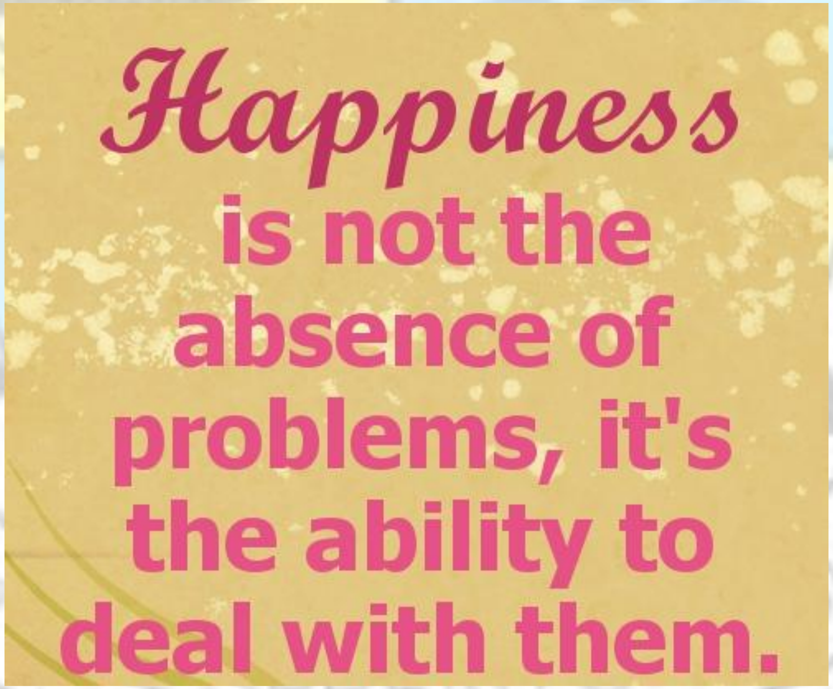
Computer Graphics

(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”, 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0 – 07 – 100472 – 6.

2. Donald D. Hearn and Baker, “Computer Graphics with OpenGL”, 4th Edition, ISBN-13: 9780136053583.

3. D. Rogers, “Procedural Elements for Computer Graphics”, 2nd 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”, 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.

2. D. Rogers, J. Adams, “Mathematical Elements for Computer Graphics”, 2nd 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/

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

Overview

2D Transformations

- Basic 2D transformations

3D Transformations

- Basic 3D transformations
- Matrix representation
- Matrix composition

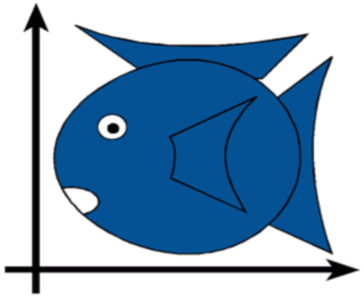
Transformation

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

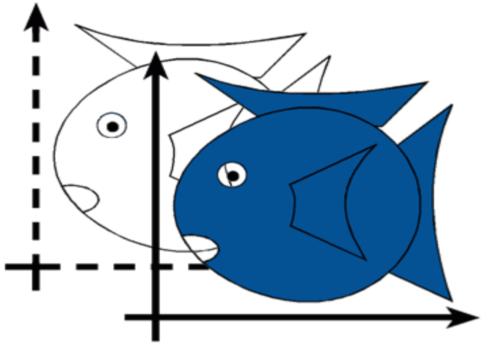
$$x' = ax + by + c$$

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

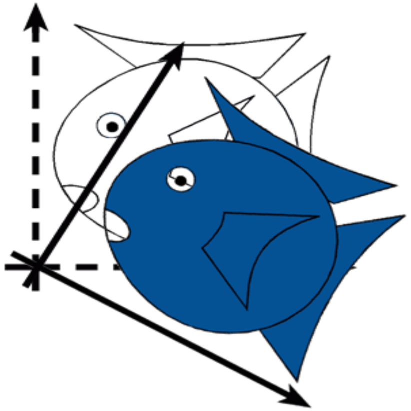
Simple Transformations



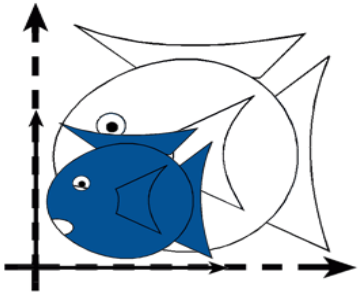
Identity



Translation



Rotation



Isotropic
(Uniform)
Scaling

Can be combined

Are these operations reversible?

Transformations are used to

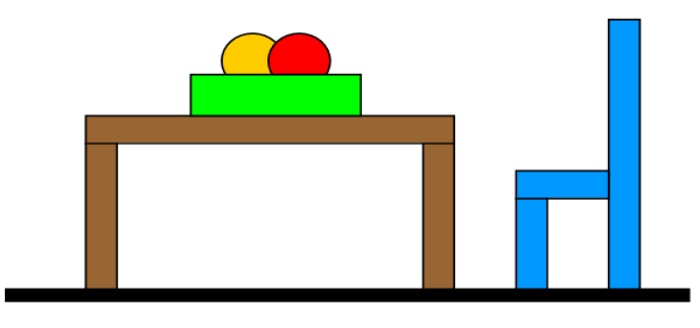
Position objects in a scene (modelling)

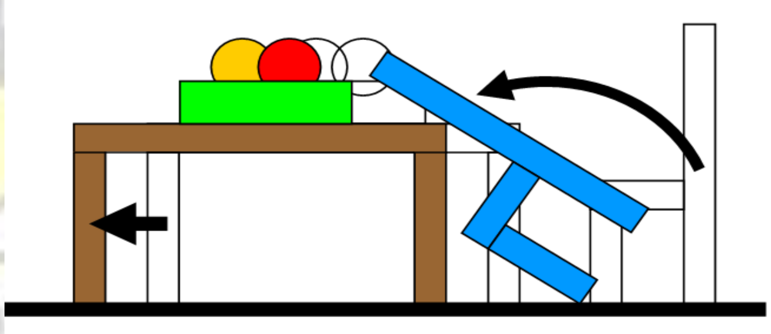
Change the shape of objects

Create multiple copies of objects

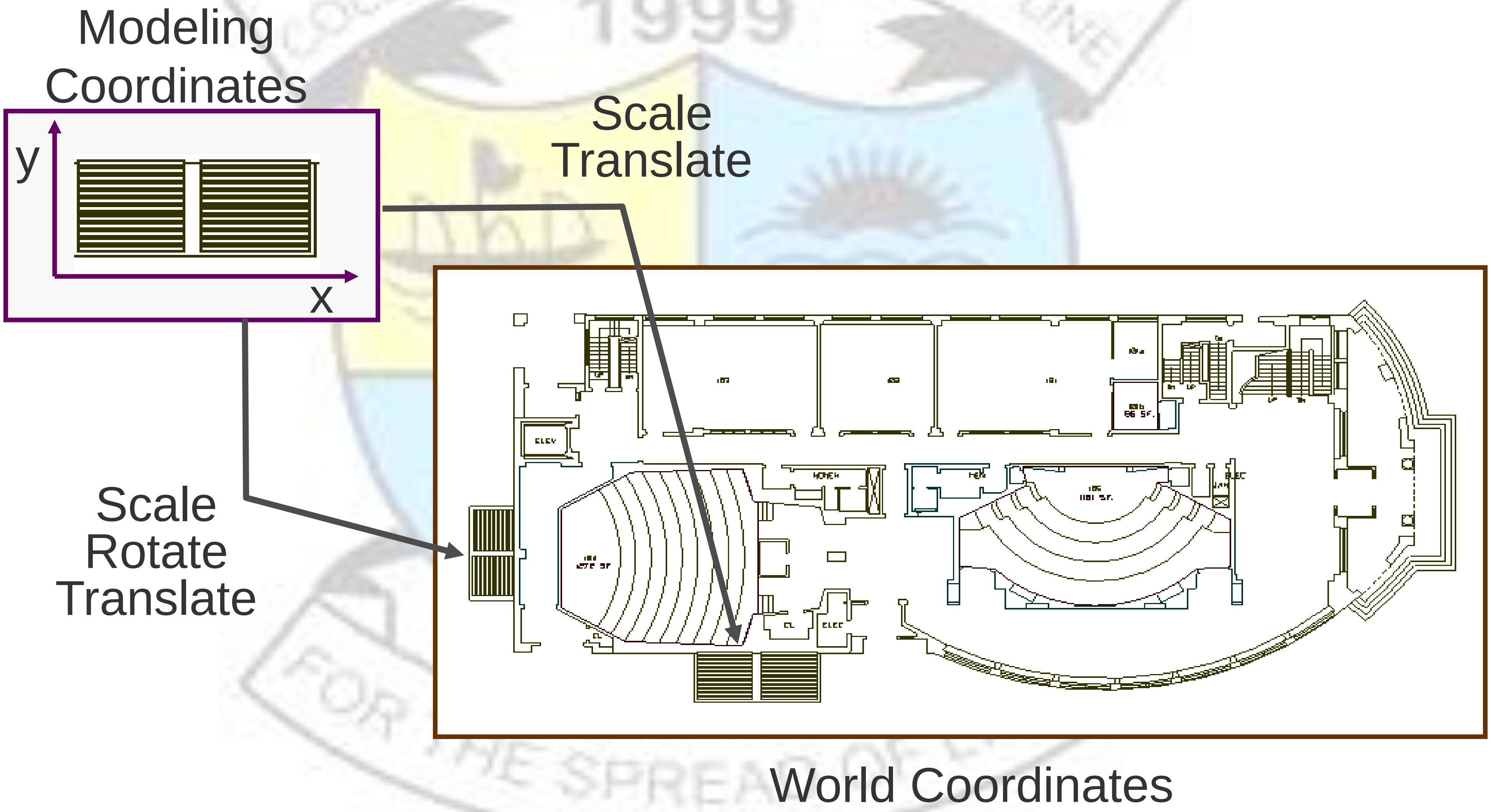
Projection for virtual cameras

Animations





2D Modeling Transformations



Modeling Coordinates

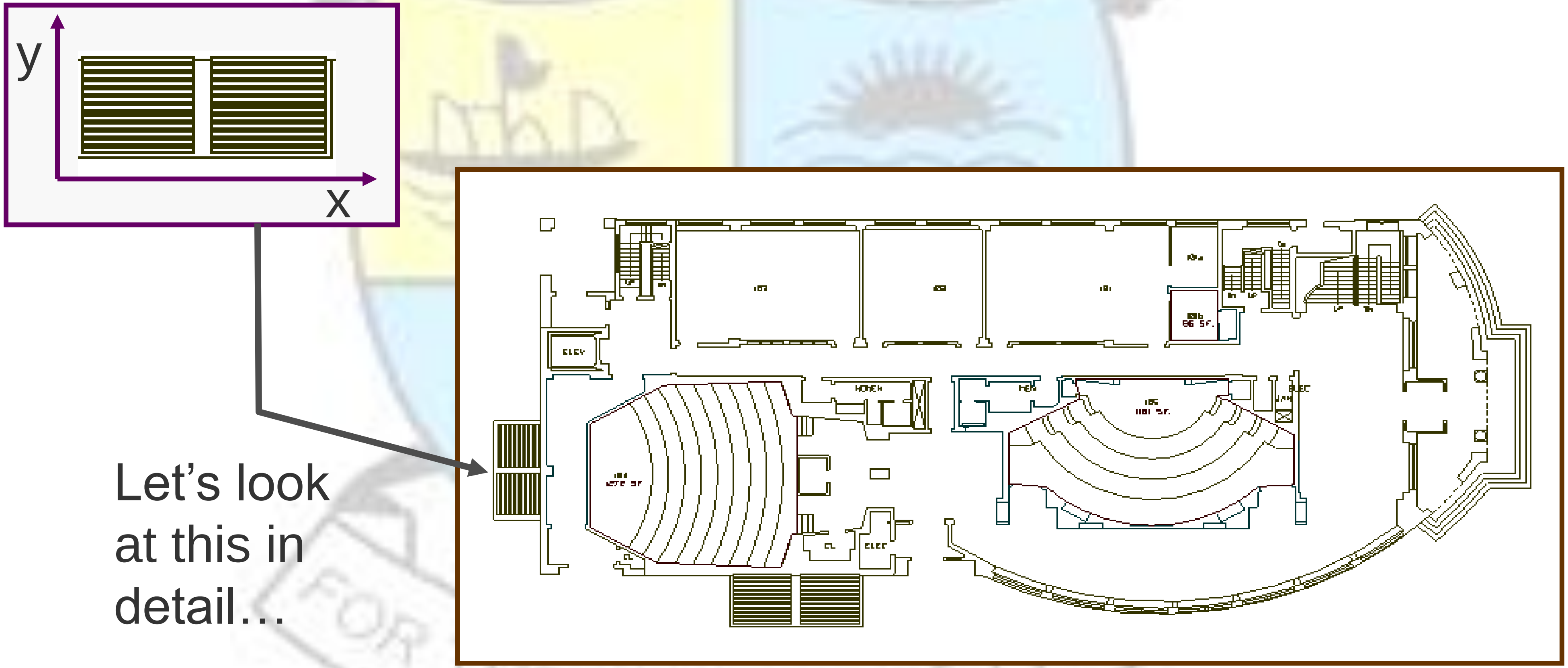
Scale
Translate

Scale
Rotate
Translate

World Coordinates

2D Modeling Transformations

Modeling Coordinates

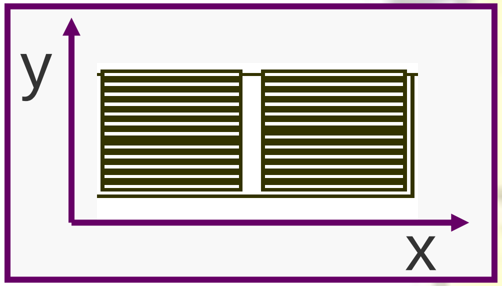


Let's look at this in detail…

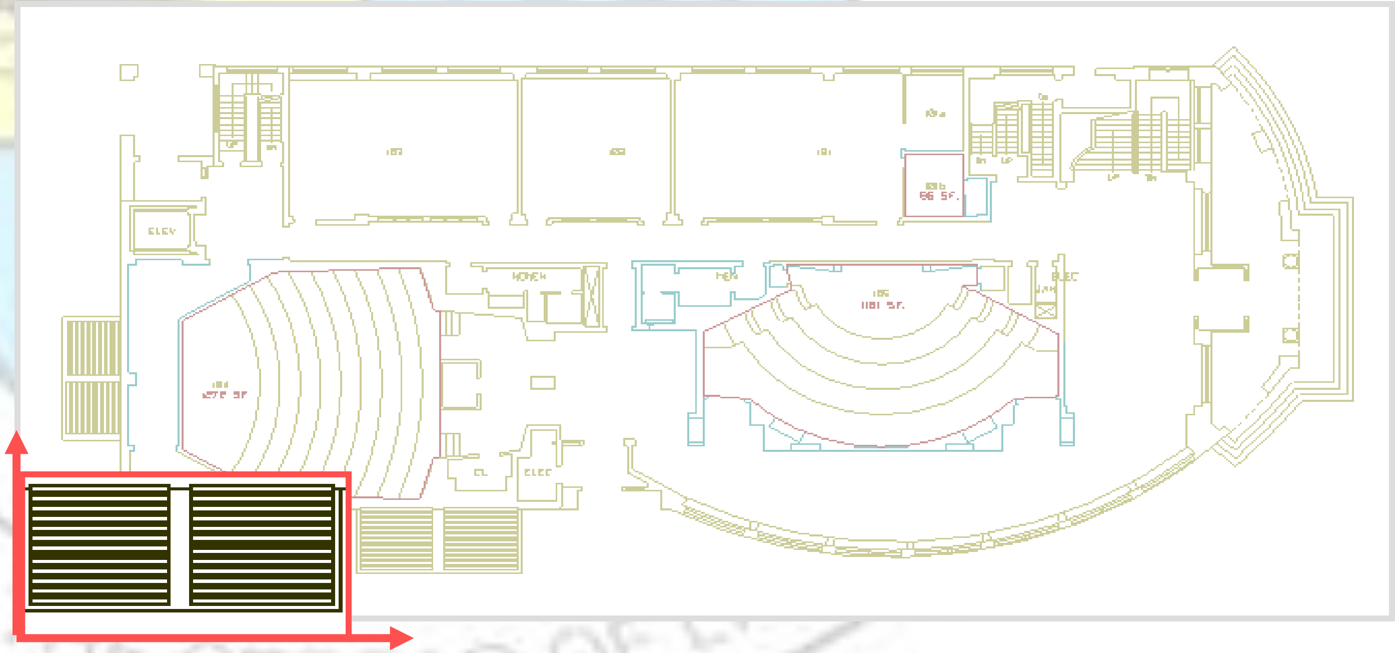
World Coordinates

2D Modeling Transformations

Modeling Coordinates

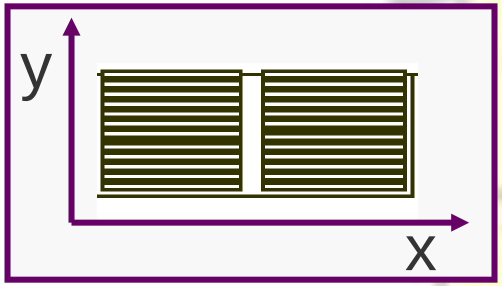


Initial location at (0, 0) with x- and y-axes aligned

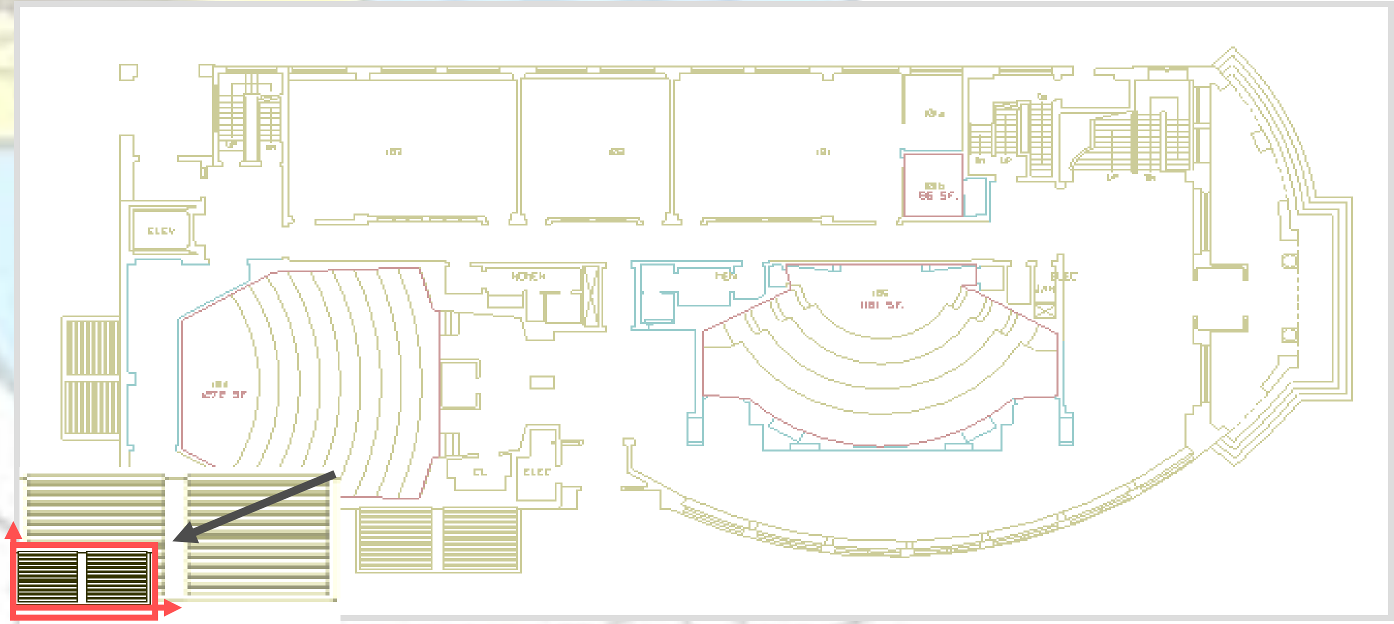


2D Modeling Transformations

Modeling Coordinates

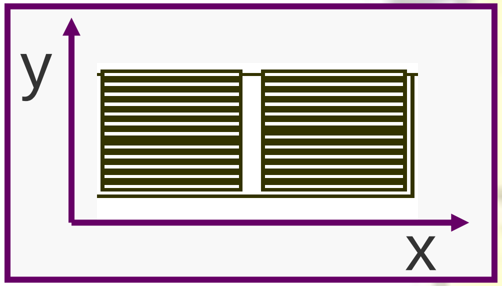


Scale .3, .3
Rotate -90
Translate 5, 3

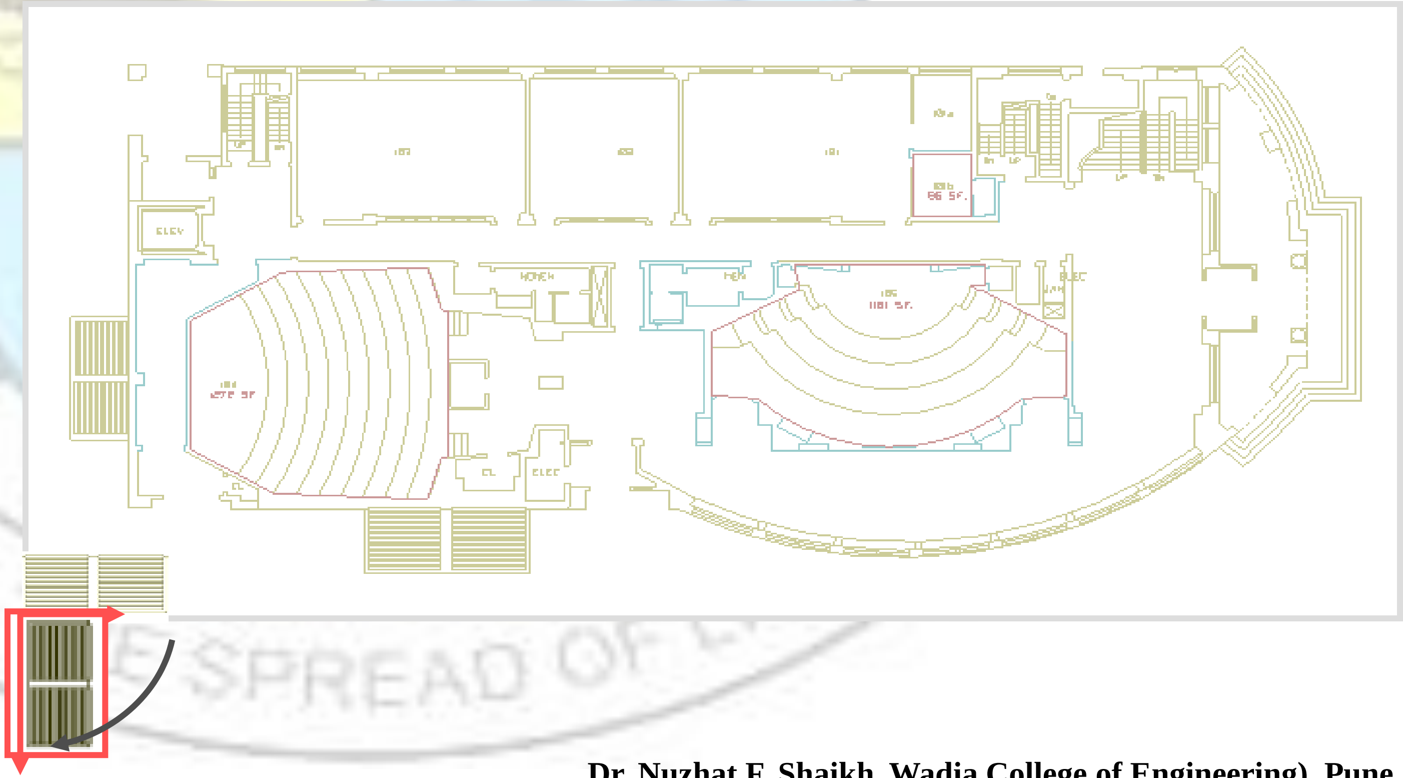


2D Modeling Transformations

Modeling Coordinates

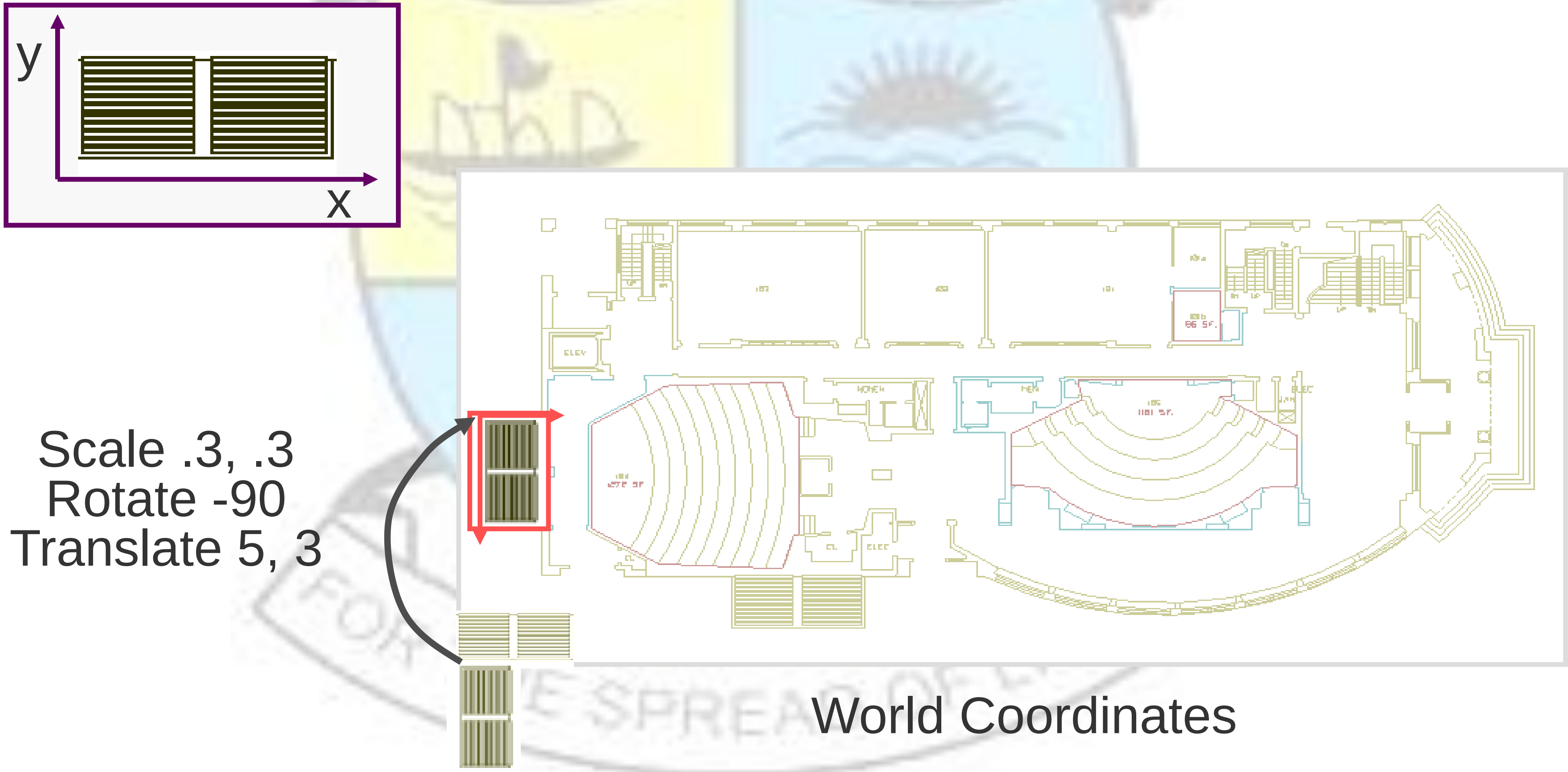


Scale .3, .3
Rotate -90
Translate 5, 3



2D Modeling Transformations

Modeling Coordinates



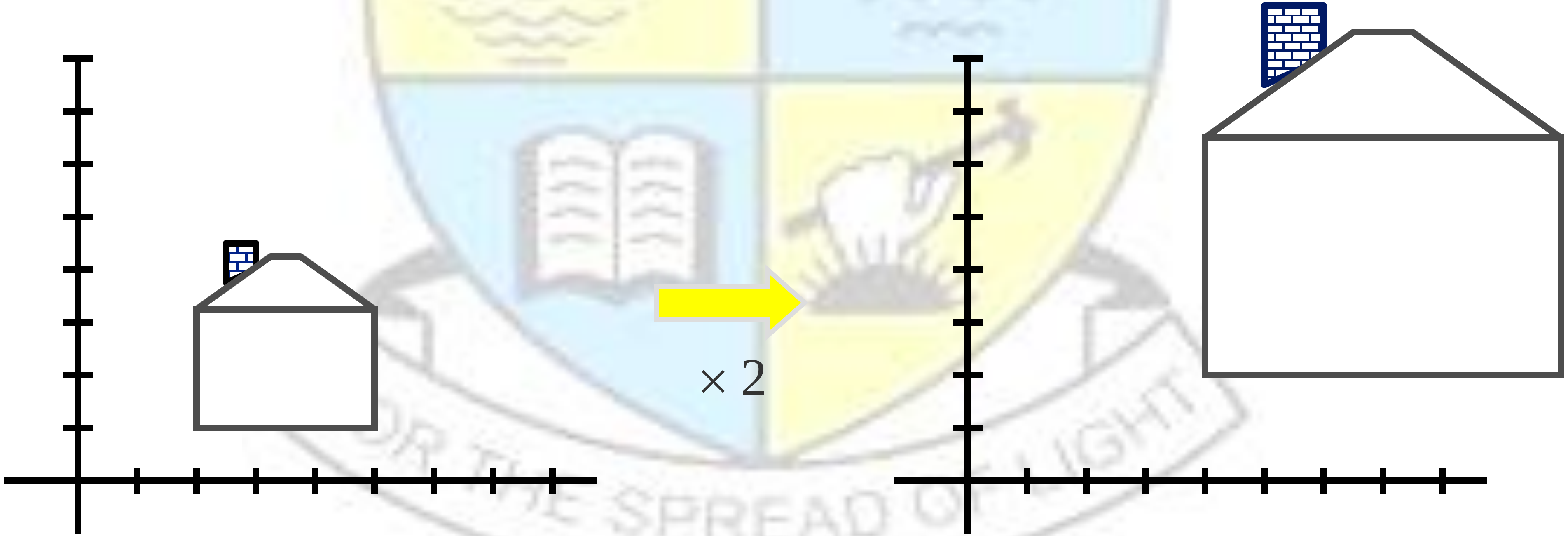
Scale .3, .3
Rotate -90
Translate 5, 3

World Coordinates

Scaling

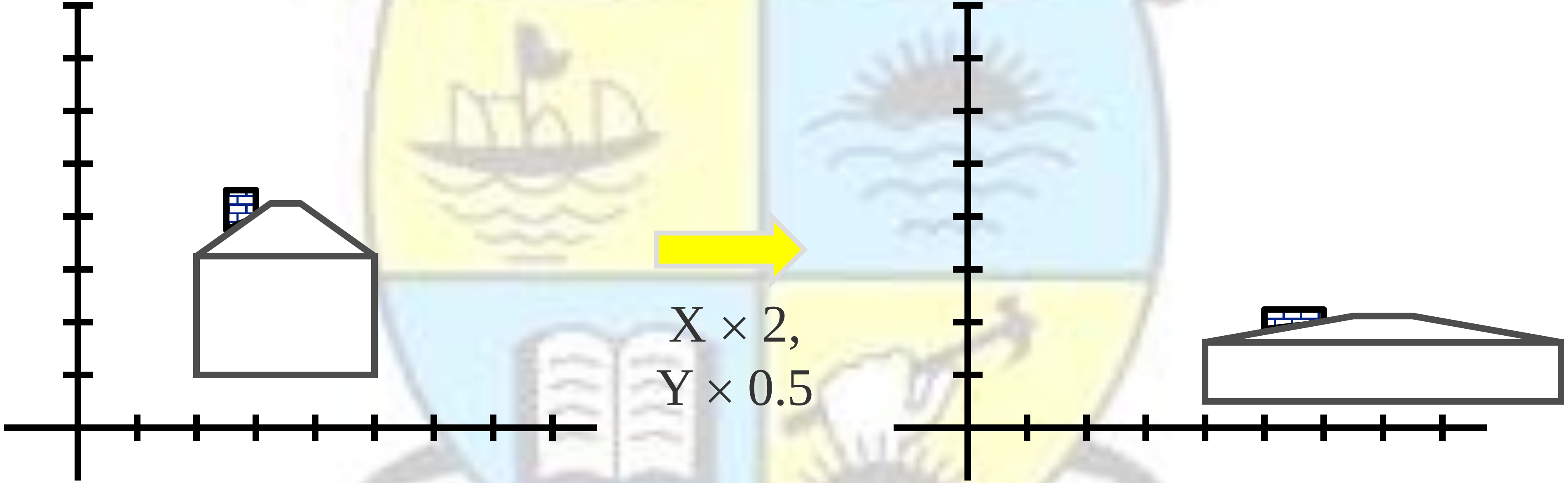
Scaling *a coordinate means multiplying each of its components by a scalar*

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



Scaling

Non-uniform scaling: different scalars per component:



X × 2,
Y × 0.5

How can we represent this in matrix form?

Scaling

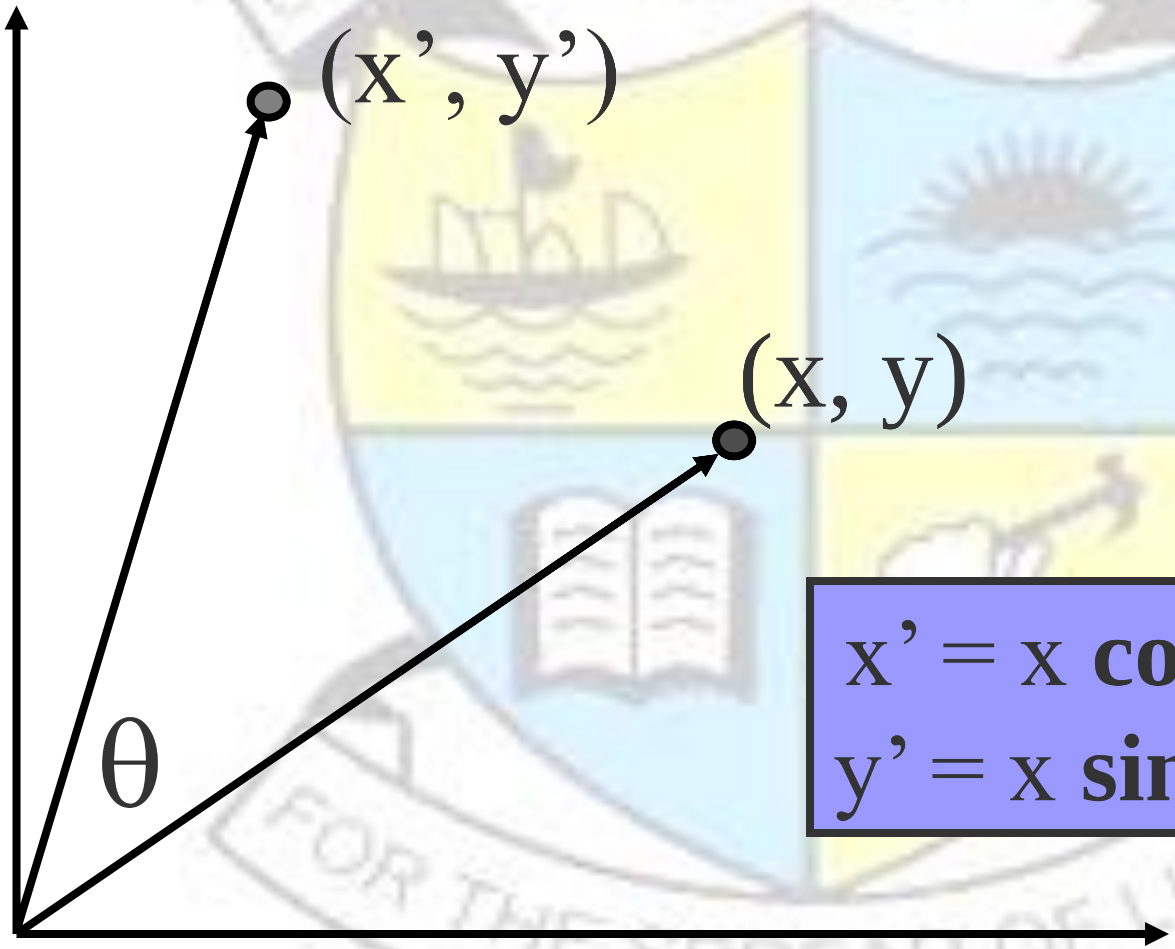
Scaling operation:

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

Or, in matrix form:

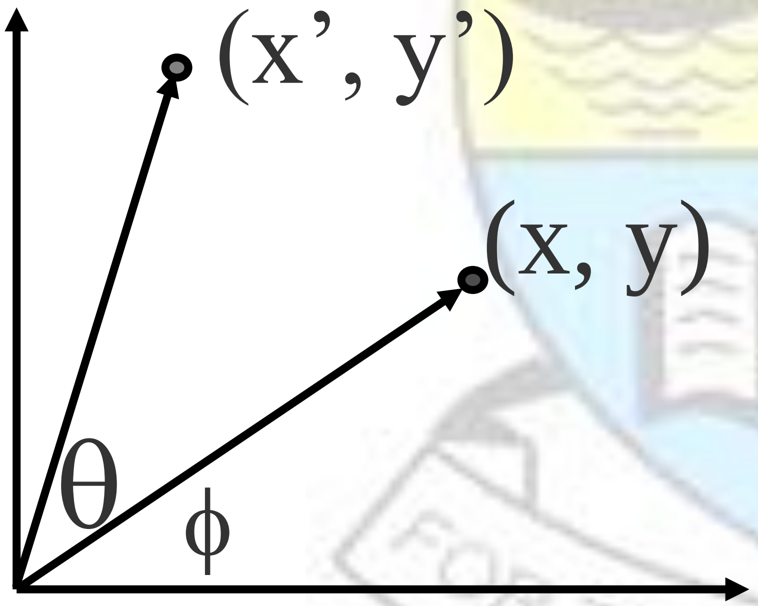
$$\boxed{\begin{bmatrix}x'\\ y'\end{bmatrix}=\underbrace{\begin{bmatrix}\overset{S_x}{a} & 0\\ 0 & \underset{S_y}{b}\end{bmatrix}}_{\textit{scaling matrix}}\begin{bmatrix}x\\ y\end{bmatrix}}$$

2-D Rotation



$$x' = x\,\mathbf{cos}(\theta) - y\,\mathbf{sin}(\theta)$$
$$y' = x\,\mathbf{sin}(\theta) + y\,\mathbf{cos}(\theta)$$

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…

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

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

2-D Rotation

This is easy to capture in matrix form:

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

Even though sin(q) and cos(q) are nonlinear functions of q,

- x’ is a linear combination of x and y
- y’ is a linear combination of x and y

Basic 2D Transformations

Translation:

- x' = x + tx
- y' = y + ty

Scale:

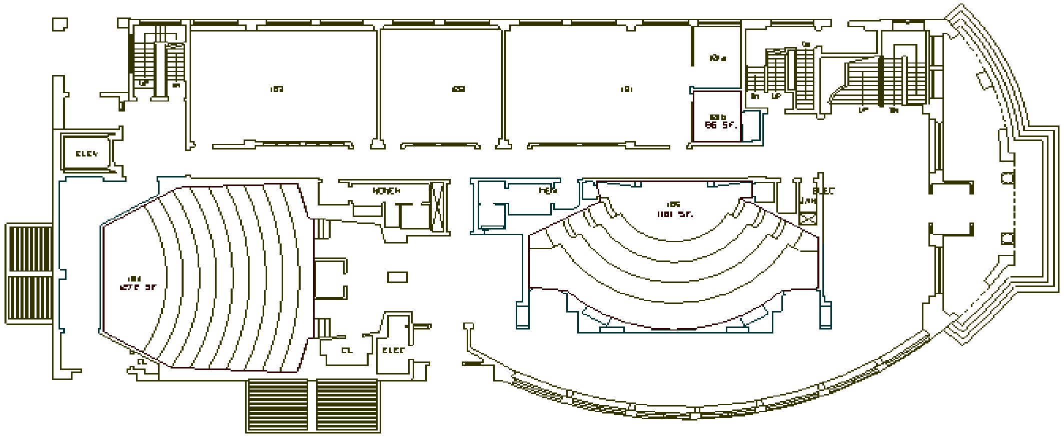
- x' = x * sx
- y' = y * sy

Shear:

- x' = x + hx*y
- y' = y + hy*x

Rotation:

- x' = x*cosQ - y*sinQ
- y' = x*sinQ + y*cosQ

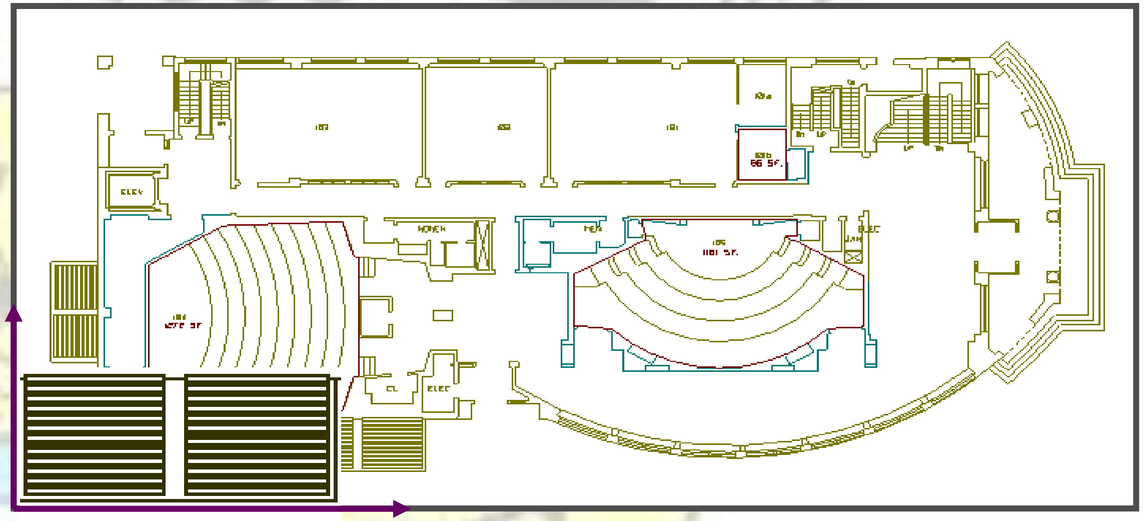


Transformations can be combined (with simple algebra)

Basic 2D Transformations

Translation:

- x’ = x + tx
- y’ = y + ty



Scale:

- x’ = x * sx
- y’ = y * sy

Shear:

- x’ = x + hx*y
- y’ = y + hy*x

Rotation:

- x’ = x*cosQ - y*sinQ
- y’ = x*sinQ + y*cosQ

Basic 2D Transformations

Translation:

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

Scale:

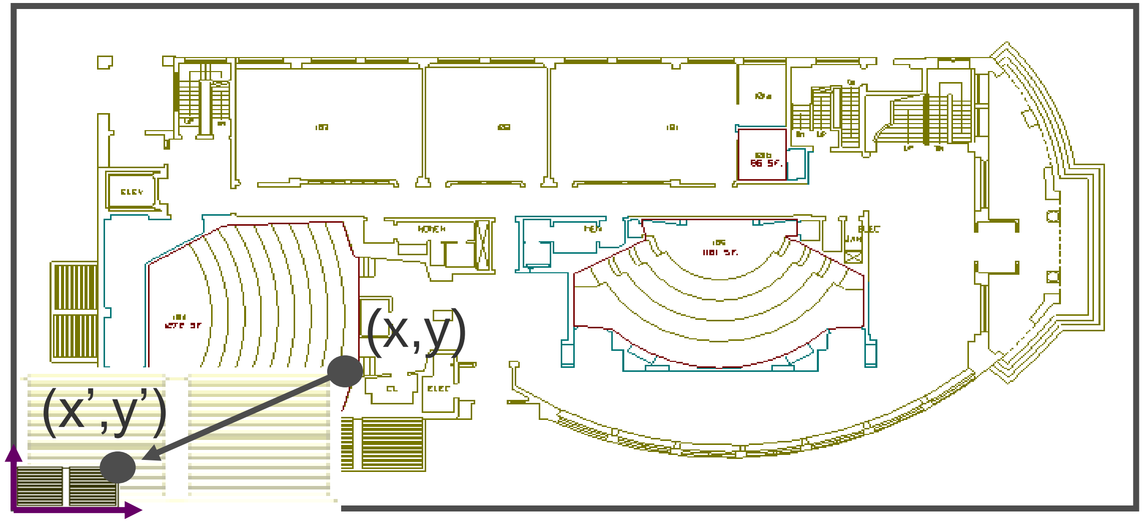
- $x' = x * s_x$
- $y' = y * s_y$

Shear:

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

Rotation:

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



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

Basic 2D Transformations

Translation:

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

Scale:

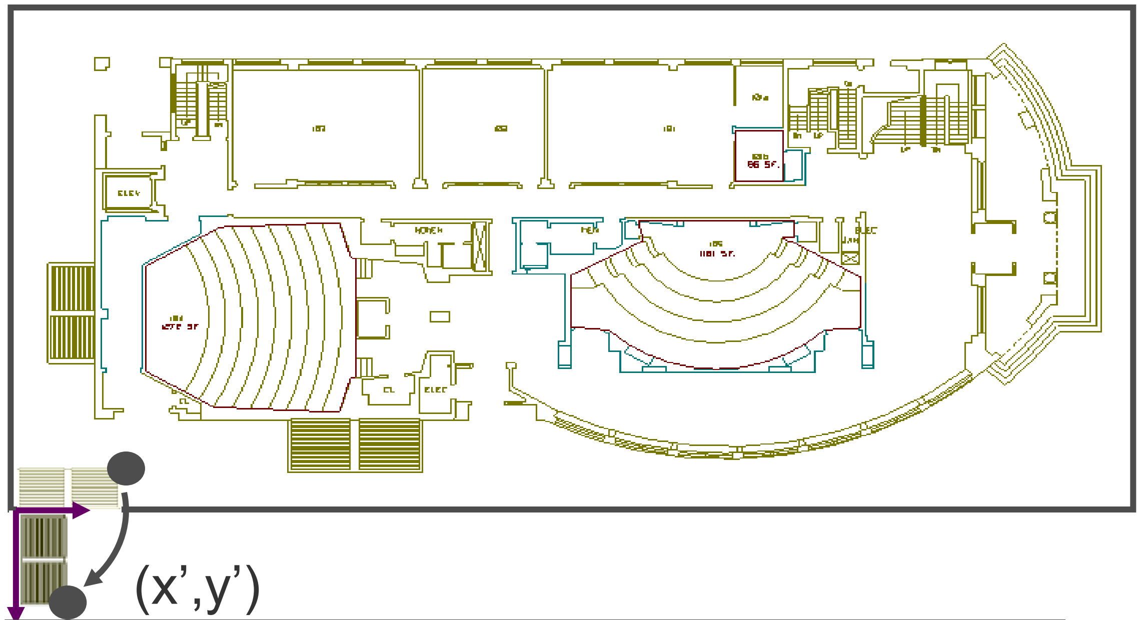
- $x' = x * s_x$
- $y' = y * s_y$

Shear:

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

Rotation:

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



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

Basic 2D Transformations

Translation:

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

Scale:

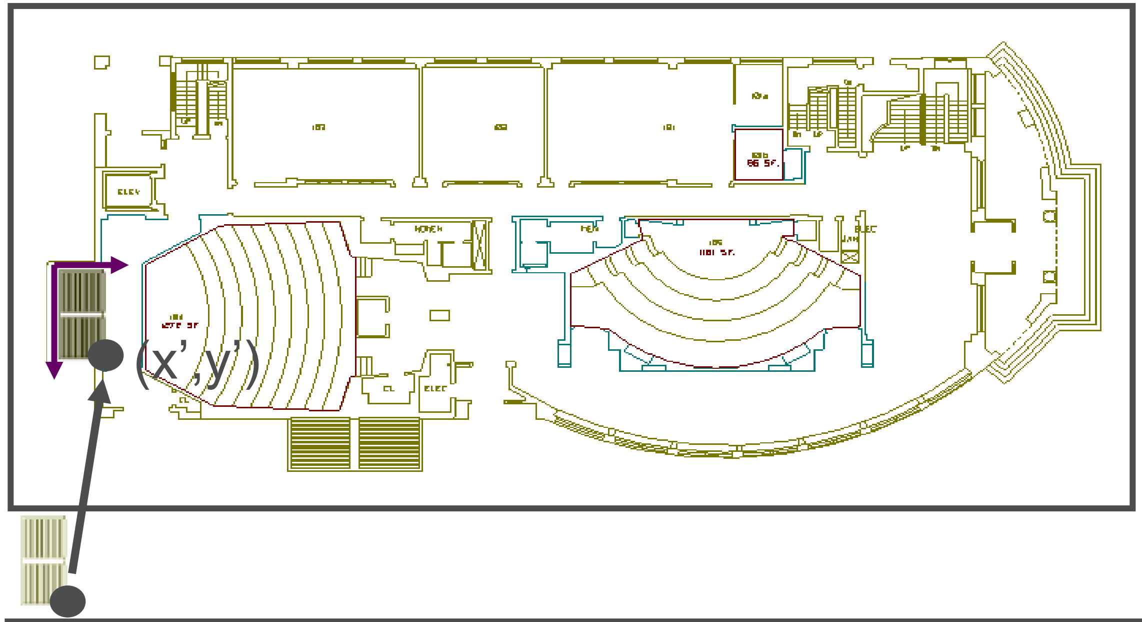
- $x' = x * s_x$
- $y' = y * s_y$

Shear:

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

Rotation:

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



$$x' = ((x * s_x) * \cos\Theta - (y * s_y) * \sin\Theta) + t_x$$
$$y' = ((x * s_x) * \sin\Theta + (y * s_y) * \cos\Theta) + t_y$$

Basic 2D Transformations

Translation:

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

Scale:

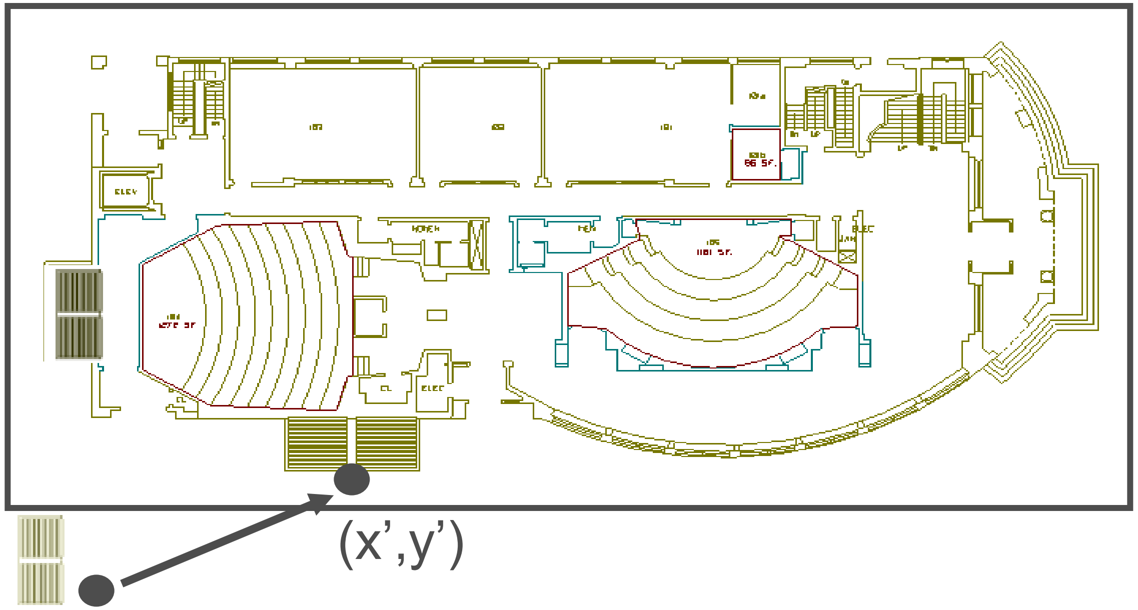
- $x' = x * s_x$
- $y' = y * s_y$

Shear:

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

Rotation:

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



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

$y' = ((x*s_x)*\sin\Theta + (y*s_y)*\cos\Theta) + t_y$

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

$\Leftrightarrow$ apply transformation to a point

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

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?

$$x' = x$$
$$y' = y$$

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

2D Scale around (0,0)?

$$\boldsymbol{x' = s_x * x}$$

$$\boldsymbol{y' = s_y * y}$$

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

2x2 Matrices

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

2D Rotate around (0,0)?

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

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

2D Shear?

$$\boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{sh_x} * \boldsymbol{y}$$
$$\boldsymbol{y'} = \boldsymbol{sh_y} * \boldsymbol{x} + \boldsymbol{y}$$

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

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?

$$x' = x + t_x$$

$$y' = y + t_y$$

NO!

DON'T TAKE REST AFTER YOUR FIRST VICTORY.
BECAUSE IF YOU FAIL IN SECOND,
MORE LIPS ARE WAITING TO SAY,
THAT YOUR FIRST VICTORY WAS JUST LUCK

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

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{\text{homogeneous coords}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

Homogeneous Coordinates

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

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

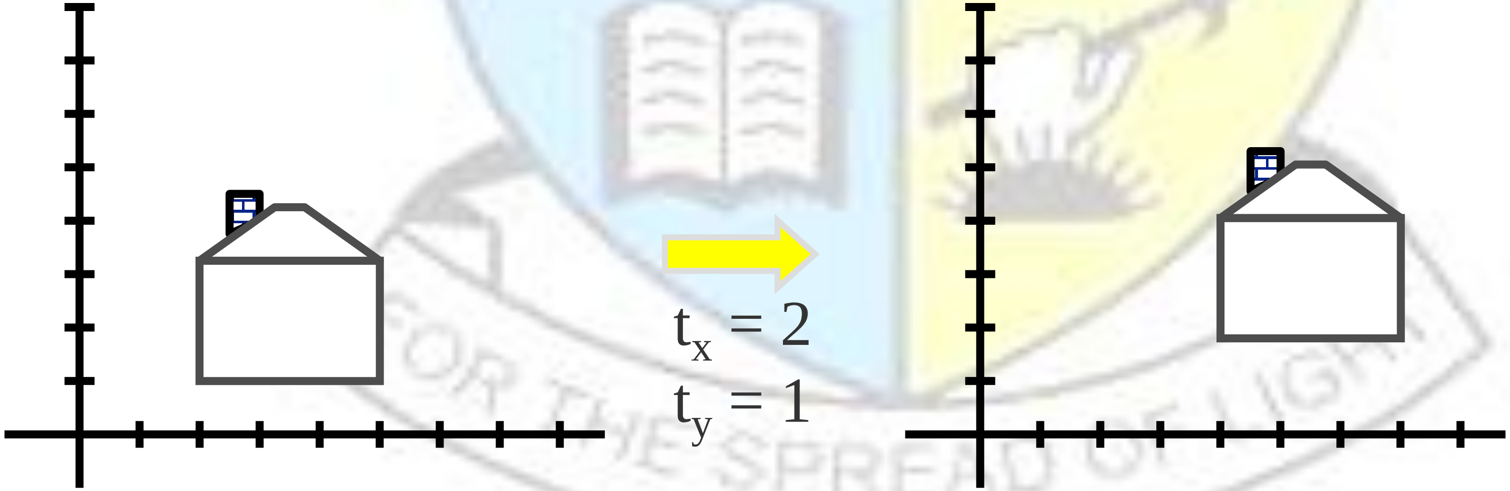
A: Using the rightmost column:

$$\mathbf{Translation} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Translation

Homogeneous Coordinates

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



$t_x = 2$

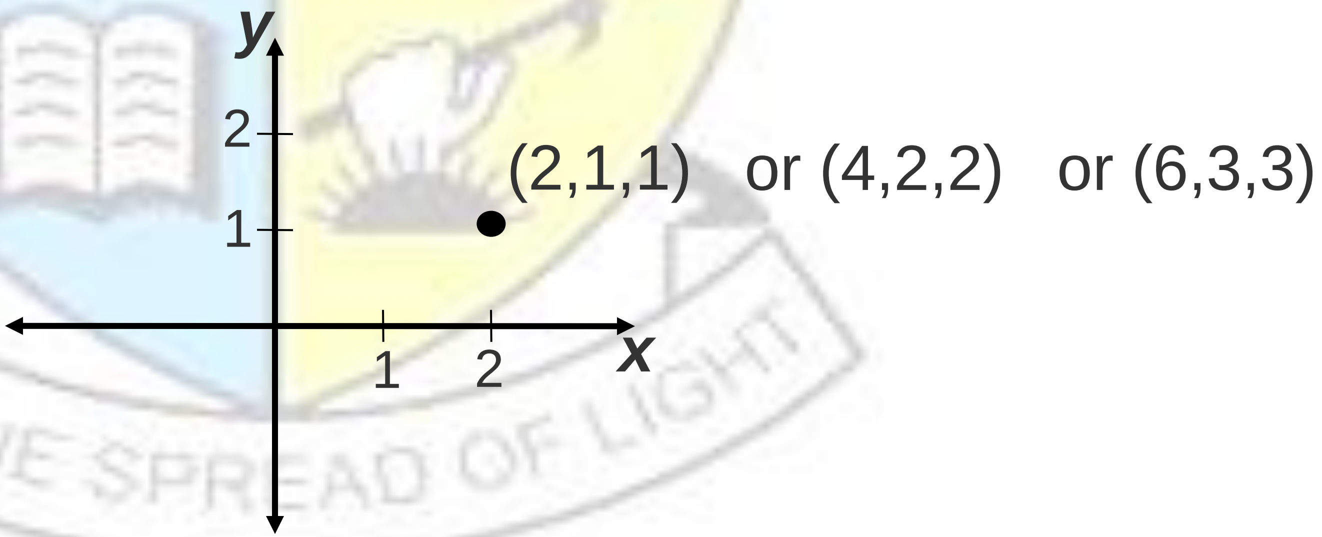
$t_y = 1$

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

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

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

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{sh}_x & 0 \\ \boldsymbol{sh}_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

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.

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Sol.
A'(0,0)
B'(9,0)
C'(9,9)
D'(0,9)

Example 1

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 2

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

THE BEGINNING OF YOUR SUCCESS IS……

AT THE END OF YOUR COMFORT ZONE !!!!!

Matrix Composition

Transformations can be combined by matrix multiplication

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

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

Matrix Composition

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

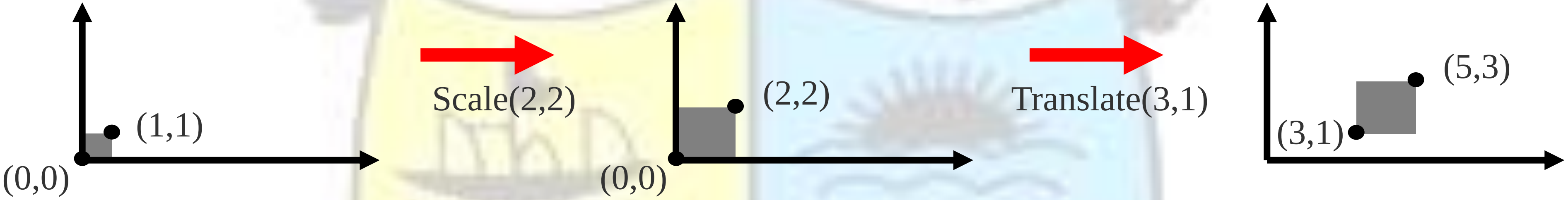
- General purpose representation
- Hardware matrix multiply

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

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

How are transforms combined?

Scale then Translate



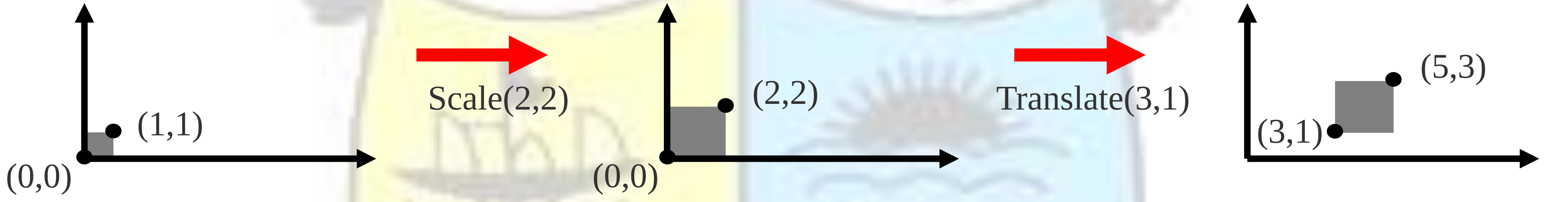
Use matrix multiplication: p' = T (S p) = TS p

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

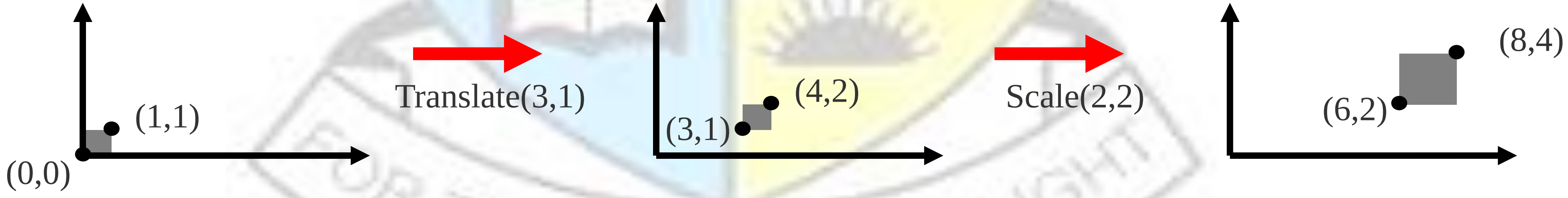
Caution: matrix multiplication is NOT commutative!

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

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

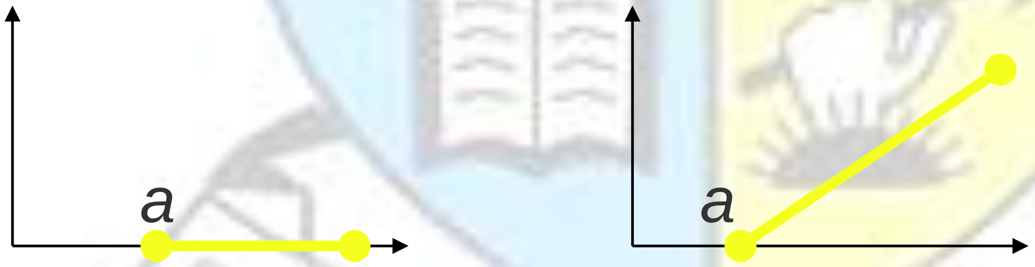
Translate then Scale: p' = S (T p) = ST p

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

Matrix Composition

What if we want to rotate *and* scale?

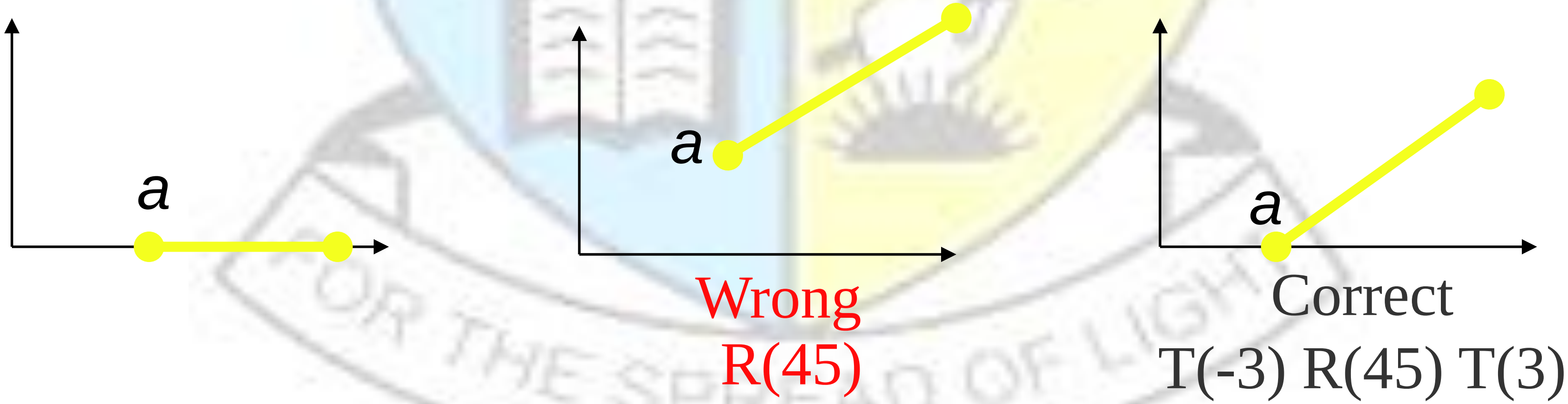
- Ex: Rotate line segment by 45 degrees about endpoint *a* *and lengthen*



Multiplication Order – Wrong Way

Our line is defined by two endpoints

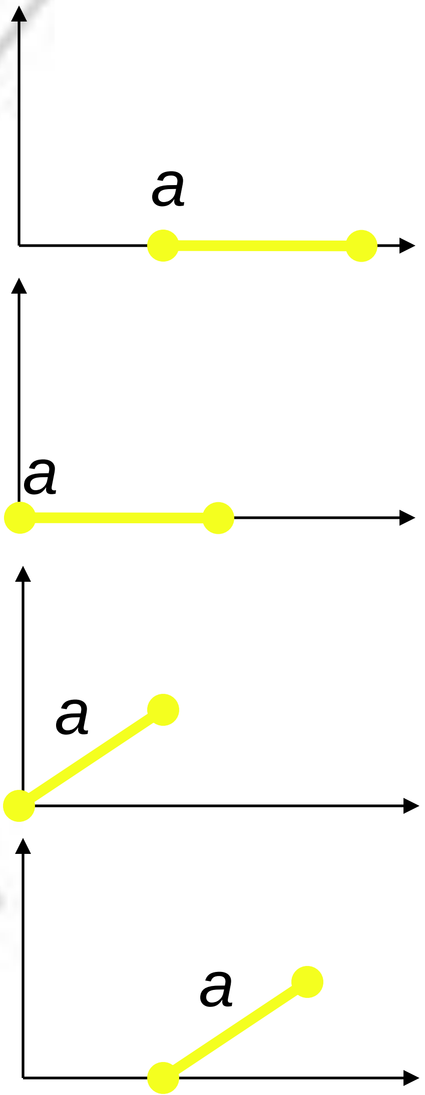
- Applying a rotation of 45 degrees, R(45), affects both points
- We could try to translate both endpoints to return endpoint *a* to its original position, but by how much?



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)



Matrix Composition

Will this sequence of operations work?

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

NO!!!

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 arbitray point

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

Three steps:

1) Move (R_x, R_y) to origin

2) Rotate θ degrees about origin

3) Move back to (R_x, R_y)

Example, step 1)

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

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

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

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

Exercise 4.11 from Harrington Pg. 143

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

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

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

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

$$sh_y = s_x \sin\Theta$$

$$s_x \cos\Theta = 1$$

$$s_y \cos\Theta = 1$$

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

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

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

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

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

Exercise 4.11 from Harrington Pg. 143

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

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

Overview

2D Transformations

- Basic 2D transformations
- Matrix representation
- Matrix composition

3D Transformations

- Basic 3D transformations
- Same as 2D

Time is what we want most,
but what we use worst.

William Penn

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

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

Identity

$$\begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ \boldsymbol{z'} \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_x & 0 & 0 & 0 \\ 0 & \boldsymbol{s}_y & 0 & 0 \\ 0 & 0 & \boldsymbol{s}_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix}$$

Scale

$$\begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ \boldsymbol{z'} \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \boldsymbol{t}_x \\ 0 & 1 & 0 & \boldsymbol{t}_y \\ 0 & 0 & 1 & \boldsymbol{t}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix}$$

Translation

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

Mirror about Y/Z plane

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:

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

Reverse Rotations

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

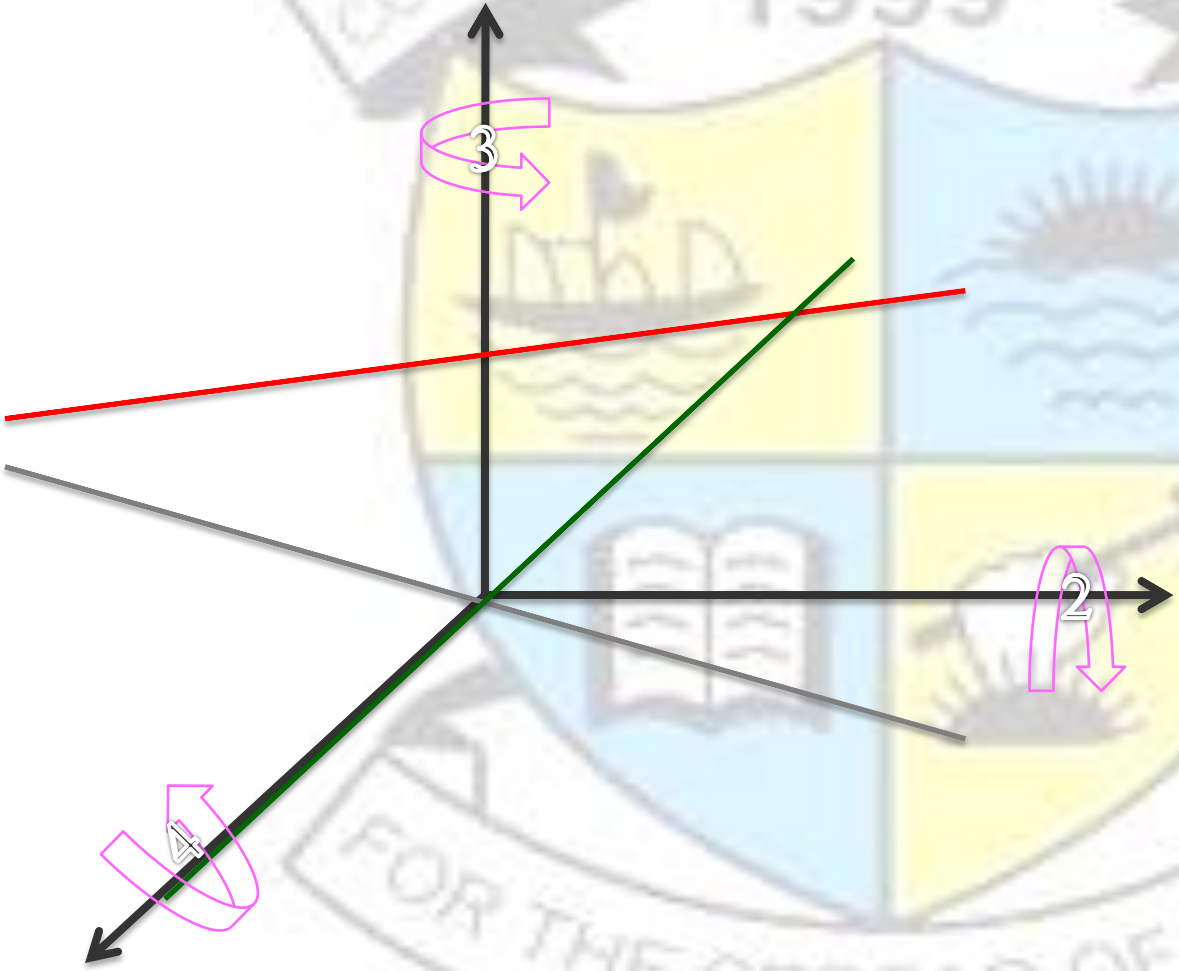
A: Apply the inverse of the rotation… $R^{-1}(\theta) = R(-\theta)$

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

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

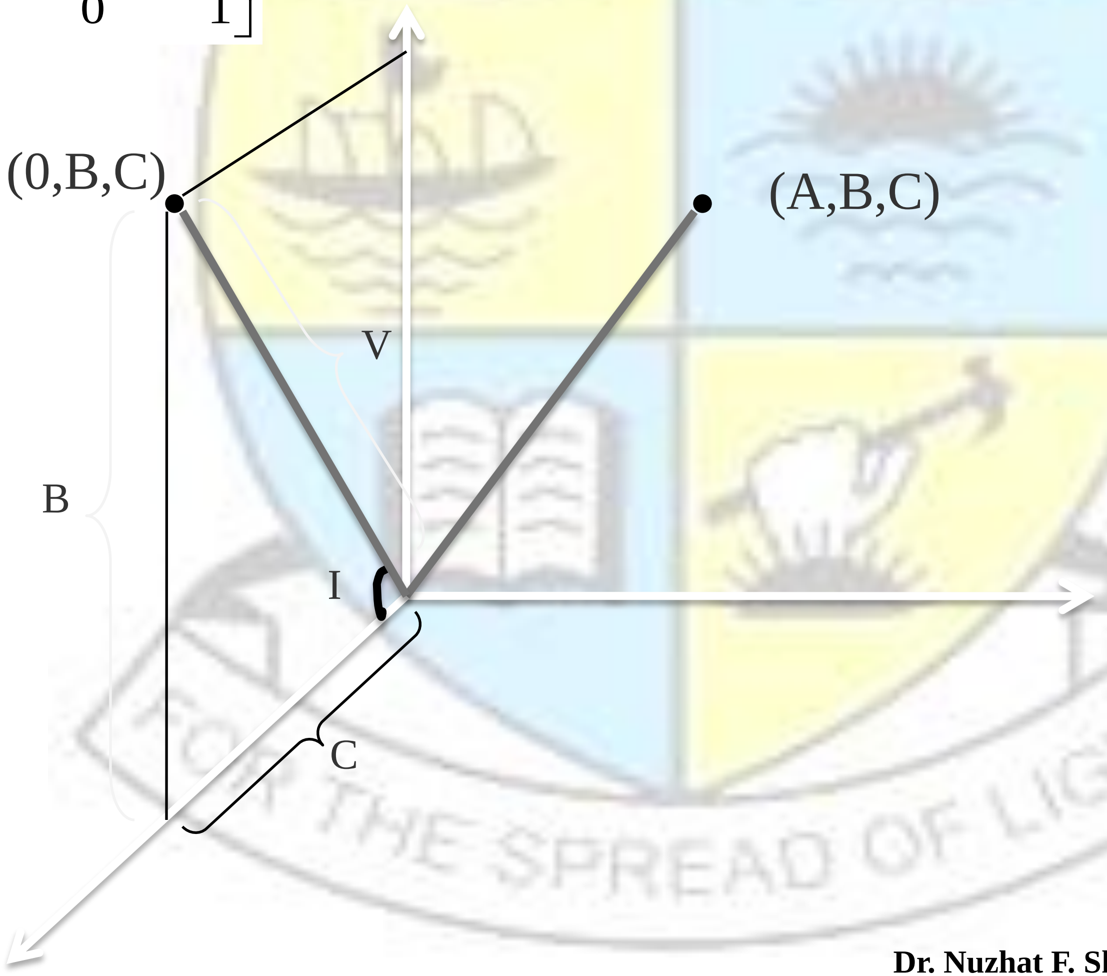
7.Inverse Translate

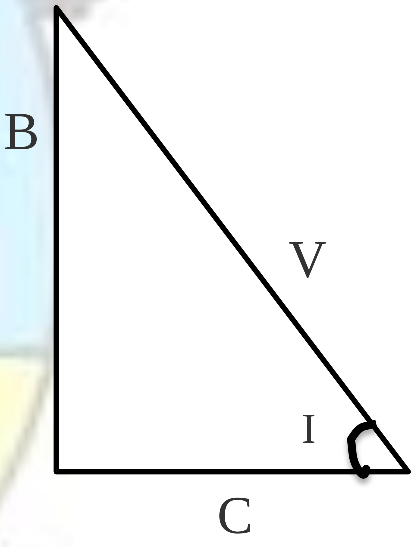
$$T = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.Rotate about x axis

6. Inverse Rotation about x axis

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





Length of shadow

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

$$\sin I = B / V$$

$$\cos I = C / V$$

3.Rotate about y axis

5. Inverse Rotation about y axis

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

Rotation about x axis has left the x coordinate unchanged.

The total length of the segment

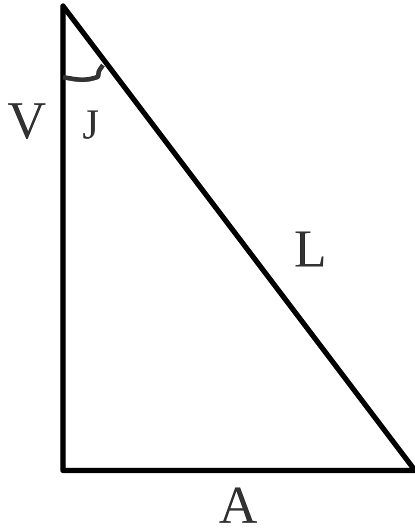
$$L = (A^2 + B^2 + C^2)^{1/2}$$

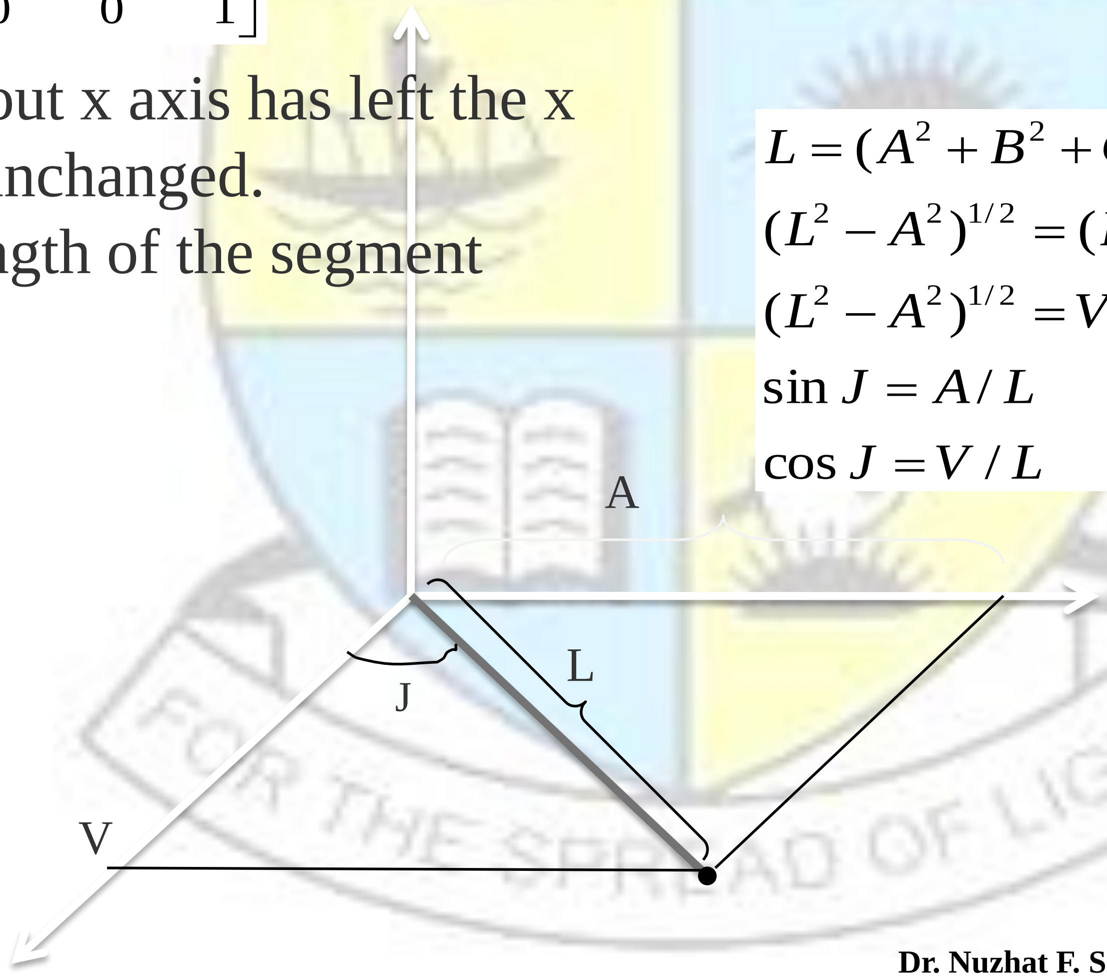
$$(L^2 - A^2)^{1/2} = (B^2 + C^2)^{1/2}$$

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

$$\sin J = A/L$$

$$\cos J = V/L$$





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

$T\ Rx\ Ry\ Rz\ Ry^{-1}\ Rx^{-1}\ T^{-1}$

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…

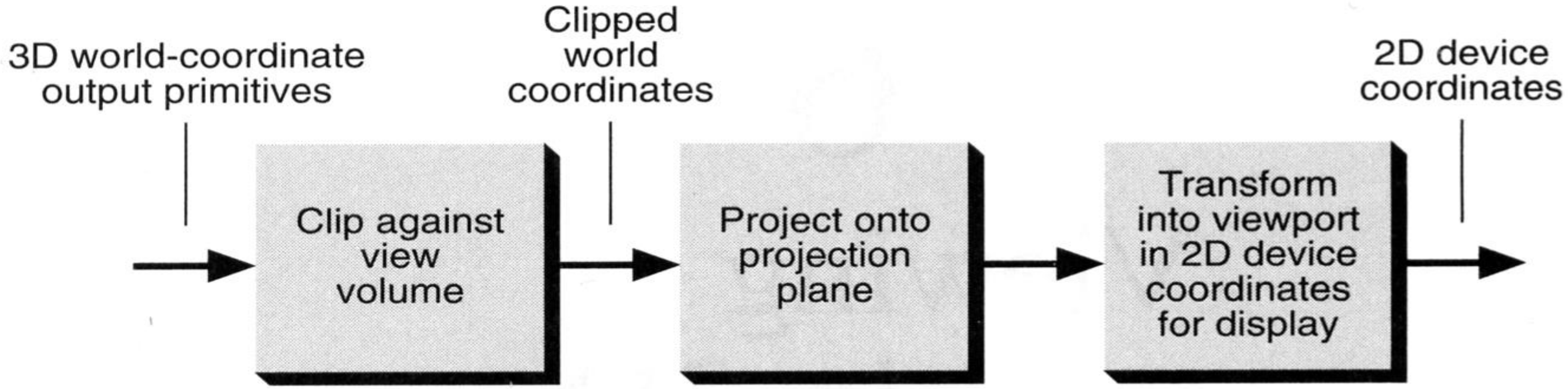
- View volume in the world
- Projection onto the 2D projection plane
- A viewport to the view surface

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

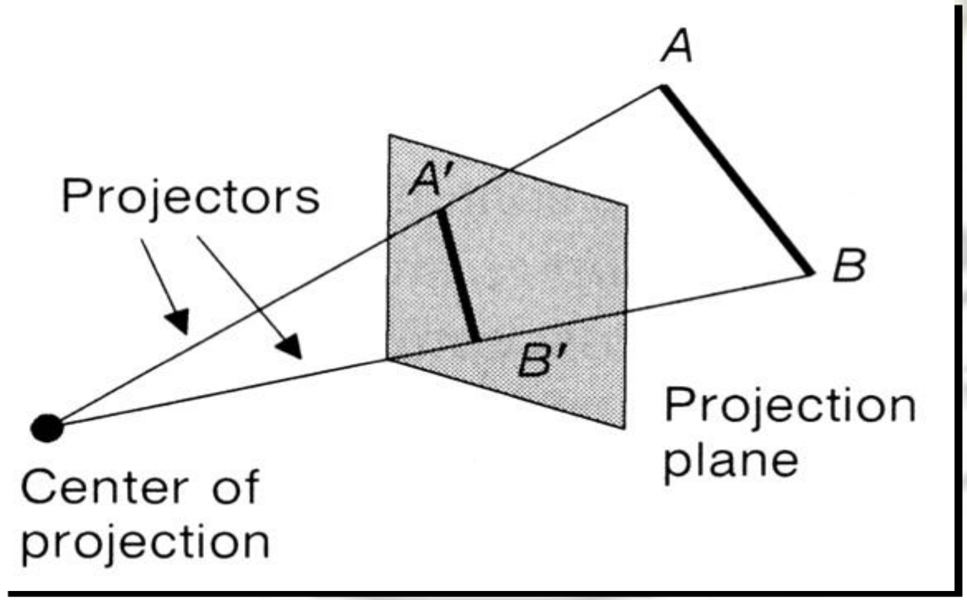


3D world-coordinate output primitives
Clipped world coordinates
2D device coordinates
Clip against view volume
Project onto projection plane
Transform into viewport in 2D device coordinates for display

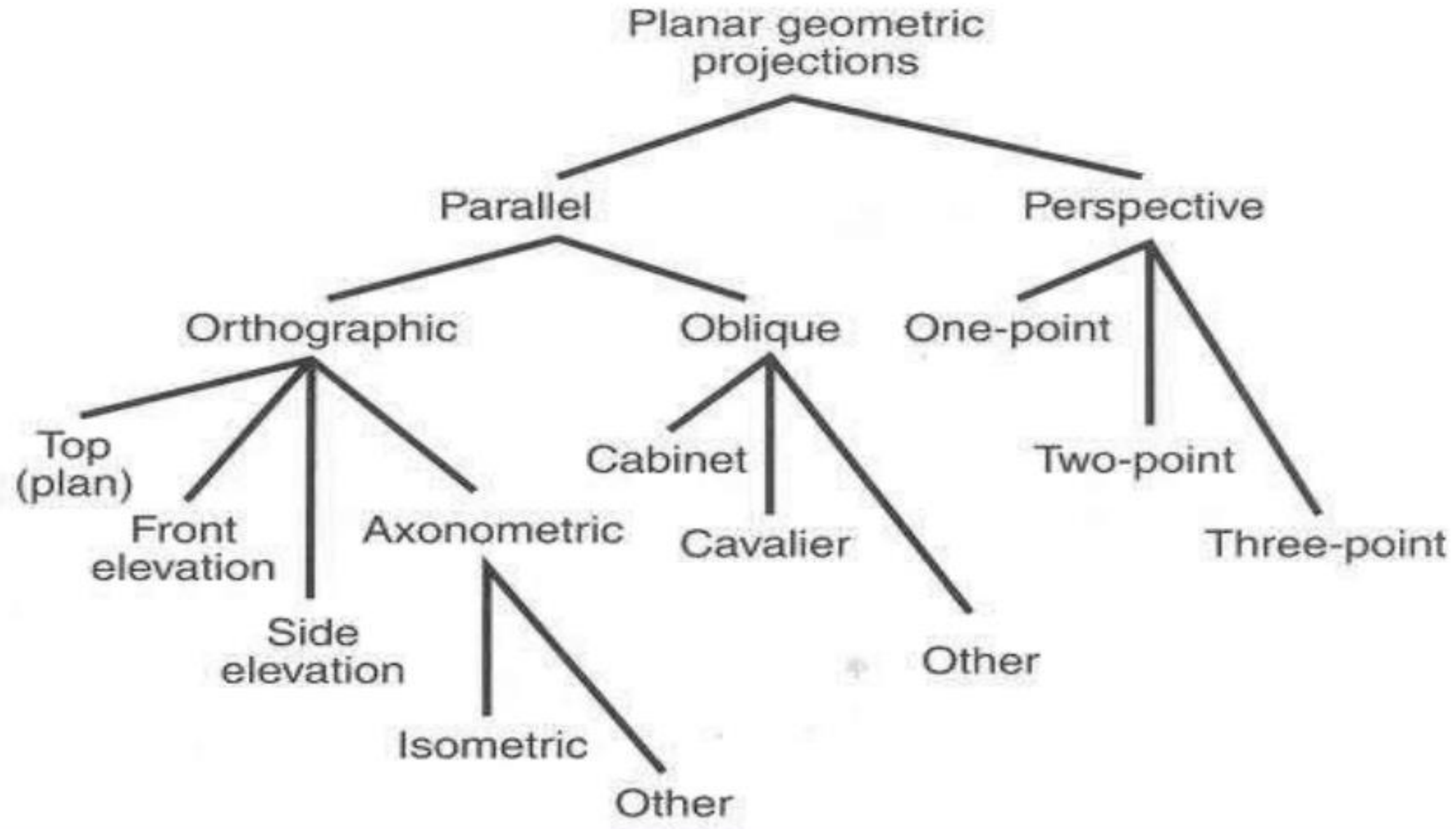
Projections

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



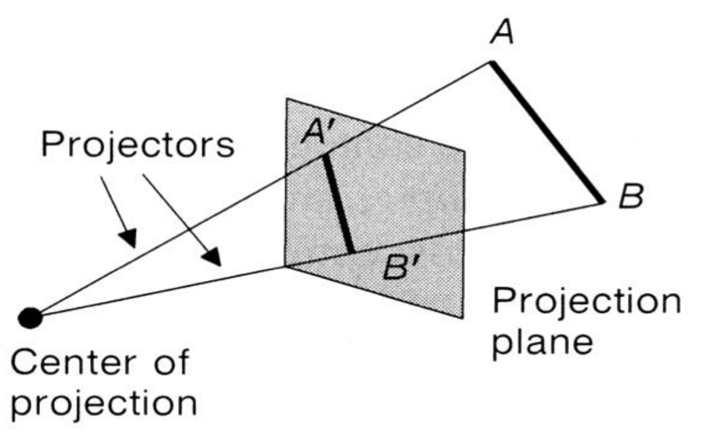
Types of projections

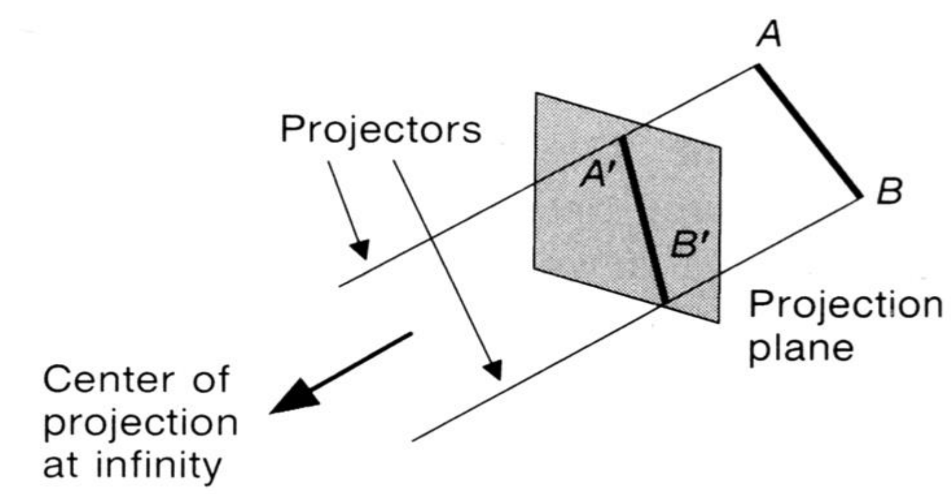
2 types of projections

- 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 != normal to the projection plane.

Parallel Projections

Orthographic (or orthogonal) projections:

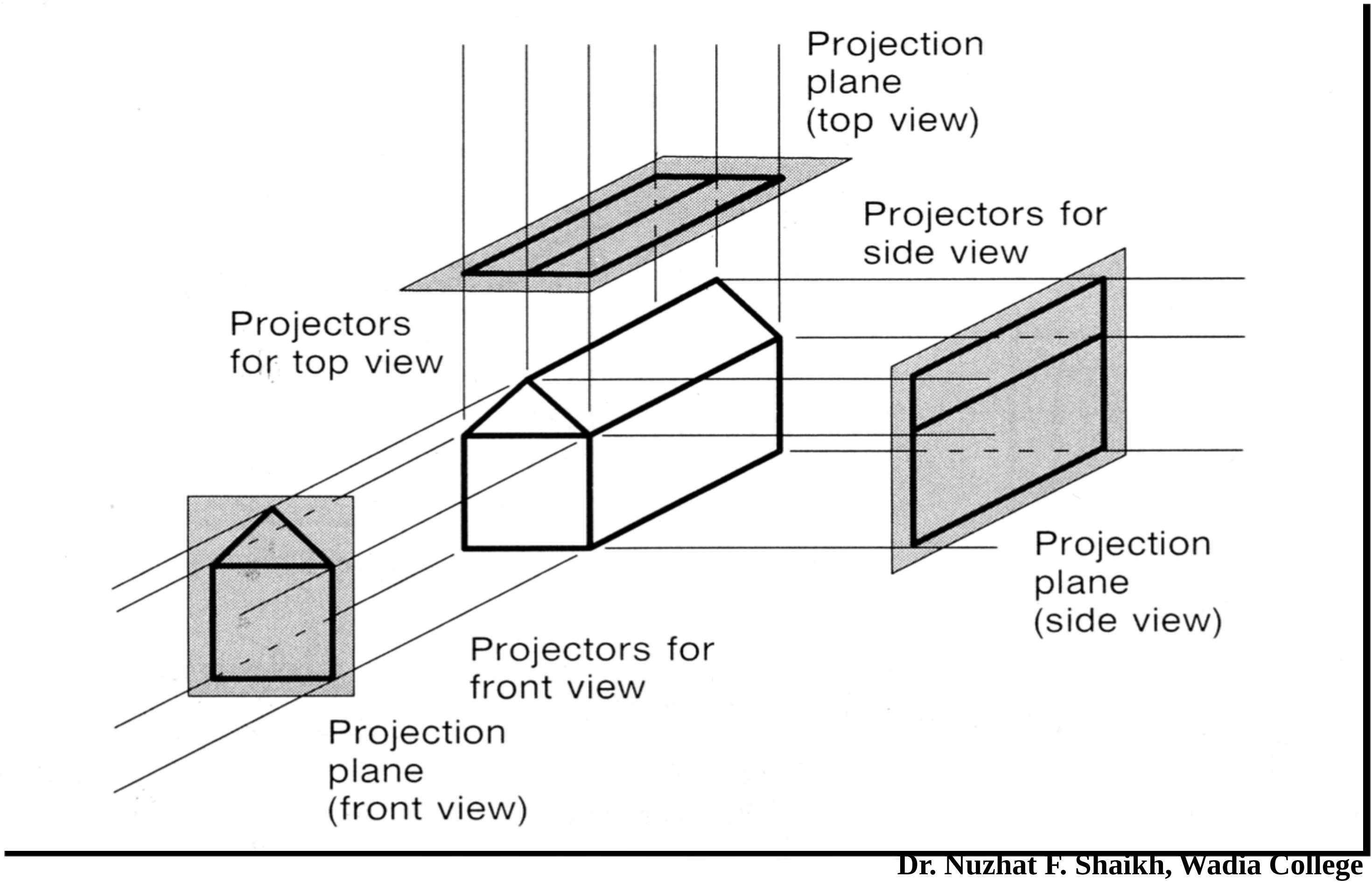
- **front elevation, top-elevation and side-elevation.**
- **all have projection plane perpendicular to a principle axes.**

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:



Projection plane (top view)
Projectors for side view
Projectors for top view
Projection plane (side view)
Projectors for front view
Projection plane (front view)

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:

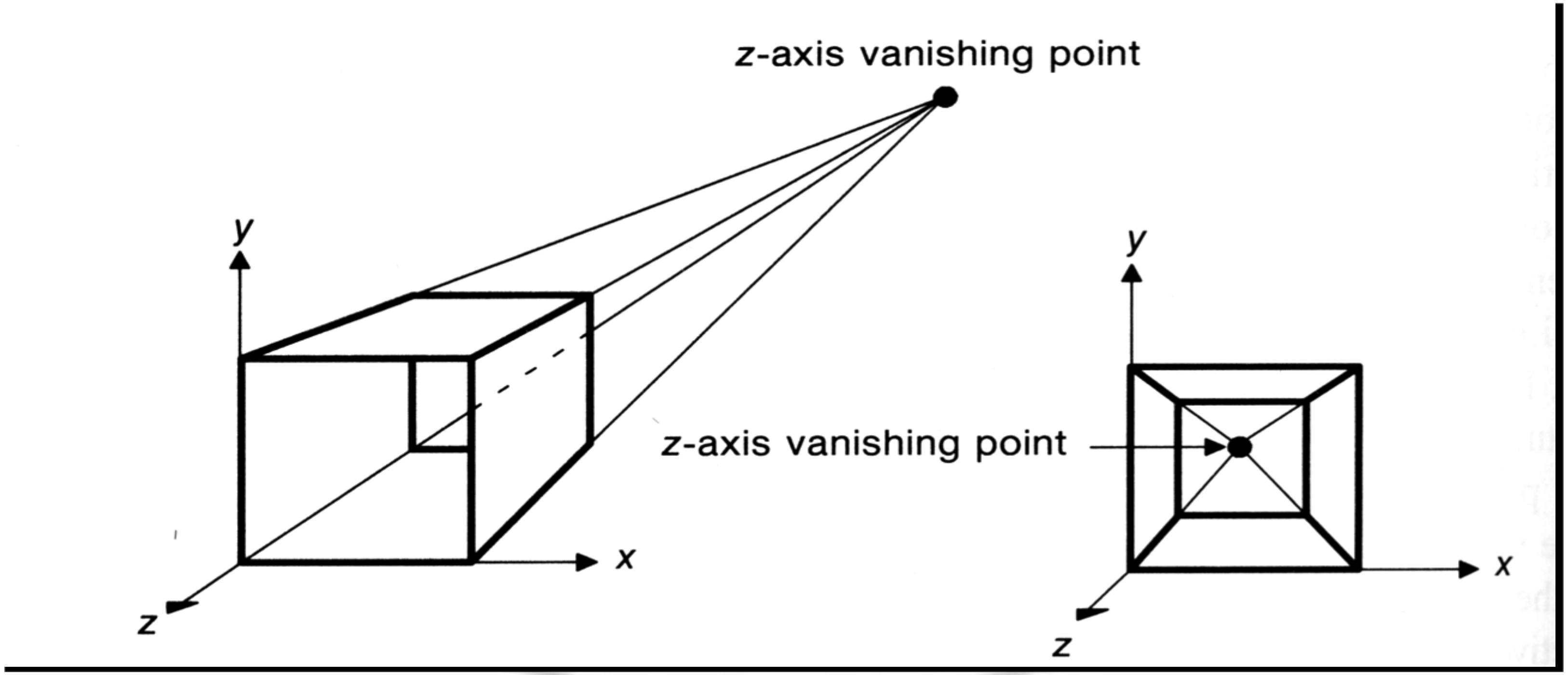
- **if z projection plane cuts the z axis: normal to it, so only z has a principle vanishing point, as x and y are parallel and have none.**

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)



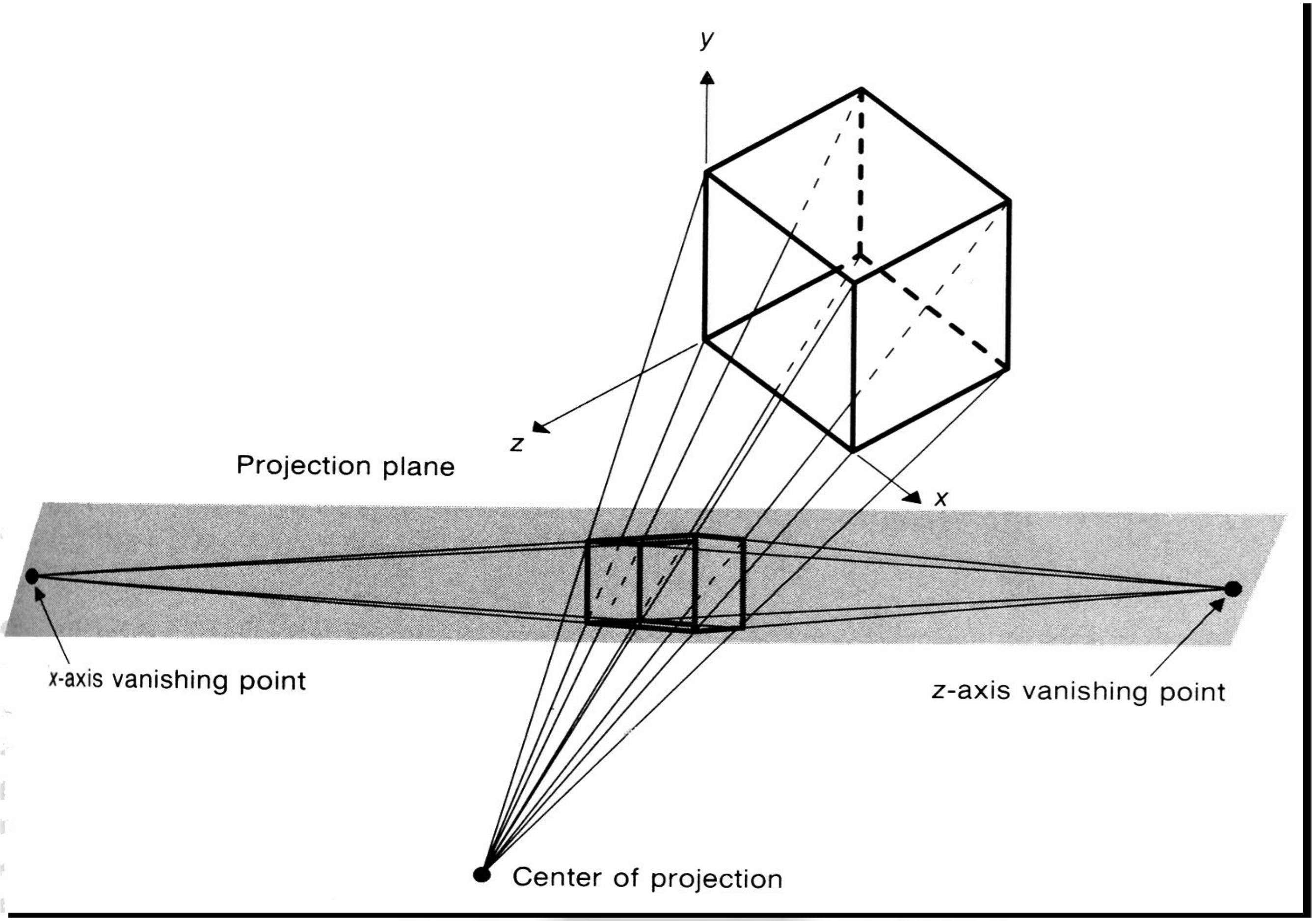
Perspective Projections

Two-point perspective projection:

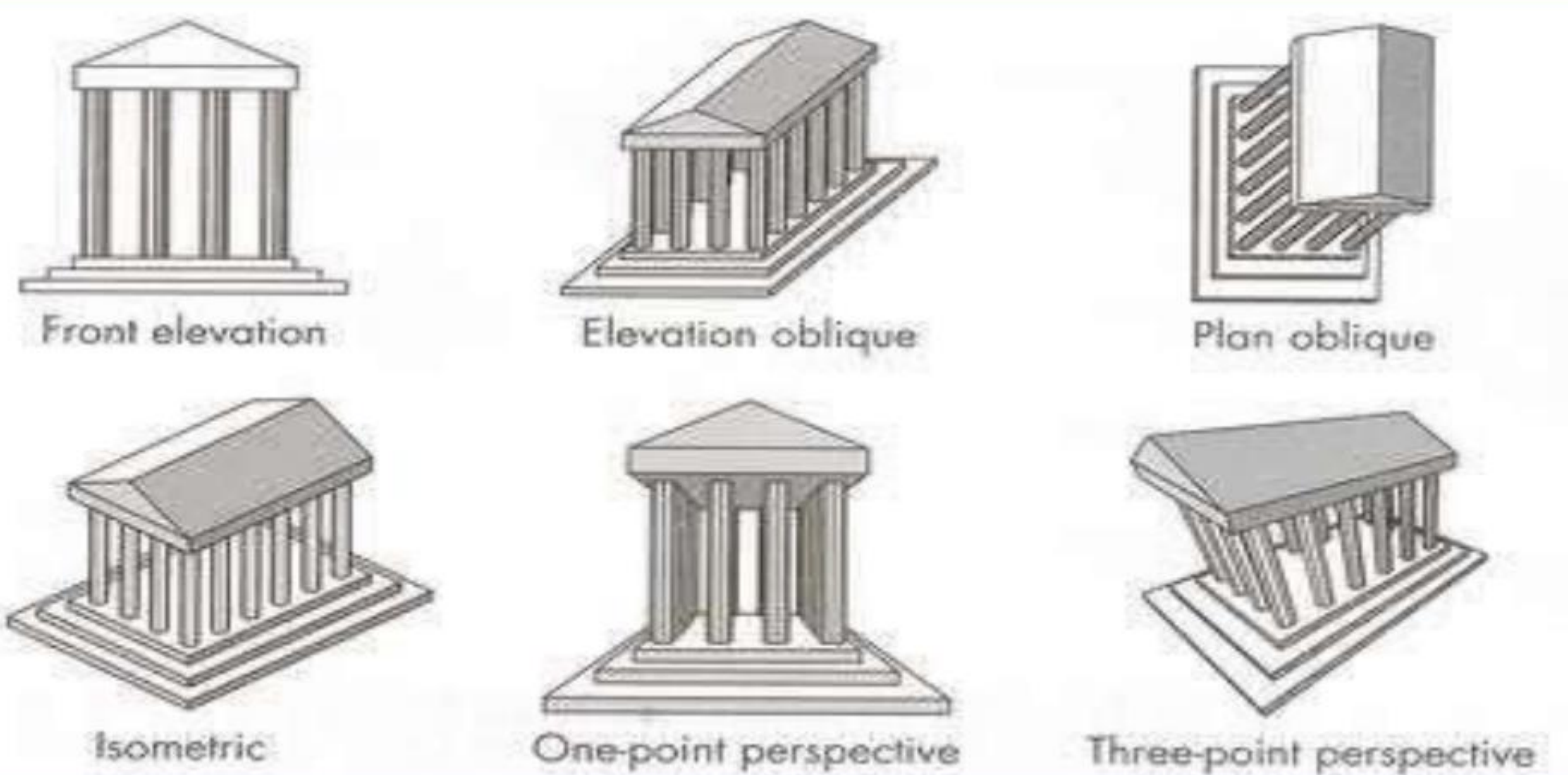
- This is often used in architectural, engineering and industrial design drawings.
- Three-point is used less frequently as it adds little extra realism to that offered by two-point perspective projection.

Perspective Projections

Two-point perspective projection:



Classical Projections



Front elevation

Elevation oblique

Plan oblique

Isometric

One-point perspective

Three-point perspective