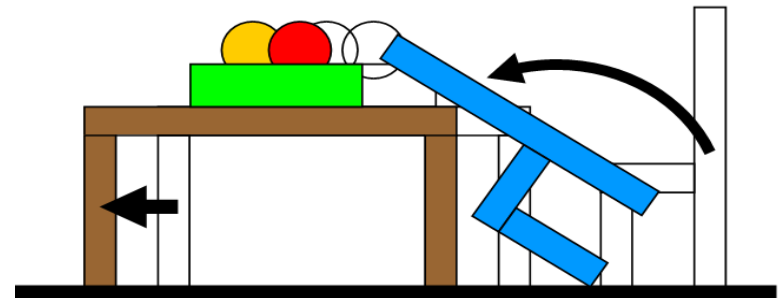
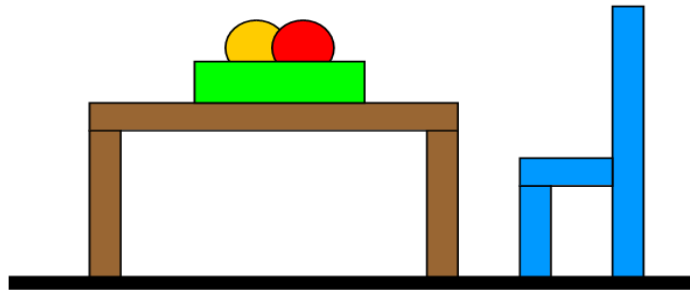
Isotropic  
(Uniform)  
Scaling

Can be combined

Are these operations reversible?

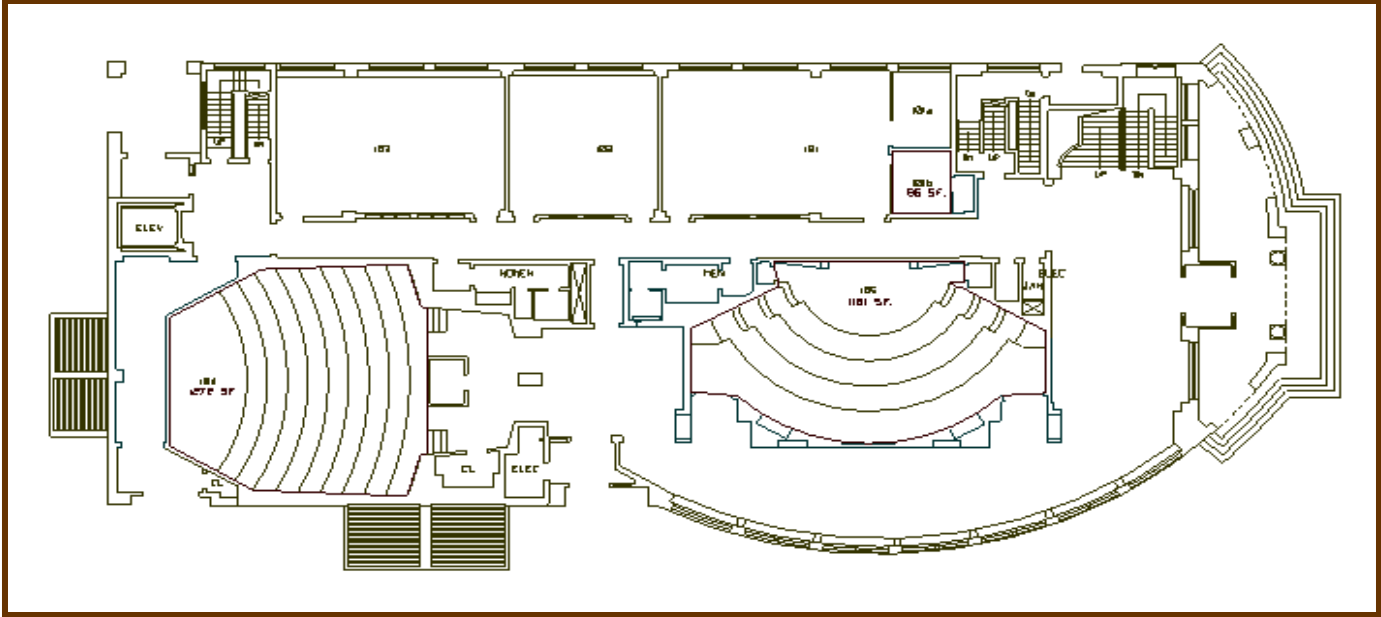
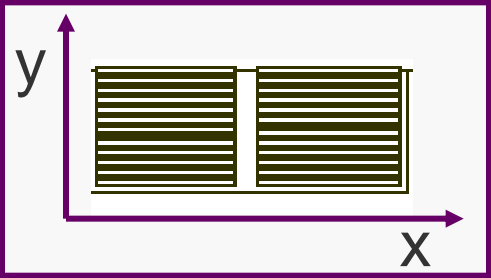
# Transformations are used to

- Position objects in a scene (modelling)
- Change the shape of objects
- Create multiple copies of objects
- Projection for virtual cameras
- Animations



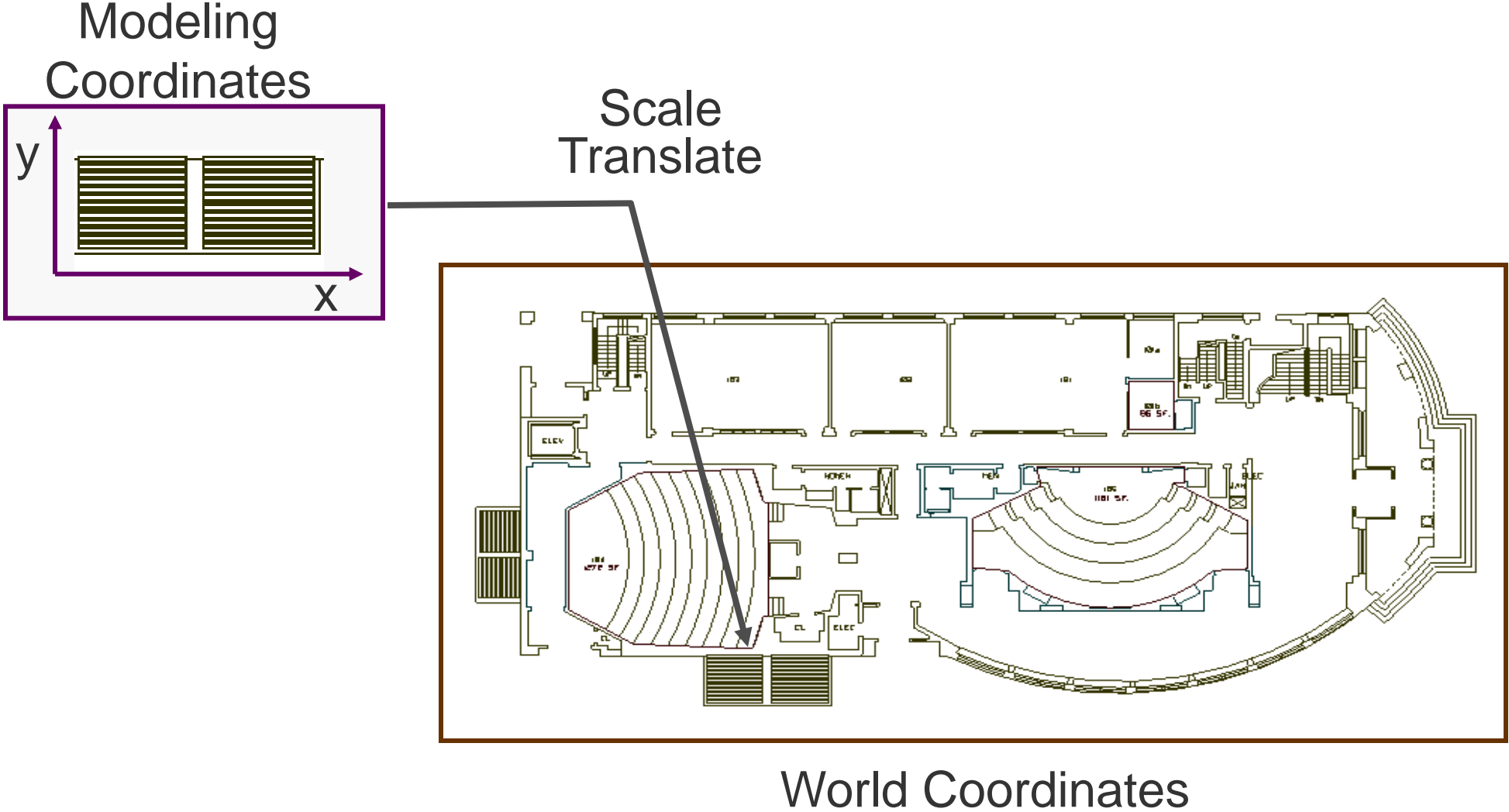
# 2D Modeling Transformations

Modeling  
Coordinates

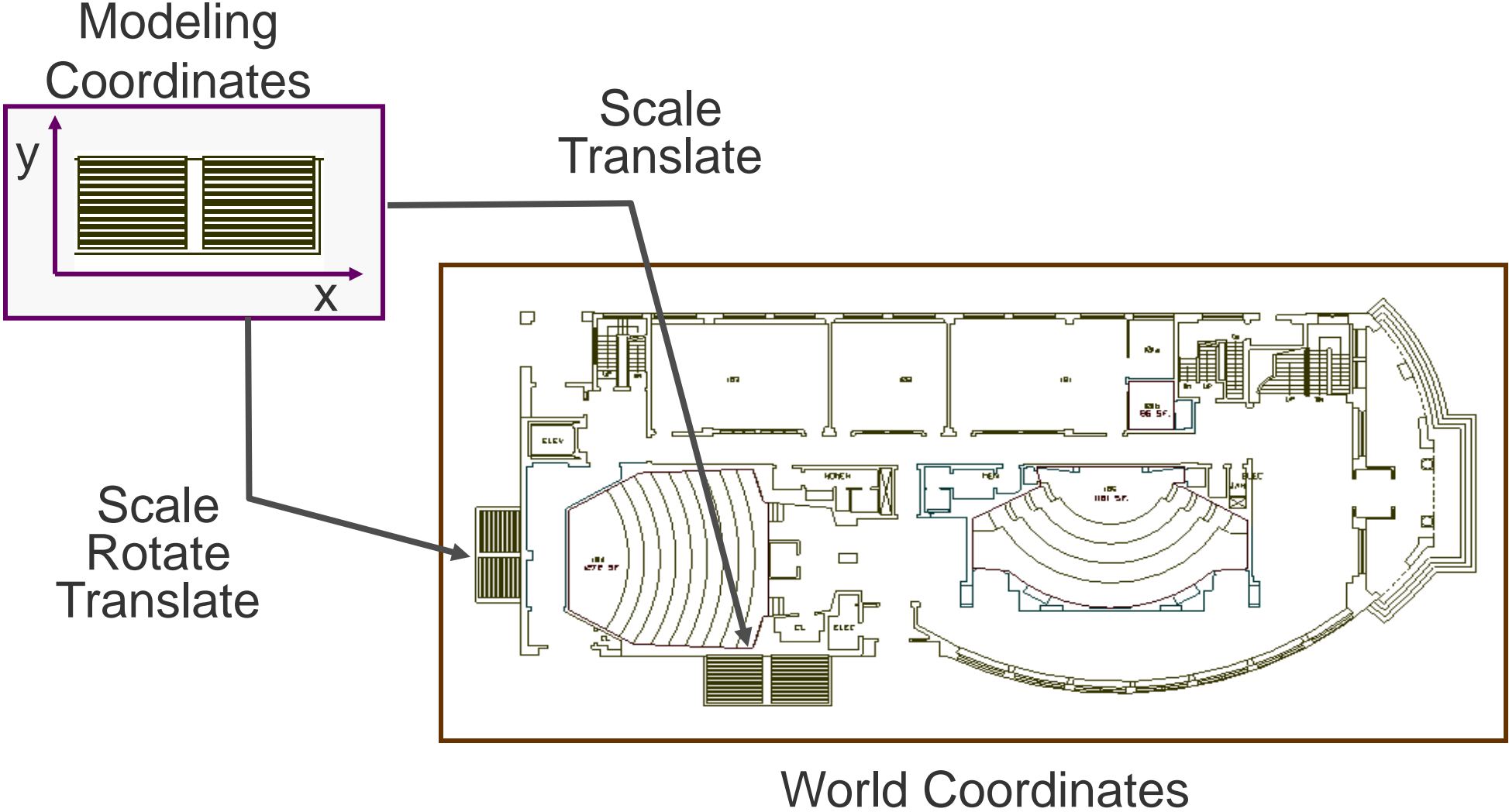


World Coordinates

# 2D Modeling Transformations

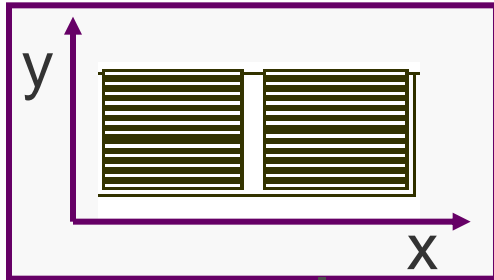


# 2D Modeling Transformations

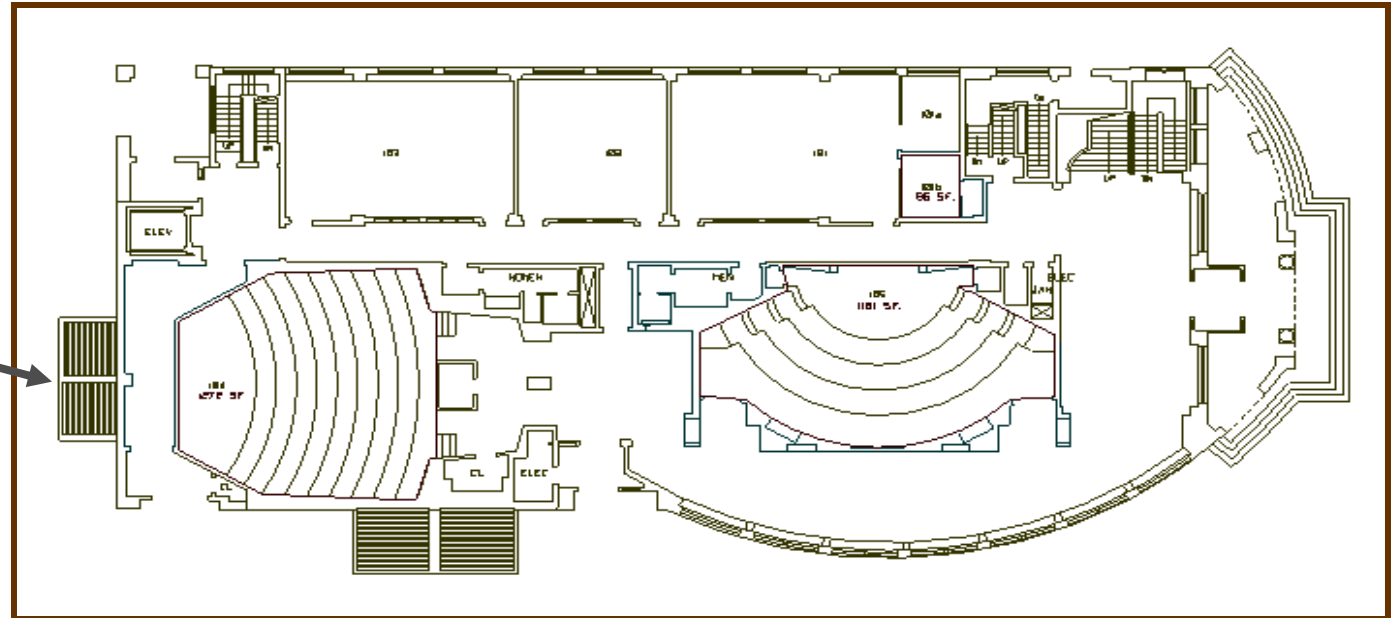


# 2D Modeling Transformations

Modeling  
Coordinates



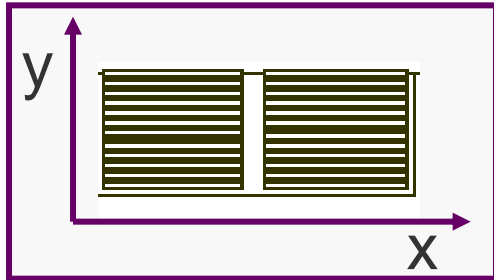
Let's look  
at this in  
detail...



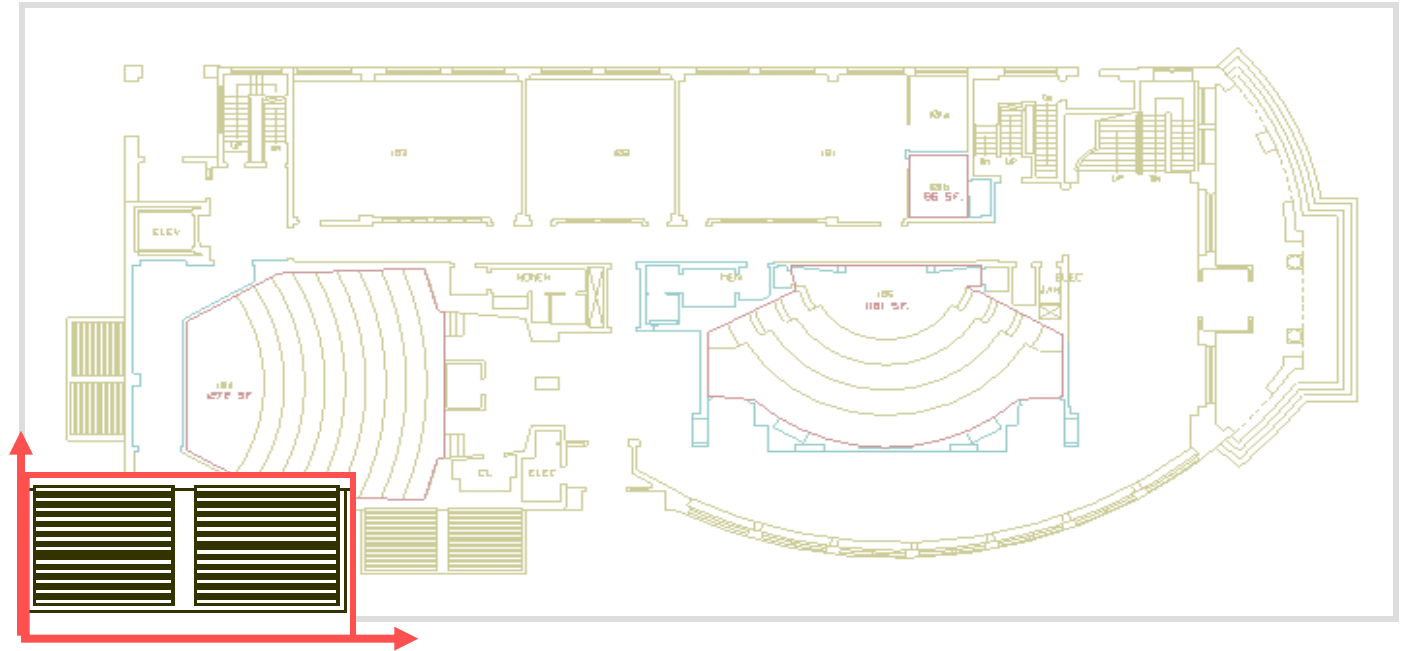
World Coordinates

# 2D Modeling Transformations

Modeling  
Coordinates

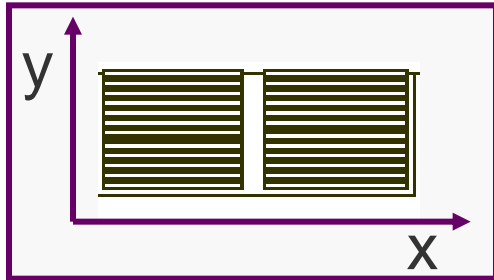


Initial location  
at (0, 0) with  
x- and y-axes  
aligned

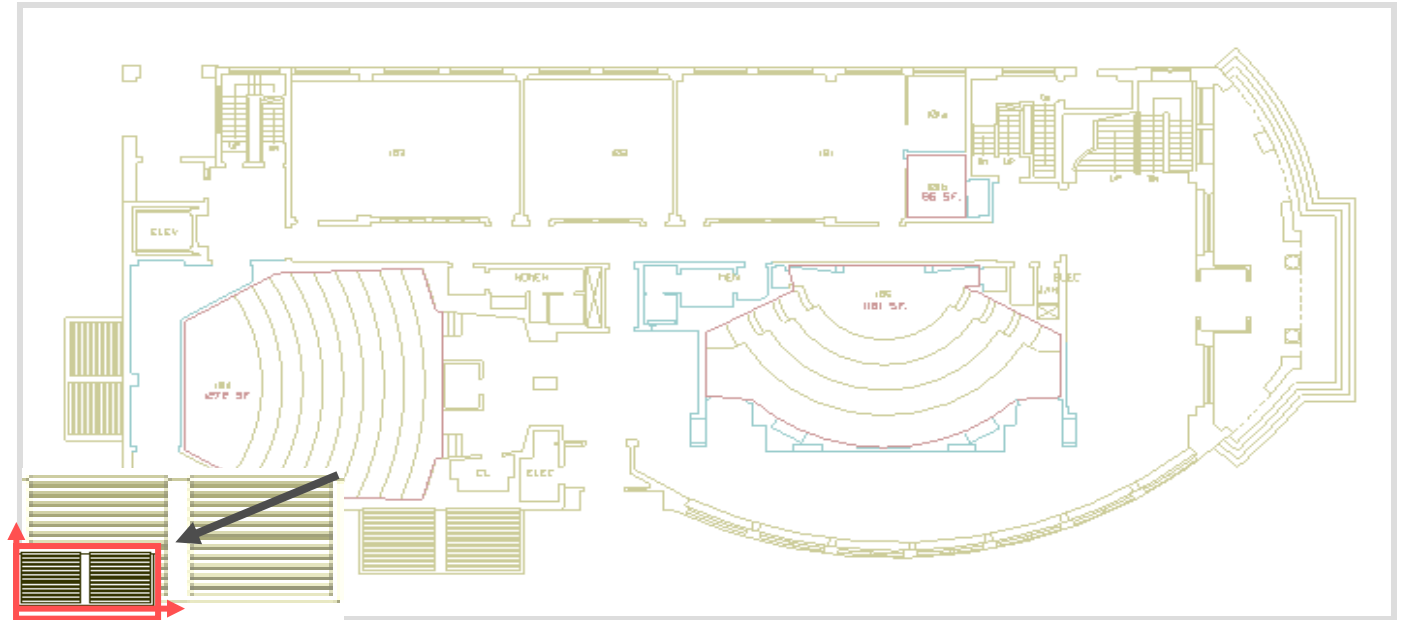


# 2D Modeling Transformations

Modeling  
Coordinates



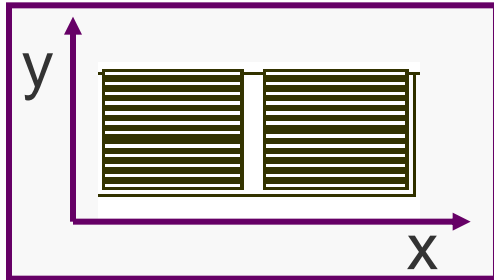
**Scale .3, .3**  
**Rotate -90**  
**Translate 5, 3**



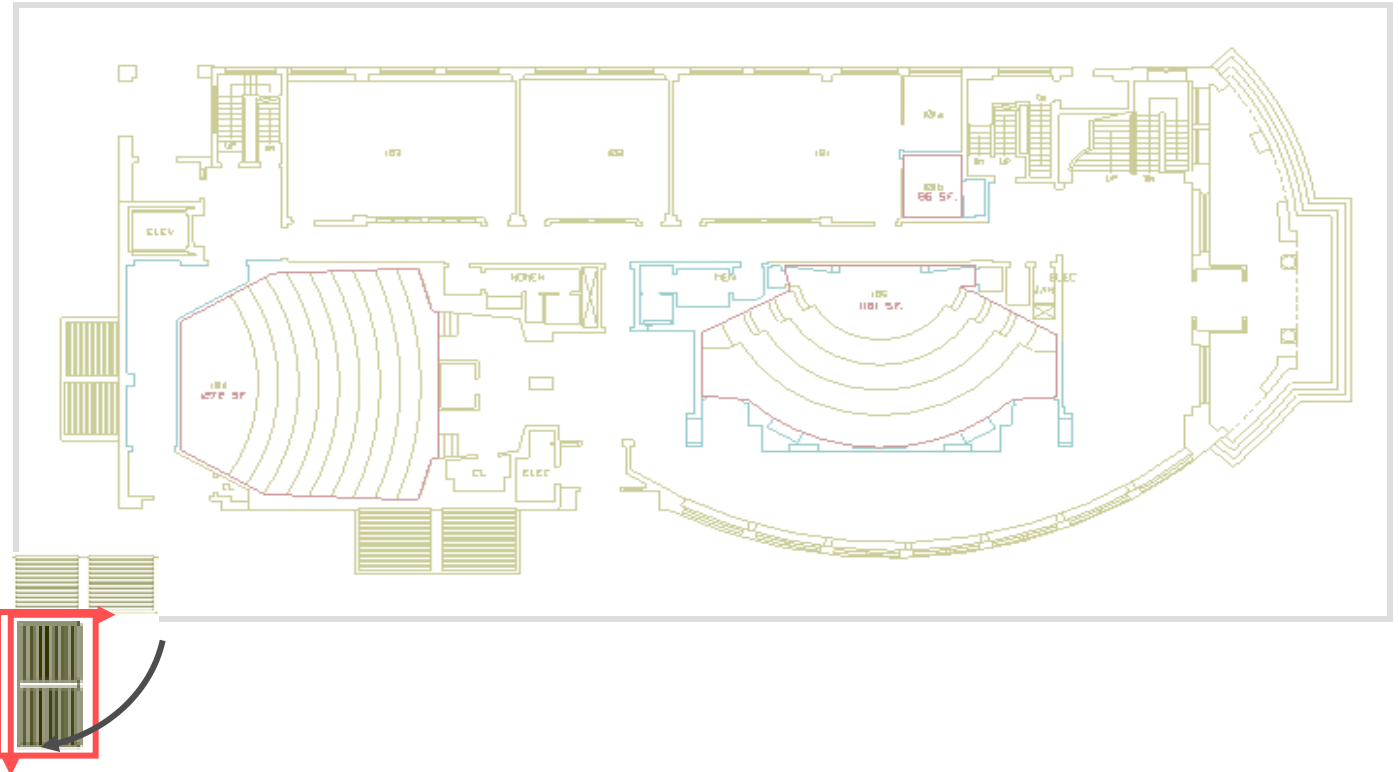


# 2D Modeling Transformations

Modeling  
Coordinates

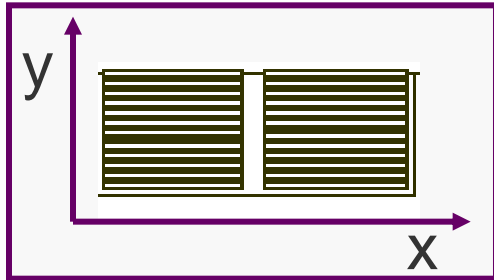


Scale .3, .3  
**Rotate -90**  
Translate 5, 3

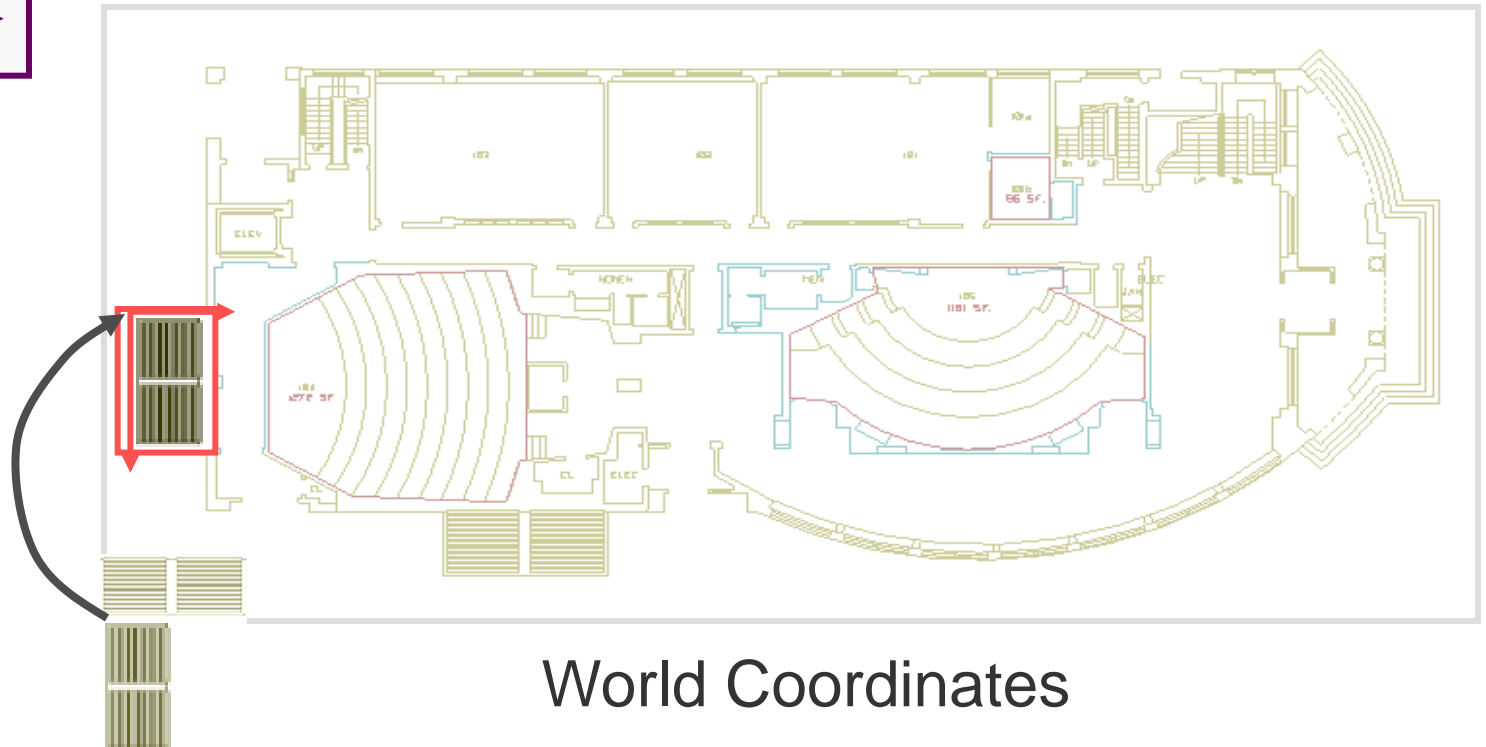


# 2D Modeling Transformations

Modeling  
Coordinates



Scale .3, .3  
Rotate -90  
Translate 5, 3

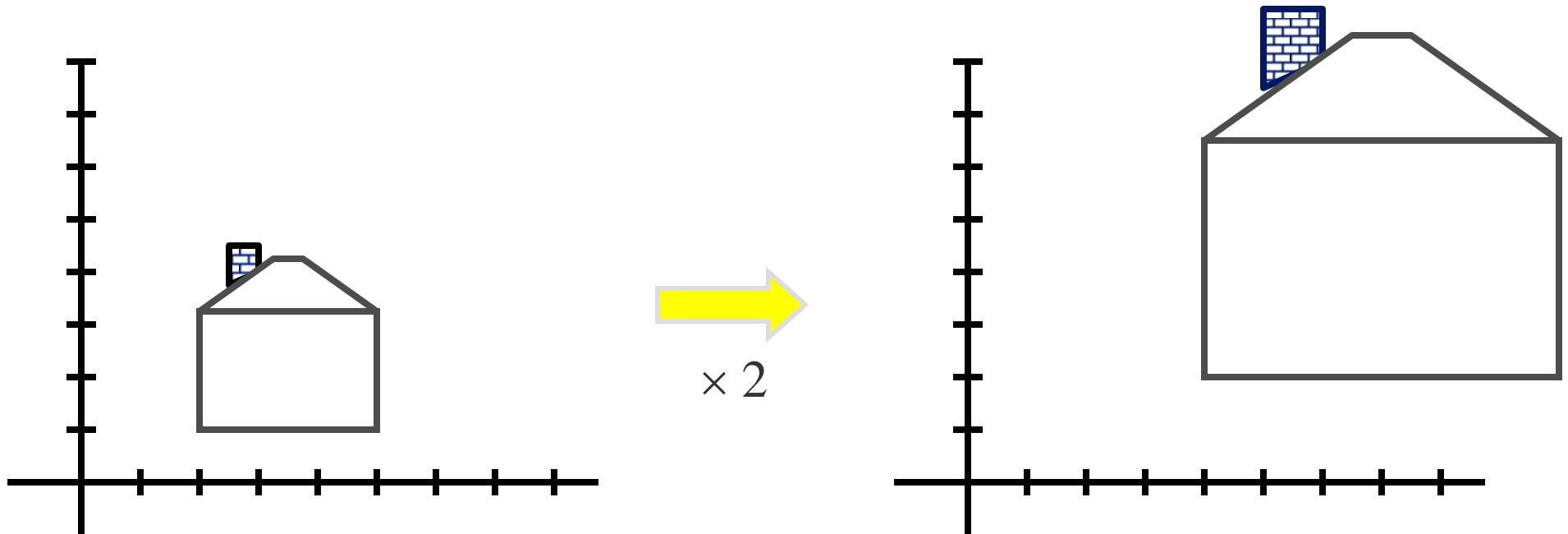


World Coordinates

# Scaling

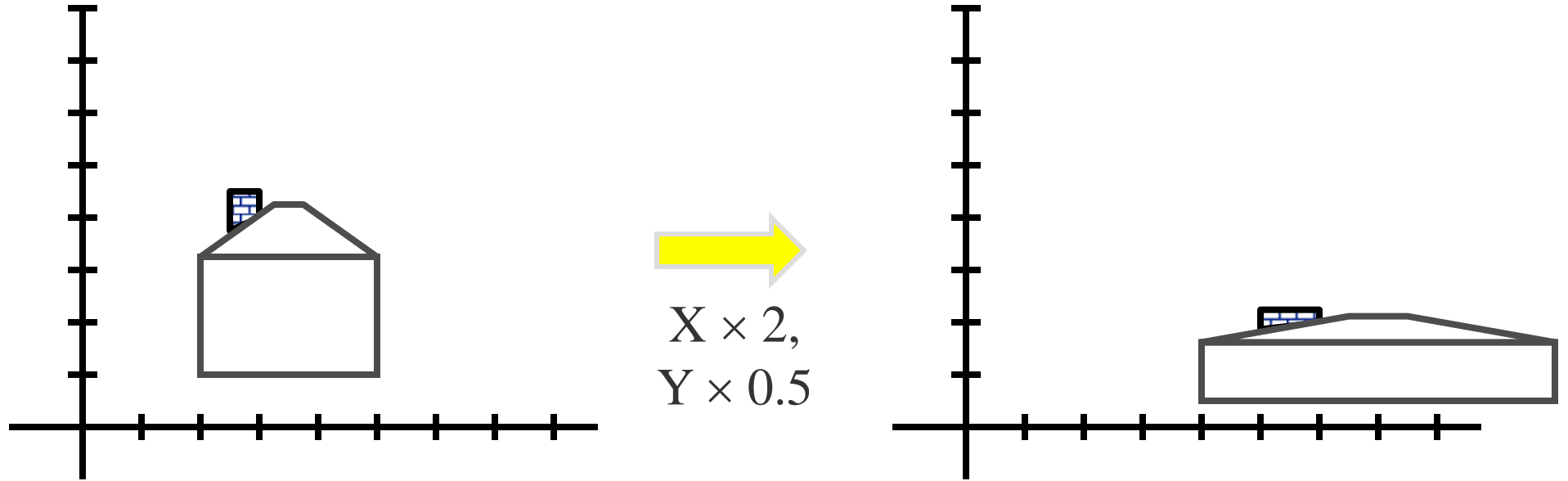
*Scaling a coordinate means multiplying each of its components by a scalar*

*Uniform scaling means this scalar is the same for all components:*



# Scaling

Non-uniform scaling: different scalars per component:



How can we represent this in matrix form?

# Scaling

*Scaling operation:*

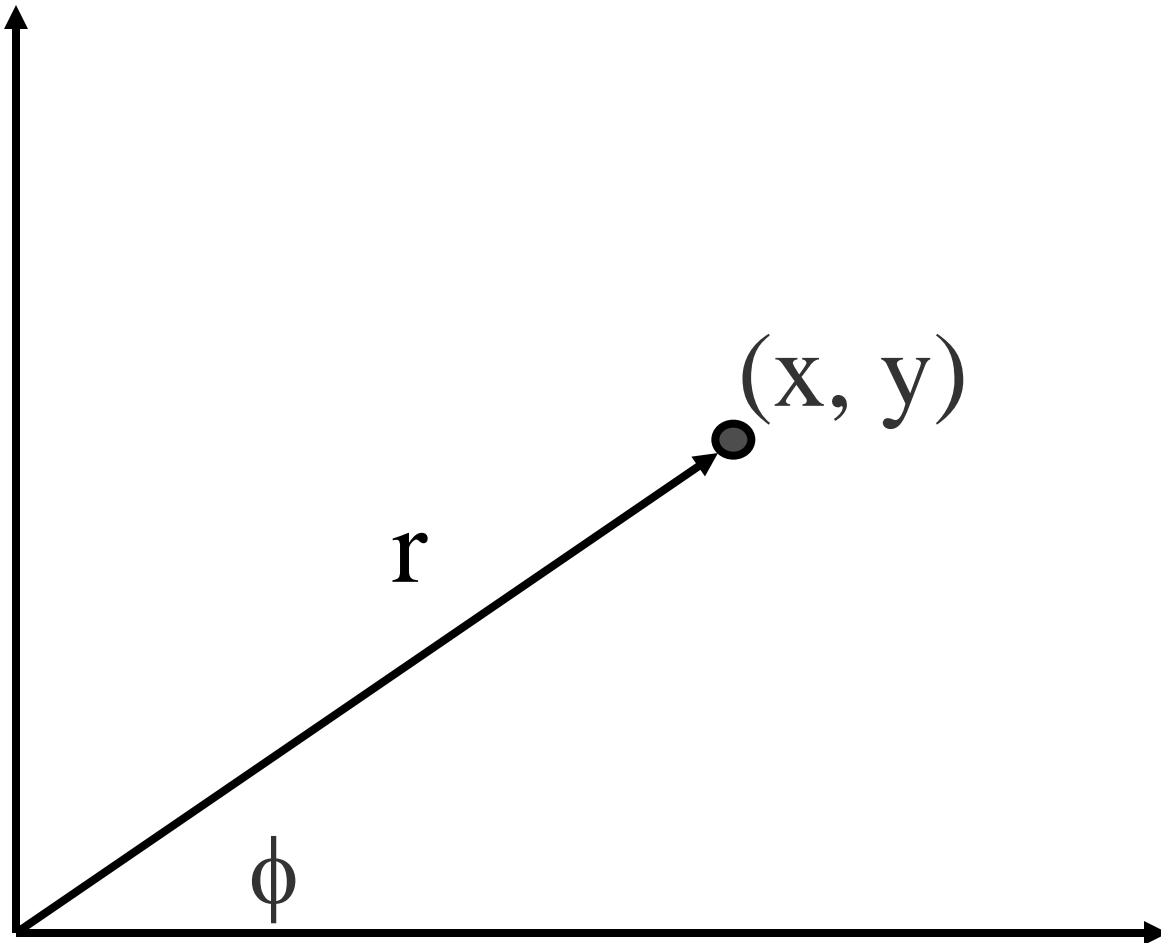
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} ax \\ by \end{bmatrix}$$

*Or, in matrix form:*

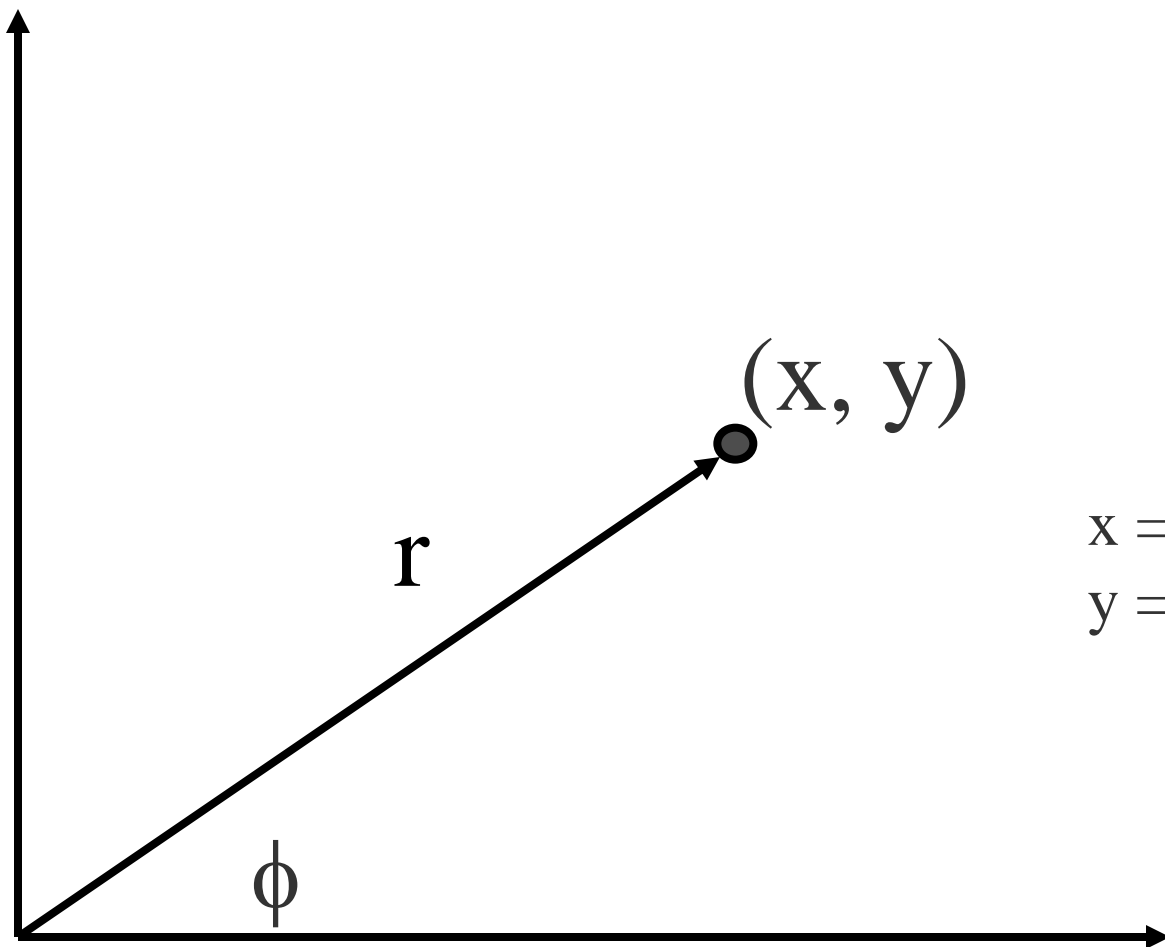
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\text{scaling matrix}} \begin{bmatrix} x \\ y \end{bmatrix}$$

$S_x$  (arrow pointing to  $a$ )  
 $S_y$  (arrow pointing to  $b$ )

# 2-D Rotation



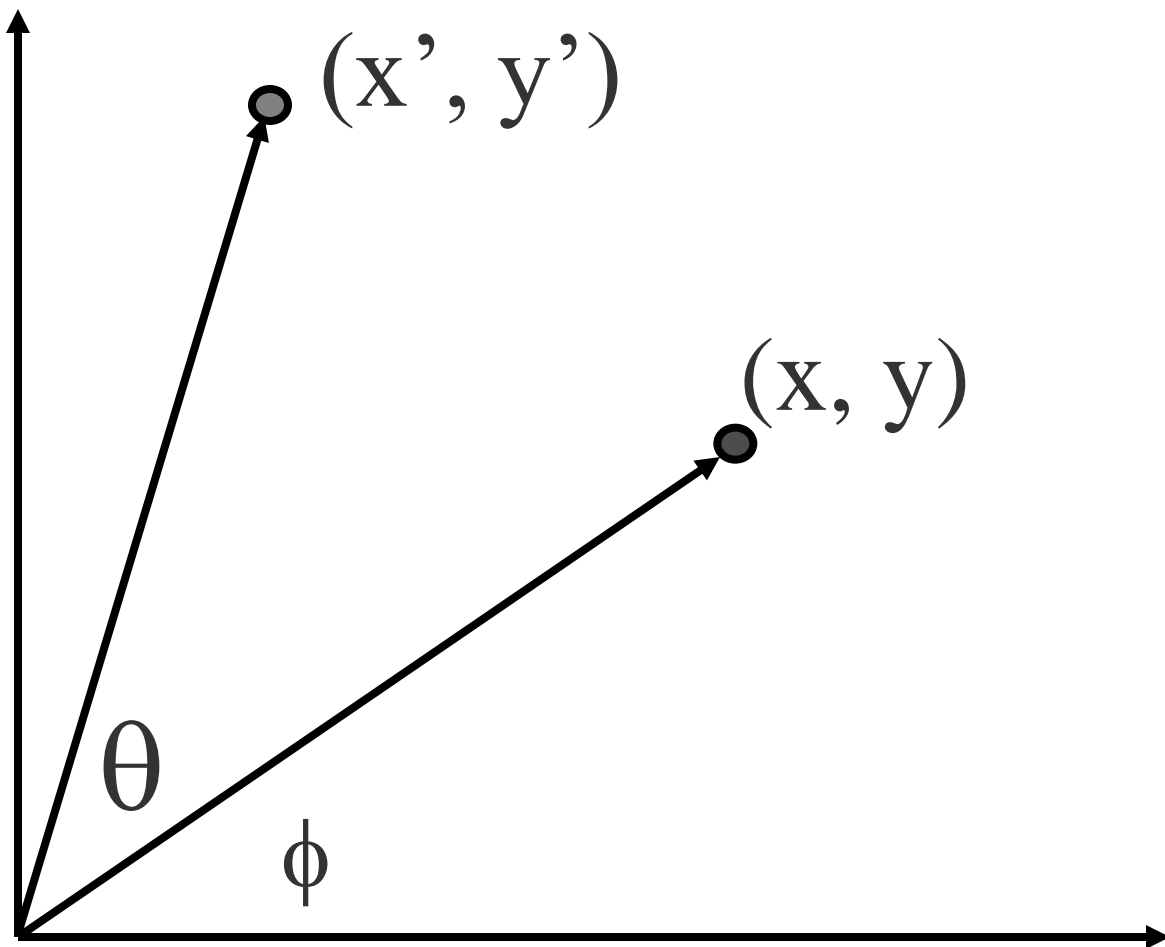
# 2-D Rotation



$$x = r \cos (\phi)$$

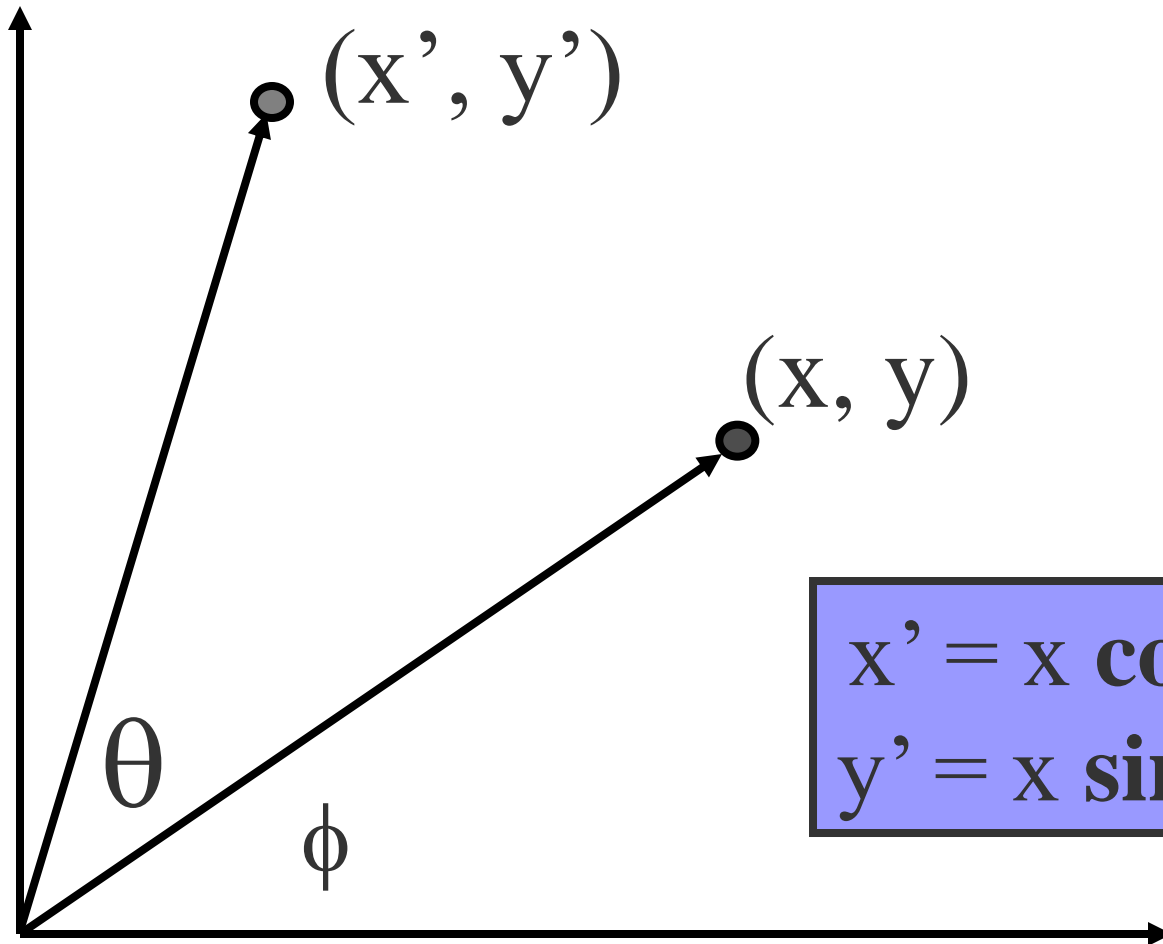
$$y = r \sin (\phi)$$

# 2-D Rotation





# 2-D Rotation

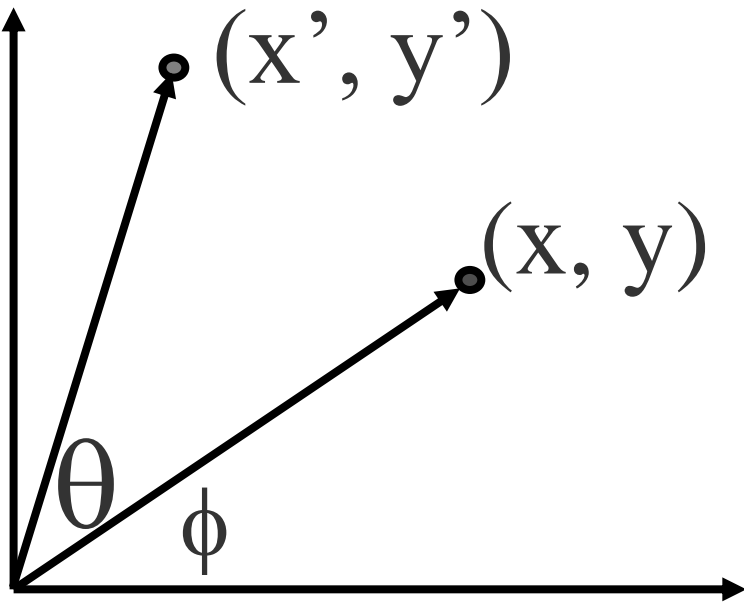


$$\begin{aligned}x' &= x \cos(\theta) - y \sin(\theta) \\y' &= x \sin(\theta) + y \cos(\theta)\end{aligned}$$

# 2-D Rotation

$$x = r \cos (\phi)$$

$$y = r \sin (\phi)$$



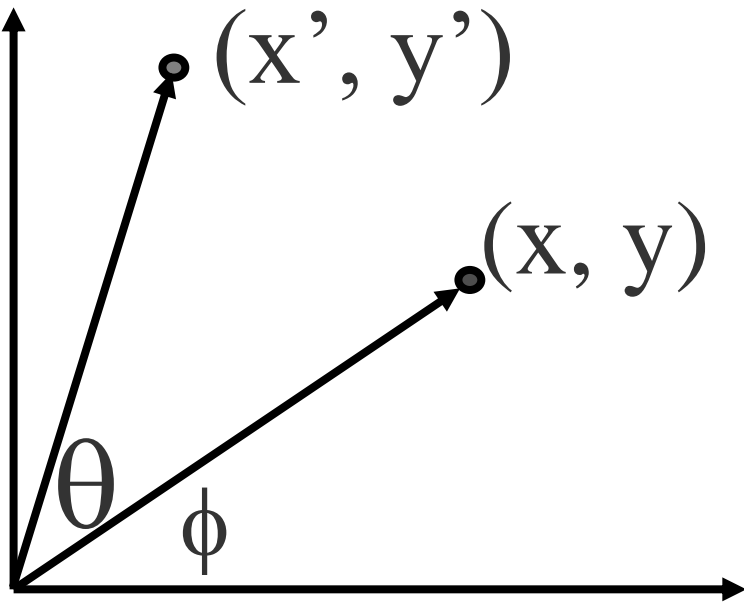
# 2-D Rotation

$$x = r \cos (\phi)$$

$$y = r \sin (\phi)$$

$$x' = r \cos (\phi + \theta)$$

$$y' = r \sin (\phi + \theta)$$



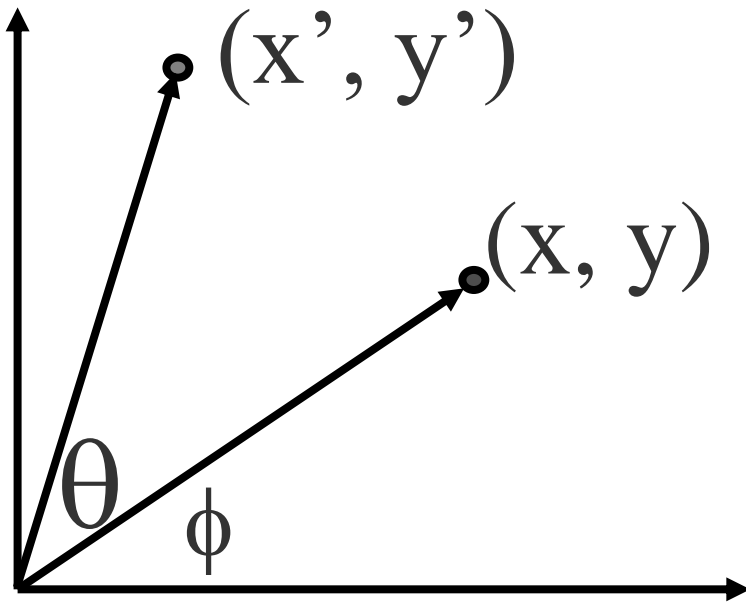
# 2-D Rotation

$$x = r \cos(\phi)$$

$$y = r \sin(\phi)$$

$$x' = r \cos(\phi + \theta)$$

$$y' = r \sin(\phi + \theta)$$

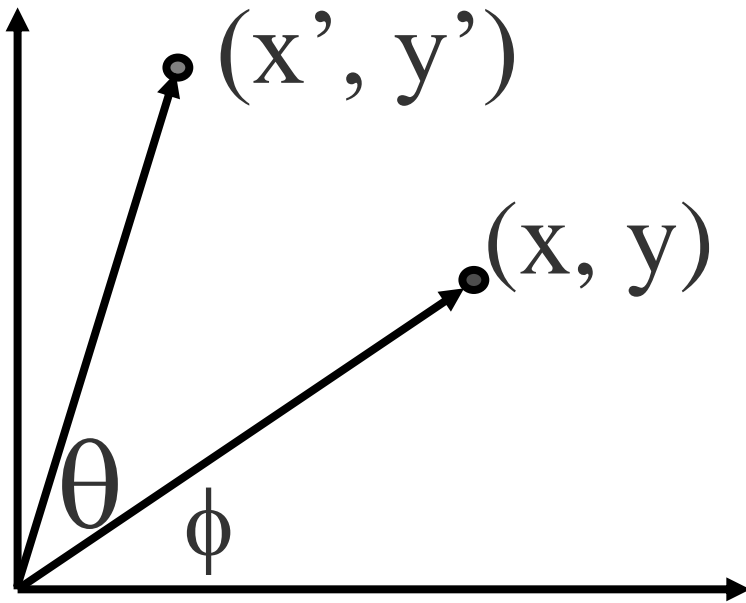


Trig Identity...

$$x' = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta)$$

$$y' = r \cos(\phi) \sin(\theta) + r \sin(\phi) \cos(\theta)$$

# 2-D Rotation



$$x = r \cos(\phi)$$

$$y = r \sin(\phi)$$

$$x' = r \cos(\phi + \theta)$$

$$y' = r \sin(\phi + \theta)$$

Trig Identity...

$$x' = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta)$$

$$y' = r \cos(\phi) \sin(\theta) + r \sin(\phi) \cos(\theta)$$

Substitute...

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

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

## 2-D Rotation

This is easy to capture in matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Even though  $\sin(\theta)$  and  $\cos(\theta)$  are nonlinear functions of  $\theta$ ,

- $x'$  is a linear combination of  $x$  and  $y$
- $y'$  is a linear combination of  $x$  and  $y$

# Basic 2D Transformations

*Translation:*

- $x' = x + t_x$
- $y' = y + t_y$

*Scale:*

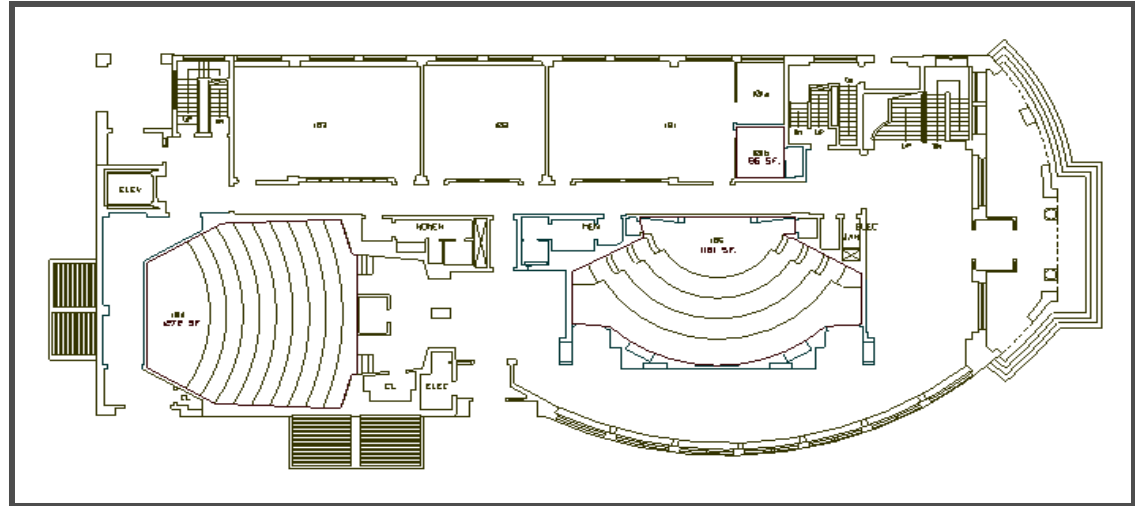
- $x' = x * s_x$
- $y' = y * s_y$

*Shear:*

- $x' = x + h_x * y$
- $y' = y + h_y * x$

*Rotation:*

- $x' = x * \cos\Theta - y * \sin\Theta$
- $y' = x * \sin\Theta + y * \cos\Theta$



**Transformations  
can be combined  
(with simple algebra)**

# Basic 2D Transformations

## *Translation:*

- $x' = x + t_x$
- $y' = y + t_y$

## *Scale:*

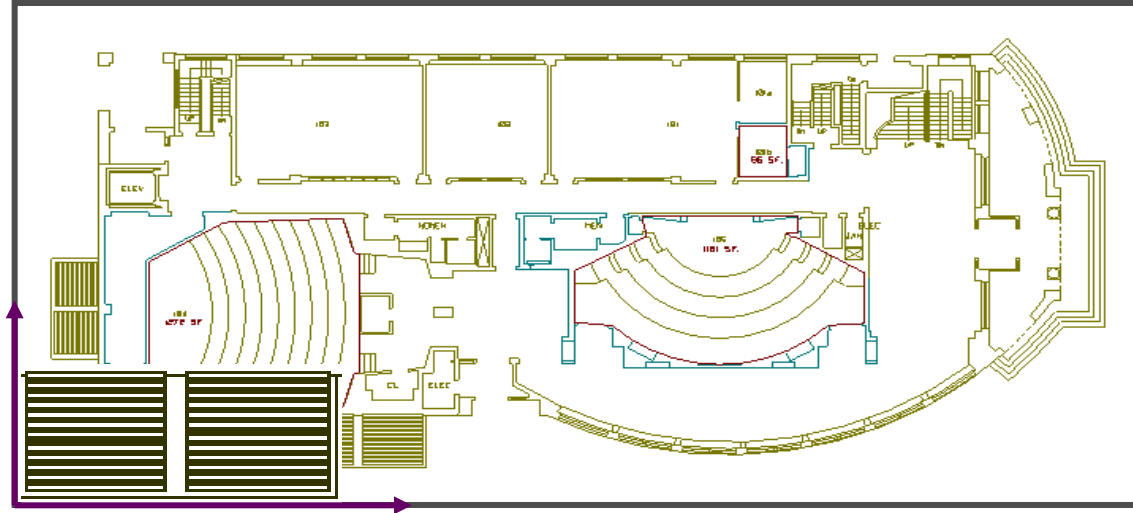
- $x' = x * s_x$
- $y' = y * s_y$

## *Shear:*

- $x' = x + h_x * y$
- $y' = y + h_y * x$

## *Rotation:*

- $x' = x * \cos\Theta - y * \sin\Theta$
- $y' = x * \sin\Theta + y * \cos\Theta$





# Basic 2D Transformations

*Translation:*

- $x' = x + t_x$
- $y' = y + t_y$

**Scale:**

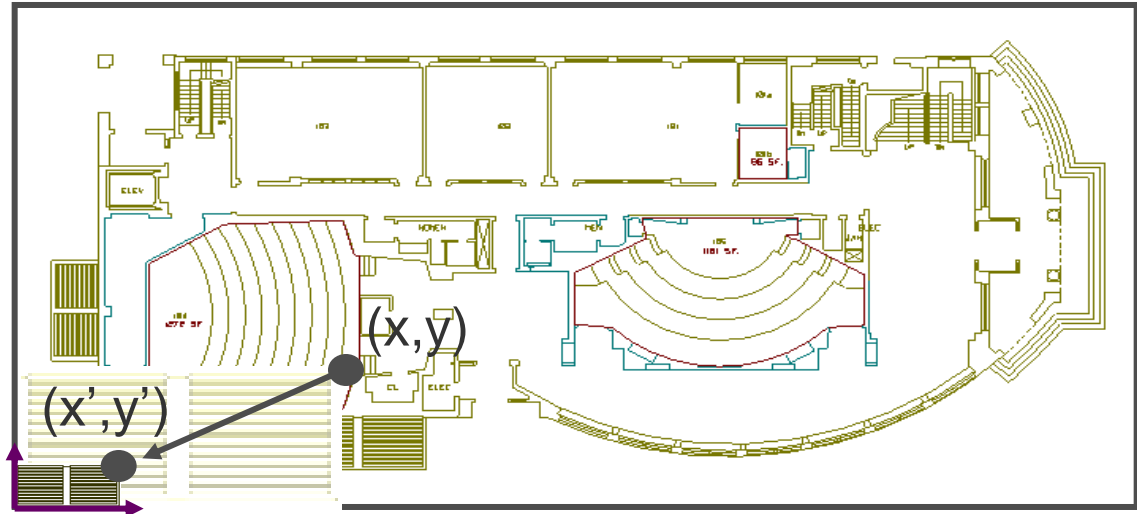
- $x' = x * s_x$
- $y' = y * s_y$

*Shear:*

- $x' = x + h_x * y$
- $y' = y + h_y * x$

*Rotation:*

- $x' = x * \cos(\Theta) - y * \sin(\Theta)$
- $y' = x * \sin(\Theta) + y * \cos(\Theta)$



$$\begin{aligned} x' &= x * s_x \\ y' &= y * s_y \end{aligned}$$

# Basic 2D Transformations

## Translation:

- $x' = x + t_x$
- $y' = y + t_y$

## Scale:

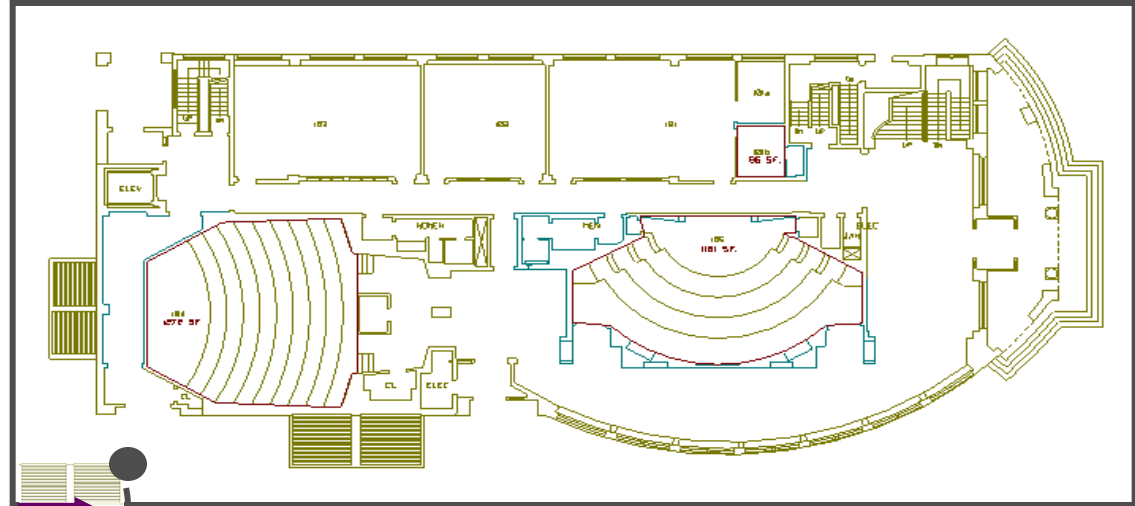
- $x' = x * s_x$
- $y' = y * s_y$

## Shear:

- $x' = x + h_x * y$
- $y' = y + h_y * x$

## Rotation:

- $x' = x * \cos\Theta - y * \sin\Theta$
- $y' = x * \sin\Theta + y * \cos\Theta$



$$\begin{aligned}x' &= (x * s_x) * \cos\Theta - (y * s_y) * \sin\Theta \\y' &= (x * s_x) * \sin\Theta + (y * s_y) * \cos\Theta\end{aligned}$$

# Basic 2D Transformations

## Translation:

- $x' = x + t_x$
- $y' = y + t_y$

## Scale:

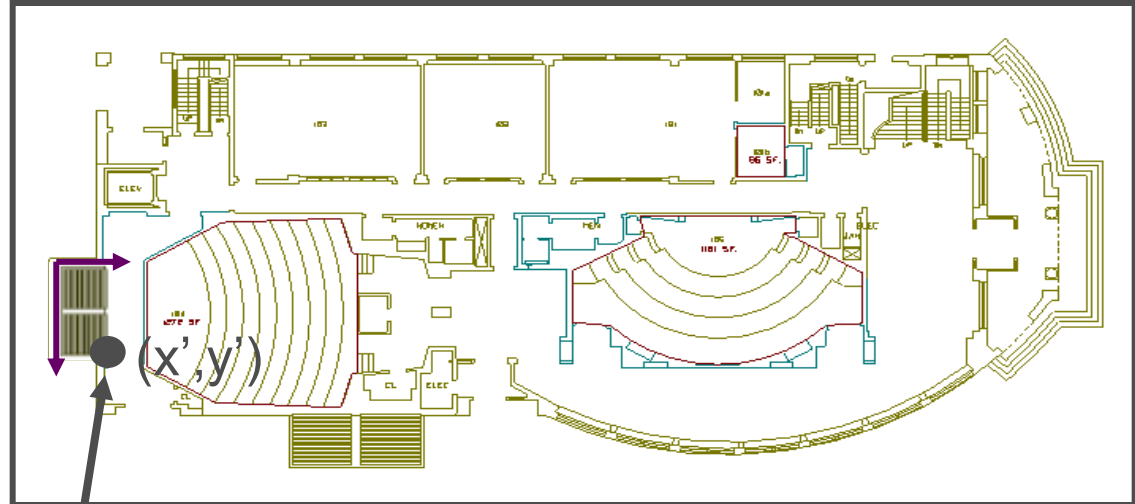
- $x' = x * s_x$
- $y' = y * s_y$

## Shear:

- $x' = x + h_x * y$
- $y' = y + h_y * x$

## Rotation:

- $x' = x * \cos\Theta - y * \sin\Theta$
- $y' = x * \sin\Theta + y * \cos\Theta$



$$\begin{aligned}x' &= ((x * s_x) * \cos\Theta - (y * s_y) * \sin\Theta) + t_x \\y' &= ((x * s_x) * \sin\Theta + (y * s_y) * \cos\Theta) + t_y\end{aligned}$$

# Basic 2D Transformations

## Translation:

- $x' = x + t_x$
- $y' = y + t_y$

## Scale:

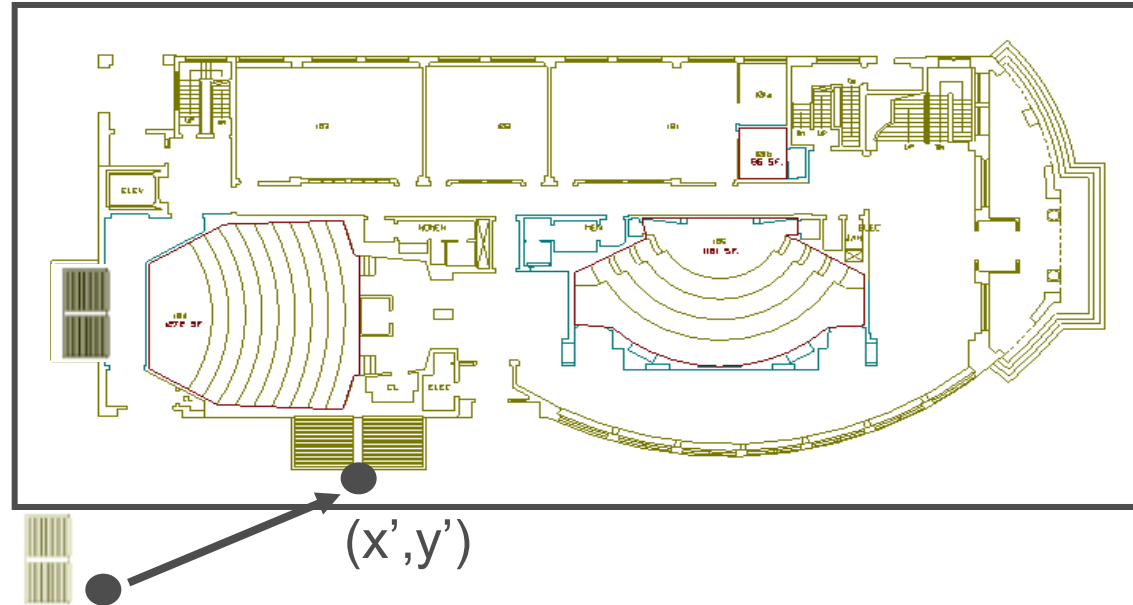
- $x' = x * s_x$
- $y' = y * s_y$

## Shear:

- $x' = x + h_x * y$
- $y' = y + h_y * x$

## Rotation:

- $x' = x * \cos\Theta - y * \sin\Theta$
- $y' = x * \sin\Theta + y * \cos\Theta$



$$\begin{aligned}x' &= ((x * s_x) * \cos\Theta - (y * s_y) * \sin\Theta) + t_x \\y' &= ((x * s_x) * \sin\Theta + (y * s_y) * \cos\Theta) + t_y\end{aligned}$$

# Overview

## *2D Transformations*

- Basic 2D transformations
- Matrix representation
- Matrix composition

## *3D Transformations*

- Basic 3D transformations
- Same as 2D

# Matrix Representation

*Represent 2D transformation by a matrix*

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

*Multiply matrix by column vector*

*⇔ apply transformation to a point*

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \begin{array}{l} x' = ax + by \\ y' = cx + dy \end{array}$$

# Matrix Representation

*Transformations combined by multiplication*

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} i & j \\ k & l \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Matrices are a convenient and efficient way to represent a sequence of transformations!

# 2x2 Matrices

*What type of transformations can be represented with a 2x2 matrix?*

2D Identity?

$$\begin{aligned}x' &= x \\ y' &= y\end{aligned} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Scale around (0,0)?

$$\begin{aligned}\mathbf{x}' &= s_x * \mathbf{x} \\ \mathbf{y}' &= s_y * \mathbf{y}\end{aligned} \quad \begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$



# 2x2 Matrices

*What type of transformations can be represented with a 2x2 matrix?*

2D Rotate around (0,0)?

$$\begin{aligned}x' &= \cos \Theta * x - \sin \Theta * y \\y' &= \sin \Theta * x + \cos \Theta * y\end{aligned}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Shear?

$$\begin{aligned}x' &= x + sh_x * y \\y' &= sh_y * x + y\end{aligned}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

*Q: How can we represent translation as a 2x2 matrix?*

$$x' = x + t_x$$

$$y' = y + t_y$$

# 2x2 Matrices

*What types of transformations can be represented with a 2x2 matrix?*

2D Translation?

$$\mathbf{x}' = \mathbf{x} + \mathbf{t}_x$$

$$\mathbf{y}' = \mathbf{y} + \mathbf{t}_y \quad \text{NO!}$$

Only linear 2D linear transformations  
can be represented with a 2x2 matrix

# Linear Transformations

*Linear transformations are combinations of ...*

- Scale,
- Rotation,
- Shear, and
- Mirror

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

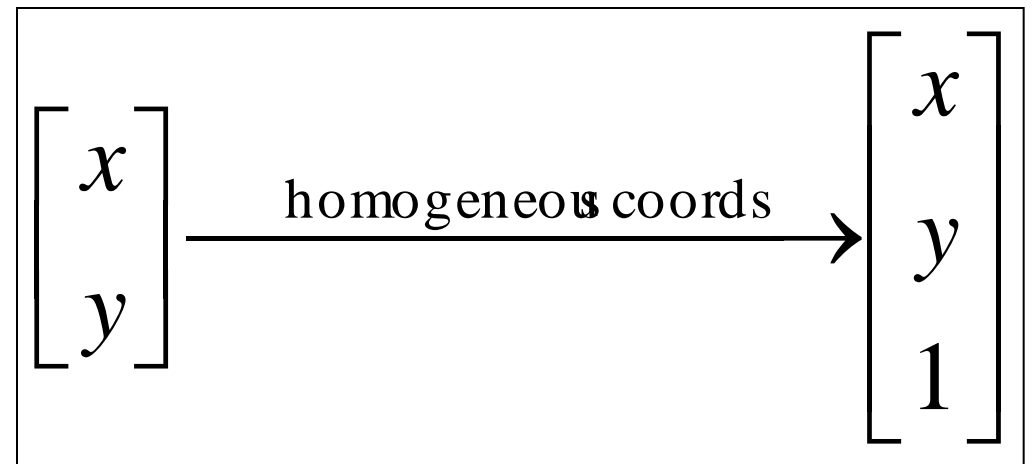
*Properties of linear transformations:*

- Origin maps to origin
- Lines map to lines
- Parallel lines remain parallel
- Ratios are preserved
- Closed under composition

# Homogeneous Coordinates

Homogeneous coordinates

- represent coordinates in 2 dimensions with a 3-vector



Homogeneous coordinates seem unintuitive, but they make graphics operations much easier

# Homogeneous Coordinates

*Q: How can we represent translation as a 3x3 matrix?*

$$\mathbf{x}' = \mathbf{x} + \mathbf{t}_x$$

$$\mathbf{y}' = \mathbf{y} + \mathbf{t}_y$$

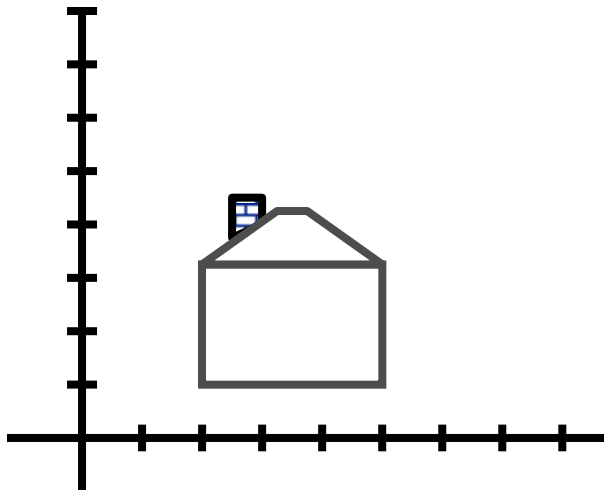
*A: Using the rightmost column:*

$$\mathbf{Translation} = \begin{bmatrix} 1 & 0 & \mathbf{t}_x \\ 0 & 1 & \mathbf{t}_y \\ 0 & 0 & 1 \end{bmatrix}$$

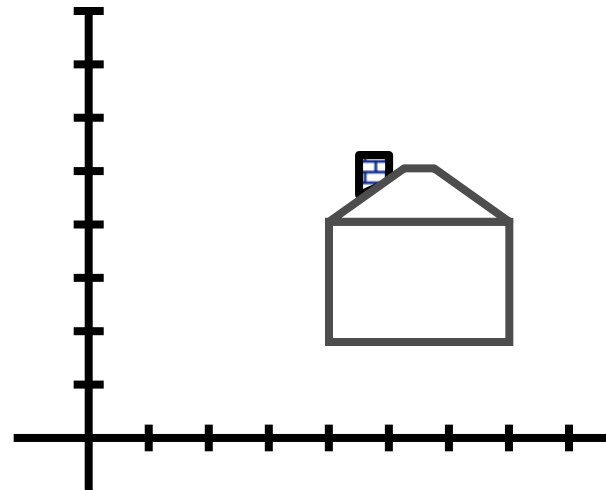
# Translation

*Homogeneous Coordinates*

$$\begin{matrix} \blacktriangledown & & \blacktriangledown & & \blacktriangledown \\ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} & = & \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} & = & \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} \end{matrix}$$



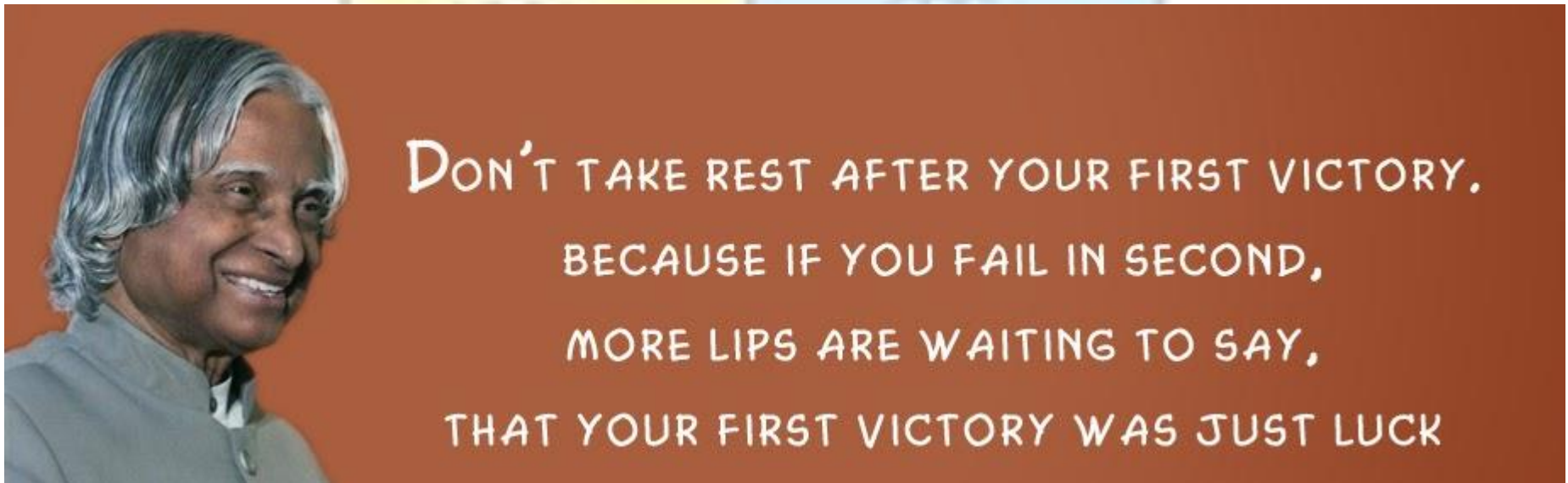
$$\begin{aligned} t_x &= 2 \\ t_y &= 1 \end{aligned}$$



# Computer Graphics

(SE Computer Engineering 2019-course)

## Unit III- 2D, 3D Transformations and Projections



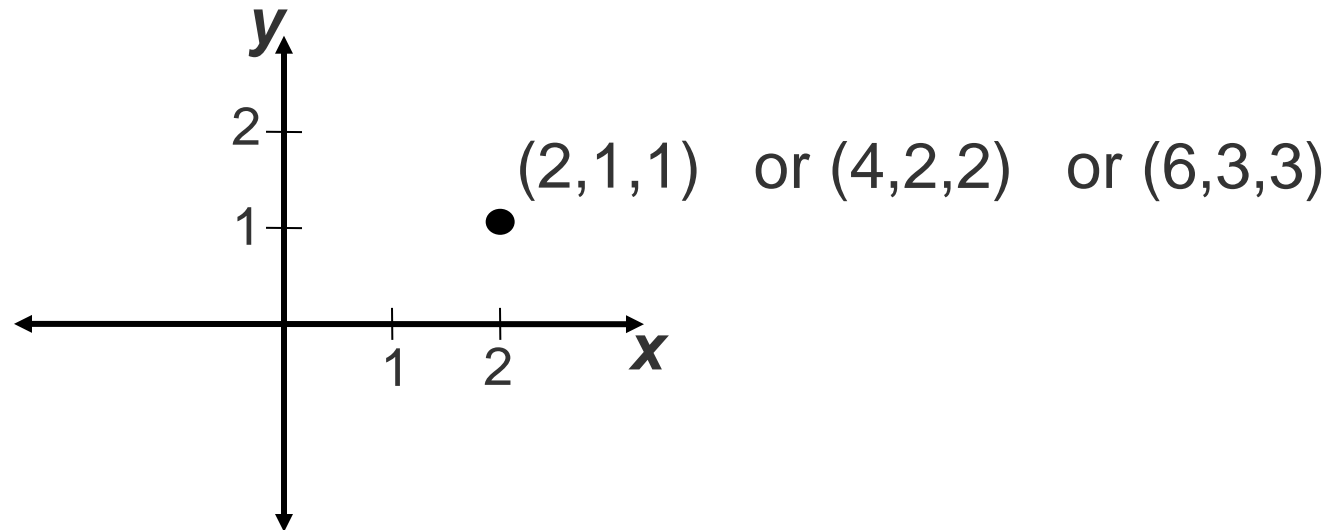


# Homogeneous Coordinates

## *Add a 3rd coordinate to every 2D point*

- $(x, y, w)$  represents a point at location  $(x/w, y/w)$
- $(x, y, 0)$  represents a point at infinity
- $(0, 0, 0)$  is not allowed

Convenient coordinate system to represent many useful transformations



# Basic 2D Transformations

## *Basic 2D transformations as 3x3 matrices*

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear

## Example 1

*Scale a square ABCD with the co-ordinates  $A(0,0)$ ,  $B(3,0)$ ,  $C(3,3)$ ,  $D(0,3)$  by 3 units in x direction and 3 units in y direction.*

## Example 1

*Scale a square ABCD with the co-ordinates A(0,0), B(3,0), C(3,3), D(0,3) by 3 units in x direction and 3 units in y direction.*

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

## Example 1

*Scale a square ABCD with the co-ordinates A(0,0), B(3,0), C(3,3), D(0,3) by 3 units in x direction and 3 units in y direction.*

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Sol.

A'(0,0)

B'(9,0)

C'(9,9)

D'(0,9)

## Example 2

*Translate a square ABCD with the co-ordinates  $A(0,0)$ ,  $B(3,0)$ ,  $C(3,3)$ ,  $D(0,3)$  by 2 units in x direction and 2 units in y direction.*

## Example 2

*Translate a square ABCD with the co-ordinates A(0,0), B(3,0), C(3,3), D(0,3) by 2 units in x direction and 2 units in y direction.*

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

## Example 2

*Translate a square ABCD with the co-ordinates A(0,0), B(3,0), C(3,3), D(0,3) by 2 units in x direction and 2 units in y direction.*

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Sol.

A'(2,2)

B'(5,2)

C'(5,5)

D'(2,5)



### **Example 3**

*Rotate a triangle ABC with the co-ordinates A(0,0), B(6,0), C(3,3) by 90° about origin in anticlockwise direction.*

### Example 3

*Rotate a triangle ABC with the co-ordinates A(0,0), B(6,0), C(3,3) by 90° about origin in anticlockwise direction.*

$$\begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

### Example 3

*Rotate a triangle ABC with the co-ordinates A(0,0), B(6,0), C(3,3) by 90° about origin in anticlockwise direction.*

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 6 & 3 \\ 0 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -3 \\ 0 & 6 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

### Example 3

*Rotate a triangle ABC with the co-ordinates A(0,0), B(6,0), C(3,3) by 90° about origin in anticlockwise direction.*

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 6 & 3 \\ 0 & 0 & 3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -3 \\ 0 & 6 & 3 \\ 1 & 1 & 1 \end{bmatrix}$$

Sol.

A'(0,0)

B'(0,6)

C'(-3,3)

# Linear Transformations

*Linear transformations are combinations of ...*

- Scale,
- Rotation,
- Shear, and
- Mirror

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

*Properties of linear transformations:*

- Origin maps to origin
- Lines map to lines
- Parallel lines remain parallel
- Ratios are preserved
- Closed under composition

# Affine Transformations

*Affine transformations are combinations of ...*

- Linear transformations, and
- Translations

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

*Properties of affine transformations:*

- **Origin does not necessarily map to origin**
- Lines map to lines
- Parallel lines remain parallel
- Ratios are preserved
- Closed under composition

# Projective Transformations

## *Projective transformations ...*

- Affine transformations, and
- Projective warps

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

## *Properties of projective transformations:*

- Origin does not necessarily map to origin
- Lines map to lines
- **Parallel lines do not necessarily remain parallel**
- Ratios are not preserved
- Closed under composition

# Overview

## *2D Transformations*

- Basic 2D transformations
- Matrix representation
- Matrix composition



# Matrix Composition

*Transformations can be combined by matrix multiplication*

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \left( \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$\mathbf{p}' = \mathbf{T}(t_x, t_y) \mathbf{R}(\Theta) \mathbf{S}(s_x, s_y) \mathbf{p}$

# Matrix Composition

*Matrices are a convenient and efficient way to represent a sequence of transformations*

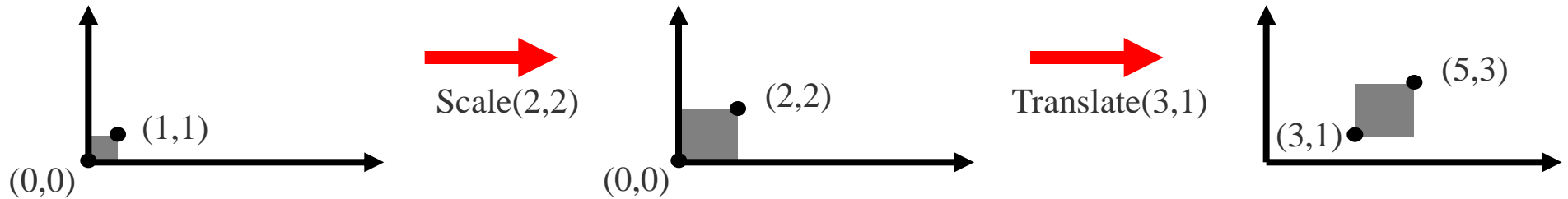
- General purpose representation
- Hardware matrix multiply

$$\mathbf{p}' = (T * (R * (S * \mathbf{p})))$$

$$\mathbf{p}' = (T * R * S) * \mathbf{p}$$

# How are transforms combined?

Scale then Translate



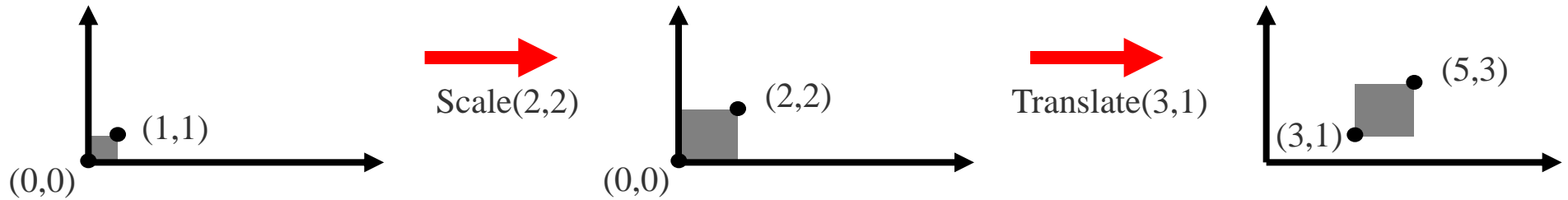
Use matrix multiplication:  $p' = T ( S p ) = TS p$

$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

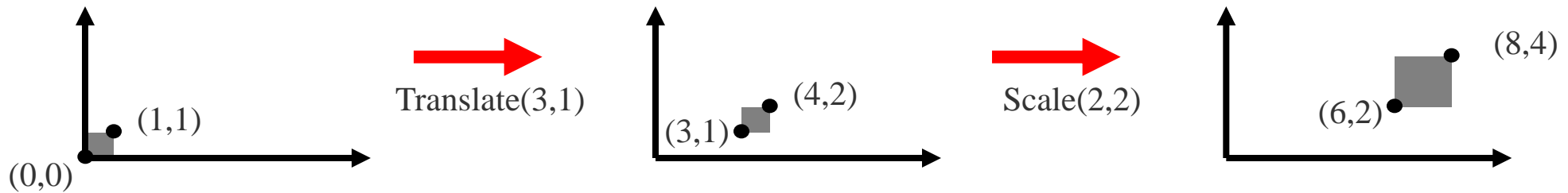
Caution: matrix multiplication is **NOT** commutative!

# Non-commutative Composition

Scale then Translate:  $p' = T ( S p ) = TS p$



Translate then Scale:  $p' = S ( T p ) = ST p$



# Non-commutative Composition

Scale then Translate:  $p' = T ( S p ) = TS p$

$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

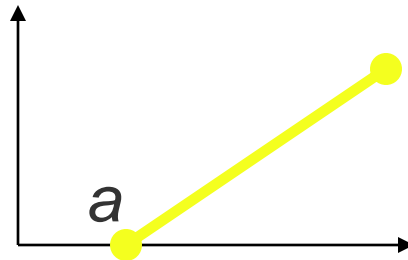
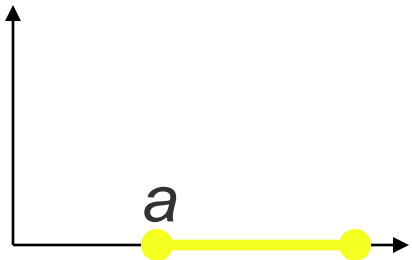
Translate then Scale:  $p' = S ( T p ) = ST p$

$$ST = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

# Matrix Composition

What if we want to rotate *and* scale?

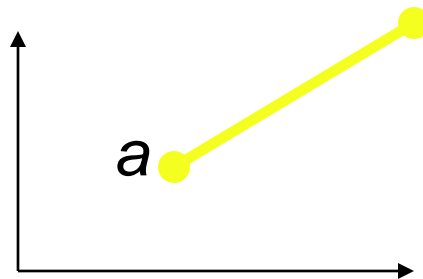
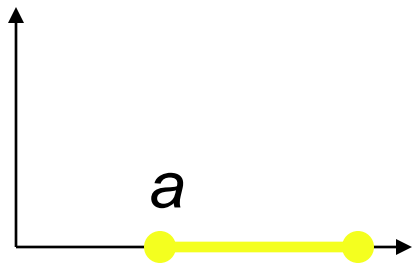
- Ex: Rotate line segment by 45 degrees about endpoint  $a$  *and* lengthen



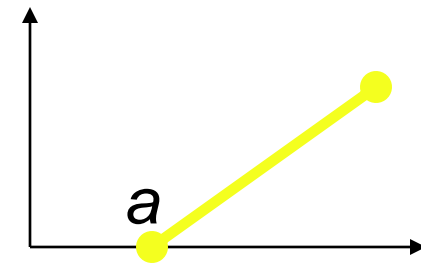
# Multiplication Order – Wrong Way

*Our line is defined by two endpoints*

- Applying a rotation of 45 degrees,  $R(45)$ , affects both points
- We could try to translate both endpoints to return endpoint  $a$  to its original position, but by how much?



Wrong  
 $R(45)$

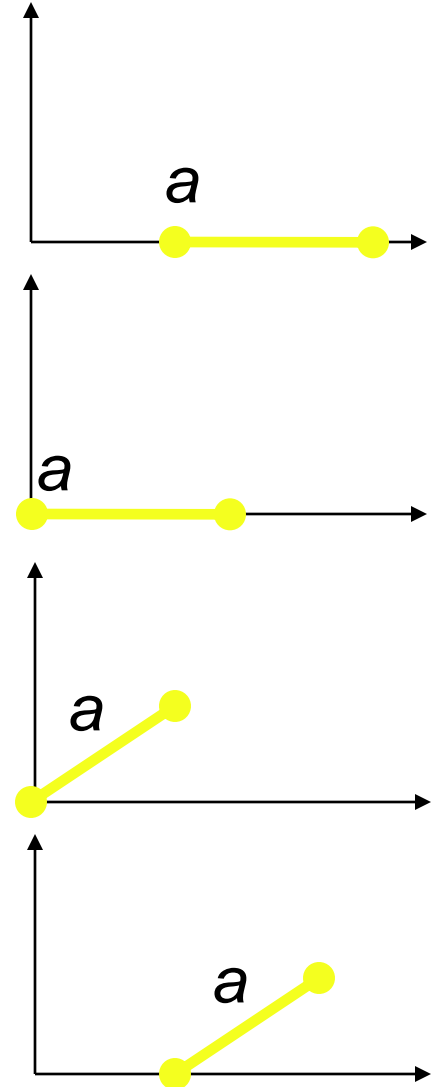


Correct  
 $T(-3) R(45) T(3)$

# Multiplication Order - Correct

Isolate endpoint  $a$  from rotation effects

- First translate line so  $a$  is at origin:  $T(-3)$
- Then rotate line 45 degrees:  $R(45)$
- Then translate back so  $a$  is where it was:  $T(3)$





# Computer Graphics

(SE Computer Engineering 2019-course)

## Unit III- 2D, 3D Transformations and Projections

“When one door closes,  
another opens; but we  
often look so long and  
so regretfully upon the  
closed door that we do  
not see the one which  
has opened for us.” –  
**Alexander Graham Bell**

# Matrix Composition

*Will this sequence of operations work?*

$$\begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(45) & -\sin(45) & 0 \\ \sin(45) & \cos(45) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ 1 \end{bmatrix} = \begin{bmatrix} a'_x \\ a'_y \\ 1 \end{bmatrix}$$

# Combining simple transformations

*More complex transformations can be combined and expressed by using one single 3x3 matrix, which is the product of all the combined simple matrices.*

*Note: The order between the matrices is crucial!*

# Combined Transformation-

## Rotation about an arbitrary point

*A point  $(x, y)$  is to be rotated  $\theta$  degrees in clockwise direction about the point  $(R_x, R_y)$ .*

*Three steps:*

*1) Move  $(R_x, R_y)$  to origin*

*2) Rotate  $\theta$  degrees about origin*

*3) Move back to  $(R_x, R_y)$*

## Example, step 1)

$$T(-R_x, -R_y) = \left( \begin{array}{ccc} 1 & 0 & -R_x \\ 0 & 1 & -R_y \\ 0 & 0 & 1 \end{array} \right)$$

---

**Example, step 2) (clockwise rotation)**

$$R(-\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

---

## Example, step 3)

$$T(R_x, R_y) = \begin{pmatrix} 1 & 0 & R_x \\ 0 & 1 & R_y \\ 0 & 0 & 1 \end{pmatrix}$$

---

## Example, result

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = T(R_x, R_y) \cdot R(-\theta) \cdot T(-R_x, -R_y) \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = M \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



# Matrix Composition

1. After correctly ordering the matrices
2. Multiply matrices together
3. What results is one matrix !
4. Multiply this matrix by the vector of each vertex
5. All vertices easily transformed with one matrix multiplication

# Basic 2D Transformations

## *Basic 2D transformations as 3x3 matrices*

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 \\ \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear

4-2 Write a  $2 \times 2$  transformation matrix for each of the following scaling transformations:

- The entire picture three times as large
- The entire picture one-third as large
- The x direction four times as large, the y direction unchanged
- The y lengths reduced to two-thirds their original value, the x lengths unchanged
- The x direction reduced to three-fourths the original, y direction increased by a factor of seven-fifths

...tion matrices for each of the scaling trans-

## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

$$\textit{Shear} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

$$\textit{Shear} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

$$\textit{Shear} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

$$\text{Shear} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$SR = \begin{bmatrix} s_x \cos \Theta & s_x \sin \Theta & 0 \\ -s_y \sin \Theta & s_y \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

$$\text{Shear} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$SR = \begin{bmatrix} s_x \cos \Theta & s_x \sin \Theta & 0 \\ -s_y \sin \Theta & s_y \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$sh_x = -s_y \sin \Theta$$

$$sh_y = s_x \sin \Theta$$

$$s_x \cos \Theta = 1$$

$$s_y \cos \Theta = 1$$

$$s_x = 1 / \cos \Theta$$

$$s_y = 1 / \cos \Theta$$

## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

$$\text{Shear} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$SR = \begin{bmatrix} s_x \cos \Theta & s_x \sin \Theta & 0 \\ -s_y \sin \Theta & s_y \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$sh_x = -s_y \sin \Theta$$

$$sh_y = s_x \sin \Theta$$

$$s_x \cos \Theta = 1$$

$$s_y \cos \Theta = 1$$

$$s_x = 1 / \cos \Theta$$

$$s_y = 1 / \cos \Theta$$

$$sh_x = -1 / \cos \Theta \times \sin \Theta = -\tan \Theta$$

$$sh_y = 1 / \cos \Theta \times \sin \Theta = \tan \Theta$$

## Exercise 4.11 from Harrington Pg. 143

- *Show how shear transformation can be expressed in terms of rotation and scales.*

$$\text{Shear} = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$SR = \begin{bmatrix} s_x \cos \Theta & s_x \sin \Theta & 0 \\ -s_y \sin \Theta & s_y \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore \text{Shear} = \begin{bmatrix} 1 & \tan \Theta & 0 \\ -\tan \Theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$sh_x = -s_y \sin \Theta$$

$$sh_y = s_x \sin \Theta$$

$$s_x \cos \Theta = 1$$

$$s_y \cos \Theta = 1$$

$$s_x = 1 / \cos \Theta$$

$$s_y = 1 / \cos \Theta$$

$$sh_x = -(1 / \cos \Theta) \times \sin \Theta = -\tan \Theta$$

$$sh_y = (1 / \cos \Theta) \times \sin \Theta = \tan \Theta$$

## Exercise 4.11 from Harrington Pg. 143

- *Show how rotation transformation can be expressed in terms of shear and scales.*

$$R = \begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Shear = \begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Computer Graphics

(SE Computer Engineering 2019-course)

## Unit III- 2D, 3D Transformations and Projections

No matter what job you have in life, your success will be determined 5% by your academic credentials, 15% by your professional experiences, and 80% by your communication skills.

# Overview

## *2D Transformations*

- Basic 2D transformations
- Matrix representation
- Matrix composition

## *3D Transformations*

- Basic 3D transformations
- Matrix representation
- Matrix composition

# 3D Transformations

## *Same idea as 2D transformations*

- Homogeneous coordinates:  $(x,y,z,w)$
- 4x4 transformation matrices

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

# Basic 3D Transformations

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Translation

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Scale



# Basic 3D Transformations

Rotate around Z axis:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 & 0 \\ \sin \Theta & \cos \Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Rotate around Y axis:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} \cos \Theta & 0 & -\sin \Theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \Theta & 0 & \cos \Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Rotate around X axis:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \Theta & -\sin \Theta & 0 \\ 0 & \sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

# Basic 3D Transformations

Shearing in X direction:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & S_y & S_z & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Shearing in Y direction:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ S_x & 1 & S_z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Shearing in Z direction:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ S_x & S_y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

# Reverse Rotations

*Q: How do you undo a rotation of  $\theta$ ,  $R(\theta)$ ?*

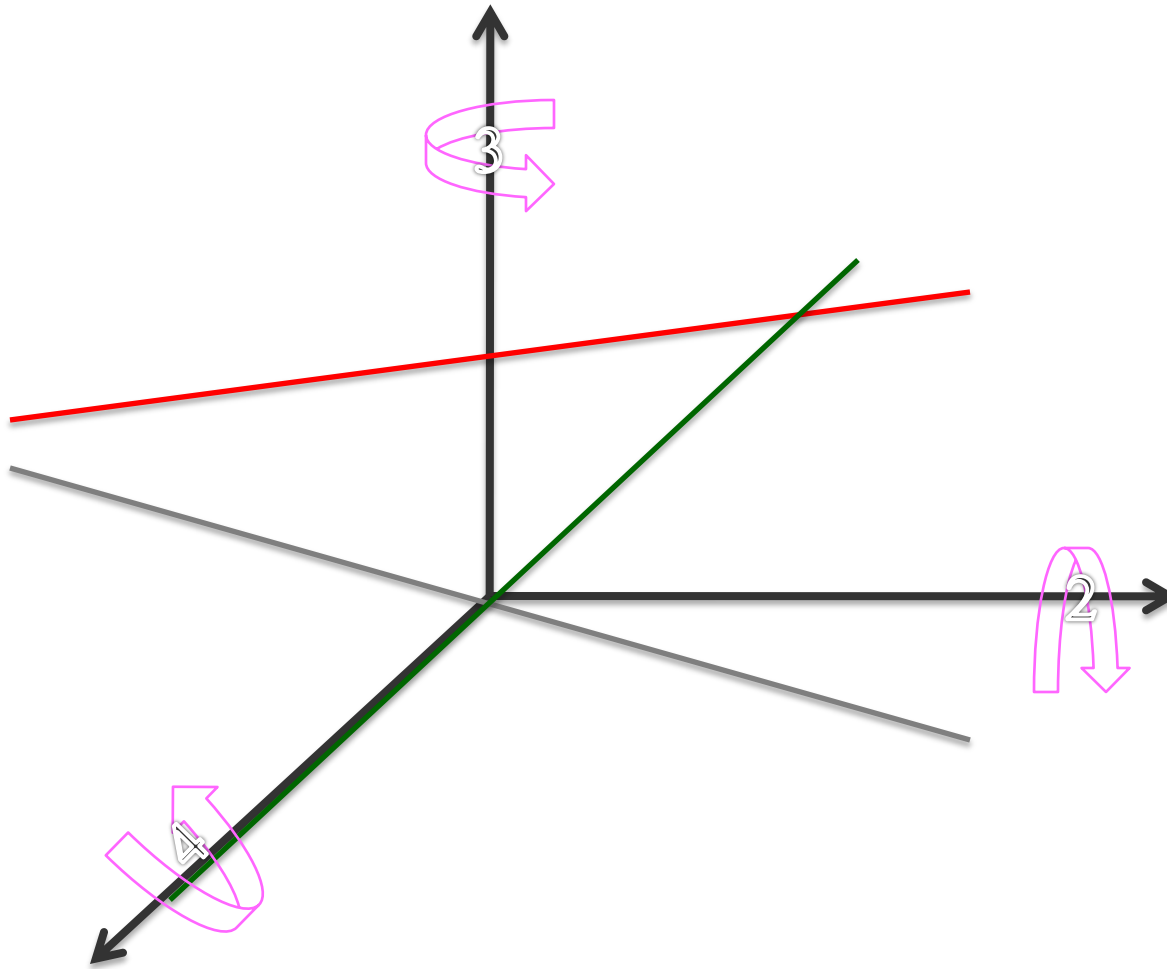
*A: Apply the inverse of the rotation...  $R^{-1}(\theta) = R(-\theta)$*

*How to construct  $R^{-1}(\theta) = R(-\theta)$*

- Inside the rotation matrix:  $\cos(\theta) = \cos(-\theta)$ 
  - *The cosine elements of the inverse rotation matrix are unchanged*
- The sign of the sine elements will flip

*Therefore...  $R^{-1}(\theta) = R(-\theta) = R^T(\theta)$*

# 3D Rotation about an arbitrary axis



1. Translate

2. Rotate about x axis

3. Rotate about y axis

4. Actual Rotation about z axis

5. Rotate about y axis (inverse)

6. Rotate about x axis (inverse)

7. Translate (inverse)

# 1. Translate

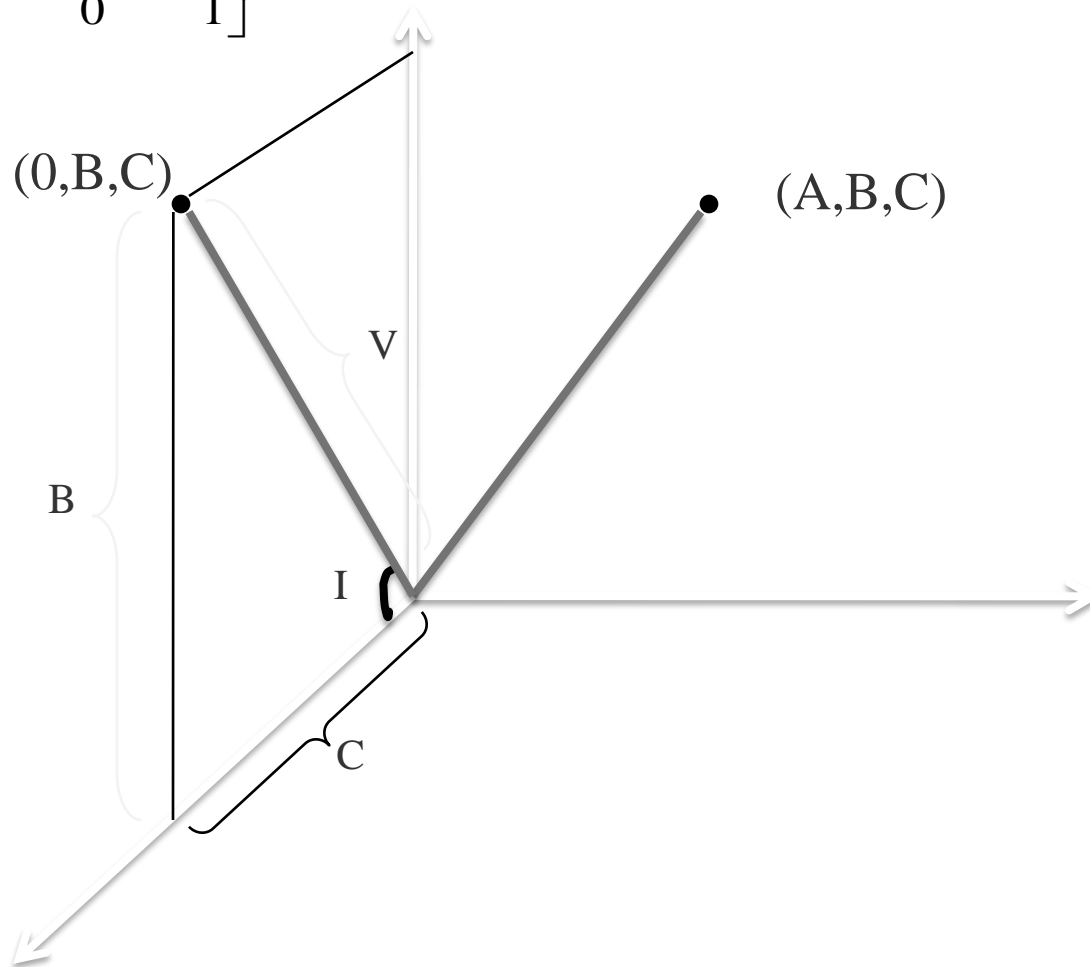
$$T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# 7. Inverse Translate

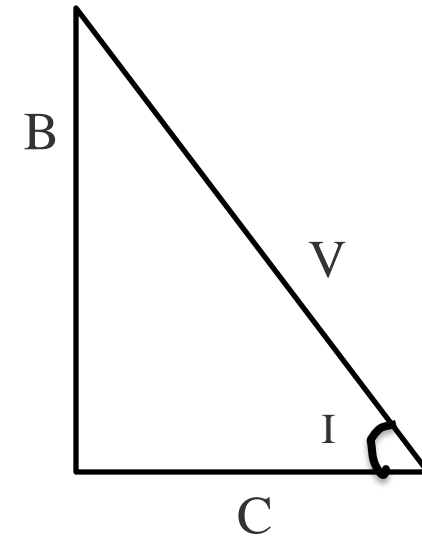
$$T = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## 2. Rotate about x axis

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



## 6. Inverse Rotation about x axis



Length of shadow

$$V = (B^2 + C^2)^{1/2}$$

$$\sin I = B/V$$

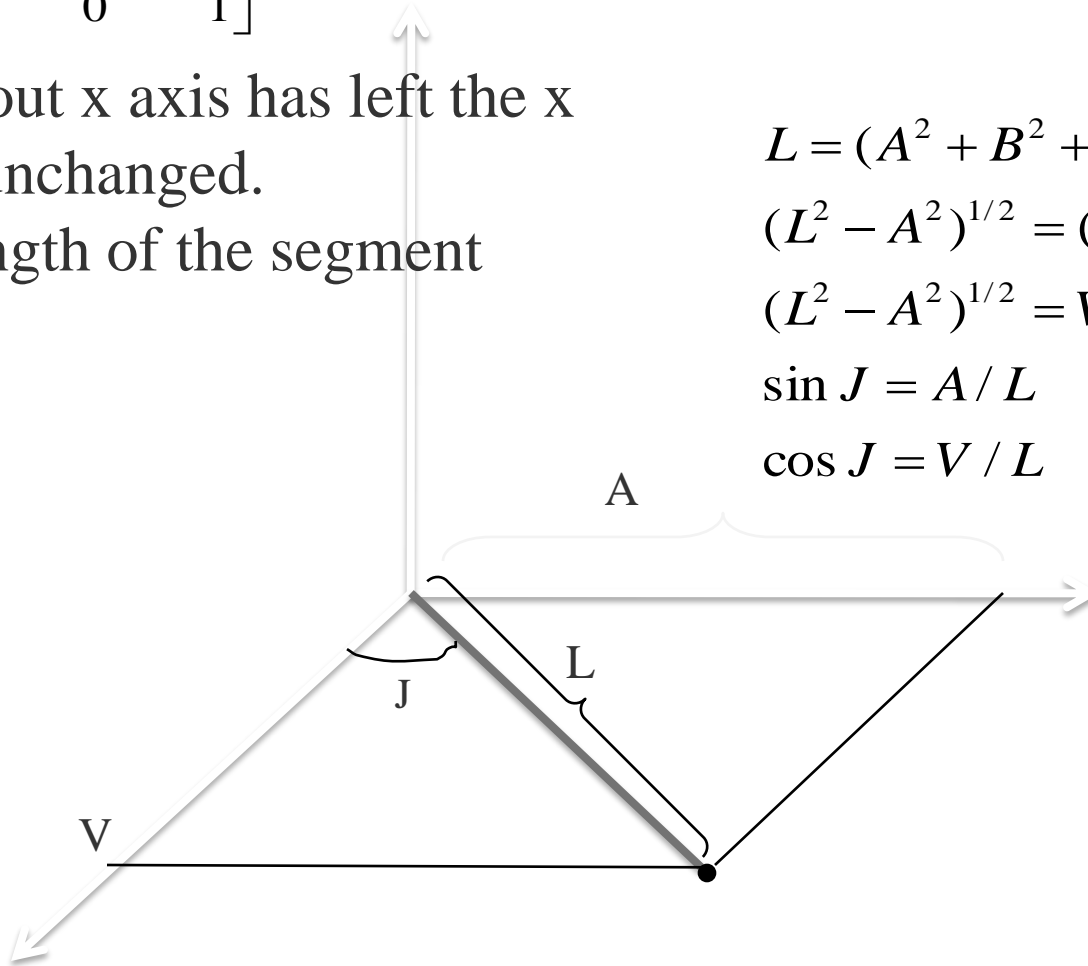
$$\cos I = C/V$$

### 3. Rotate about y axis

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

Rotation about x axis has left the x coordinate unchanged.

The total length of the segment



### 5. Inverse Rotation about y axis

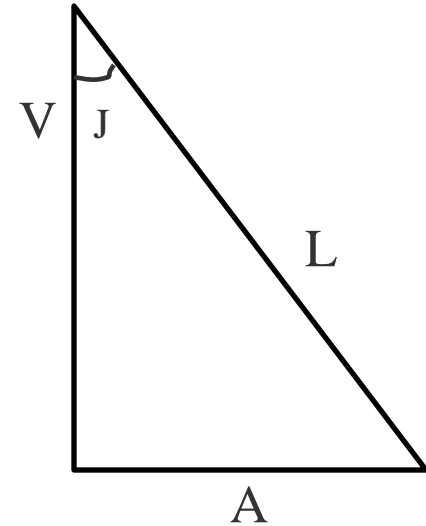
$$L = (A^2 + B^2 + C^2)^{1/2}$$

$$(L^2 - A^2)^{1/2} = (B^2 + C^2)^{1/2}$$

$$(L^2 - A^2)^{1/2} = V$$

$$\sin J = A / L$$

$$\cos J = V / L$$



## 4. Rotate around Z axis by angle $\Theta$

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} \cos \Theta & -\sin \Theta & 0 & 0 \\ \sin \Theta & \cos \Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Actual transformation for a rotation  $\Theta$  about an arbitrary axis is given by the product of the following transformations

$$T R_x R_y R_z R_y^{-1} R_x^{-1} T^{-1}$$



# Summary

## *Coordinate systems*

- World vs. modeling coordinates

## *2-D and 3-D transformations*

- Trigonometry and geometry
- Matrix representations
- Linear vs. affine transformations

## *Matrix operations*

- Matrix composition

# Viewing in 3D



# Projections

**Display device (a screen) is 2D...**

- How do we map 3D objects to 2D space?

**2D to 2D is straight forward...**

- 2D window to world.. and a viewport on the 2D surface.
- Clip what won't be shown in the 2D window, and map the remainder to the viewport.

***3D to 2D is more complicated...***

- Solution : Transform 3D objects on to a 2D plane using projections

# Projections

## In 3D...

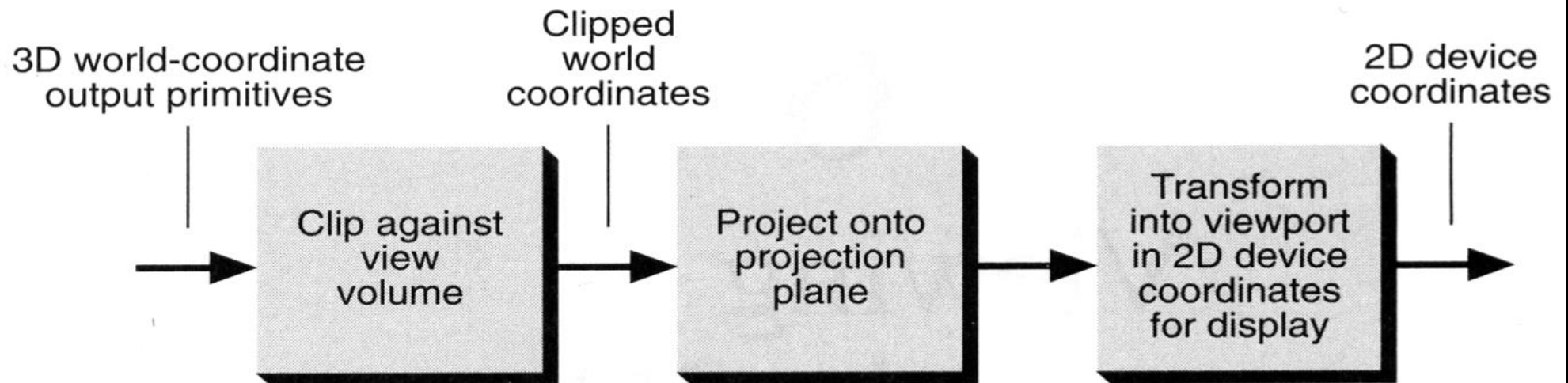
- View volume in the world
- Projection onto the 2D projection plane
- A viewport to the view surface

## Process...

- 1... clip against the view volume,
- 2... project to 2D plane, or window,
- 3... map to viewport.

# Projections

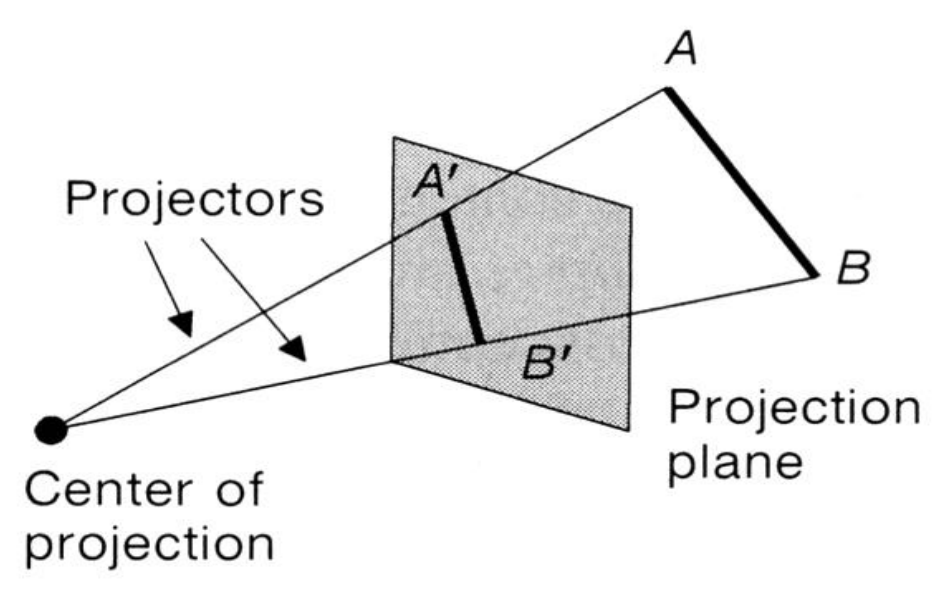
## Conceptual Model of the 3D viewing process



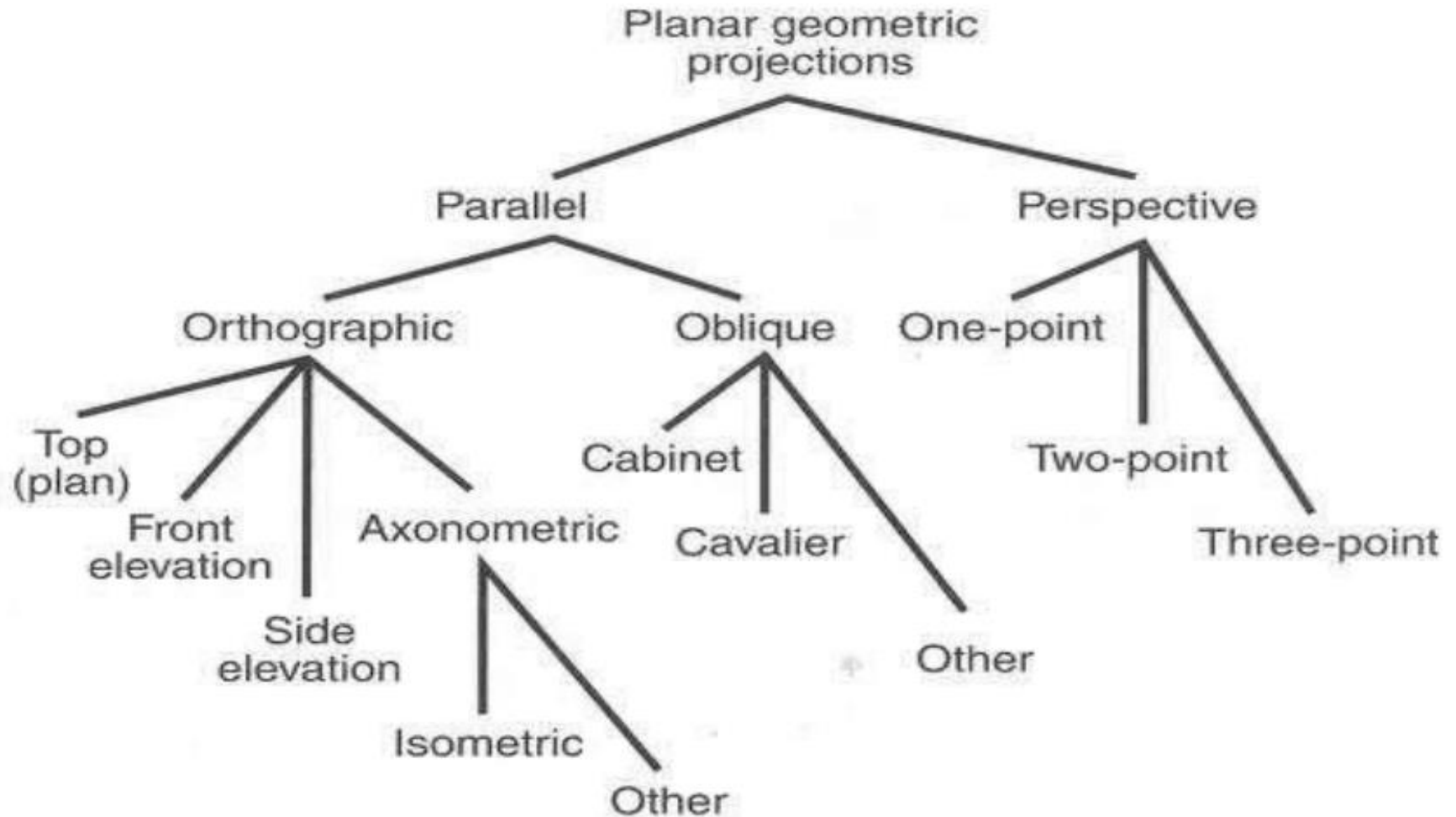
# Projections

## Projections: key terms...

- Projection from 3D to 2D is defined by straight projection rays (projectors) emanating from the 'center of projection', passing through each point of the object, and intersecting the 'projection plane' to form a projection.



# Taxonomy of Projections



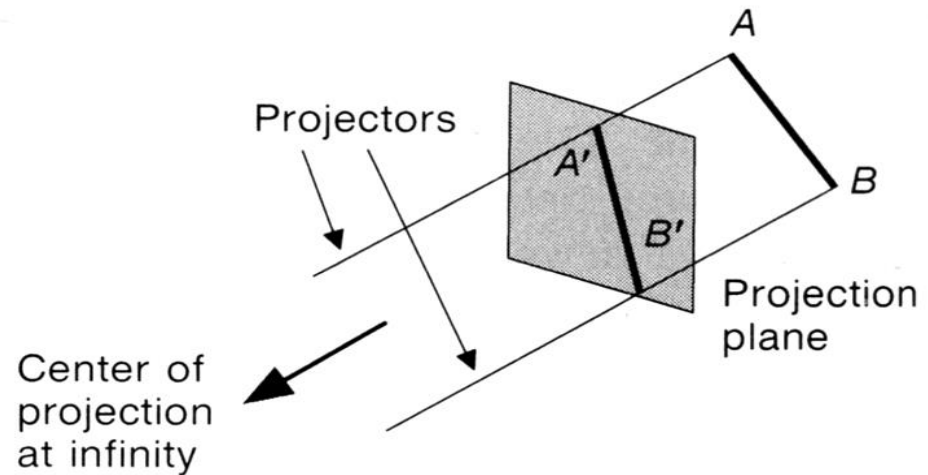
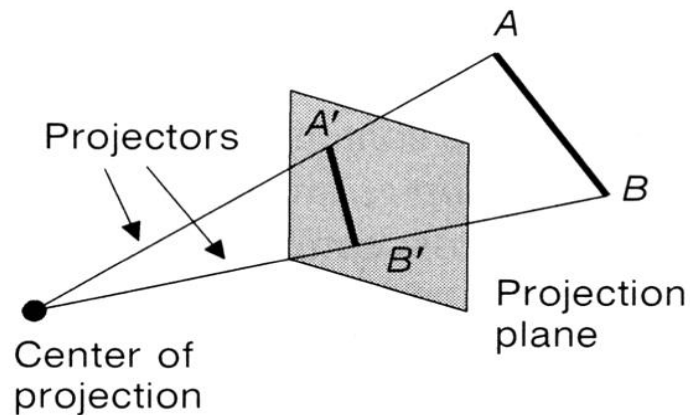
# Types of projections

## 2 types of projections

- perspective and parallel.

**Key factor is the center of projection.**

- if distance to center of projection is finite : perspective
- if infinite : parallel





# Perspective v Parallel

## Perspective:

- visual effect is similar to human visual system...
- has 'perspective foreshortening'
  - *size of object varies inversely with distance from the center of projection.*
- angles only remain intact for faces parallel to projection plane.

## Parallel:

- less realistic view because of no foreshortening
- however, parallel lines remain parallel.
- angles only remain intact for faces parallel to projection plane.

# Parallel Projections

## 2 principle types:

- orthographic and oblique.

### Orthographic :

- direction of projection = normal to the projection plane.

### Oblique :

- direction of projection  $\neq$  normal to the projection plane.

# Parallel Projections

## Orthographic (or orthogonal) projections:

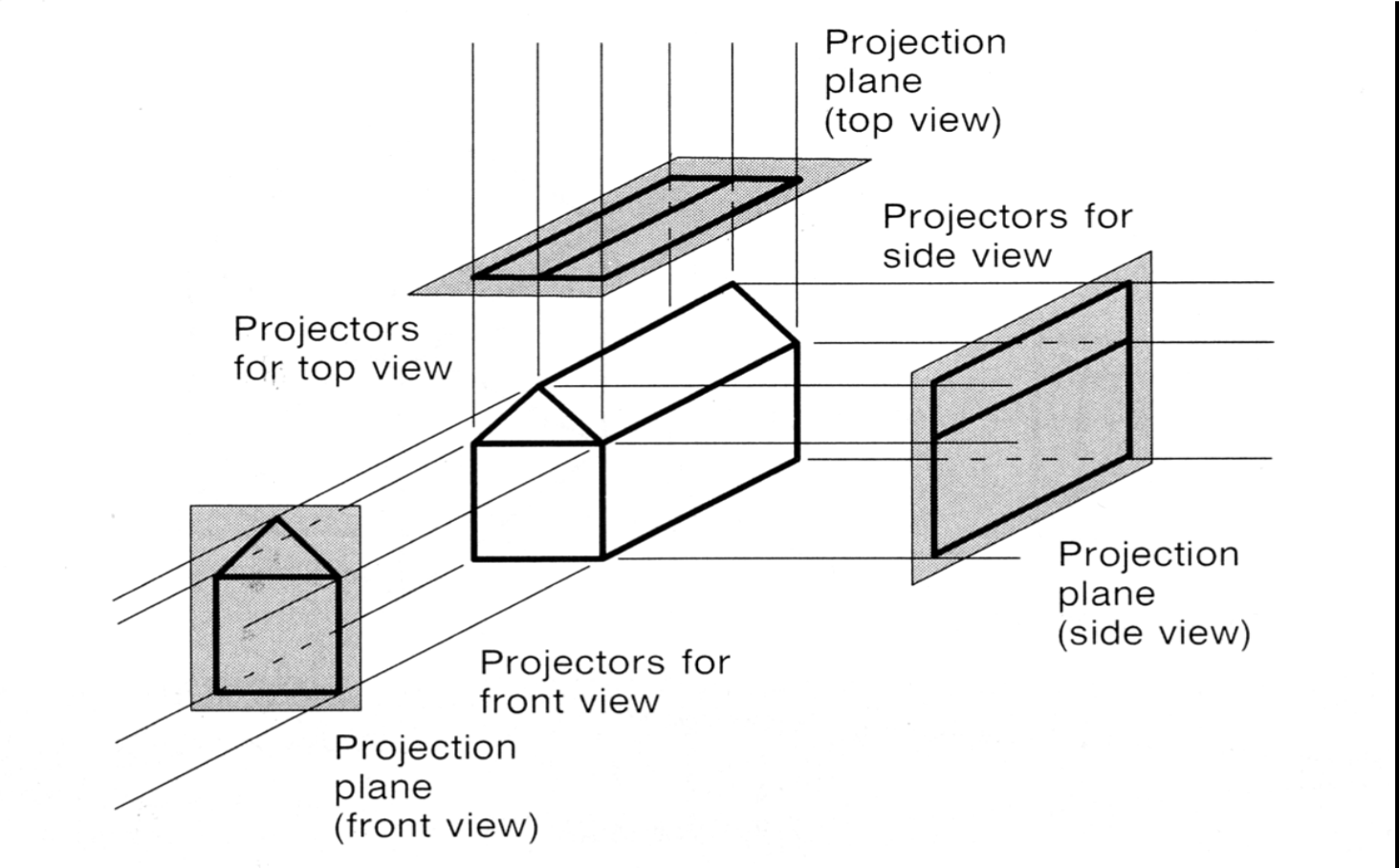
- front elevation, top-elevation and side-elevation.
- all have projection plane perpendicular to a principle axes.

**Useful because angle and distance measurements can be made...**

**However, As only one face of an object is shown, it can be hard to create a mental image of the object, even when several views are available.**

# Parallel Projections

## Orthogonal projections:



# Parallel Projections

## Oblique parallel projections

- Objects can be visualised better than with orthographic projections
- Can measure distances, but not angles\*
  - \* *Can only measure angles for faces of objects parallel to the plane*

## 2 common oblique parallel projections:

- Cavalier and Cabinet

# Parallel Projections

## Cavalier:

- The direction of the projection makes a 45 degree angle with the projection plane.
- Because there is no foreshortening, this causes an exaggeration of the z axes.

# Parallel Projections

## Cabinet:

- The direction of the projection makes a 63.4 degree angle with the projection plane. This results in foreshortening of the z axis, and provides a more “realistic” view.

# **Perspective Projections**

**Any parallel lines not parallel to the projection plane, converge at a vanishing point.**

- **There are an infinite number of these, 1 for each of the infinite amount of directions line can be oriented.**

**If a set of lines are parallel to one of the three principle axes, the vanishing point is called an axis vanishing point.**

- **There are at most 3 such points, corresponding to the number of axes cut by the projection plane.**



# Perspective Projections

## Example:

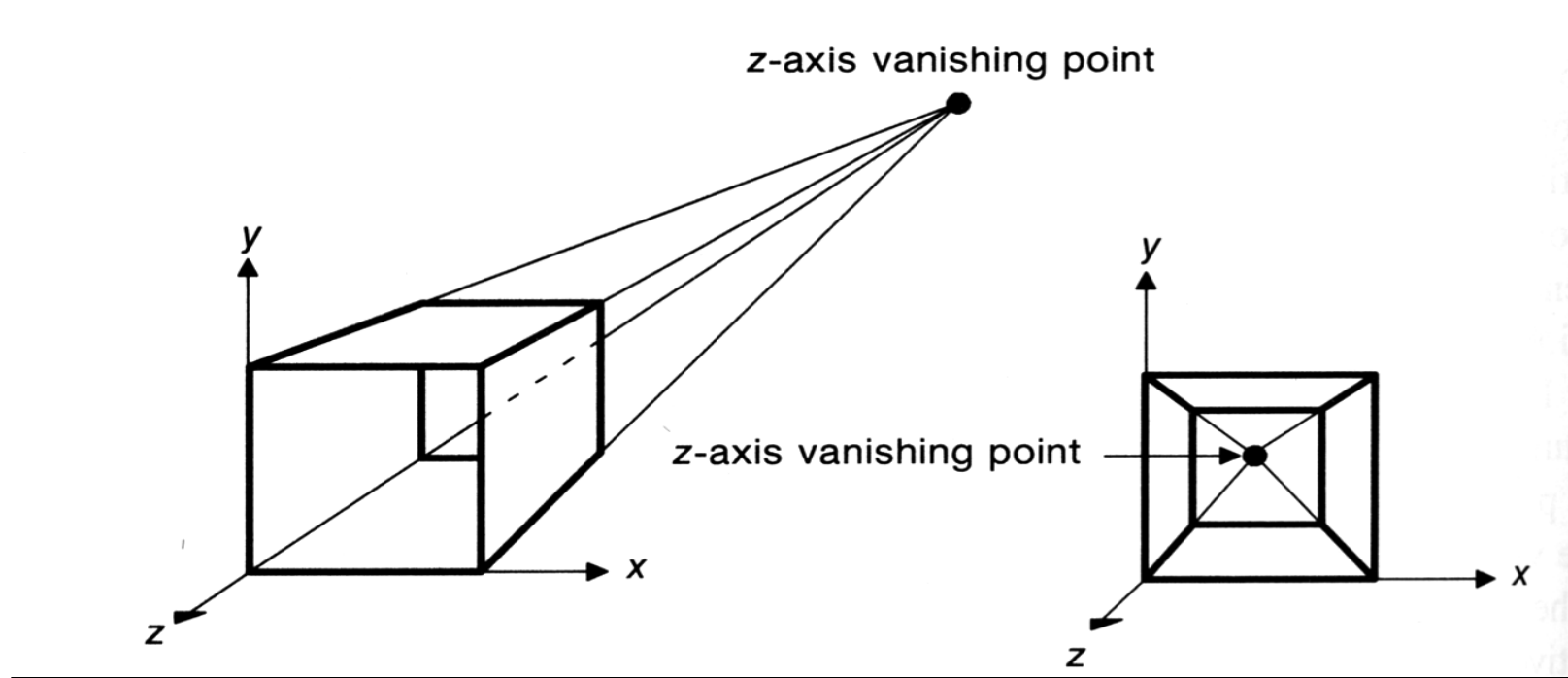
- if z projection plane cuts the z axis: normal to it, so only z has a principle vanishing point, as x and y are parallel and have none.

**Can categorise perspective projections by the number of principle vanishing points, and the number of axes the projection plane cuts.**

# Perspective Projections

example of a one-point perspective projection of a cube.

- (note: x and y parallel lines do not converge)



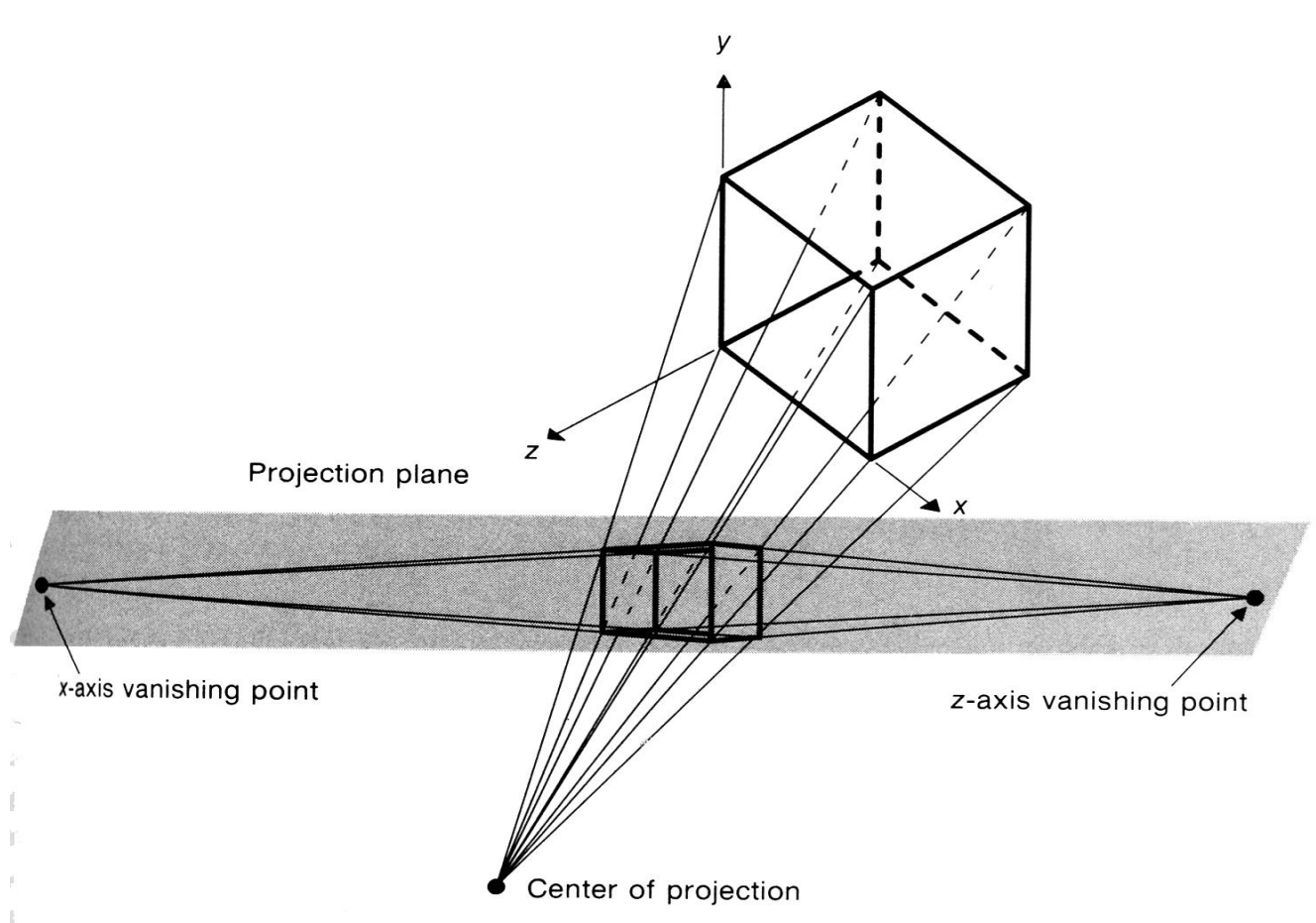
# Perspective Projections

## Two-point perspective projection:

- This is often used in architectural, engineering and industrial design drawings.
- Three-point is used less frequently as it adds little extra realism to that offered by two-point perspective projection.

# Perspective Projections

## Two-point perspective projection:



# Classical Projections



Front elevation



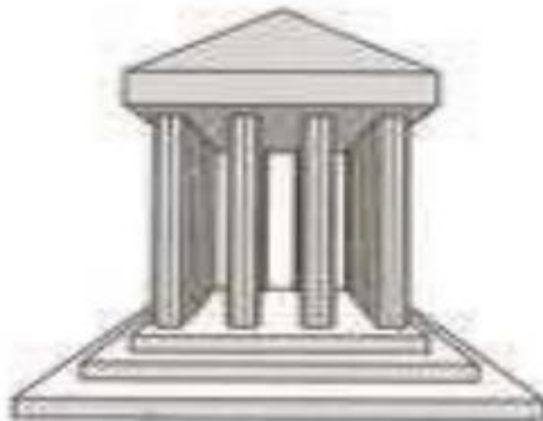
Elevation oblique



Plan oblique



Isometric



One-point perspective



Three-point perspective

1. Find transformation matrix to rotate an object about origin by 45 degree
  - i) in the counter-clockwise direction.
  - ii) Find new coordinates of the point (8,4) after rotation.
  
2. Consider the square A(1,0), B(0,0), C(0,1), D(1,1). Rotate the square ABCD by 45° anticlockwise about point A(1,0).
  
3. A triangle is defined by  $\begin{bmatrix} 2 & 4 & 4 \\ 2 & 2 & 4 \end{bmatrix}$ . Find transformed coordinates after the following transformation
  - (i) 90° rotation about the origin.
  
4. Perform 45° rotation of a triangle A(0, 0), B(1, 1) and C(5, 2). Find transformed coordinates after rotation, (a) About origin, (b) About P(- 1, - 1).
  
5. Perform 90 degree rotation of triangle with vertices A(2,2), B(4,2) and C(3,3)
  - i). About origin
  - ii). About reference point (- 2,2)
  
6. A cube is defined by 8 vertices: A(0, 0, 0), B(2, 0, 0), C(2, 2, 0), D(0, 2, 0), E(0, 0, 2), F(0, 2, 2), G(2,0,2), H(2, 2, 2). Perform following 3D transformations on the cube.
  - (i) Translation by  $T = [t_x, t_y, t_z] = [5 \ 3 \ 4]$
  - (ii) Scaling by  $S = [s_x, s_y, s_z] = [1 \ 2 \ 0.5]$