

Computer Graphics

Lecture 4 *Transformations*

M.C. Escher – Smaller and Smaller
(1956)



Overview

2D Transformations

- Basic 2D transformations
- Matrix representation
- Matrix composition

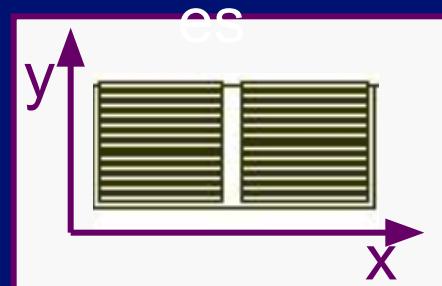
3D Transformations

- Basic 3D transformations
- Same as 2D



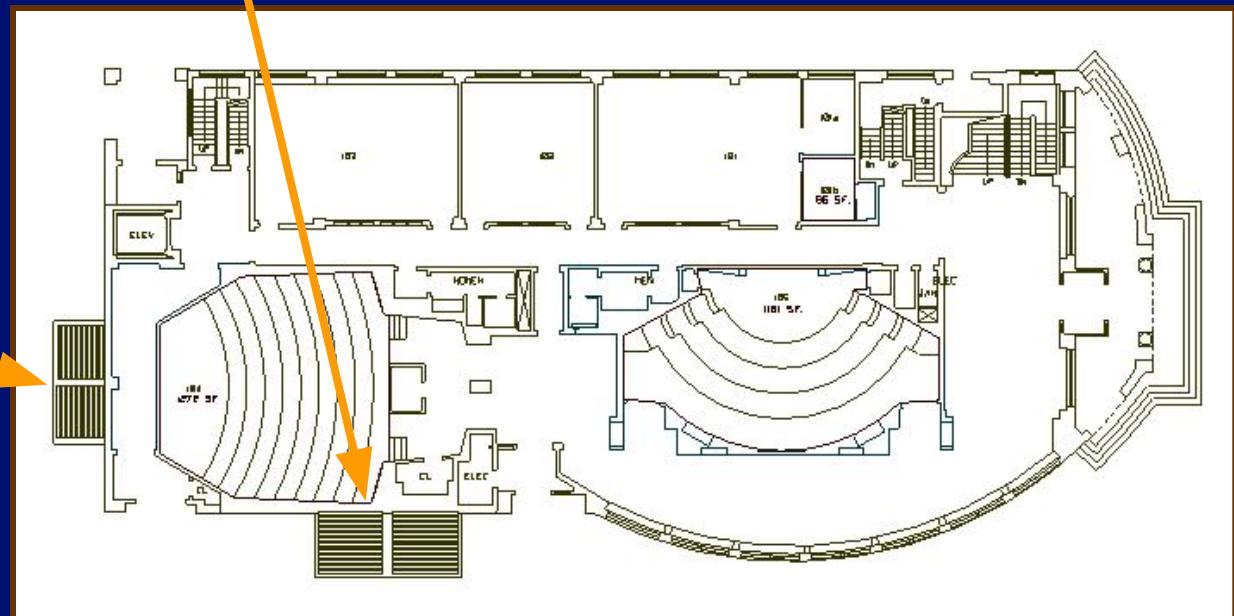
2D Modeling Transformations

Modeling
Coordinate



Scale
Translate

Scale
Rotate
Translate

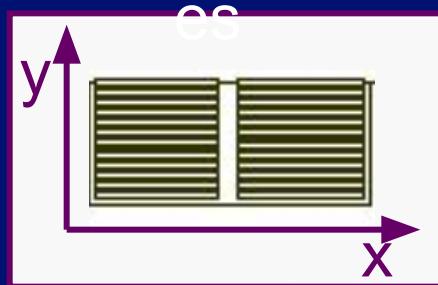


World
Coordinates

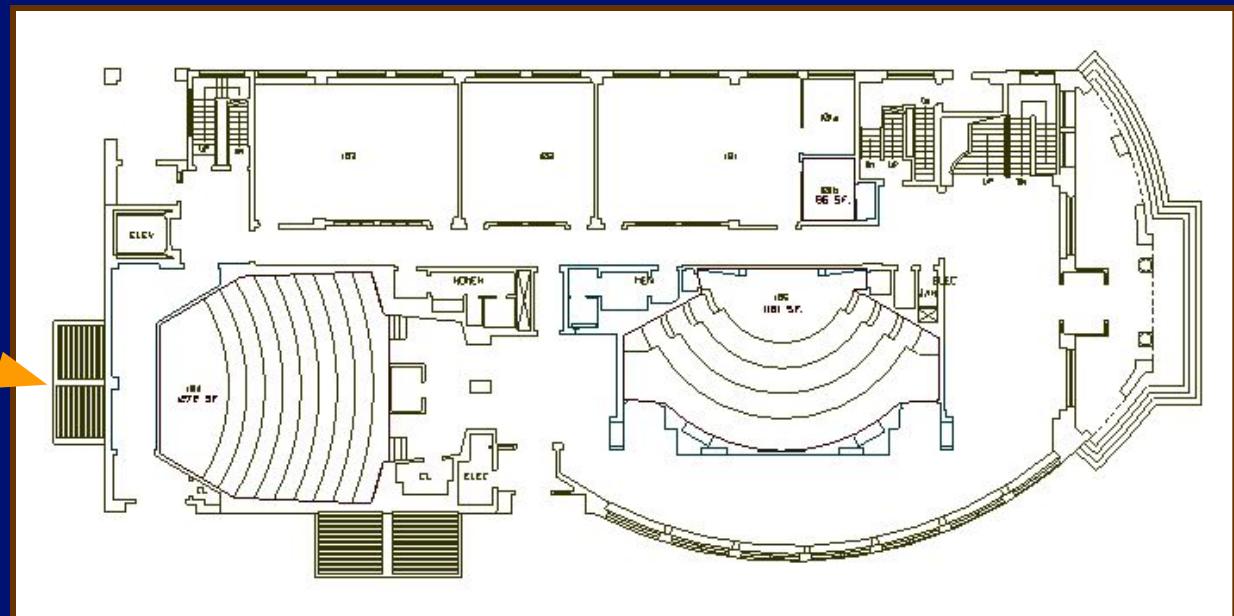


2D Modeling Transformations

Modeling
Coordinate



Let's look
at this in
detail...

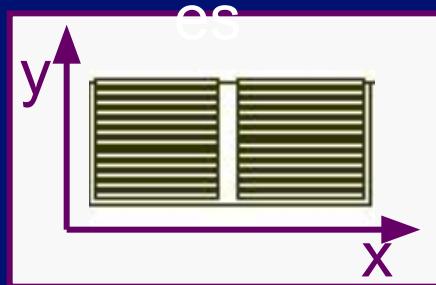


World
Coordinates

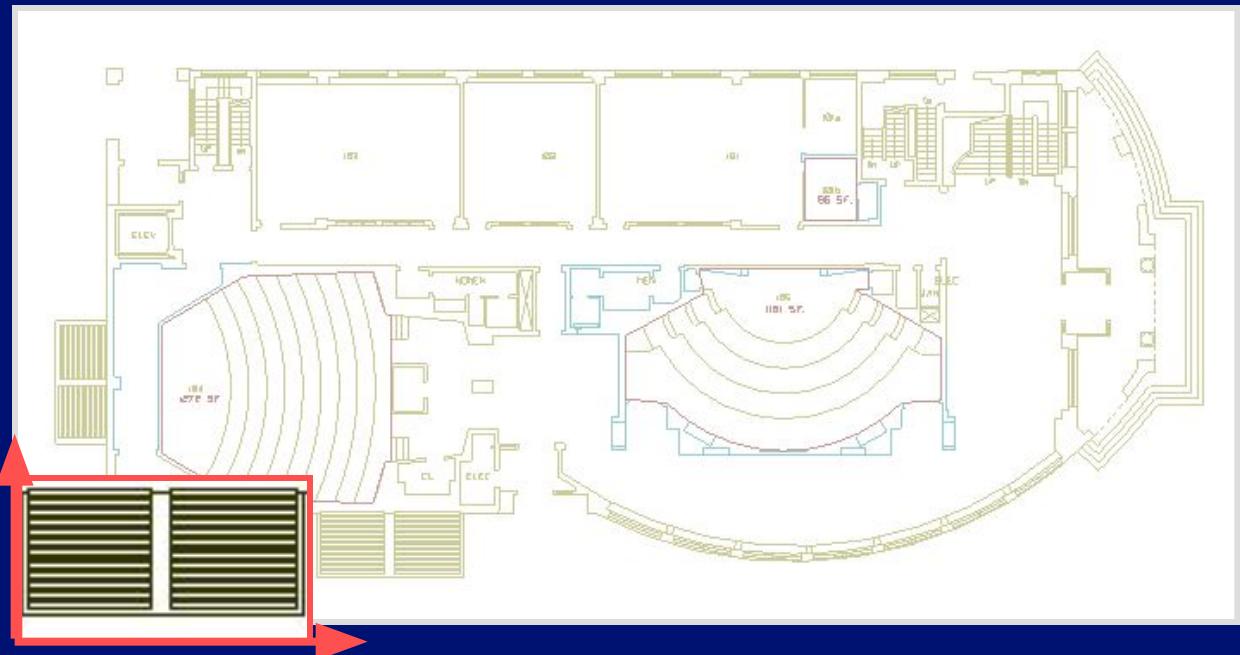


2D Modeling Transformations

Modeling Coordinate



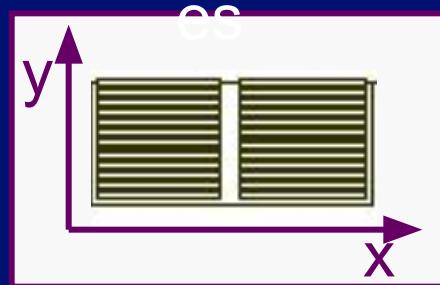
Initial
location
at $(0, 0)$
with
 x - and
 y -axes
aligned



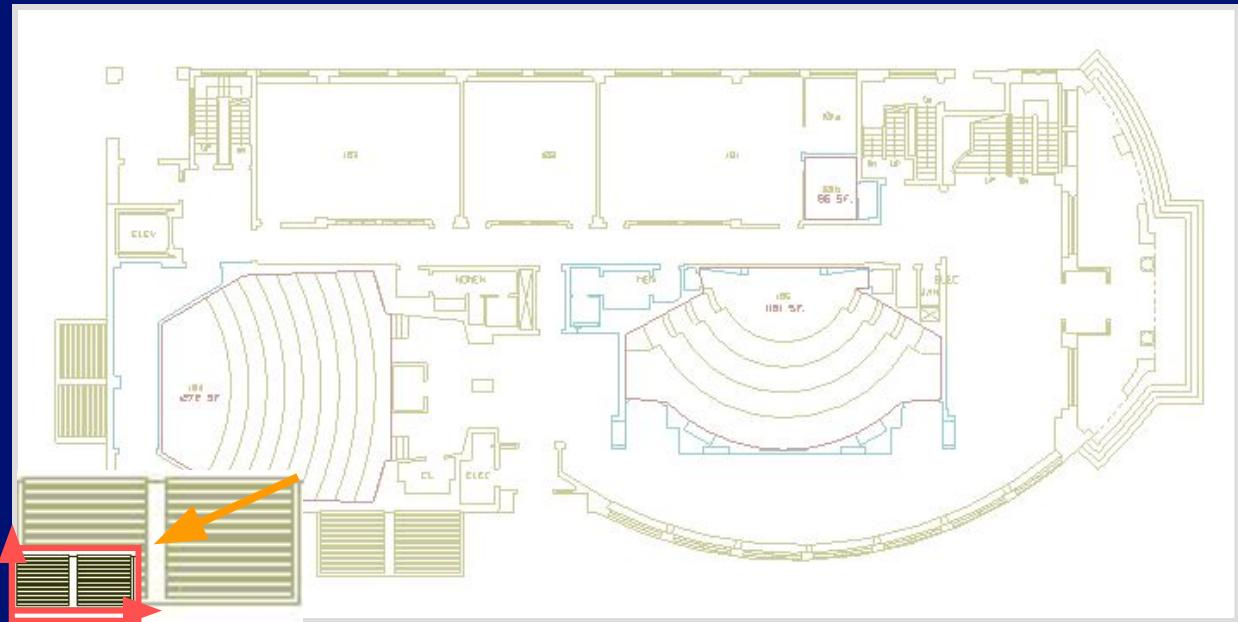


2D Modeling Transformations

Modeling Coordinate



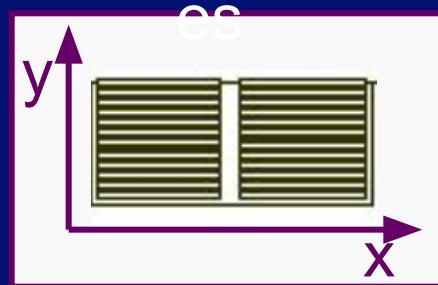
Scale .3, .3
Rotate -90
Translate 5,
3



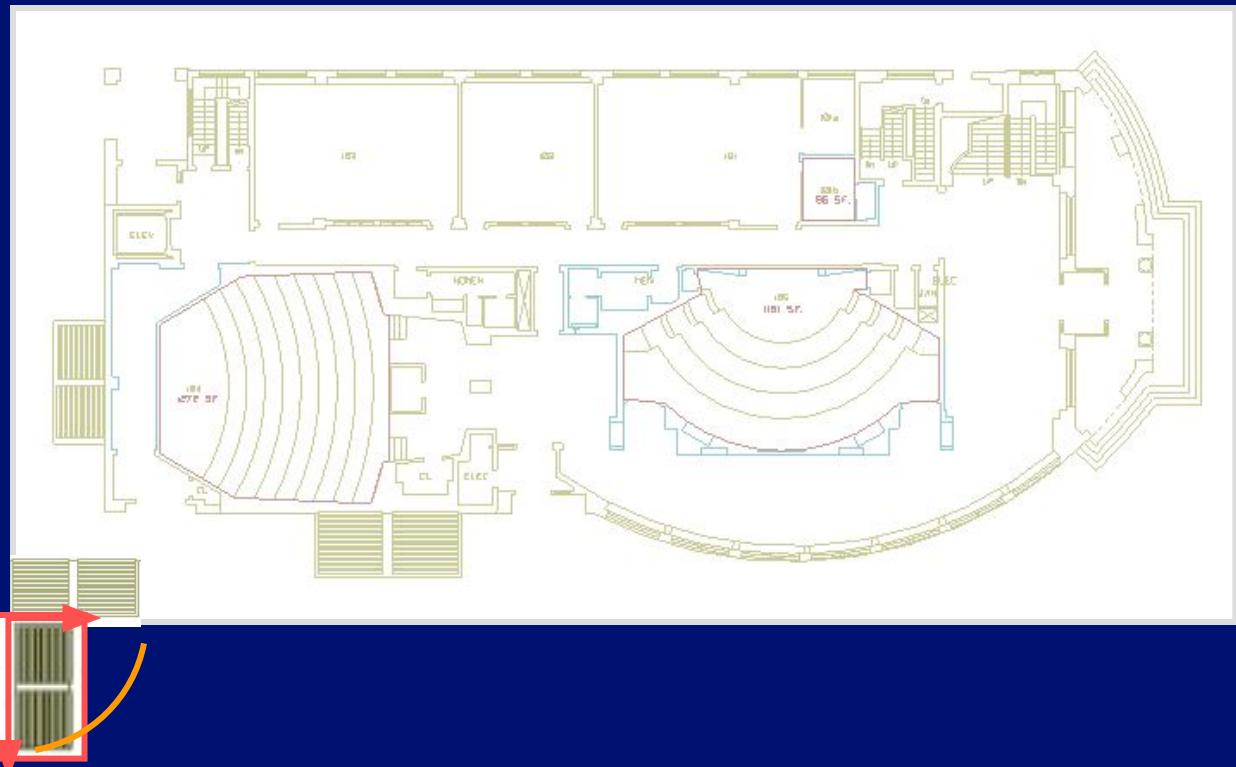


2D Modeling Transformations

Modeling Coordinate



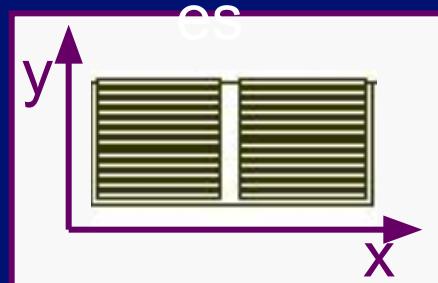
Scale .3, .3
Rotate -90
Translate 5, 3



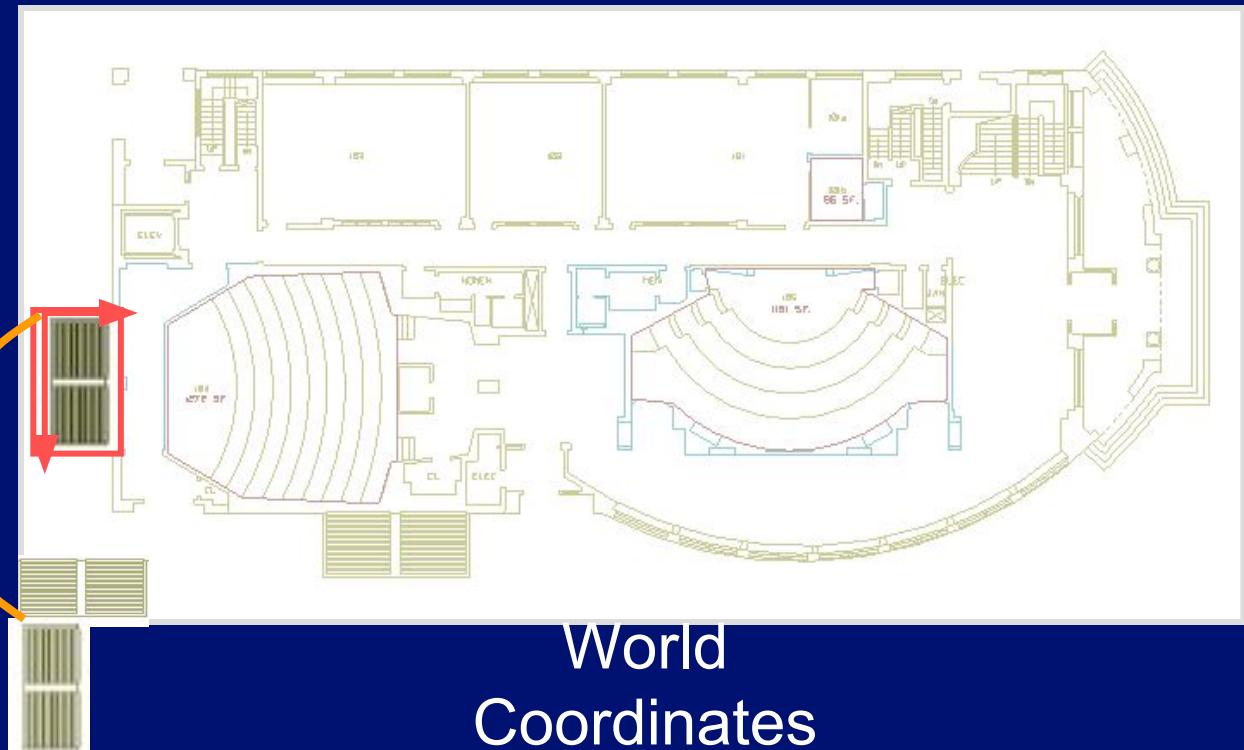


2D Modeling Transformations

Modeling Coordinate



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



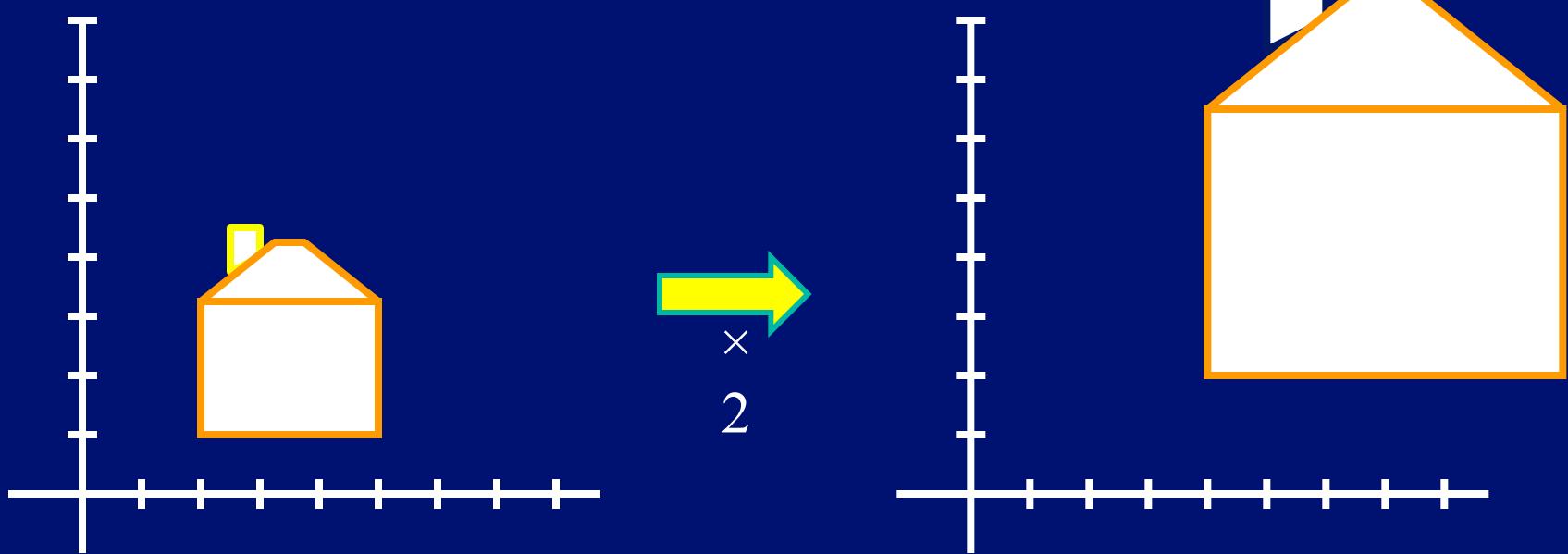
World
Coordinates



Scaling

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

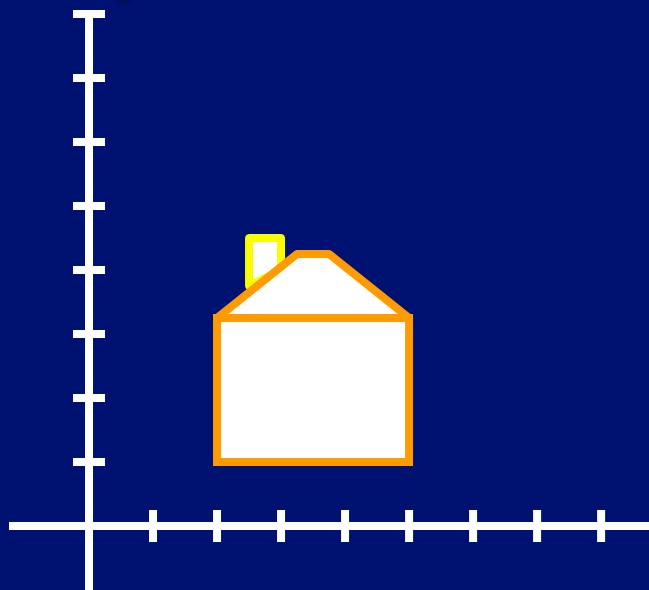
Uniform scaling means this scalar is the same for all components:



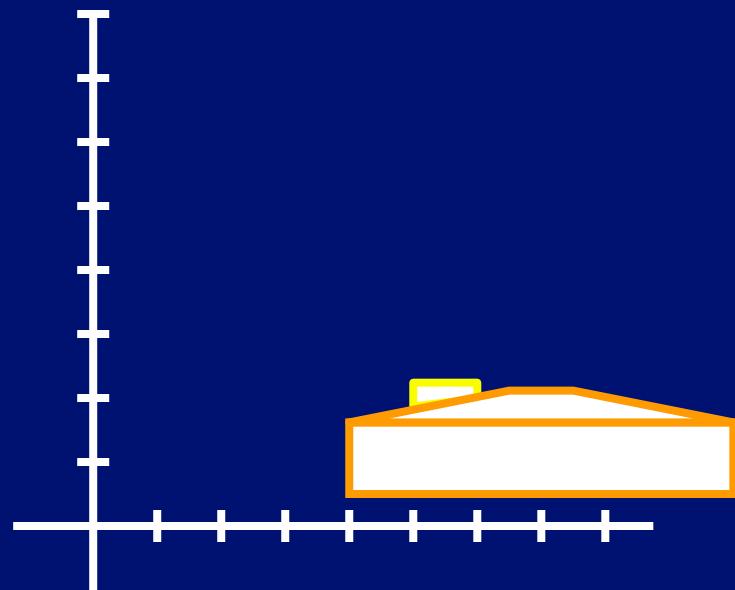


Scaling

Non-uniform scaling: *different scalars per component:*



$$\begin{array}{l} \text{X} \times 2, \\ \text{Y} \times \\ 0.5 \end{array}$$



How can we represent this in matrix form?



Scaling

Scaling operation:

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

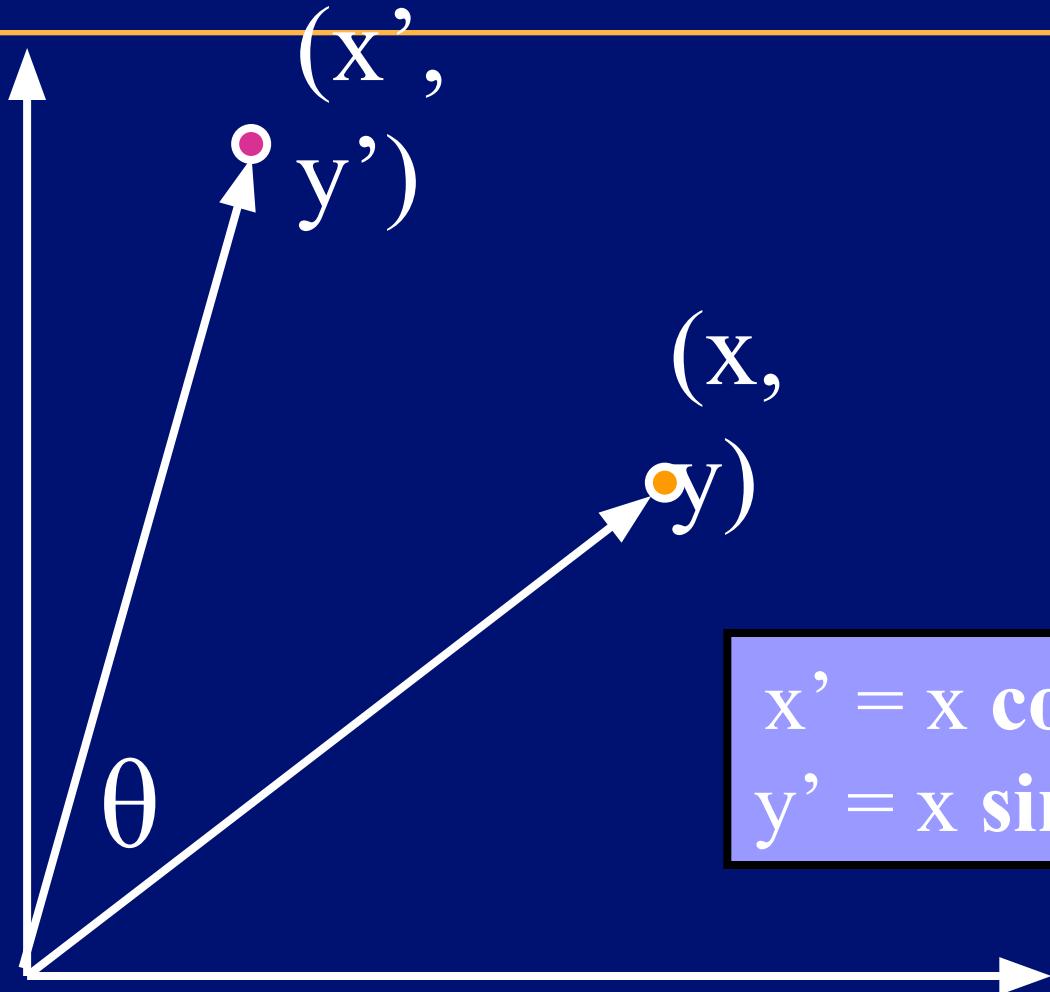
Or, in matrix form:

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

scaling
matrix



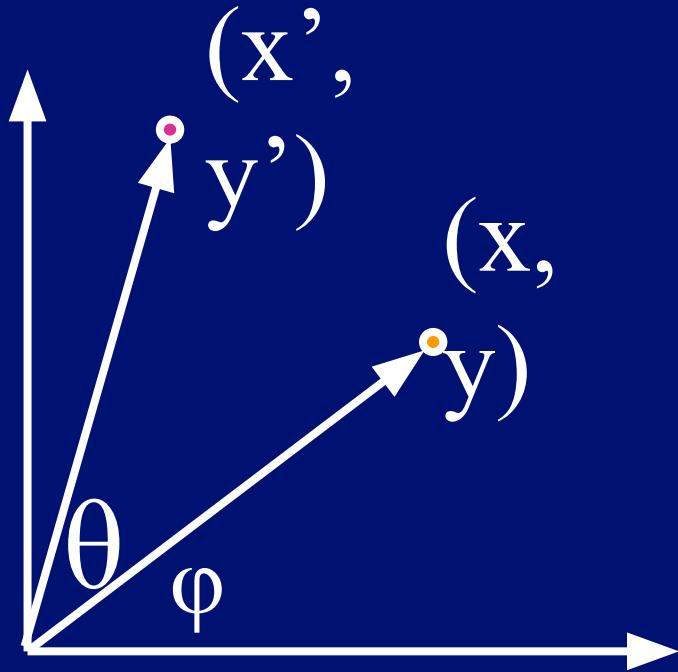
2-D Rotation



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



2-D Rotation



$$\left\{ \begin{array}{l} x = r \cos (\varphi) \\ y = r \sin (\varphi) \\ x' = r \cos (\varphi + \theta) \\ y' = r \sin (\varphi + \theta) \end{array} \right.$$

Trig Identity...

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

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 θ ,

- 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:

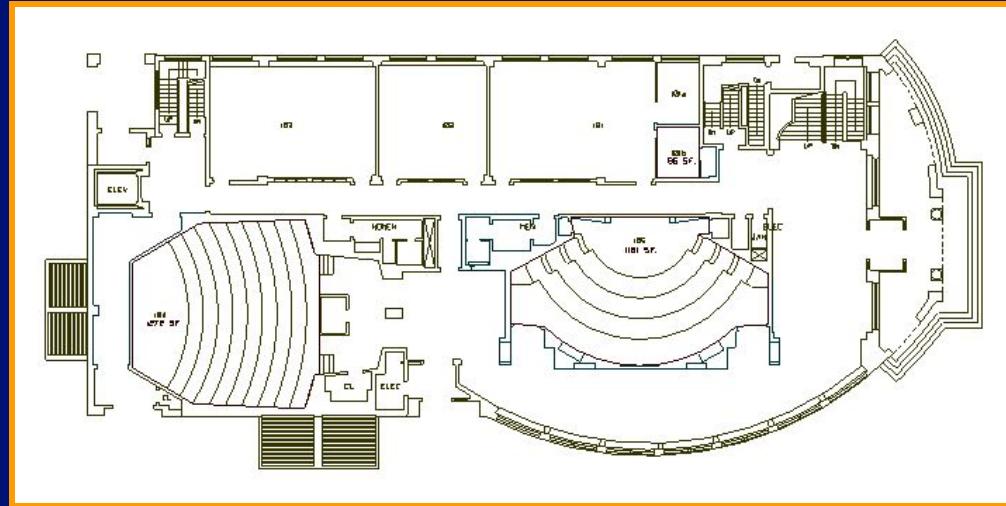
- $x' = x * s_x$
- $y' = y * s_y$

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:

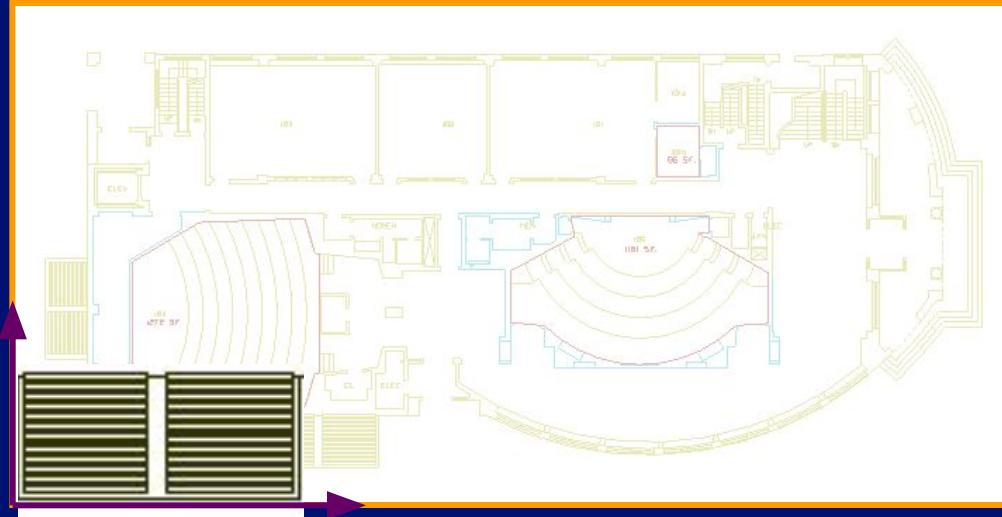
- $x' = x * s_x$
- $y' = y * s_y$

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:

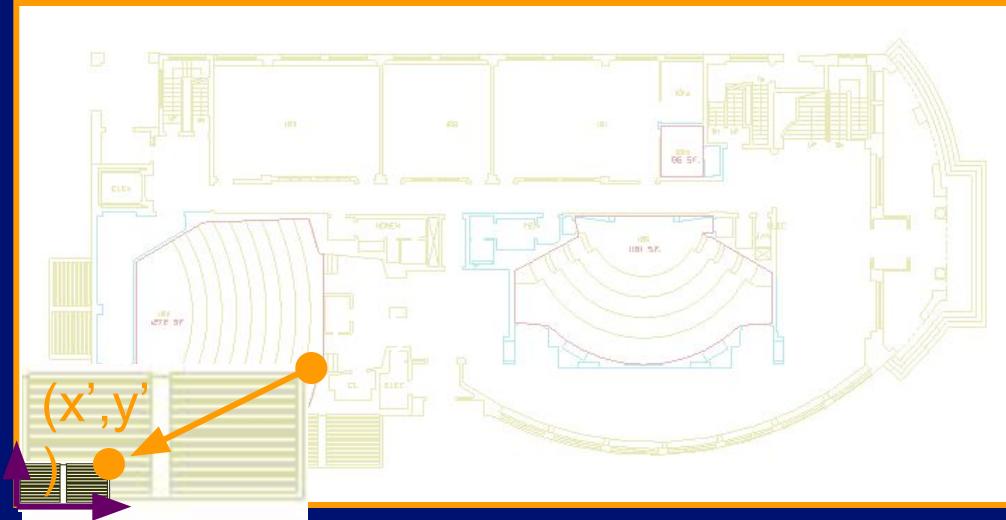
- $x' = x + t_x$
- $y' = y + t_y$

Scale:

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

Shear:

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



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

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:

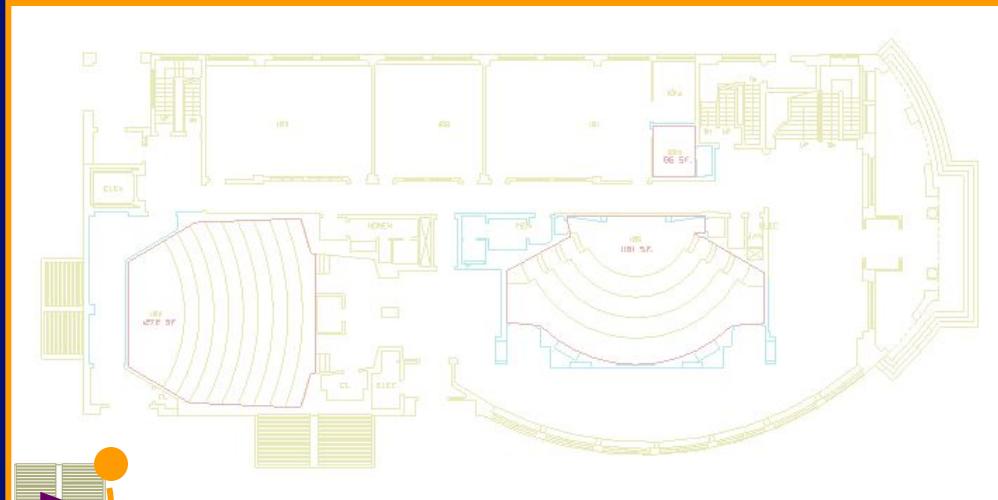
- $x' = x * s_x$
- $y' = y * s_y$

Shear:

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

Rotation:

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



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



Basic 2D Transformations

Translation:

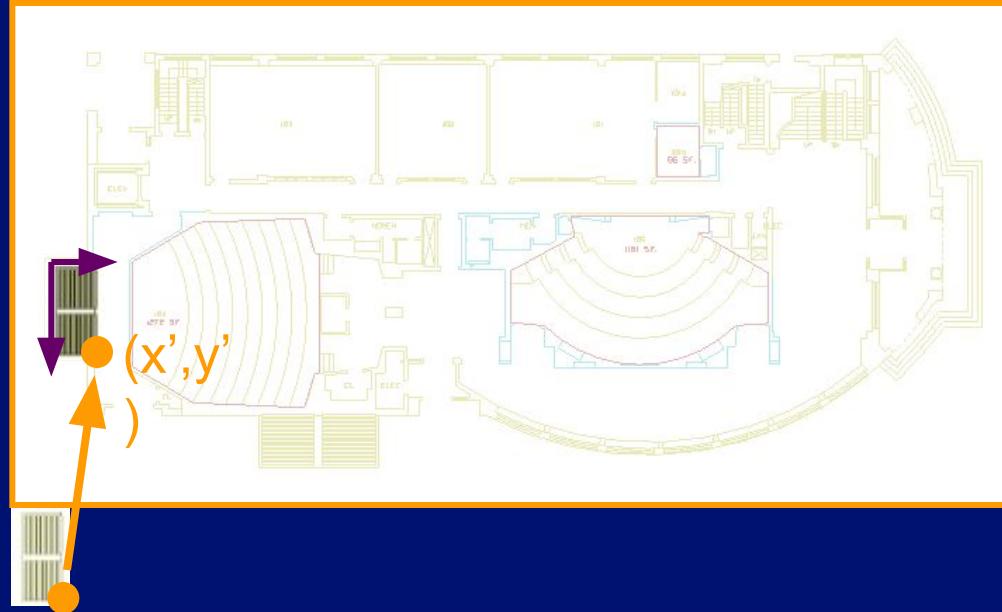
- $x' = x + t_x$
- $y' = y + t_y$

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:

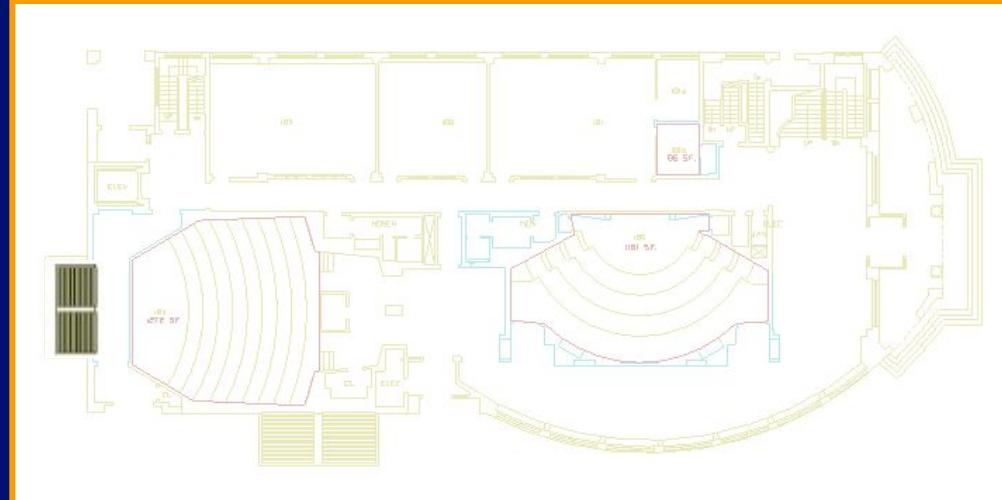
- $x' = x + t_x$
- $y' = y + t_y$

Scale:

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

Shear:

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



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

Rotation:

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



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 point*

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



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 types 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)? $s_x * x$

$$y' = s_y * y$$

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



2x2 Matrices

What types 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?

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

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



2x2 Matrices

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

2D Mirror about Y
axis?

$$x' = -x$$

$$y' = y$$

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

2D Mirror over
(0,0)?

$$x' = -x$$

$$y' = -y$$

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



2x2 Matrices

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

2D

Translation?

$$x' = x + t_x$$

NO

$$y' = y + t_y$$

!

Only linear 2D 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:

- Satisfies: $T(s_1\mathbf{p}_1 + s_2\mathbf{p}_2) = s_1T(\mathbf{p}_1) + s_2T(\mathbf{p}_2)$
- Origin maps to origin
- Lines map to lines
- Parallel lines remain parallel
- Ratios are preserved
- Associative but not commutative



Homogeneous Coordinates

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

$$x' = x + t_x$$

$$y' = y + t_y$$



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 3×3 matrix?

$$x' = x + t_x$$

$$y' = y + t_y$$

A: Using the rightmost column:

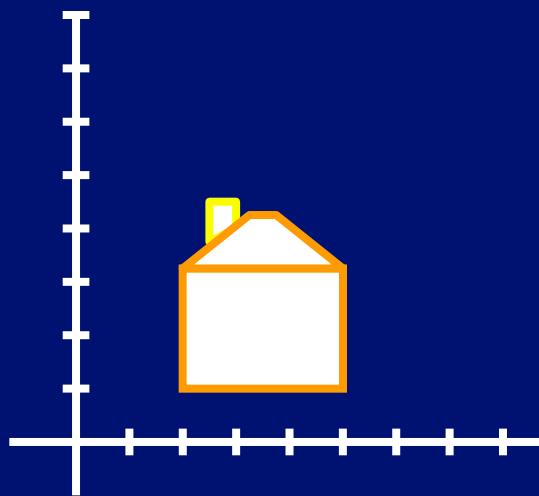
$$\text{Translation} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



Translation

Example of translation

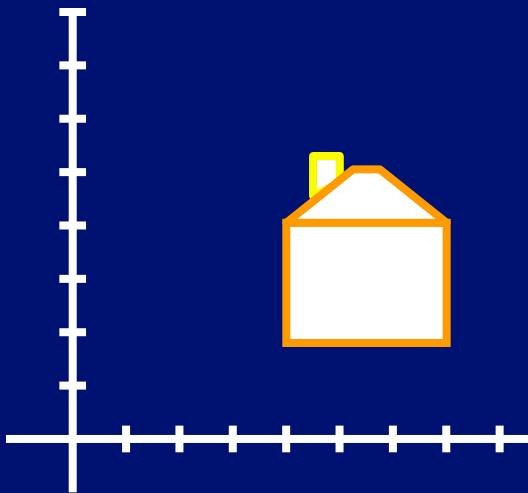
α



$$\begin{matrix} t_x \\ t_y \end{matrix} = \begin{matrix} 2 \\ 1 \end{matrix}$$

Homogeneous Coordinates

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

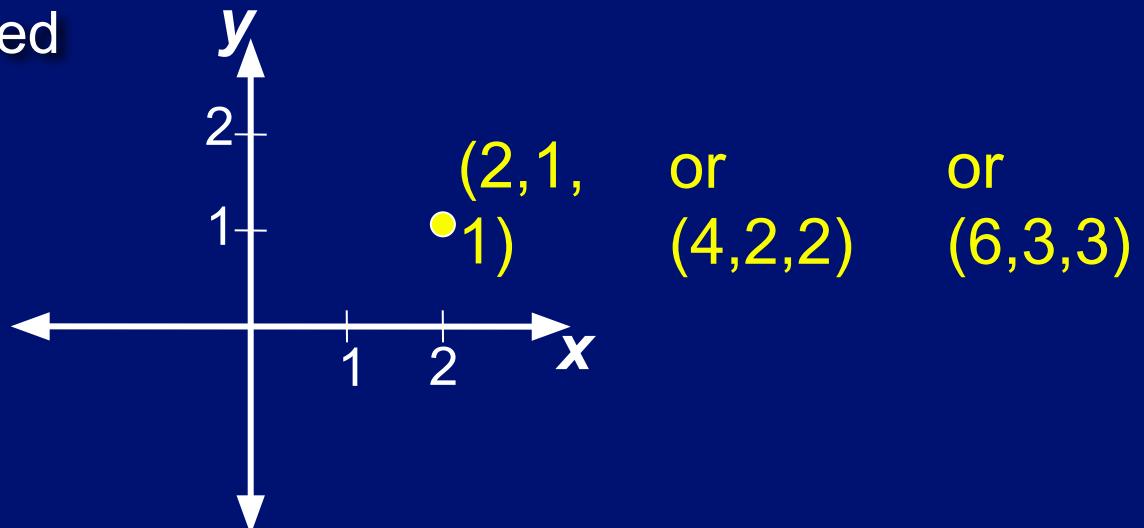




Homogeneous Coordinates

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

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

Translat
e

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

Scal
e

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

Rotat
e

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

Shea
r



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
- Associative but not commutative



Overview

2D Transformations

- Basic 2D transformations
- Matrix representation
- Matrix composition

3D Transformations

- Basic 3D transformations
- Same as 2D



Matrix Composition

Transformations can be combined by matrix multiplication

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

$$\begin{array}{lll} p' = T(t_x, t_y) & R(\Theta) & S(s_x, s_y) \\ p & & \end{array}$$



Matrix Composition

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

- General purpose representation
- Hardware matrix multiply

$$\begin{aligned} p' &= (T * (R * (S * p))) \\ p' &= (T * R * S) * p \end{aligned}$$



Matrix Composition

Be aware: order of transformations matters

- Matrix multiplication is not commutative

$$p' = T * R * S *$$

$p \longleftrightarrow$

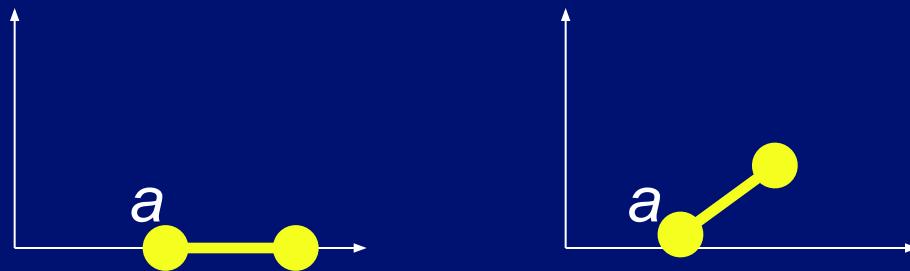
“Global”	“Local”



Matrix Composition

What if we want to rotate *and* translate?

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

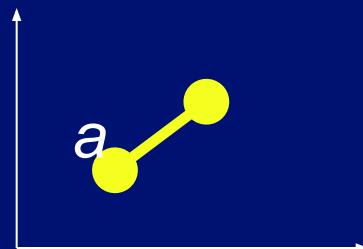
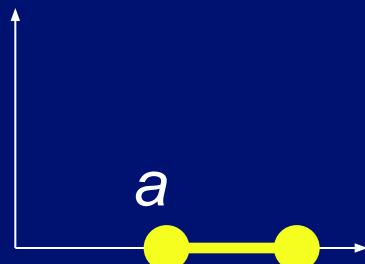


Multiplication Order – Wrong Way

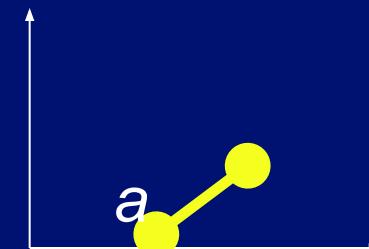


Our line is defined by two endpoints

- 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
 $R(45)$
)



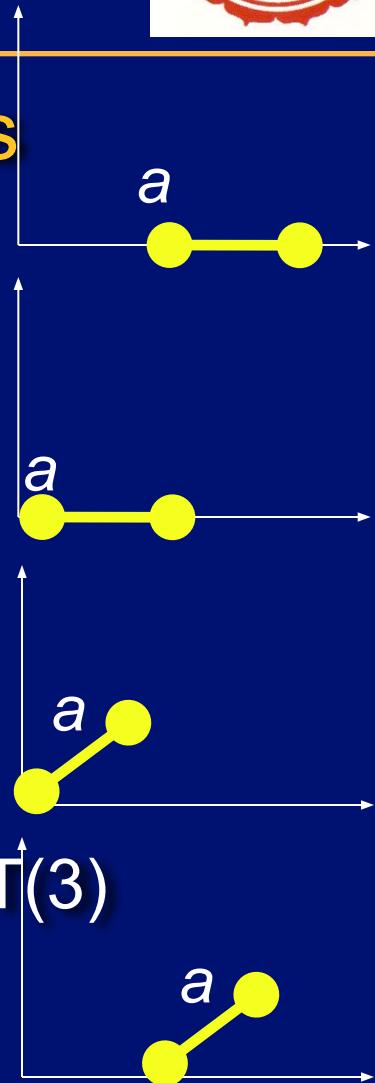
Correct
 $T(-3) R(45)$
 $T(3)$



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: $T(3)$





Matrix Composition

Will this sequence of operations work?

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

$$\begin{aligned} p' &= (T * (R * (S * p))) \\ p' &= (T * R * S) * p \end{aligned}$$



Matrix Composition

*After correctly ordering the matrices
Multiply matrices together*

What results is one matrix – store it (on stack)!

Multiply this matrix by the vector of each vertex

All vertices easily transformed with one matrix multiply



Reflection about an arbitrary line

Translate the line as well as the object so that the line passes through origin

Rotate the line and the object about the origin until the line is coincident with one of the coordinate axes

Reflect the object through the coordinate axis

Rotate back

Translate back



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: $\sin(-\theta) = -\sin(\theta)$

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



Overview

2D Transformations

- Basic 2D transformations
- Matrix representation
- Matrix composition

3D Transformations

- Basic 3D transformations
- Same as 2D



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
Matrix

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

Scaling
Matrix

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

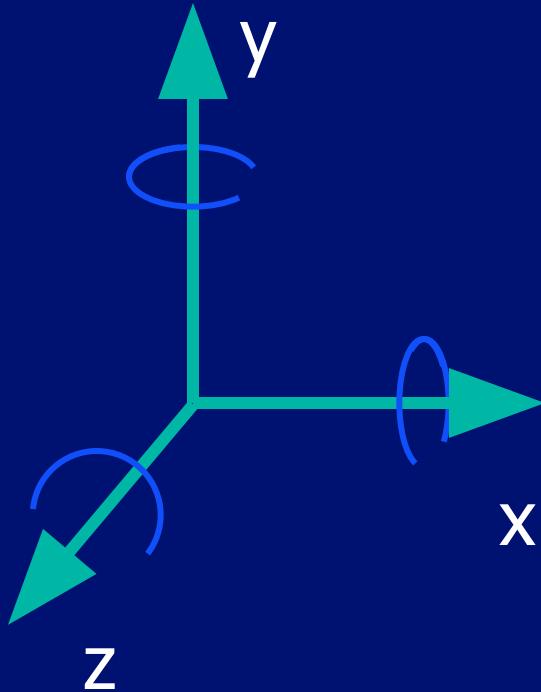
Translation
Matrix

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

Mirror about Y/Z
plane



3-D Rotation About an Axis



Positive rotation is counterclockwise, when looking from positive direction along an axis.



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:

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



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