Dr. George Karraz, Ph. D.

Geometric transformations

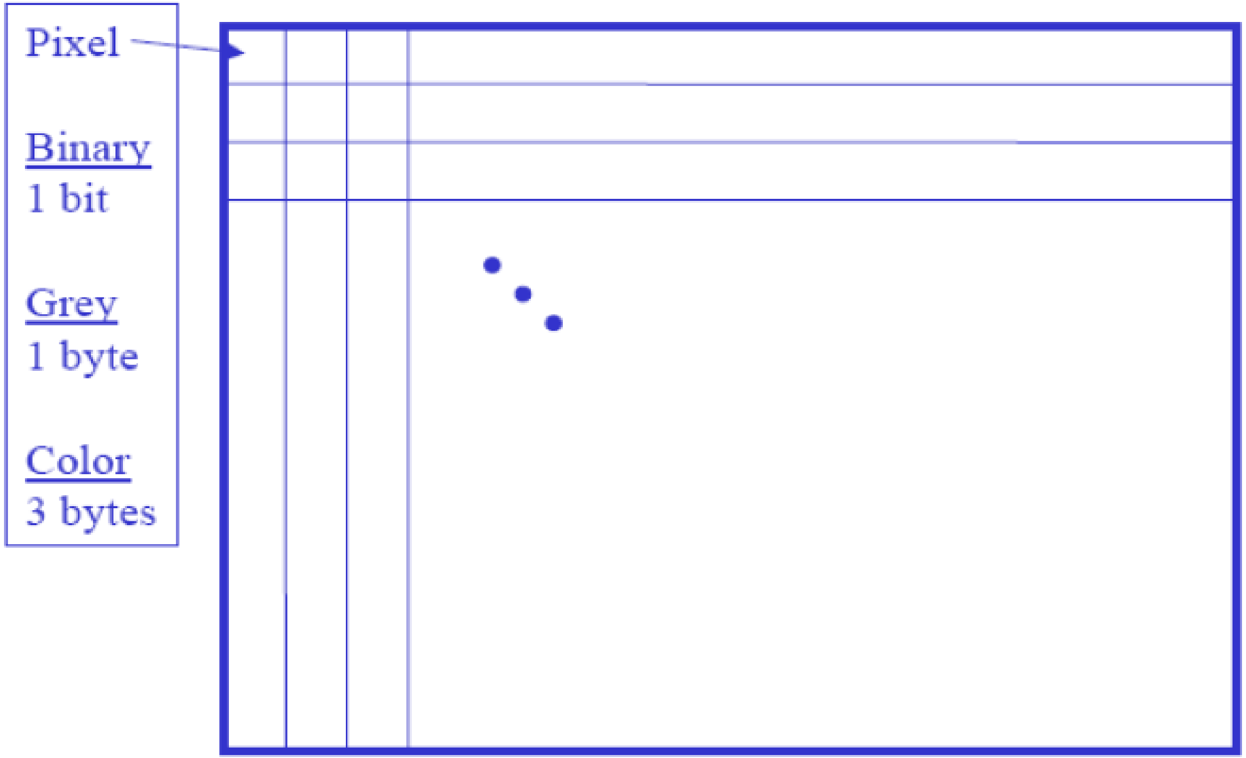
Review some basics of linear algebra and geometric transformations

Outline

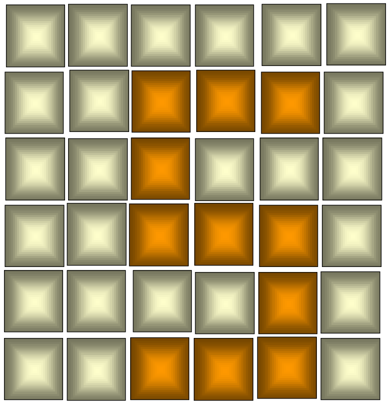
- Representation
- Basics of linear algebra
- Homogeneous Coordinates
- Geometrical transformations

Representation

- Digital Pictures are 2D arrays (matrices) of numbers
- Each pixel is a measure of the brightness (intensity of light)
  - that falls on an area of an sensor (typically a CCD chip)

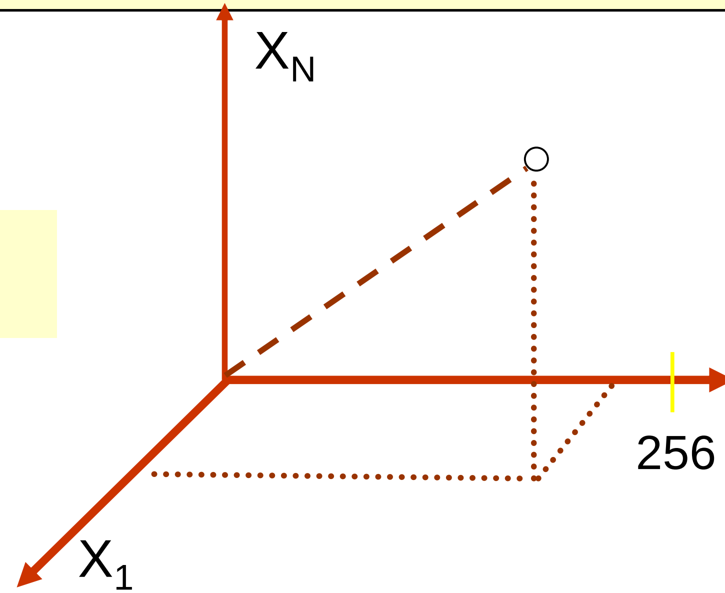


Picture as a vector in Dimension N



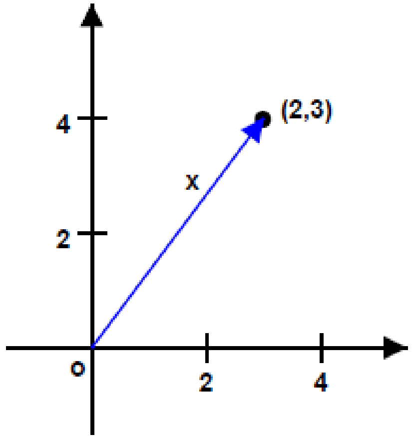
Appearance

Vector of dimension N



Vectors in $\mathbb{R}^n$

- We can think of vectors as points in a multidimensional space with respect to some coordinate system
- Ordered set of numbers
- Example in two dimensions $(\mathbb{R}^2)$:



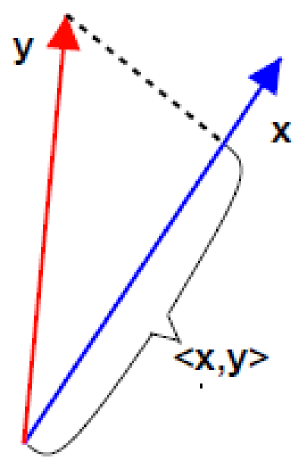
Vectors in $\mathbb{R}^n$

- Notation:

$$\mathbf{x} \in \mathbb{R}^n, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathbf{x} = (x_1, x_2, \ldots, x_n)^\mathsf{T}$$

Scalar Product

- A product of two vectors
- Amounts to projection of one vector onto the other
- Example in 2D:



The shown segment has length <x, y>, if x and y are unit vectors.

Scalar Product

- Various notations:
  - $\langle \mathbf{x}, \mathbf{y} \rangle$
  - $\mathbf{x}^T \mathbf{y}$
  - $\mathbf{x}\ \mathbf{y}$ or $\mathbf{x} \cdot \mathbf{y}$
- Other names: dot product, inner product

Scalar Product in $\mathbb{R}^n$

- Definition:

$$\mathbf{x}, \mathbf{y} \in \mathbb{R}^n : \ \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i \cdot y_i$$

- In terms of angles:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \cdot \|\mathbf{y}\| \cdot \cos(\angle(\mathbf{x}, \mathbf{y}))$$

- Other properties: commutative, associative, distributive

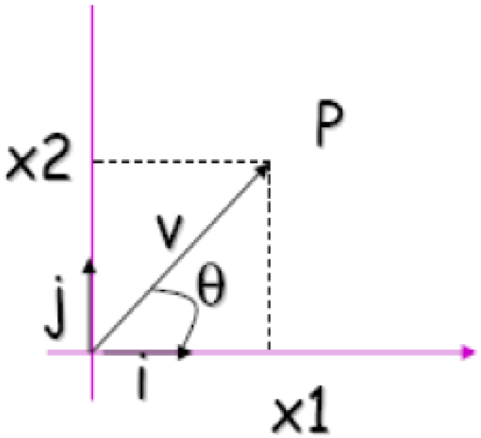
Basis

- A basis is a linearly independent set of vectors that spans the “whole space”. I. e., we can write every vector in our space as linear combination of vectors in that set.
- Every set of n linearly independent vectors in $\mathbb{R}^n$ is a basis of $\mathbb{R}^n$
- *Orthogonality*: Two non-zero vectors x and y are orthogonal if x.y = 0

- A basis is called
  - *orthogonal*, if every basis vector is orthogonal to all other basis vectors
  - *orthonormal*, if additionally all basis vectors have length 1.

Bases

- Orthonormal basis:



$$\mathbf{i} = (1,0) \qquad \|\mathbf{i}\| = 1$$

$$\mathbf{j} = (0,1) \qquad \|\mathbf{j}\| = 1$$

$$\mathbf{i} \cdot \mathbf{j} = 0$$

$$\mathbf{v} = (x_1, x_2) \qquad \mathbf{v} = x_1.\mathbf{i} + x_2.\mathbf{j}$$

$$\mathbf{v}.\mathbf{i} = (x_1.\mathbf{i} + x_2.\mathbf{j}).\mathbf{i} = x_1.1 + x_2.0 = x_1$$

$$\mathbf{v}.\mathbf{j} = (x_1.\mathbf{i} + x_2.\mathbf{j}).\mathbf{j} = x_1.0 + x_2.1 = x_2$$

Overview

2D Transformations

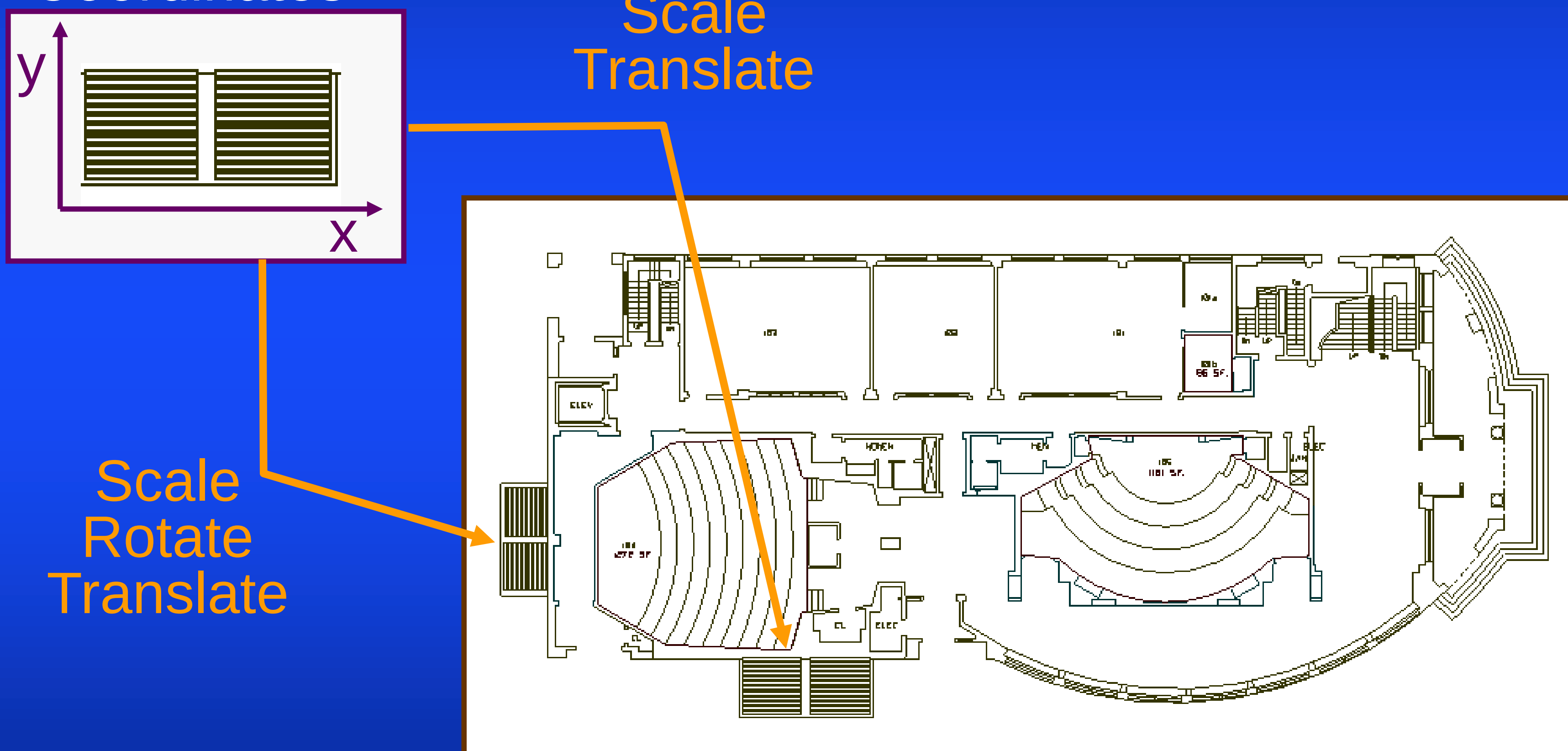
- Basic 2D transformations
- Matrix representation
- Matrix composition

3D Transformations

- Basic 3D transformations
- Same as 2D

2D Modeling Transformations

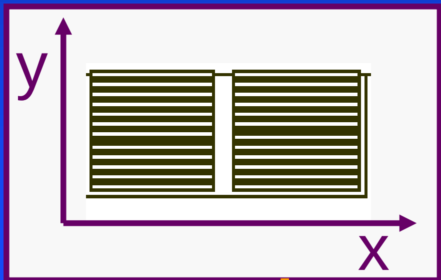
Modeling
Coordinates



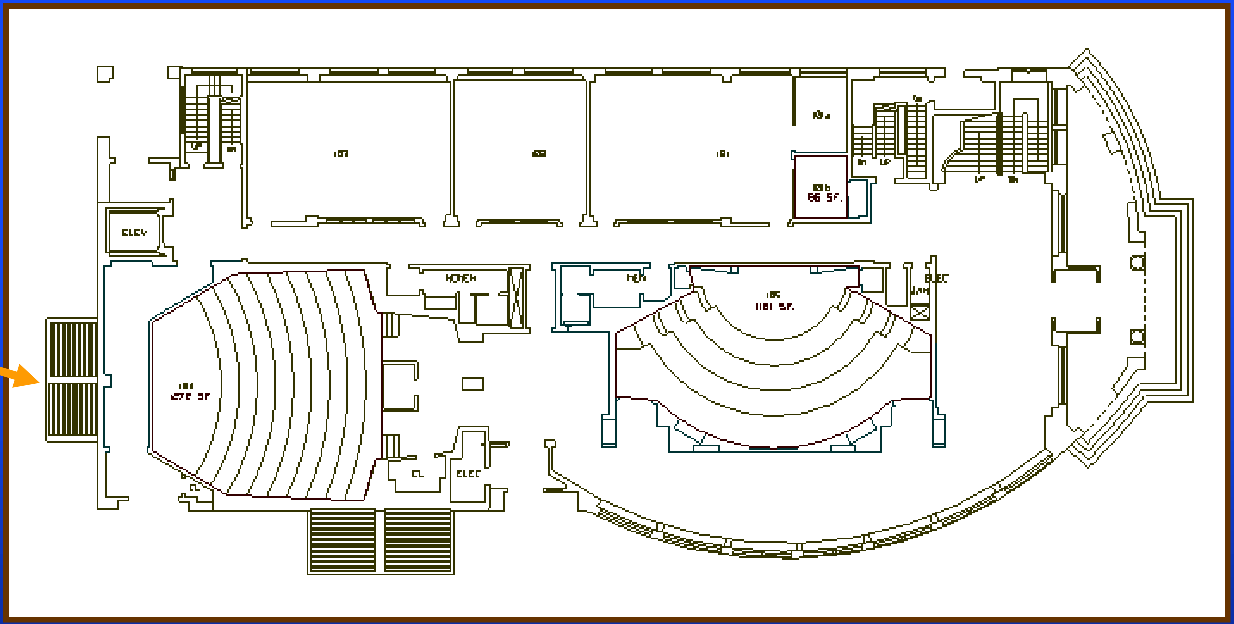
World Coordinates

2D Modeling Transformations

Modeling Coordinates



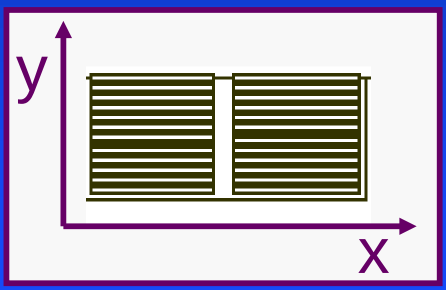
Let's look at this in detail…



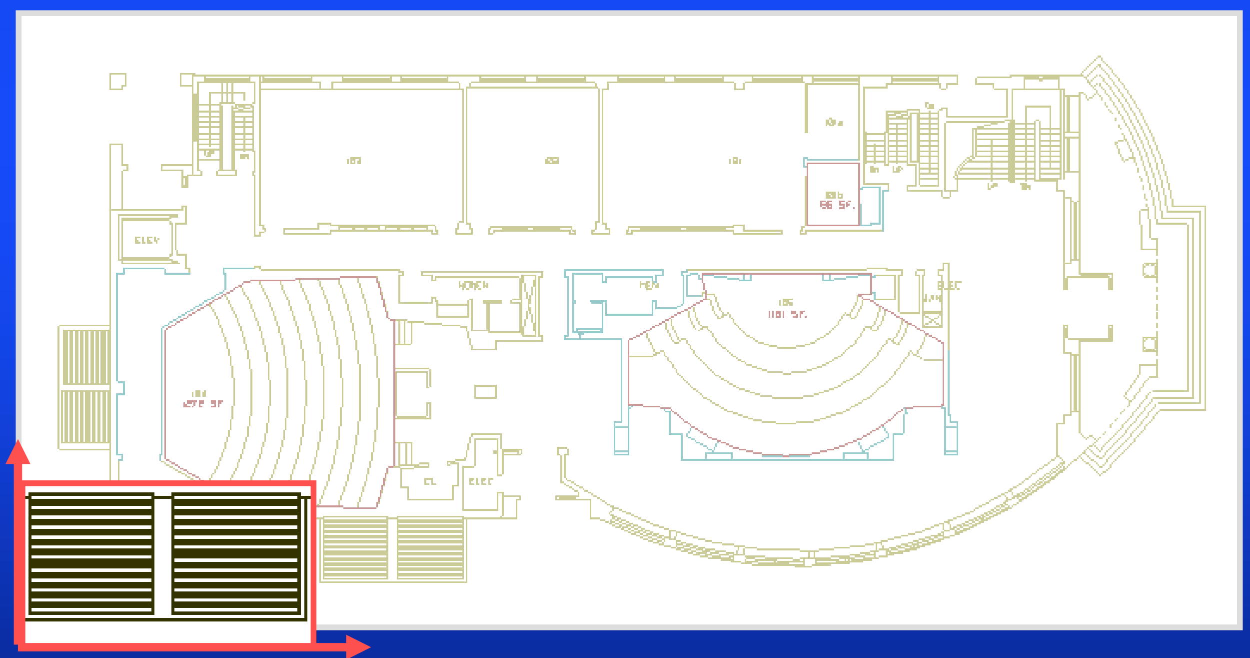
World Coordinates

2D Modeling Transformations

Modeling Coordinates

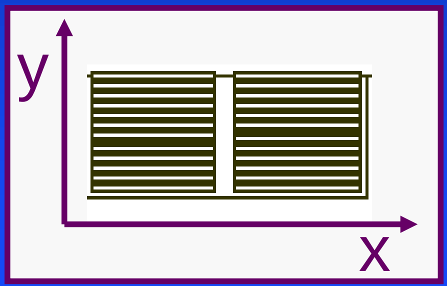


Initial location at (0, 0) with x- and y-axes aligned



2D Modeling Transformations

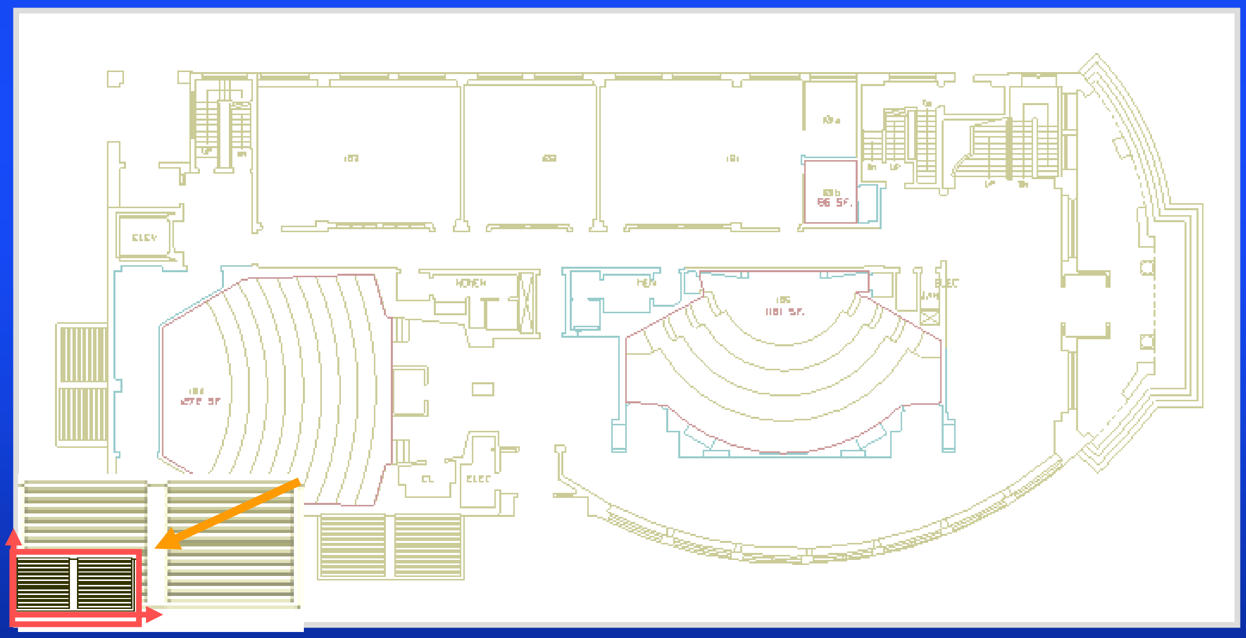
Modeling Coordinates



Scale .3, .3

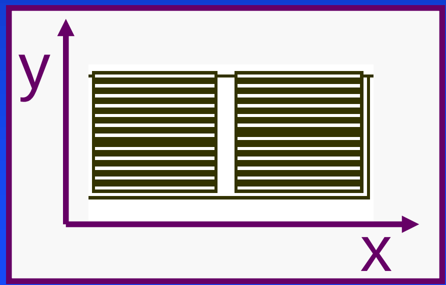
Rotate -90

Translate 5, 3

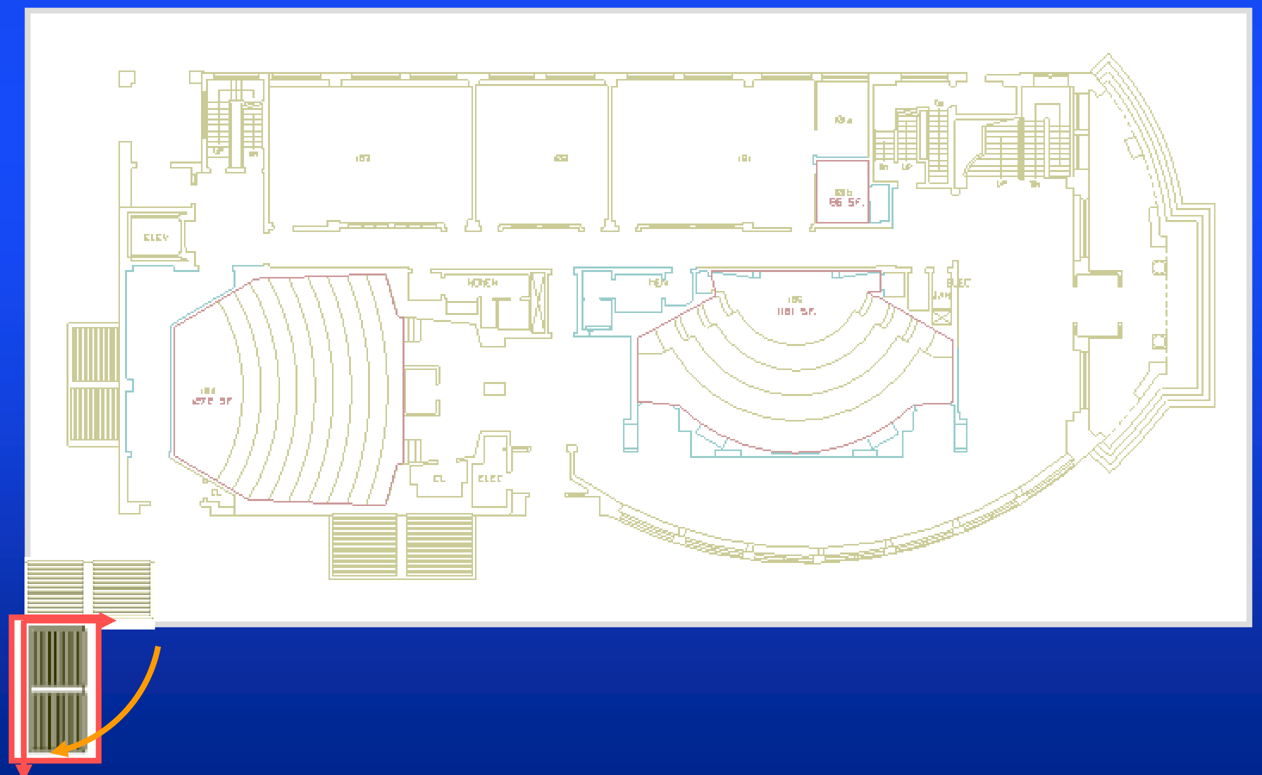


2D Modeling Transformations

Modeling Coordinates

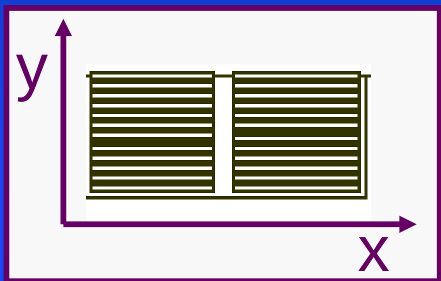


Scale .3, .3
Rotate -90
Translate 5, 3

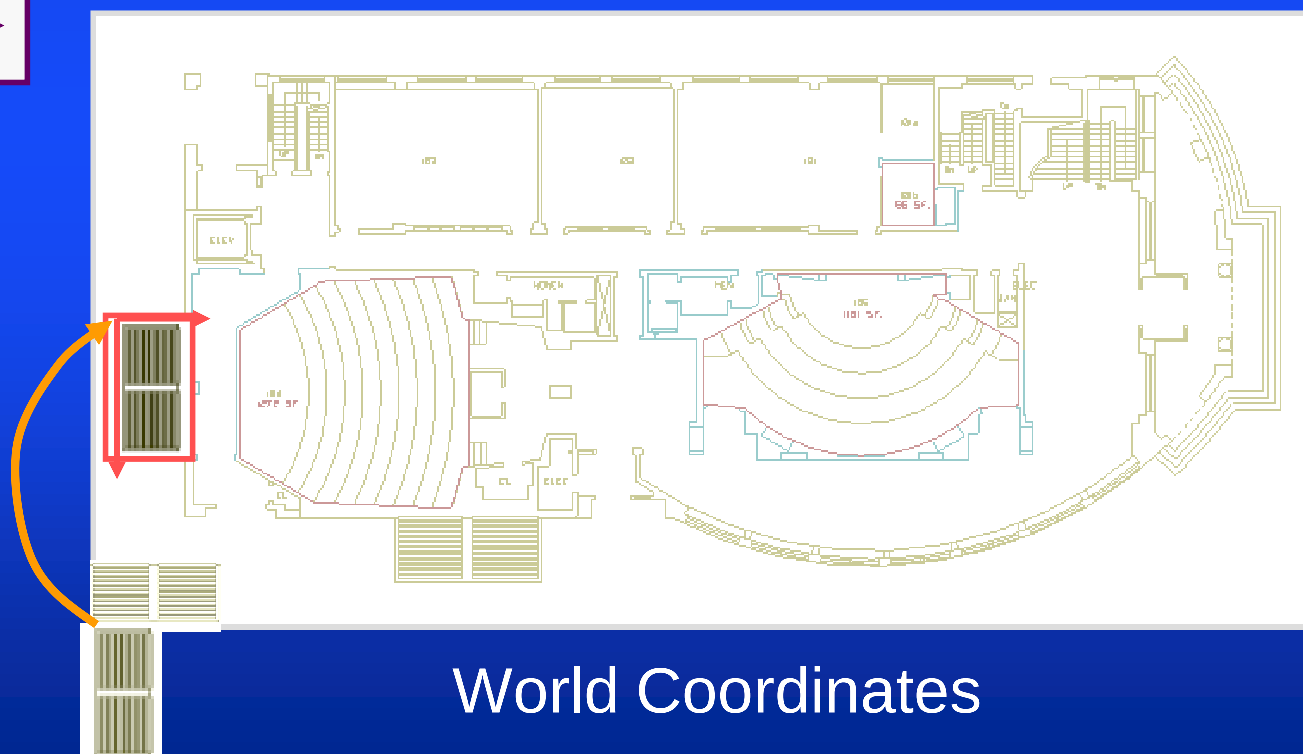


2D Modeling Transformations

Modeling Coordinates

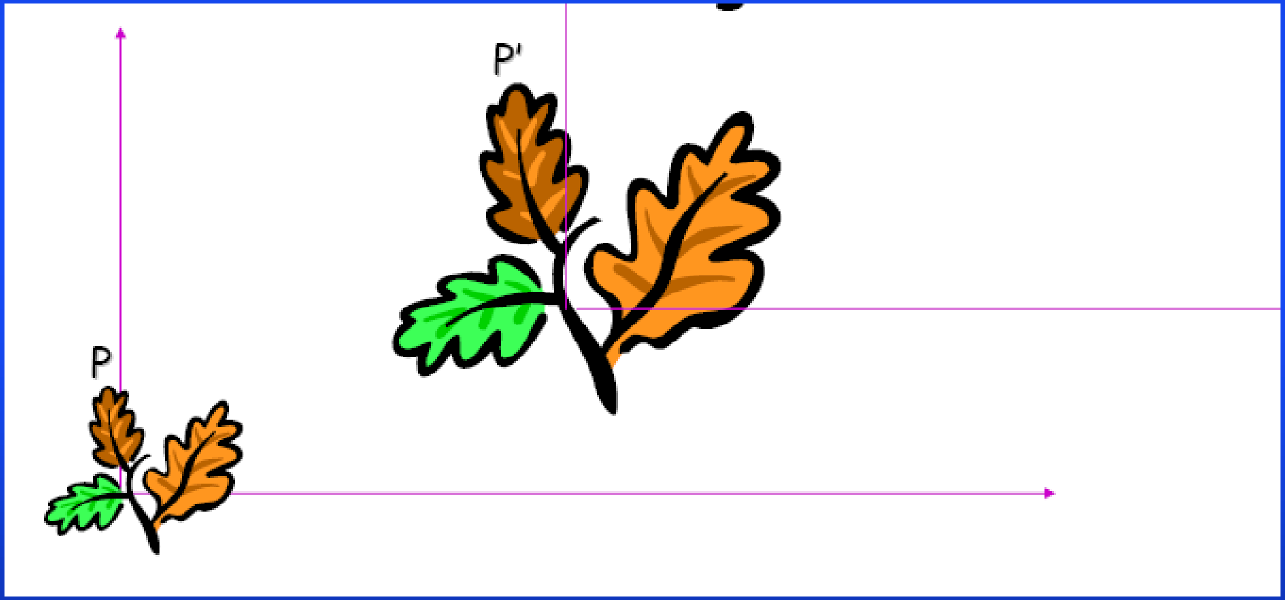


Scale .3, .3
Rotate -90
Translate 5, 3



World Coordinates

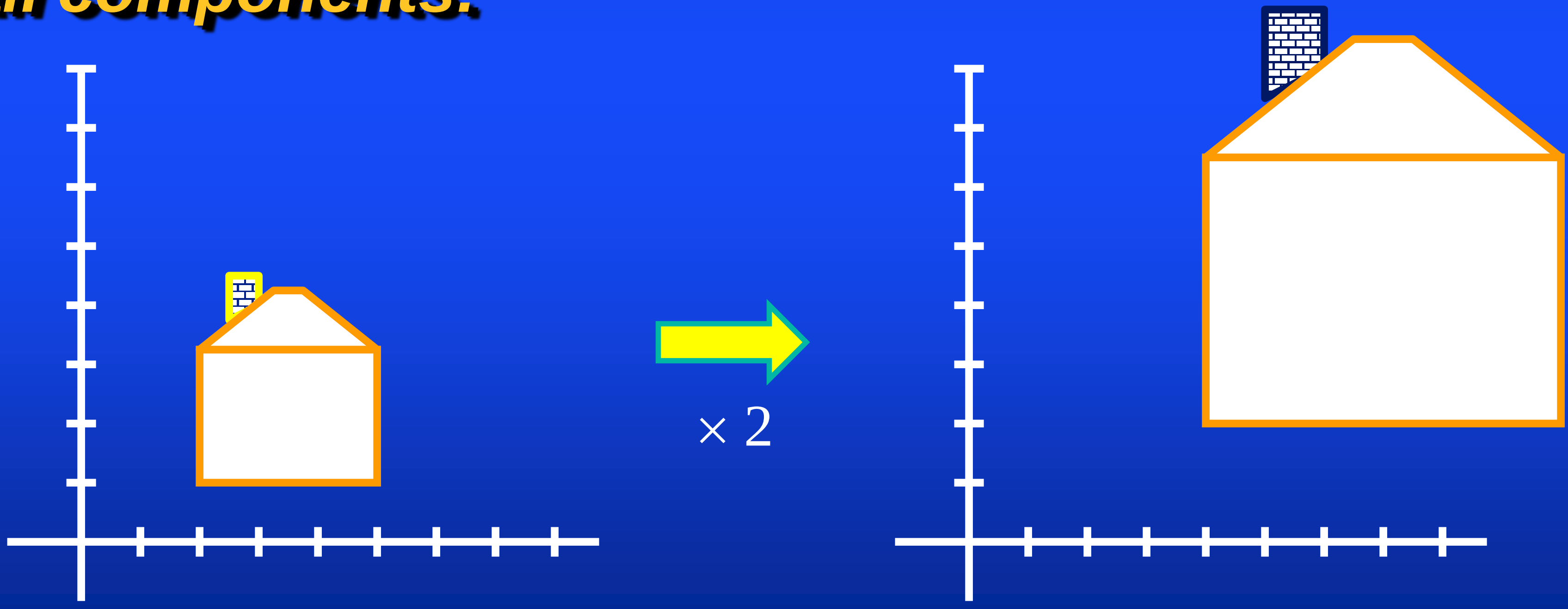
Scaling



Scaling

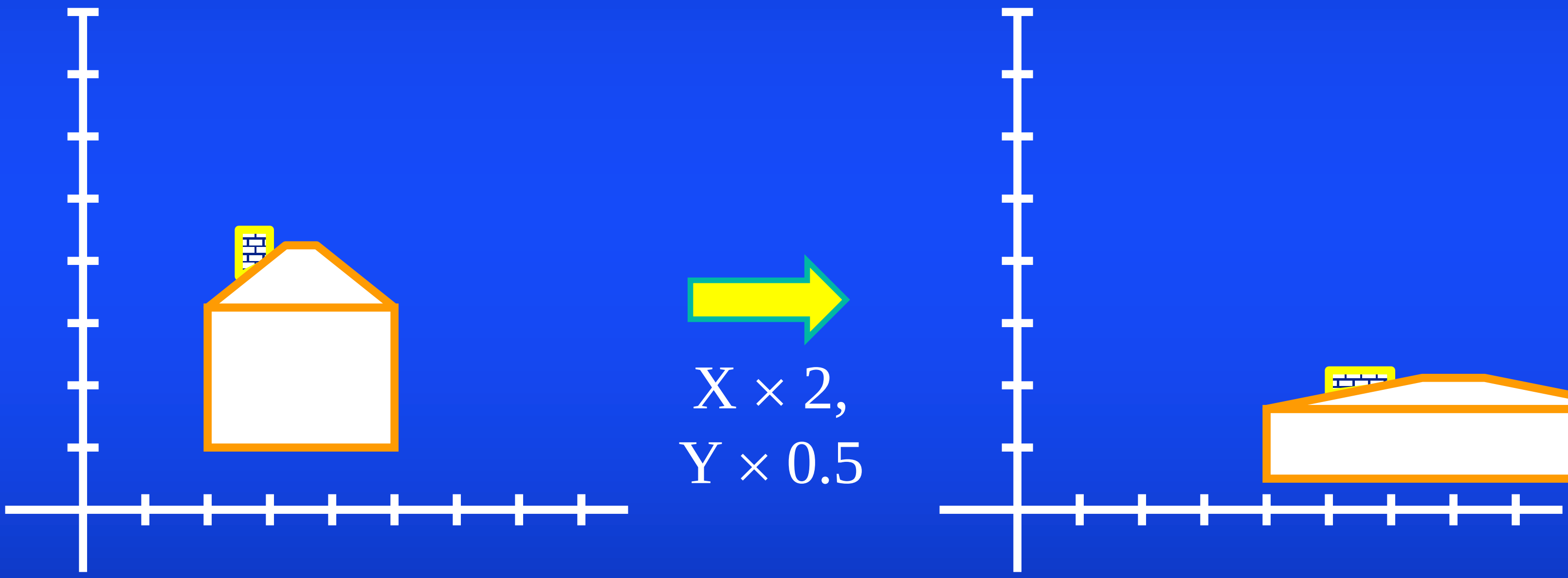
Scaling *a coordinate means multiplying each of its components by a scalar*

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:

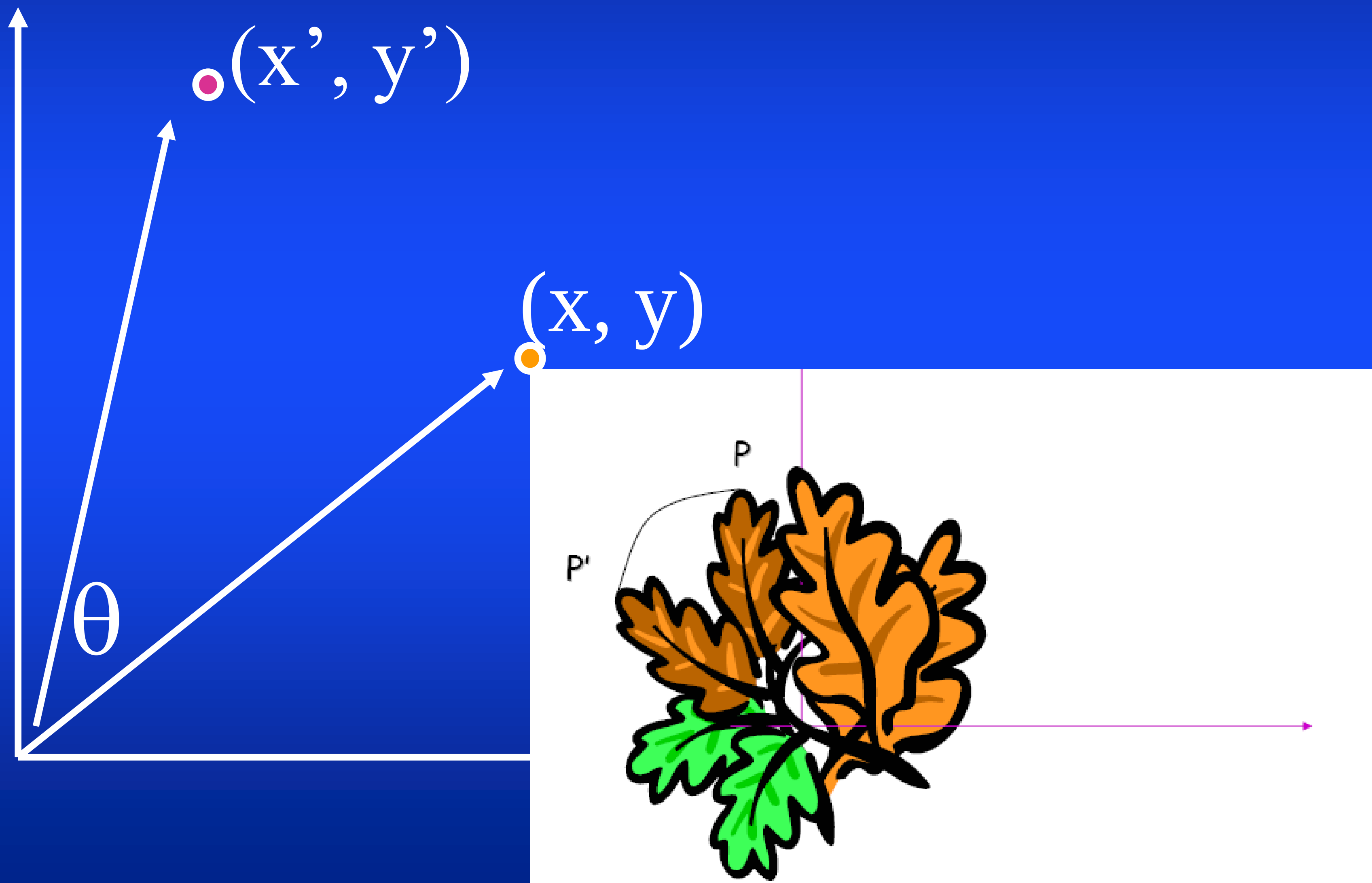
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} ax \\ by \end{bmatrix}$$

Or, in matrix form:

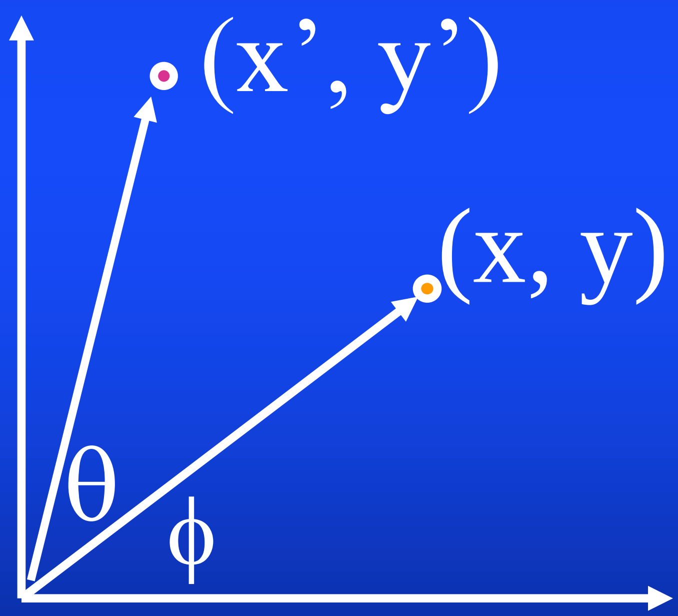
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\text{scaling matrix}} \begin{bmatrix} x \\ y \end{bmatrix}$$

scaling matrix

2-D Rotation



2-D Rotation



$x = r \cos(\phi)$

$y = r \sin(\phi)$

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

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

Trig Identity…

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

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

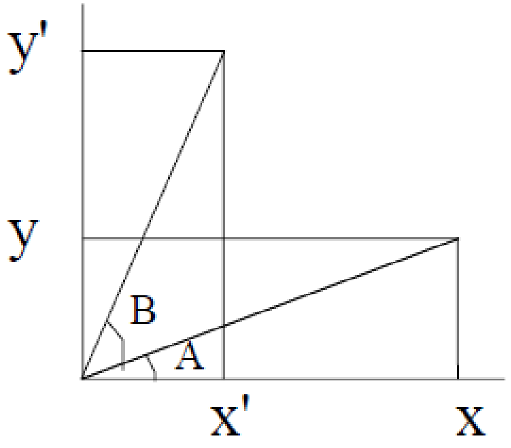
Substitute…

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

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

Geometric Transformations

Rotation Equations:



$$x = r.\cos A$$
$$y = r.\sin A$$

$$x' = r.\cos(A+B) = r.\cos A.\cos B - r.\sin A.\sin B$$
$$y' = r.\sin(A+B) = r.\sin A.\cos B + r.\cos A.\sin B$$

$$x' = x.\cos B - y.\sin B$$
$$y' = y.\cos B + x.\sin B$$

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

$$\sin(A + B) = \sin A \cos B + \cos A \sin B$$
$$\cos(A + B) = \cos A \cos B - \sin A \sin B$$

2-D Rotation

This is easy to capture in matrix form:

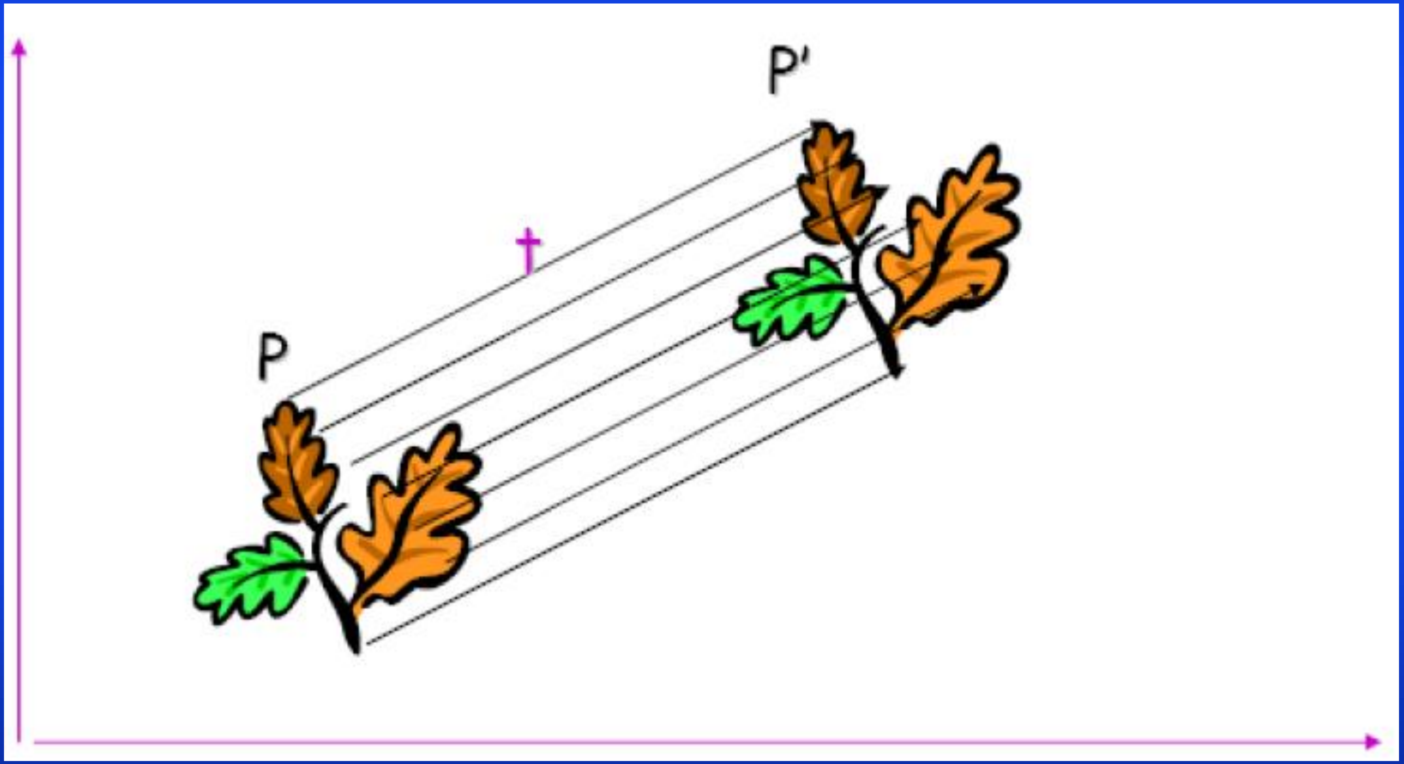
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

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

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

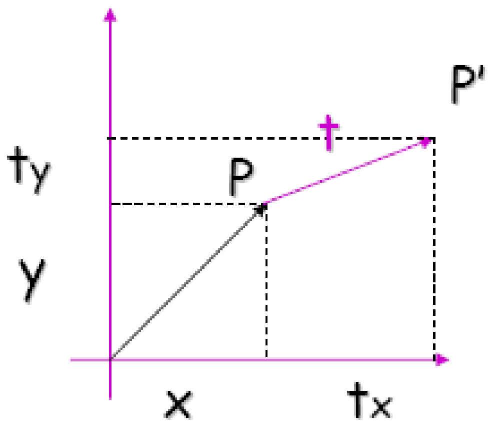
Geometric Transformations

2D Translation:



Geometric Transformations

2D Translation Equation:



$$\mathbf{P} = (x, y)$$

$$\mathbf{t} = (t_x, t_y)$$

$$\mathbf{P}' = (x + t_x, y + t_y) = \mathbf{P} + \mathbf{t}$$

Basic 2D Transformations

Translation:

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

Scale:

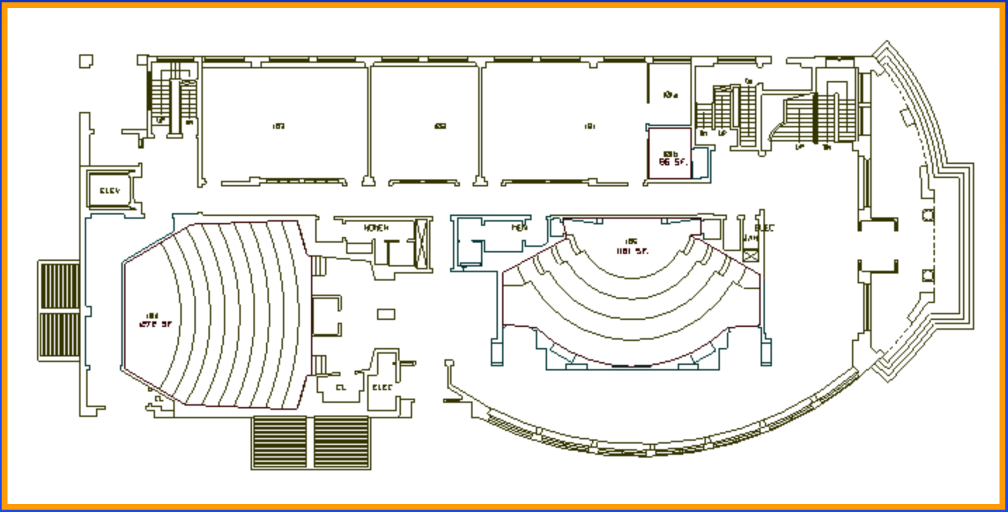
- $x' = x * s_x$
- $y' = y * s_y$

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:

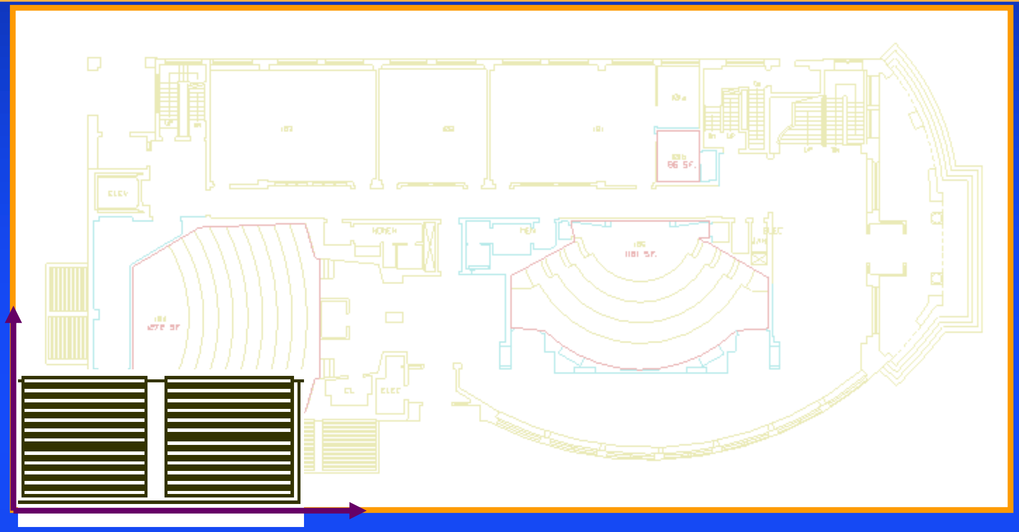
- $x' = x * s_x$
- $y' = y * s_y$

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:

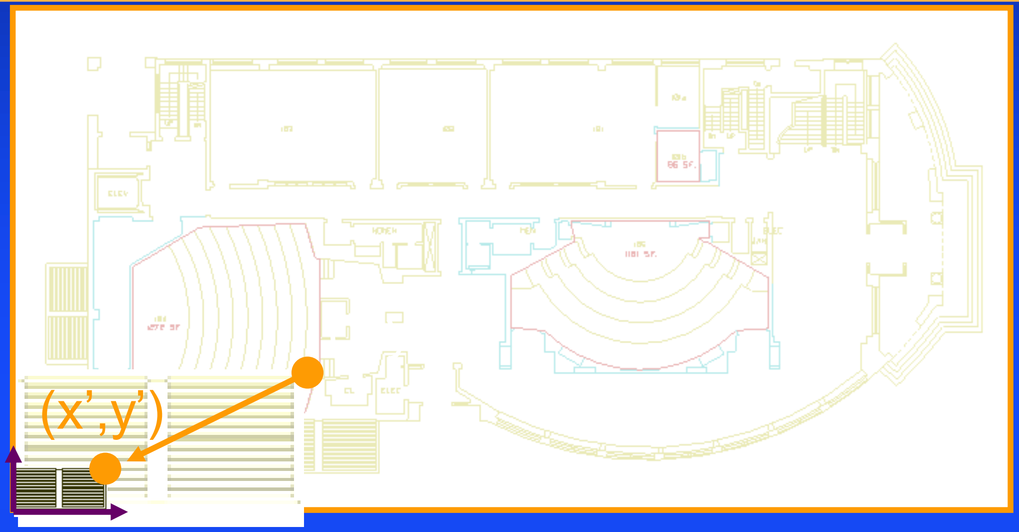
- $x' = x * s_x$
- $y' = y * s_y$

Shear:

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

Rotation:

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



$x' = x * s_x$

$y' = y * s_y$

Basic 2D Transformations

Translation:

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

Scale:

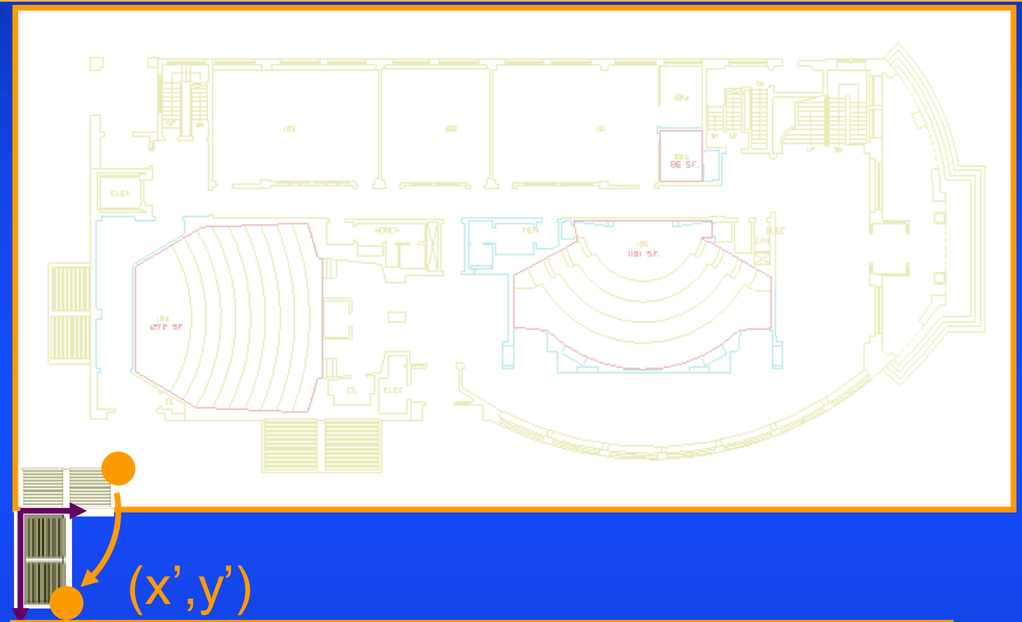
- $x' = x * s_x$
- $y' = y * s_y$

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:

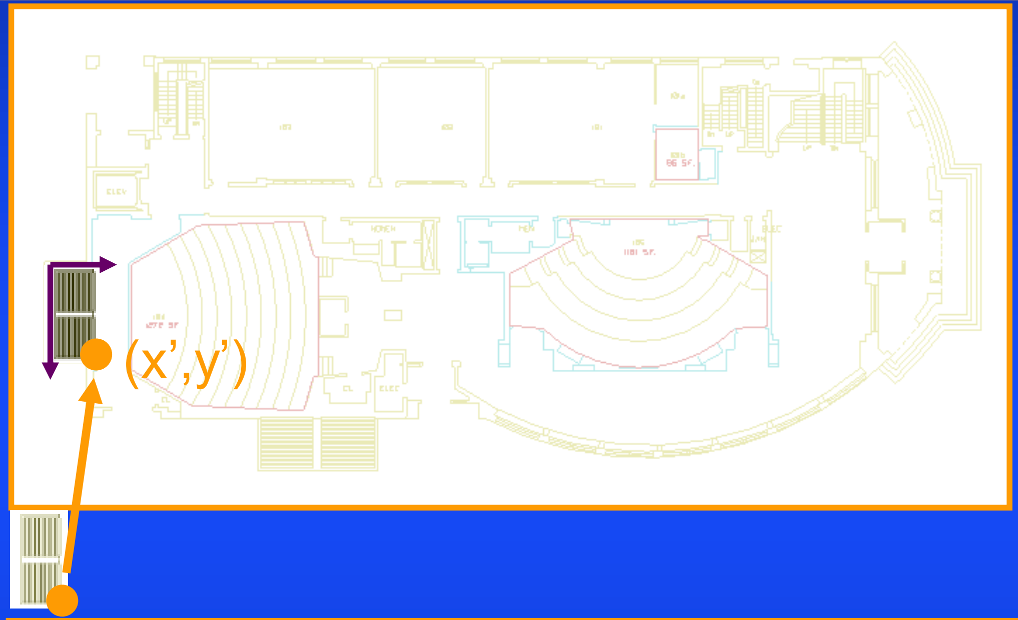
- $x' = x * s_x$
- $y' = y * s_y$

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:

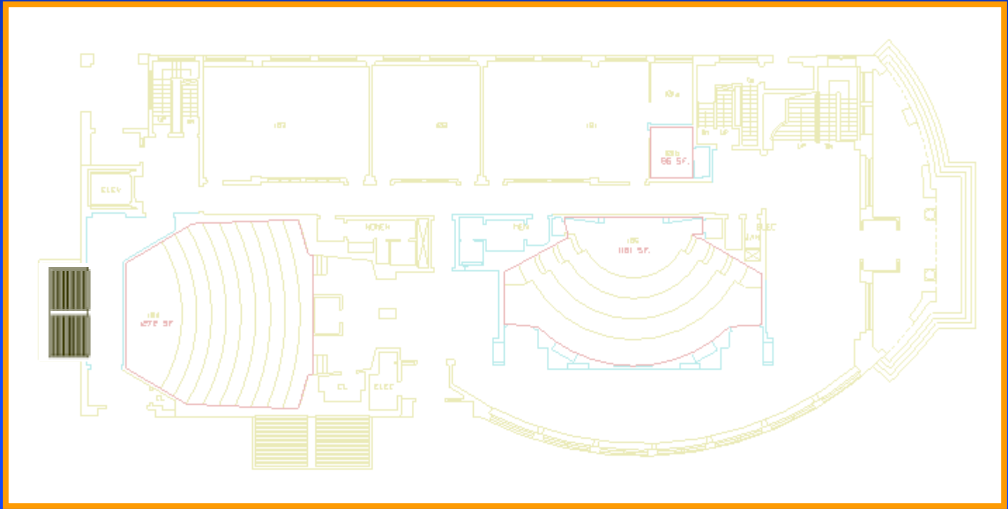
- $x' = x * s_x$
- $y' = y * s_y$

Shear:

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

Rotation:

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



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

Outline

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

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

$$x' = ax + by$$

$$y' = cx + dy$$

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)?

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

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

2x2 Matrices

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

2D Rotate around (0,0)?

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

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

2D Shear?

$$x' = x + sh_x * y$$
$$y' = sh_y * x + y$$

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

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?

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

NO!

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
- Closed under composition

Homogeneous Coordinates

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

$$x' = x + t_x$$

$$y' = y + t_y$$

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:

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

Homogeneous Coordinates

Homogeneous Coordinates

Multiply the coordinates by a non-zero scalar and add an extra coordinate equal to that scalar. For example,

$$(x, y) \rightarrow (x \cdot z, y \cdot z, z) \quad z \neq 0$$

$$(x, y, z) \rightarrow (x \cdot w, y \cdot w, z \cdot w, w) \quad w \neq 0$$

NOTE: If the scalar is 1, there is no need for the multiplication!

Homogeneous Coordinates

Homogeneous Coordinates → Back to Cartesian Coordinates

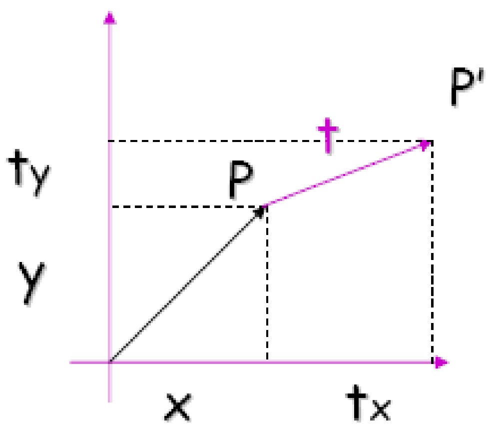
Divide by the last coordinate and eliminate it. For example,

$$(x, y, z) \quad z \neq 0 \rightarrow (x/z, y/z)$$

$$(x, y, z, w) \quad w \neq 0 \rightarrow (x/w, y/w, z/w)$$

2D Translation using Homogeneous Coordinates

2D Translation using Homogeneous Coordinates



$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{t} = (t_x, t_y) \rightarrow (t_x, t_y, 1)$$

$$\mathbf{P'} \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{T}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

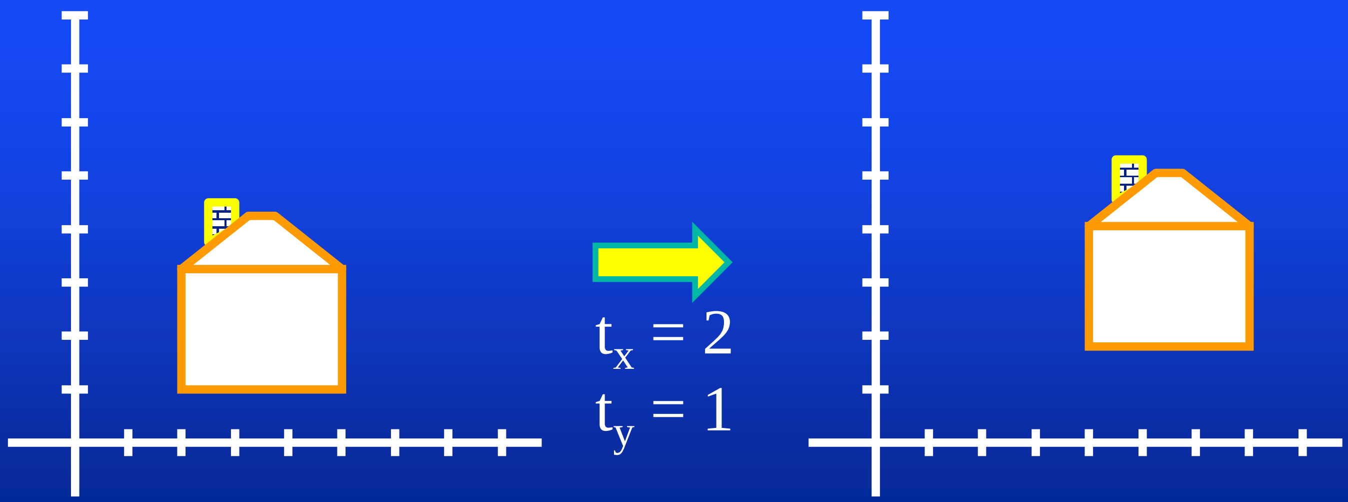
$$\mathbf{P'} = \mathbf{T} \cdot \mathbf{P}$$

Translation

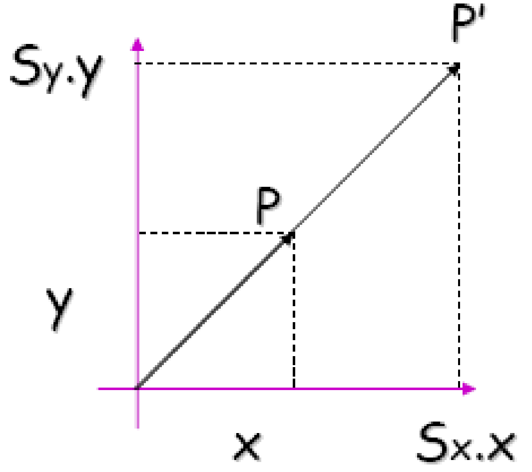
Example of translation

Homogeneous Coordinates

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



Scaling Equation



$$\mathbf{P} = (x, y) \rightarrow (x, y, 1)$$

$$\mathbf{P}' = (s_x x, s_y y) \rightarrow (s_x x, s_y y, 1)$$

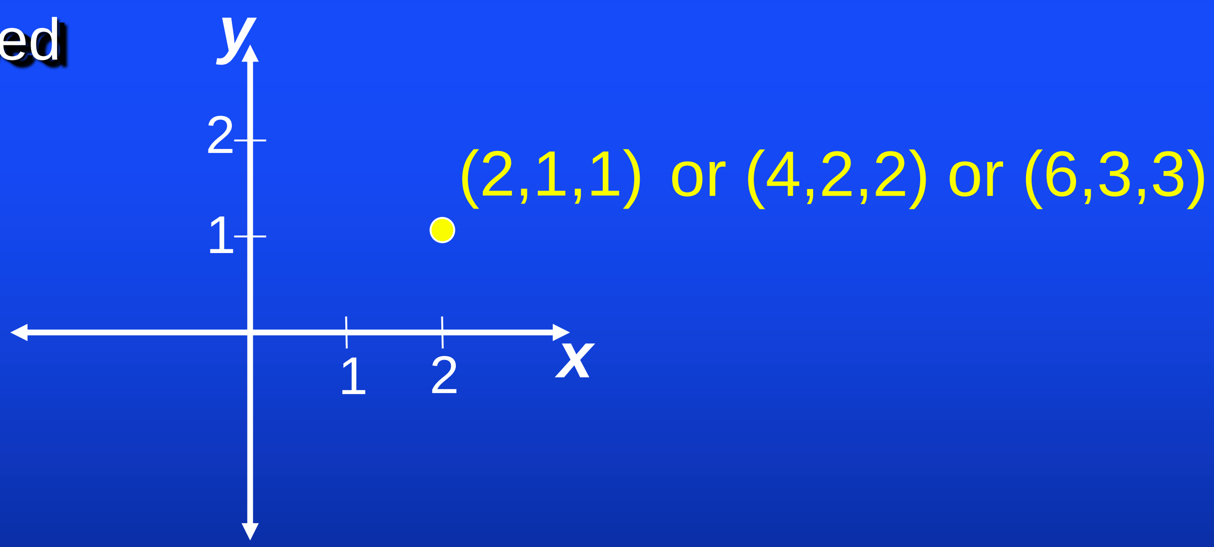
$$\mathbf{P}' \rightarrow \begin{bmatrix} s_x x \\ s_y y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{S}} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$

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

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

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

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

Outline

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

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

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

Matrix Composition

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

- General purpose representation
- Hardware matrix multiply

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

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

Matrix Composition

Be aware: order of transformations matters

– Matrix multiplication is not commutative

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

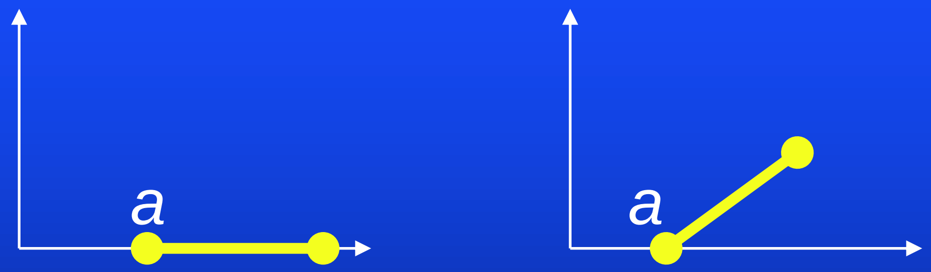
“Global” ⟷ “Local”

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?

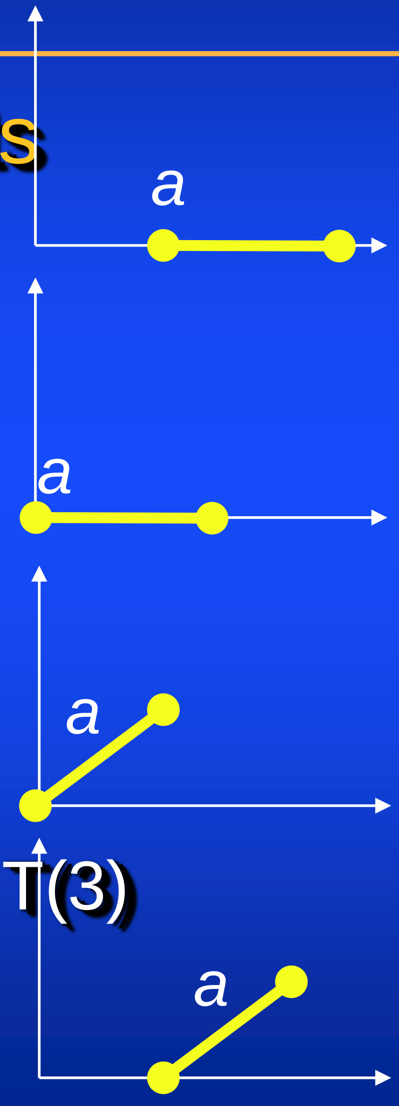
Wrong
R(45)

Correct
1. T(-3)
2. R(45)
3. 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?

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

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

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

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

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

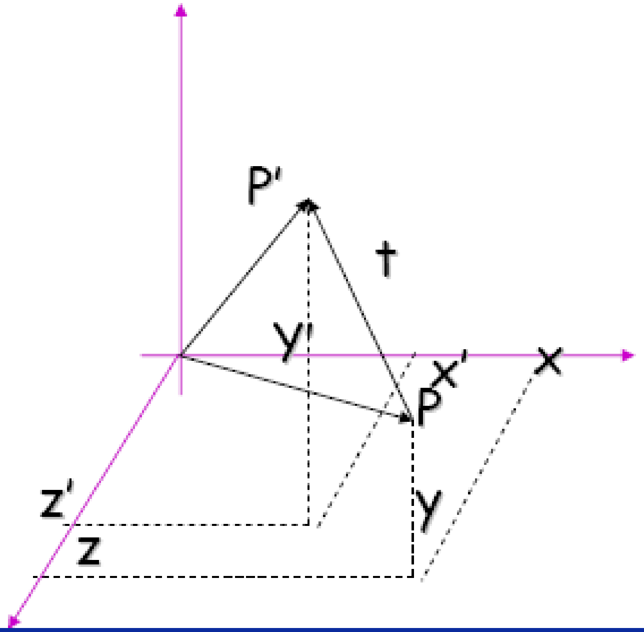
$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Mirror about Y/Z plane

Geometric Transformations

3D Translation of Points:

Translate by a vector $t=(t_x,t_y,t_x)^T$:



$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Basic 3D Transformations

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

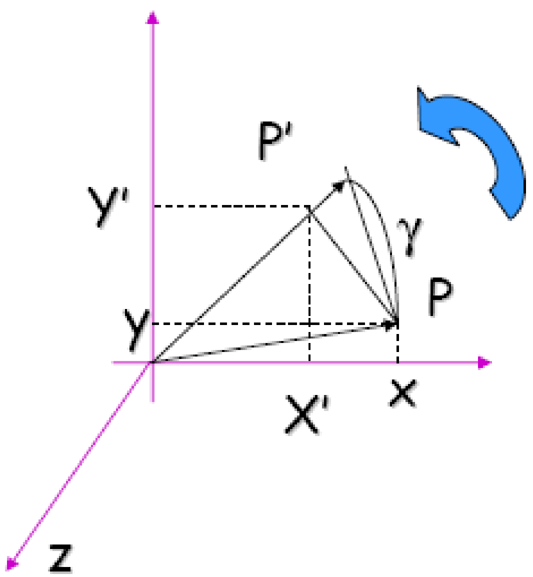
Rotate around Y axis:
$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ \boldsymbol{z}' \\ \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\Theta & 0 & \cos\Theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \boldsymbol{w} \end{bmatrix}$$

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

Geometric Transformations

3D Rotation of Points:

Rotation around the coordinate axes, counter-clockwise:

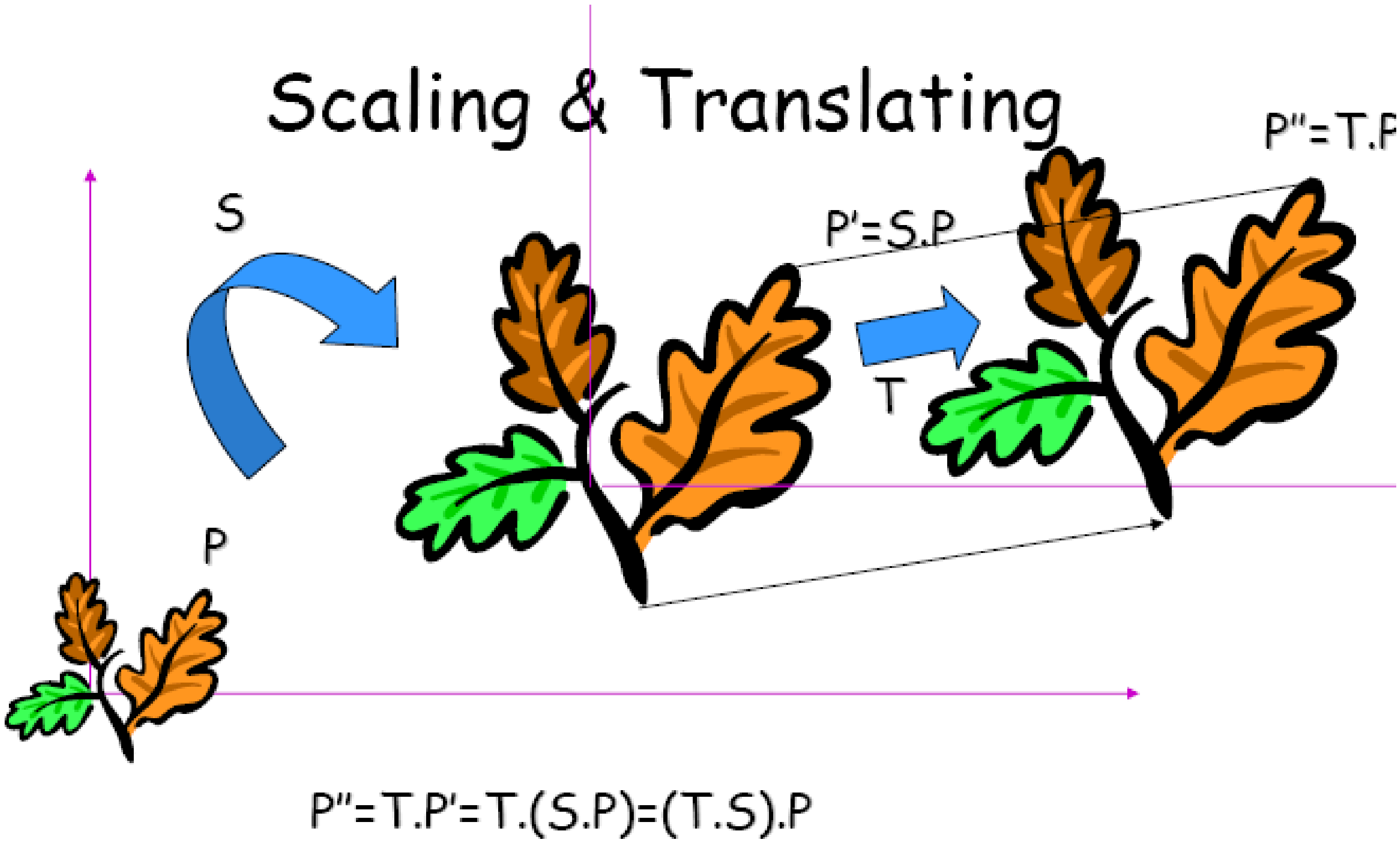


$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Geometric Transformations



Geometric Transformations

- Scaling & Translating equations

$$P'' = T.P' = T.(S.P) = (T.S).P$$

$$\mathbf{P''} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

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

Geometric Transformations

Translating & Scaling
≠ Scaling & Translating

P''=S.P'=S.(T.P)=(S.T).P

$$\mathbf{P''} = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{P} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} =$$

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

Geometric Transformations

- Scaling, Translating & Rotating:

Order matters!

$$P' = S.P$$
$$P'' = T.P' = (T.S).P$$
$$P''' = R.P'' = R.(T.S).P = (R.T.S).P$$

$$R.T.S \neq R.S.T \neq T.S.R \ldots$$