Scene Graphs & Modeling Transformations

COS 426, Fall 2022

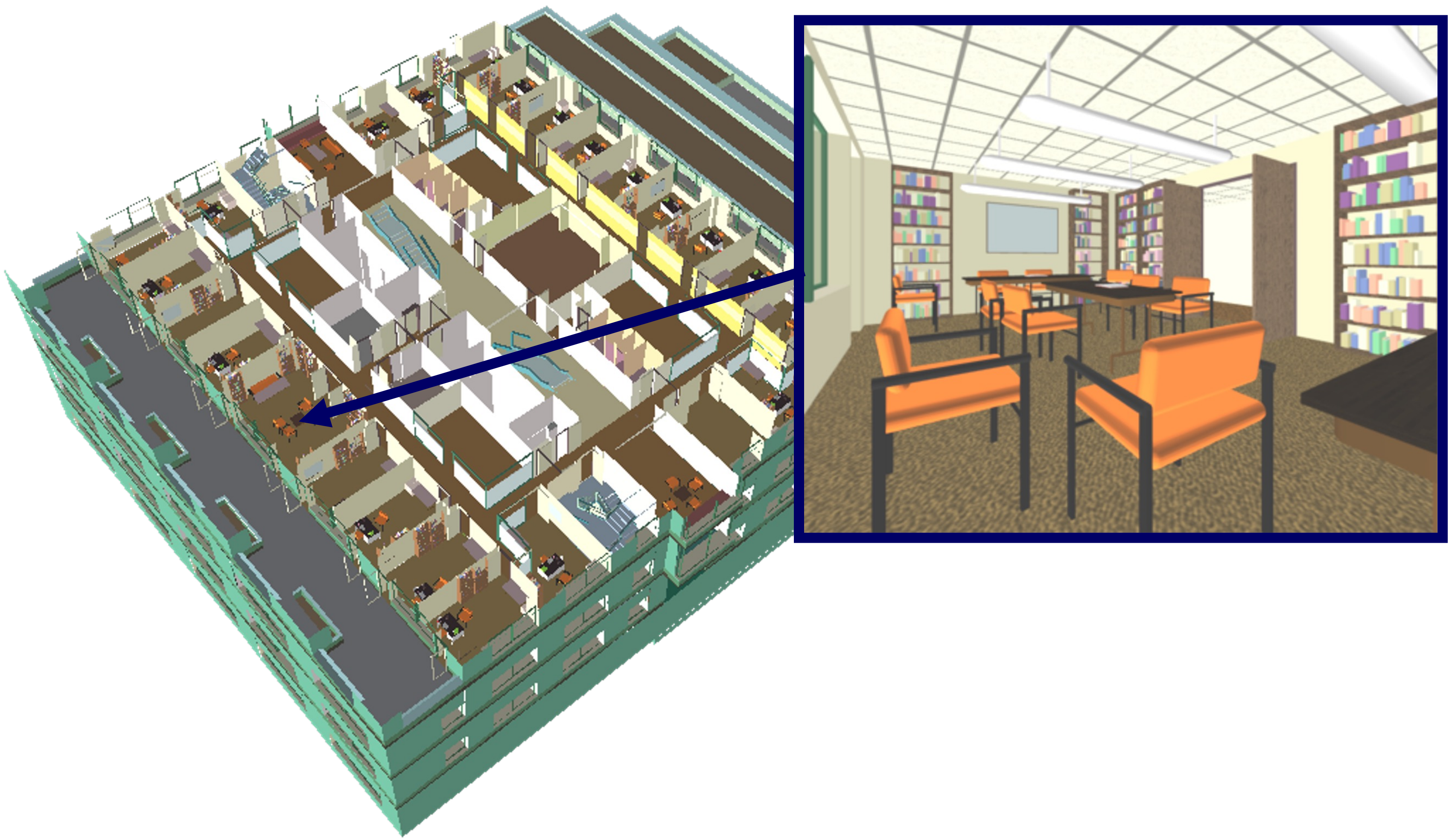
PRINCETON UNIVERSITY

3D Object Representations

- Points
  - Range image
  - Point cloud
- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Application specific

3D Object Representations

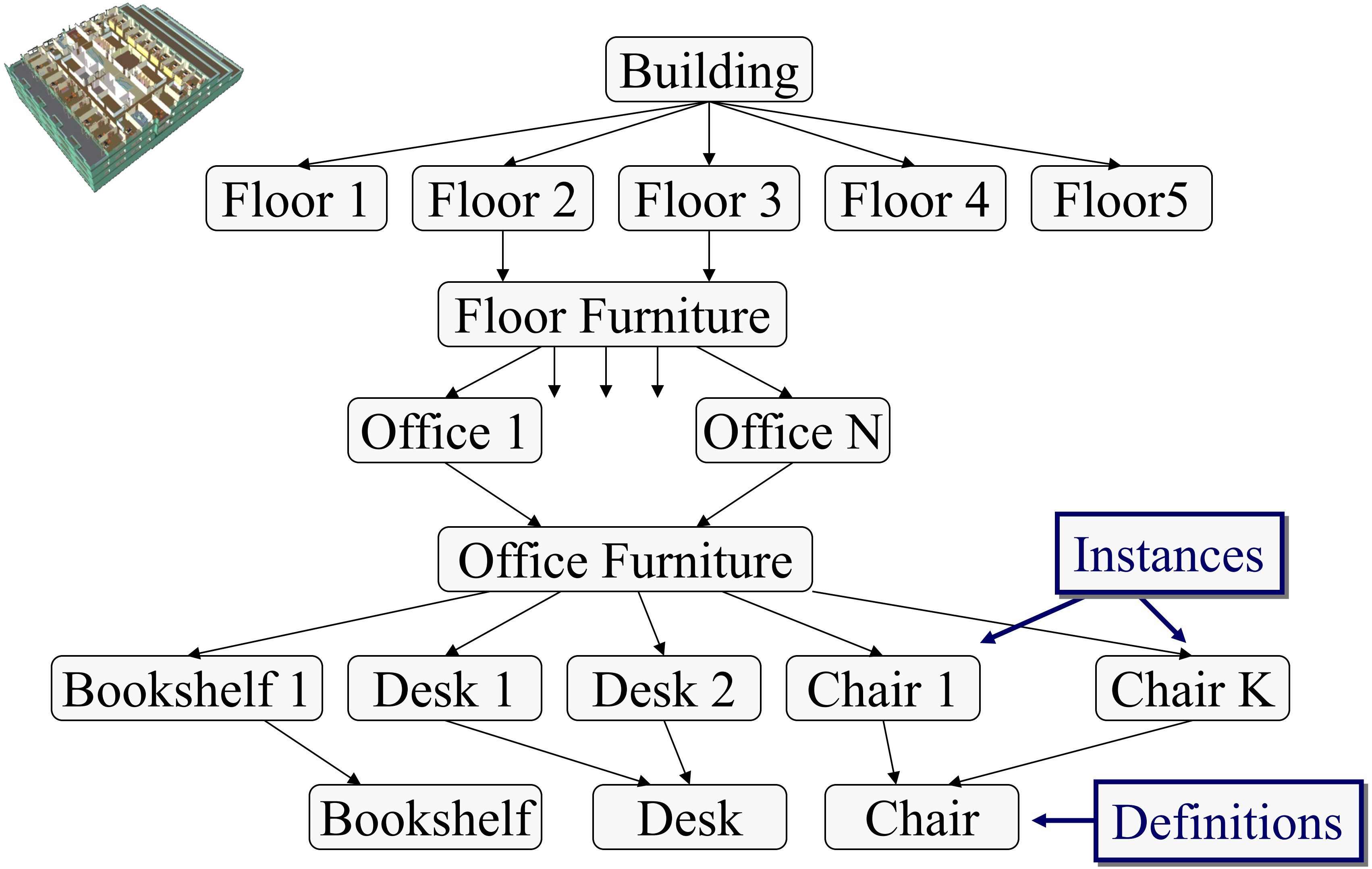
- What object representation is best for this?



Overview

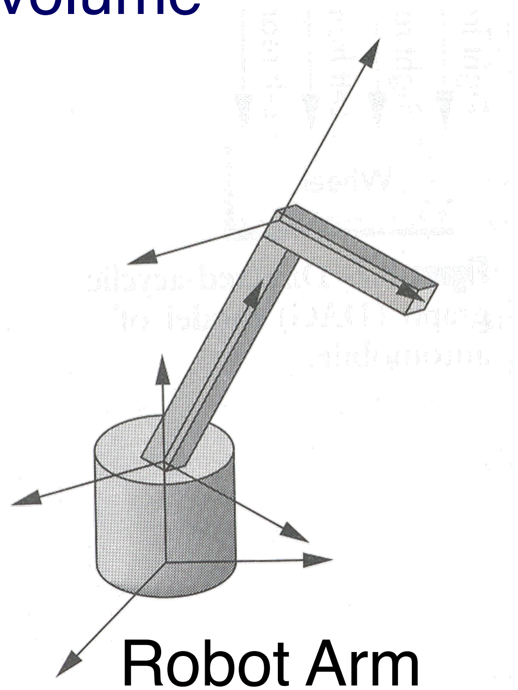
- Scene graphs
  - Geometry & attributes
  - Transformations
  - Bounding volumes
- Transformations
  - Basic 2D transformations
  - Matrix representation
  - Matrix composition
  - 3D transformations

Scene Graphs

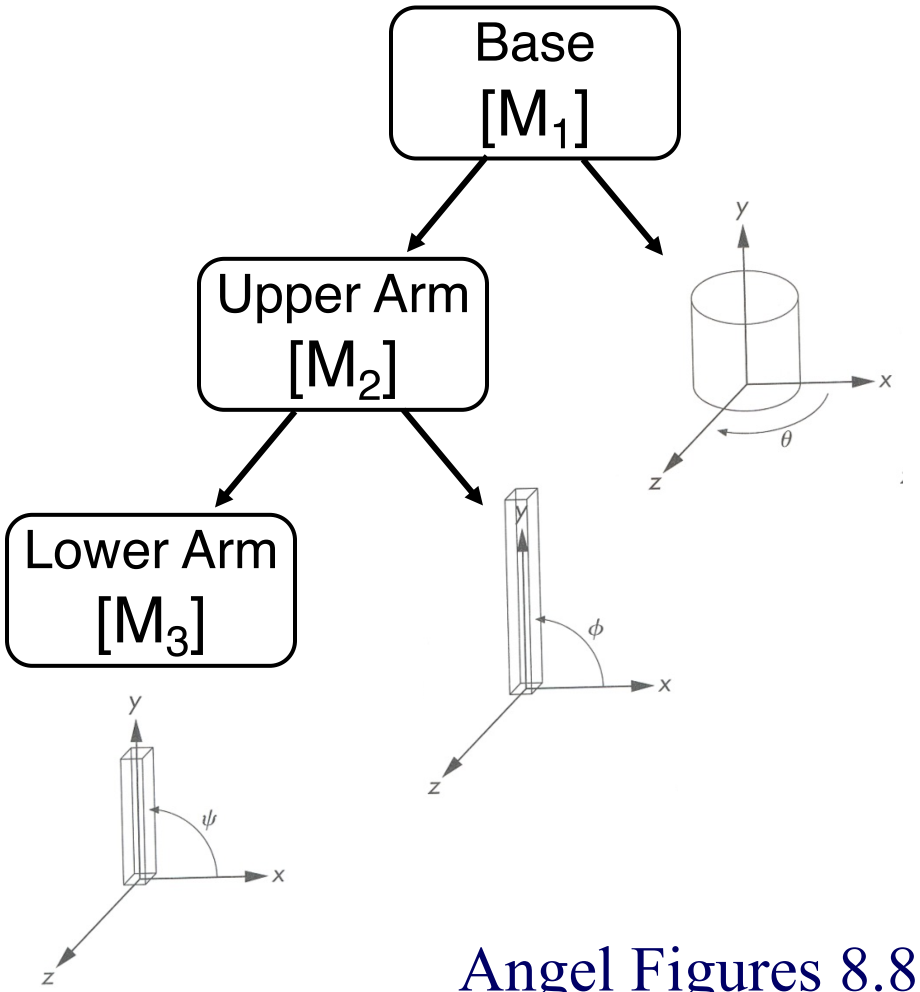


Scene Graphs

- Hierarchy (DAG) of nodes, where each node may have:
  - Geometry representation
  - Modeling transformation
  - Parents and/or children
  - Bounding volume



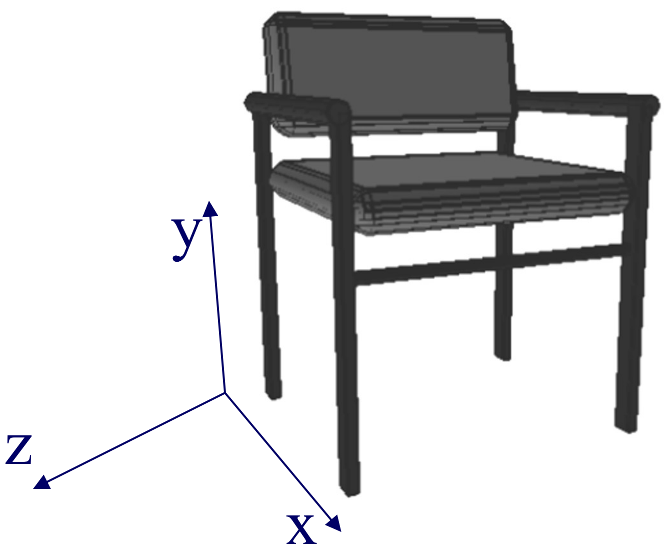
Robot Arm



Angel Figures 8.8 & 8.9

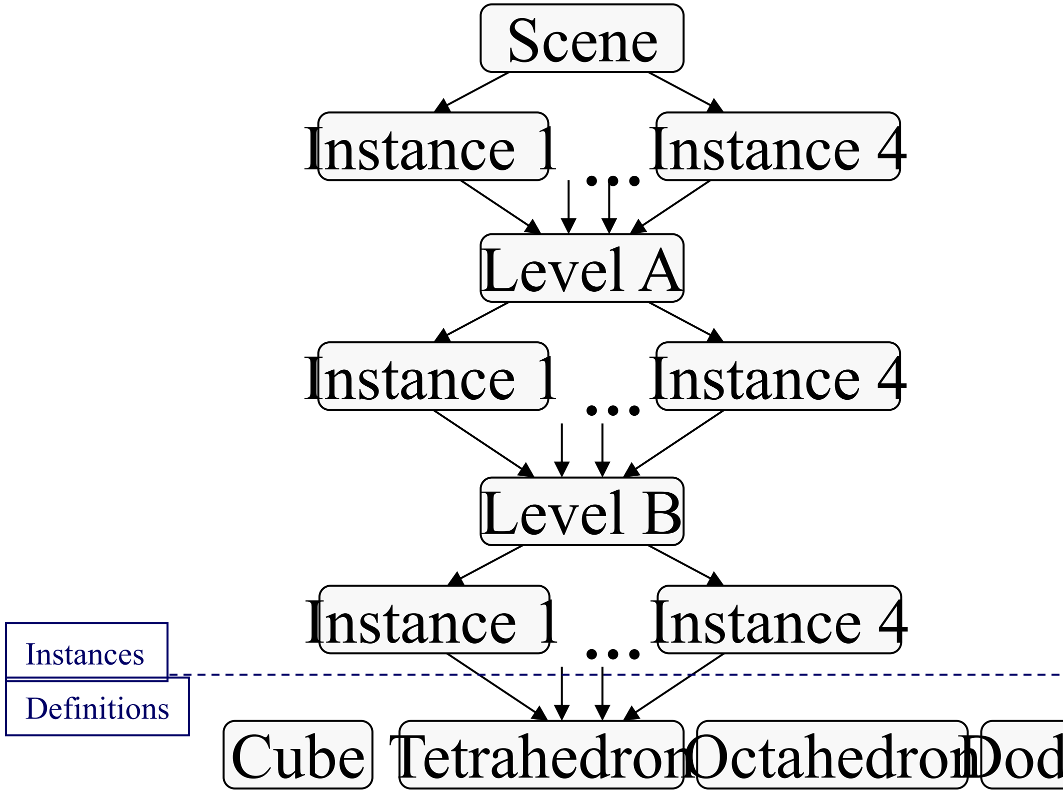
Scene Graphs

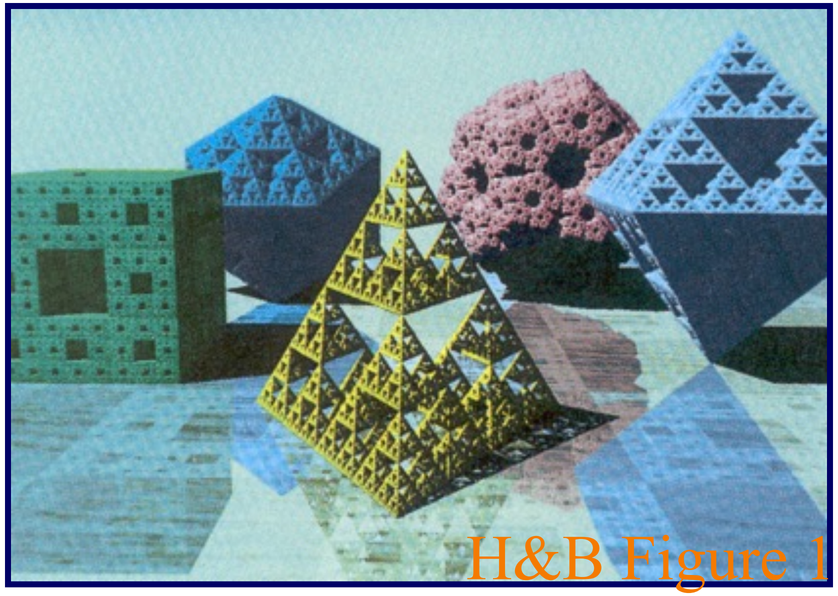
- Advantages
  - Allows definitions of objects in own coordinate systems



Scene Graphs

- Advantages
  - Allows definitions of objects in own coordinate systems
  - Allows use of object definition multiple times in a scene

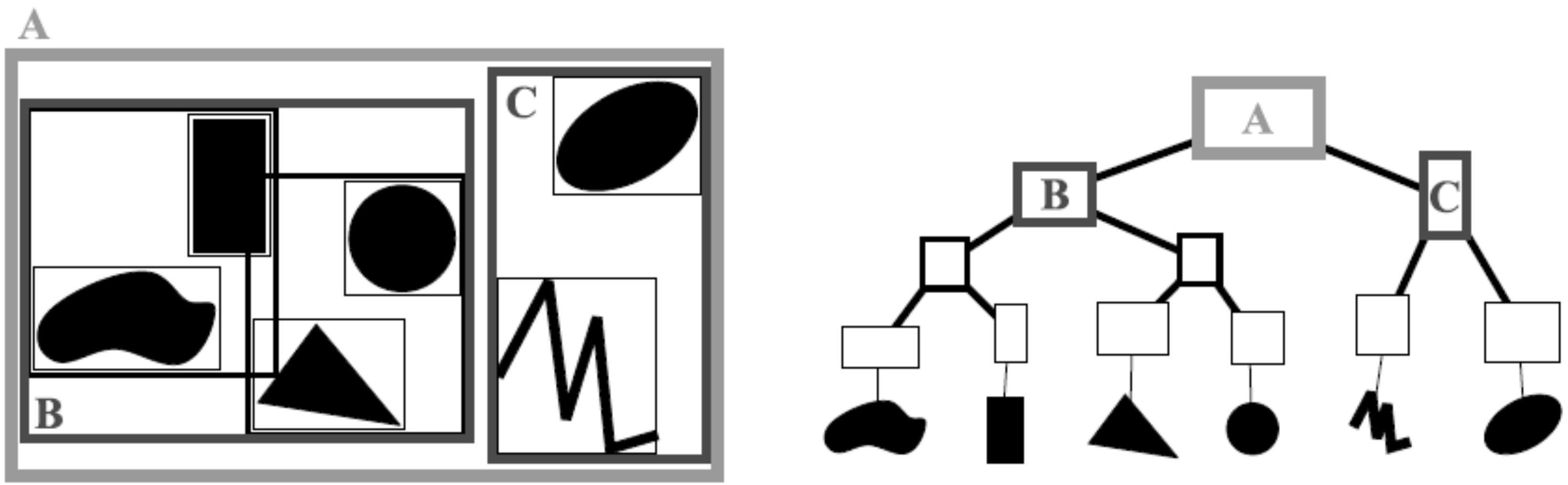




H&B Figure 109

Scene Graphs

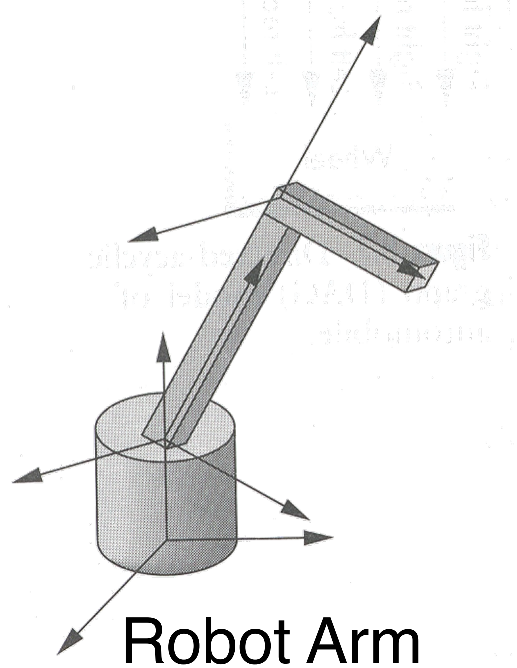
- Advantages
  - Allows definitions of objects in own coordinate systems
  - Allows use of object definition multiple times in a scene
  - Allows hierarchical processing (e.g., intersections)



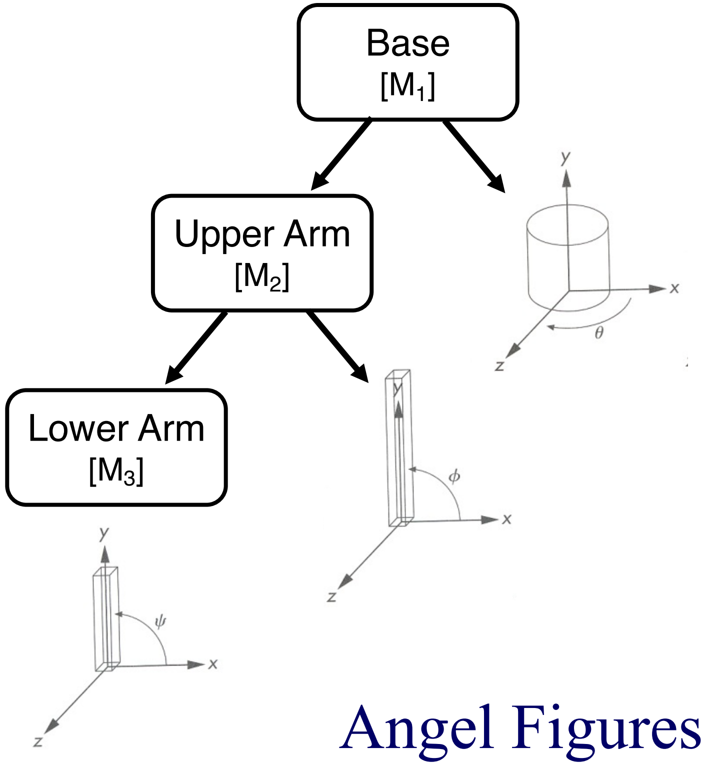
Haverkort

Scene Graphs

- Advantages
  - Allows definitions of objects in own coordinate systems
  - Allows use of object definition multiple times in a scene
  - Allows hierarchical processing (e.g., intersections)
  - Allows articulated animation



Robot Arm



Angel Figures 8.8 & 8.9

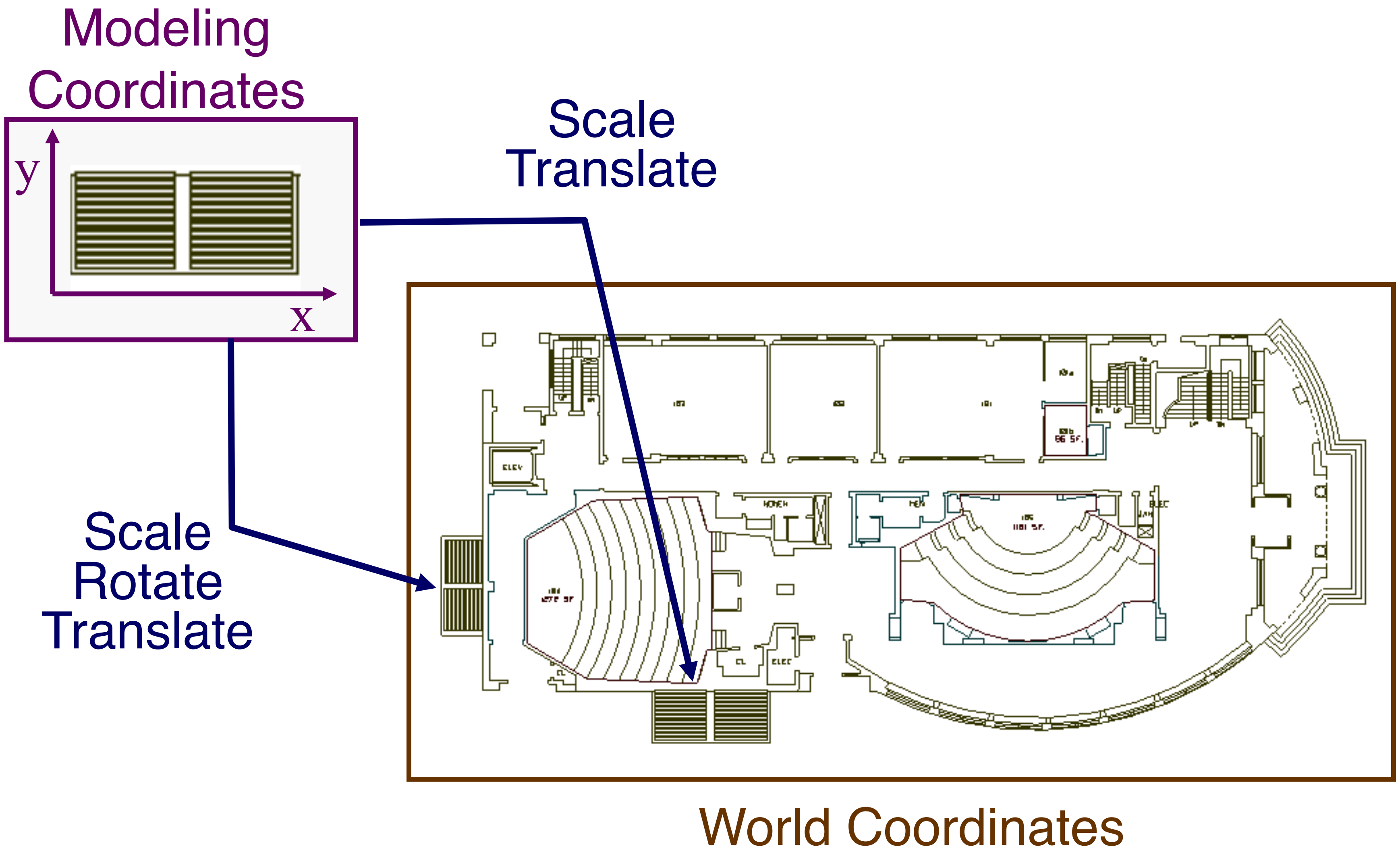
Scene Graph Example

Pixar

Overview

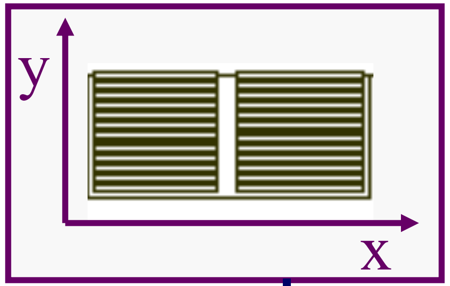
- Scene graphs
  - Geometry & attributes
  - Transformations
  - Bounding volumes
- Transformations
  - Basic 2D transformations
  - Matrix representation
  - Matrix composition
  - 3D transformations

2D Modeling Transformations

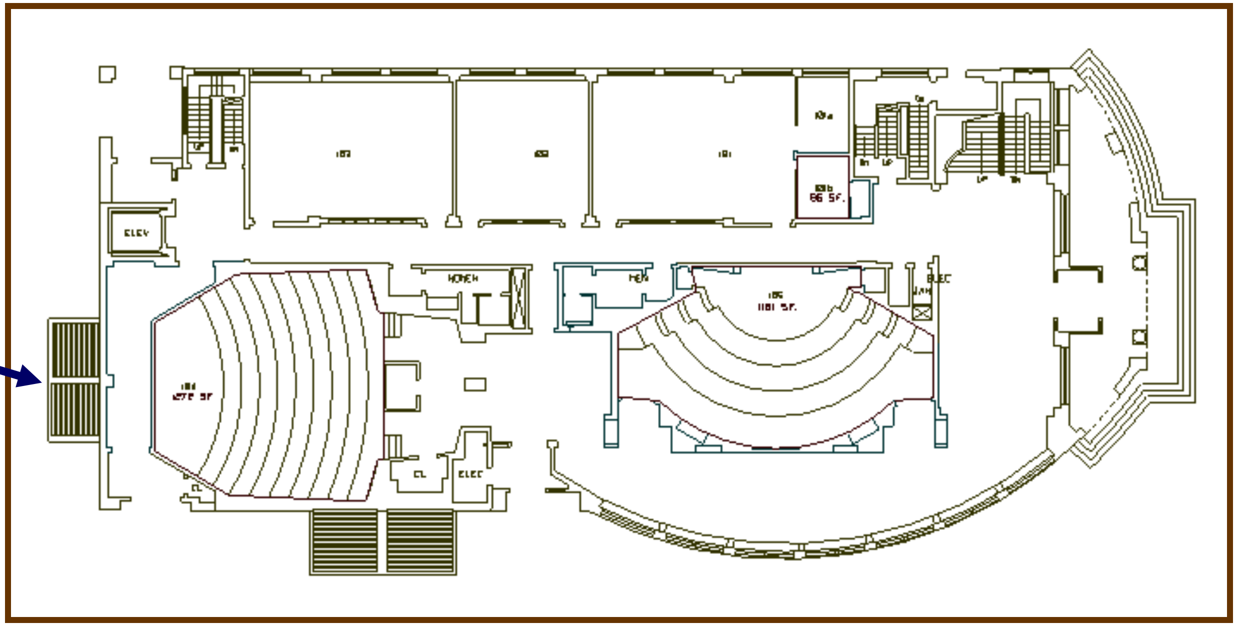


2D Modeling Transformations

Modeling Coordinates



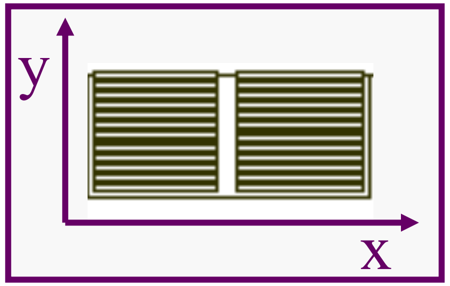
Let's look at this in detail…

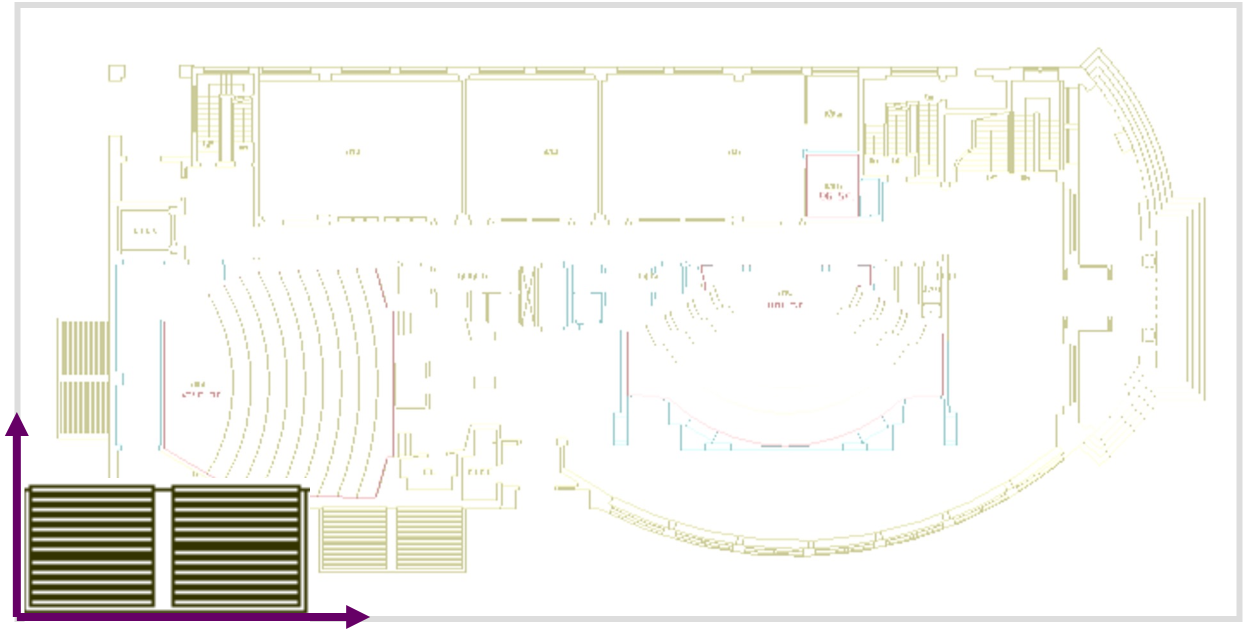


World Coordinates

2D Modeling Transformations

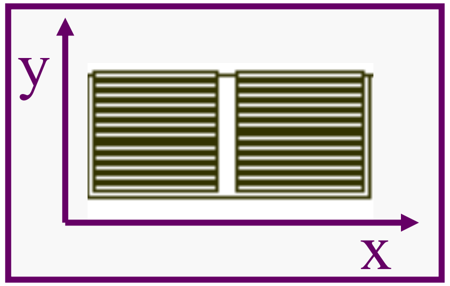
Modeling
Coordinates



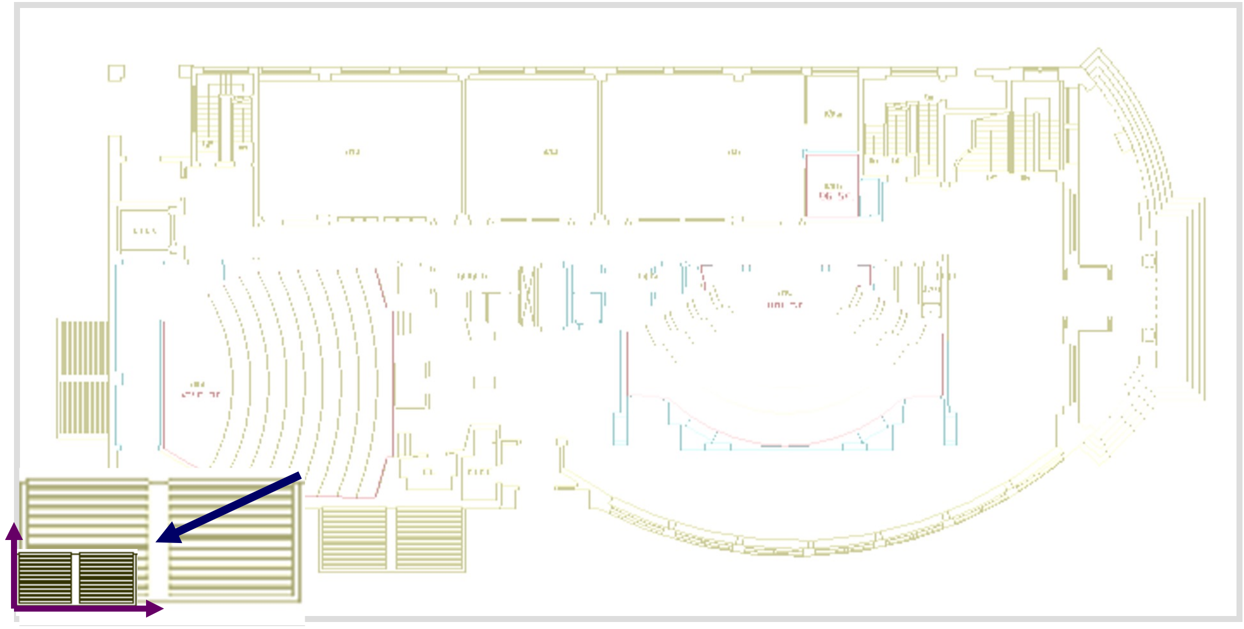


2D Modeling Transformations

Modeling Coordinates

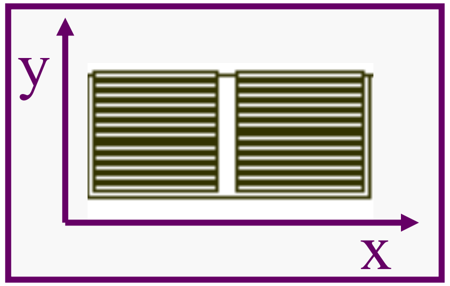


Scale .3, .3

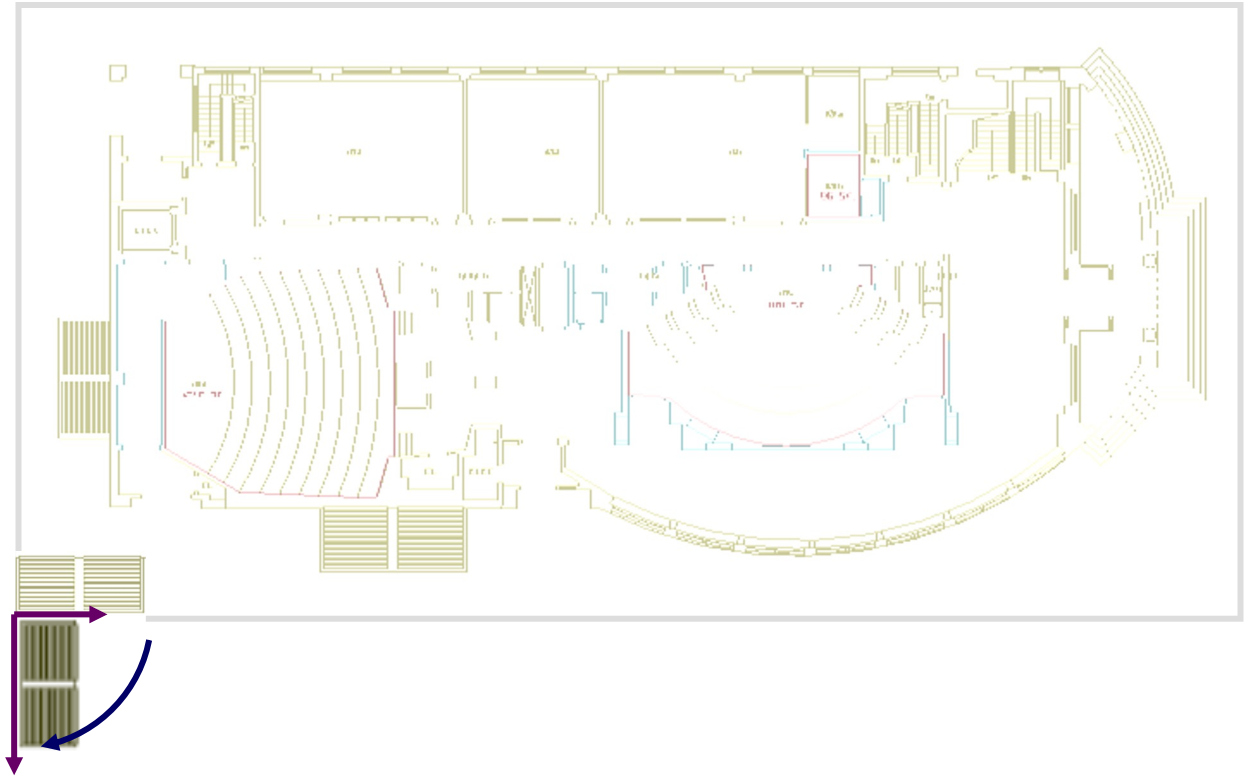


2D Modeling Transformations

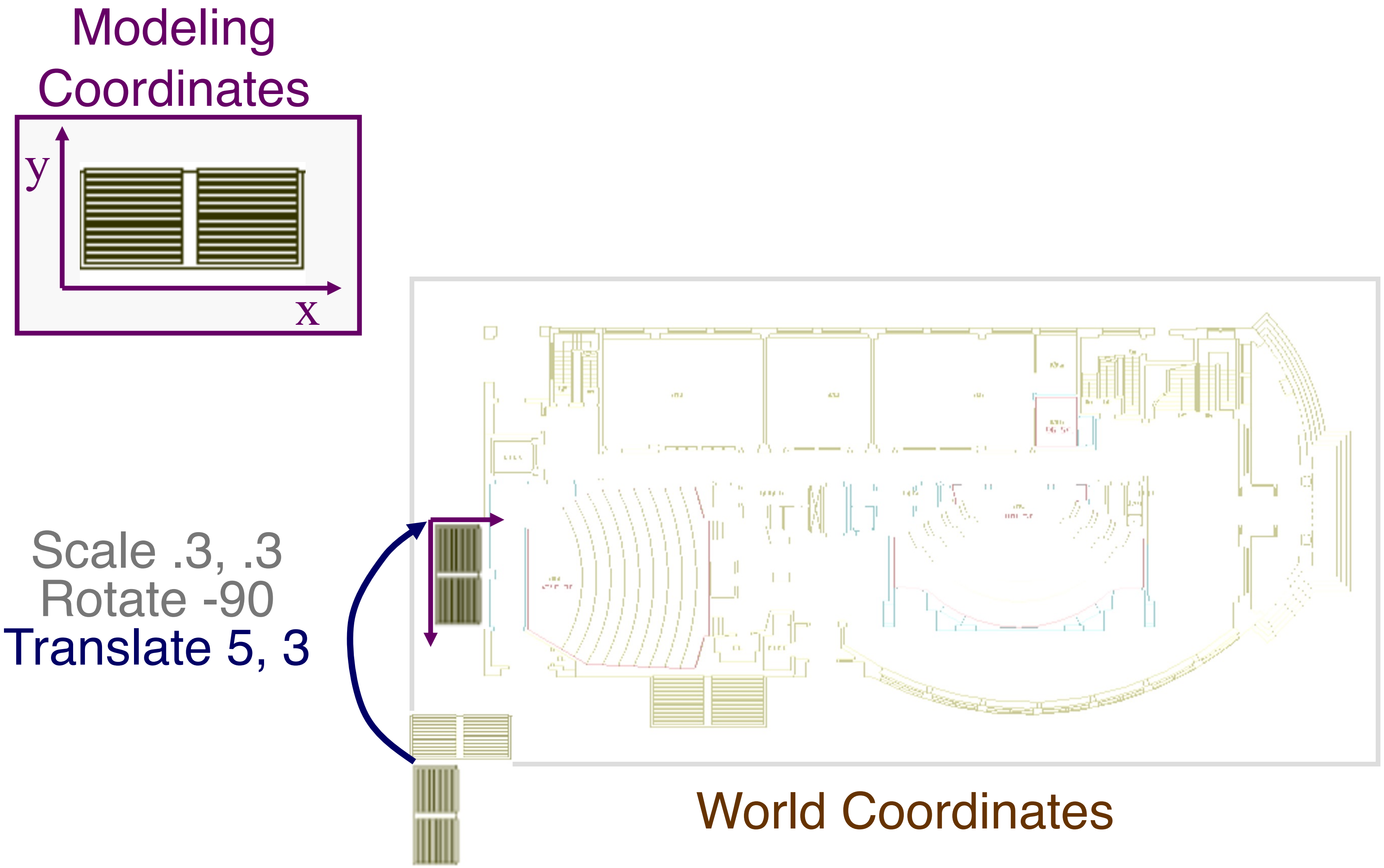
Modeling
Coordinates



Scale .3, .3
Rotate -90

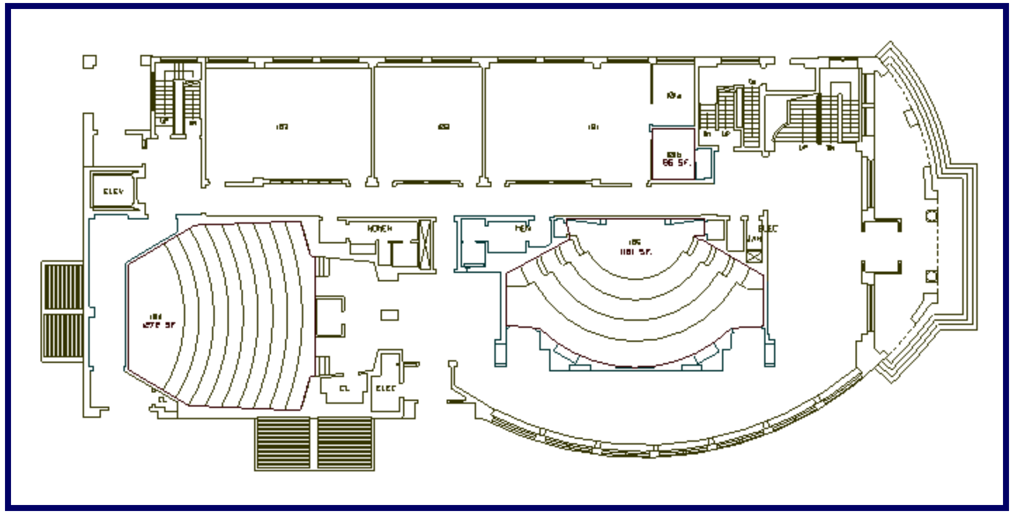


2D Modeling Transformations



Basic 2D Transformations

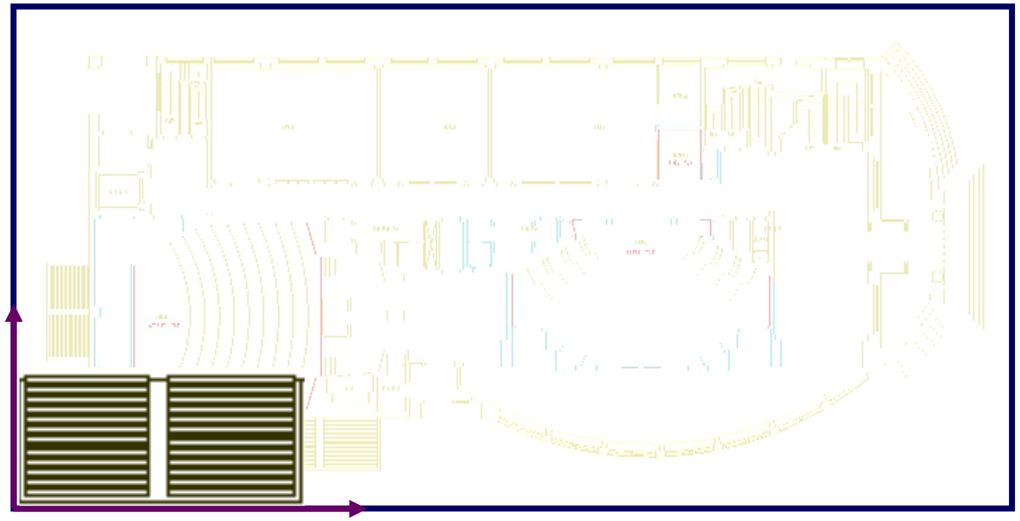
- Translation:
  - $x' = x + tx$
  - $y' = y + ty$
- Scale:
  - $x' = x * sx$
  - $y' = y * sy$
- Shear:
  - $x' = x + hx*y$
  - $y' = y + hy*x$
- Rotation:
  - $x' = x*\cos\Theta - y*\sin\Theta$
  - $y' = x*\sin\Theta + y*\cos\Theta$



Transformations can be combined (with simple algebra)

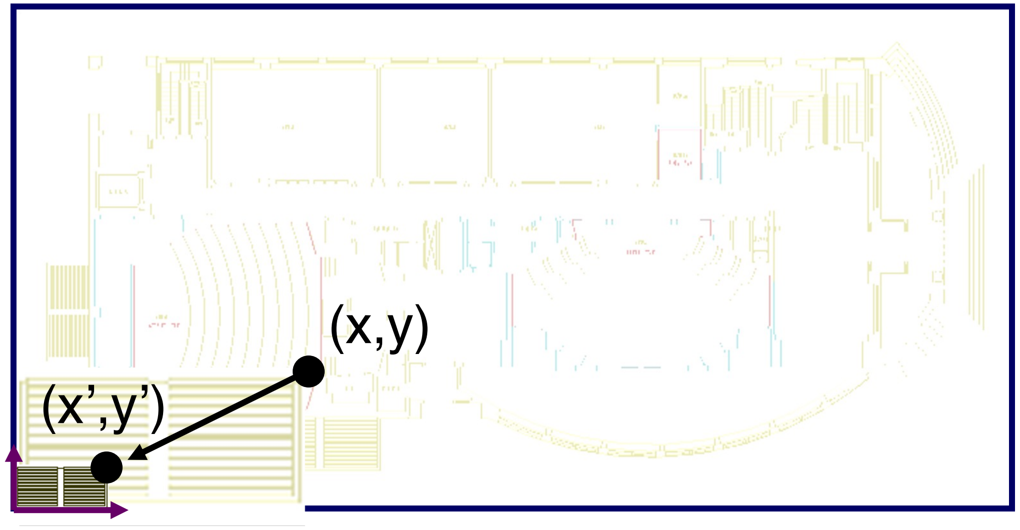
Basic 2D Transformations

- Translation:
  - x' = x + tx
  - y' = y + ty
- Scale:
  - x' = x * sx
  - y' = y * sy
- Shear:
  - x' = x + hx*y
  - y' = y + hy*x
- Rotation:
  - x' = x*cosΘ - y*sinΘ
  - y' = x*sinΘ + y*cosΘ



Basic 2D Transformations

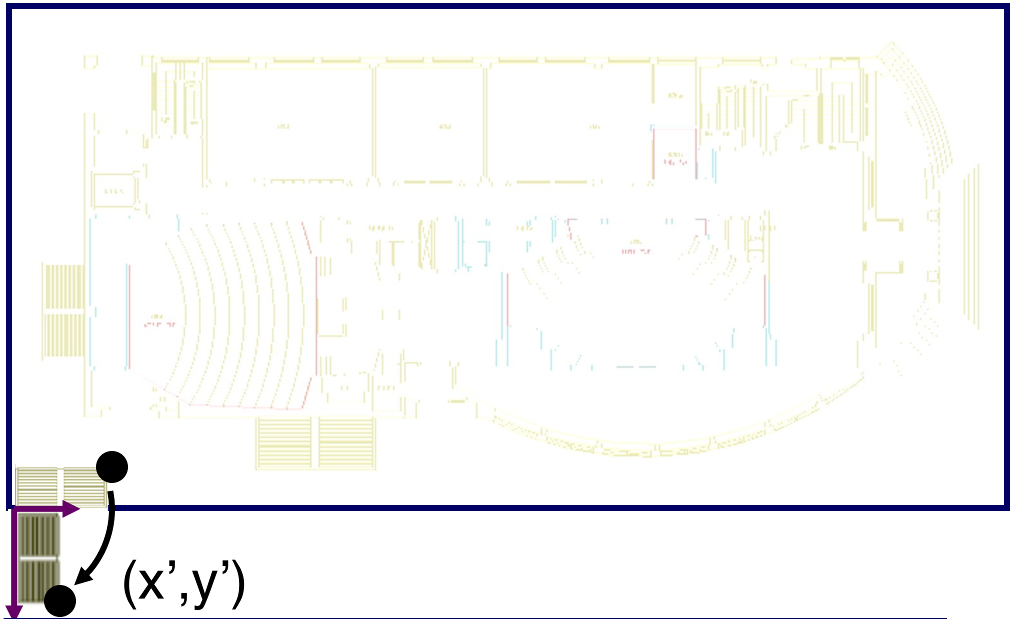
- Translation:
  - x' = x + tx
  - y' = y + ty
- Scale:
  - x' = x * sx
  - y' = y * sy
- Shear:
  - x' = x + hx*y
  - y' = y + hy*x
- Rotation:
  - x' = x*cosΘ - y*sinΘ
  - y' = x*sinΘ + y*cosΘ



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

Basic 2D Transformations

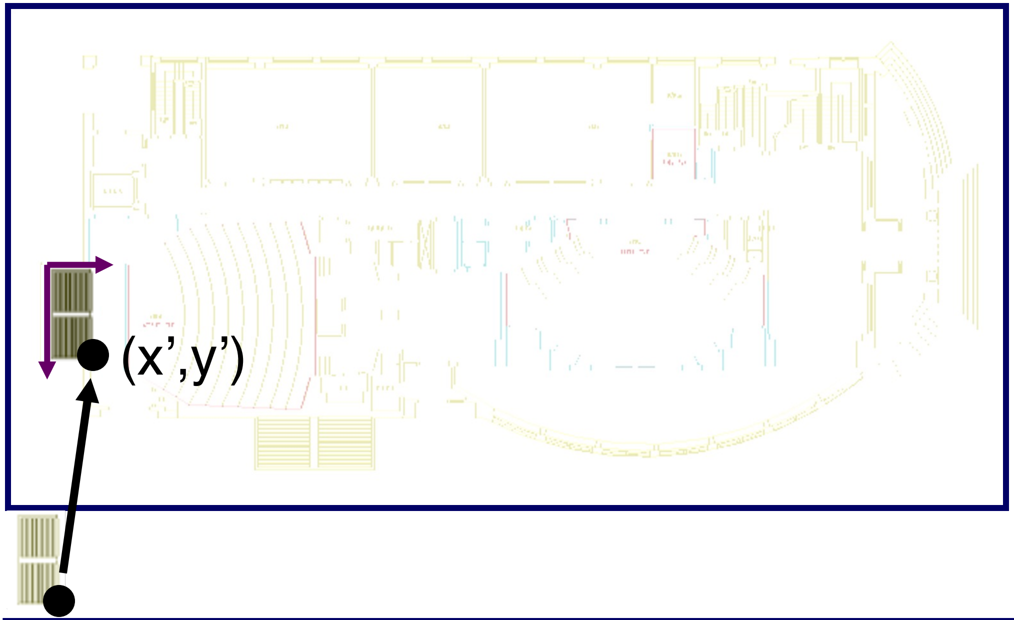
- Translation:
  - x' = x + tx
  - y' = y + ty
- Scale:
  - x' = x * sx
  - y' = y * sy
- Shear:
  - x' = x + hx*y
  - y' = y + hy*x
- Rotation:
  - x' = x*cosΘ - y*sinΘ
  - y' = x*sinΘ + y*cosΘ



x' = (x*sx)*cosΘ – (y*sy)*sinΘ
y' = (x*sx)*sinΘ + (y*sy)*cosΘ

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



$$x' = ((x*sx)*\cos\Theta - (y*sy)*\sin\Theta) + tx$$
$$y' = ((x*sx)*\sin\Theta + (y*sy)*\cos\Theta) + ty$$

Overview

- Scene graphs
  - Geometry & attributes
  - Transformations
  - Bounding volumes
- Transformations
  - Basic 2D transformations
  - Matrix representation
  - Matrix composition
  - 3D transformations

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 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)?

$$x' = sx * x$$
$$y' = sy * y$$

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

2x2 Matrices

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

2D Rotate around (0,0)?

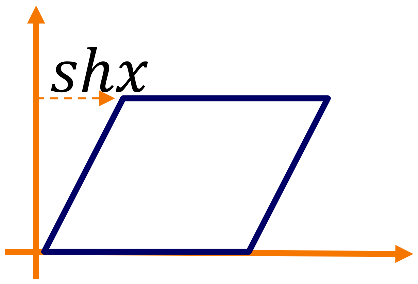
$$x' = \cos\Theta * x - \sin\Theta * y$$
$$y' = \sin\Theta * x + \cos\Theta * y$$

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

2D Shear?

$$x' = x + shx * y$$
$$y' = shy * x + y$$

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



2x2 Matrices

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

2D Mirror over 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?

$$x' = x + tx$$
$$y' = y + ty$$

NO.

Only *linear* 2D transformations can be represented with a 2×2 matrix

Linear Transformations

- 2D 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: $T(s_1\mathbf{p}_1 + s_2\mathbf{p}_2) = s_1T(\mathbf{p}_1) + s_2T(\mathbf{p}_2)$ and $T(c\mathbf{p}_1) = cT(\mathbf{p}_1)$

Linear Transformations

- Properties of linear transformations:
    - Origin maps to origin
    - Points at infinity stay at infinity
    - Lines map to lines
    - Parallel lines remain parallel
    - Ratios are preserved
    - Closed under composition

Now, lets model 2D Translation

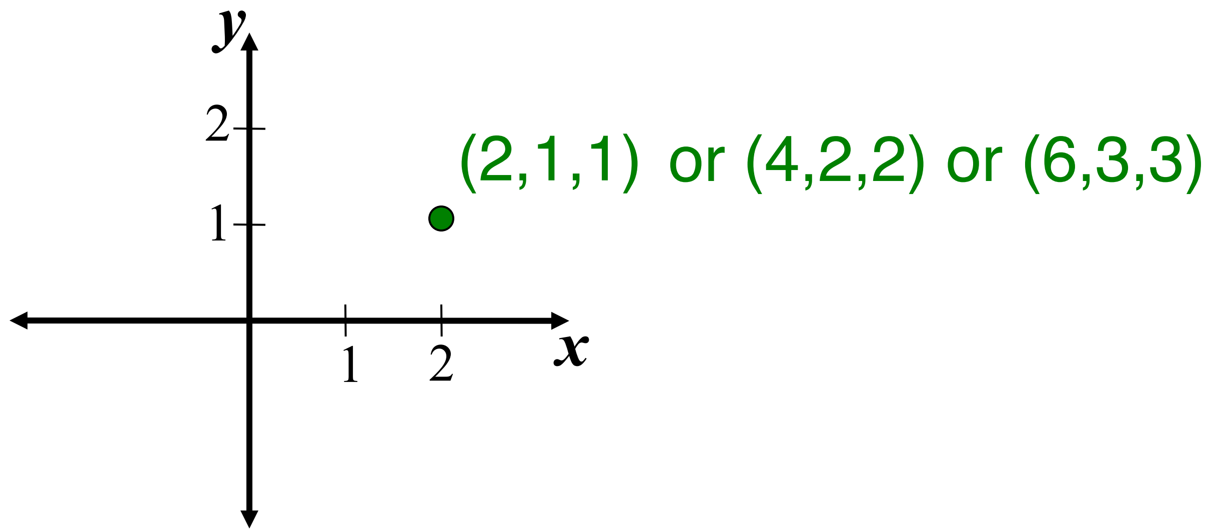
- 2D translation represented by a 3x3 matrix
  - Point represented with *homogeneous coordinates*

$$x' = x + tx$$
$$y' = y + ty$$

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

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
  - (x, 0, 0) and (0, y, 0) are not allowed



Convenient coordinate system to
represent many useful transformations

Basic 2D Transformations

- Basic 2D transformations as 3x3 matrices

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

Translate

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

Scale

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

Rotate

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

Shear

Affine Transformations

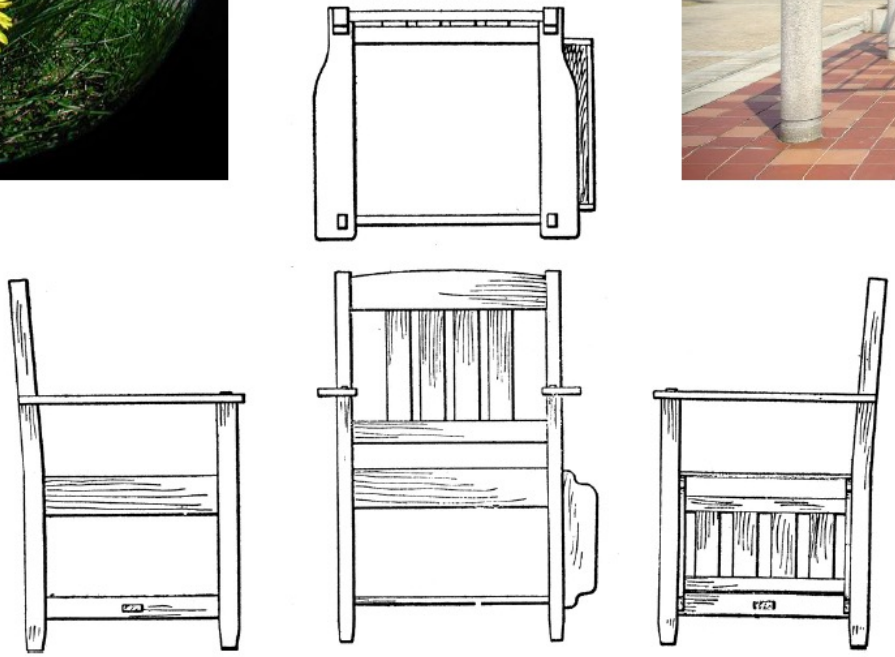
- Affine transformations are combinations of …
  - Linear transformations, and
  - Translations

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

- Properties of affine transformations:
  - Origin does not necessarily map to origin
  - Points at infinity remain at infinity
  - Lines map to lines
  - Parallel lines remain parallel
  - Ratios are preserved
  - Closed under composition

Projective Transformations

- The world is in 3D, the screen is flat. How to *Project*?

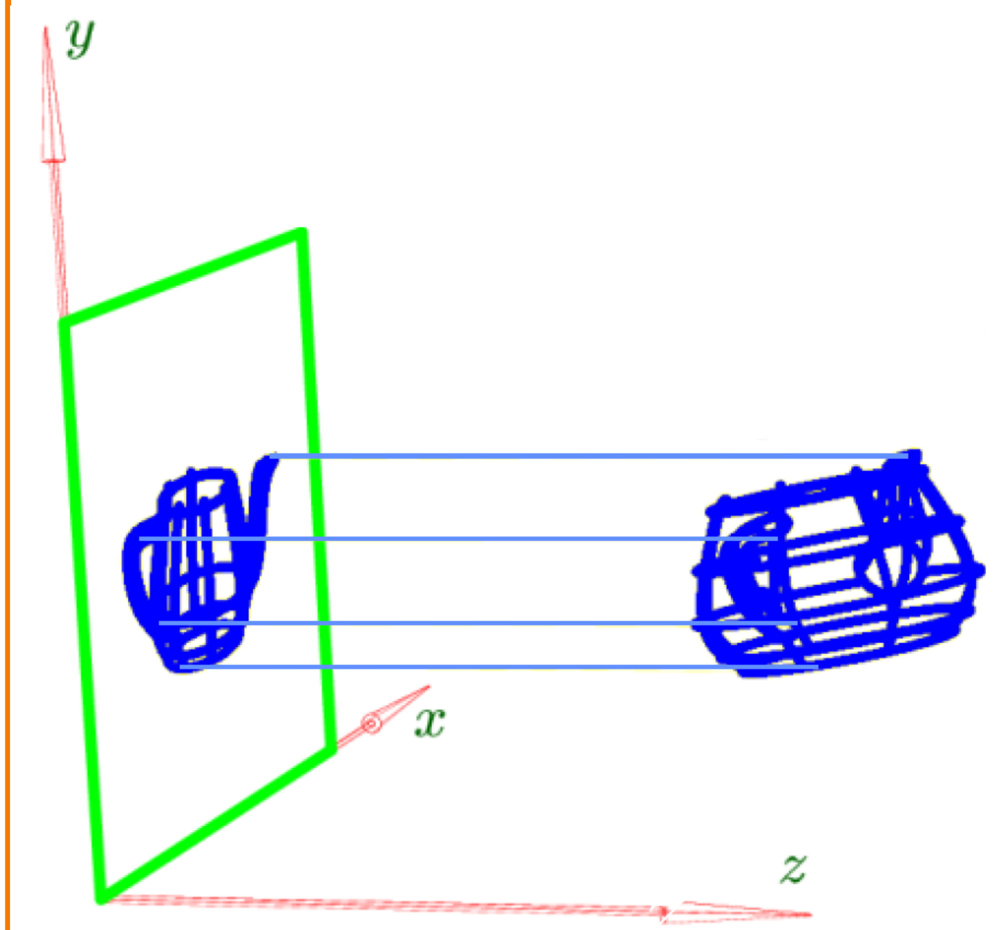


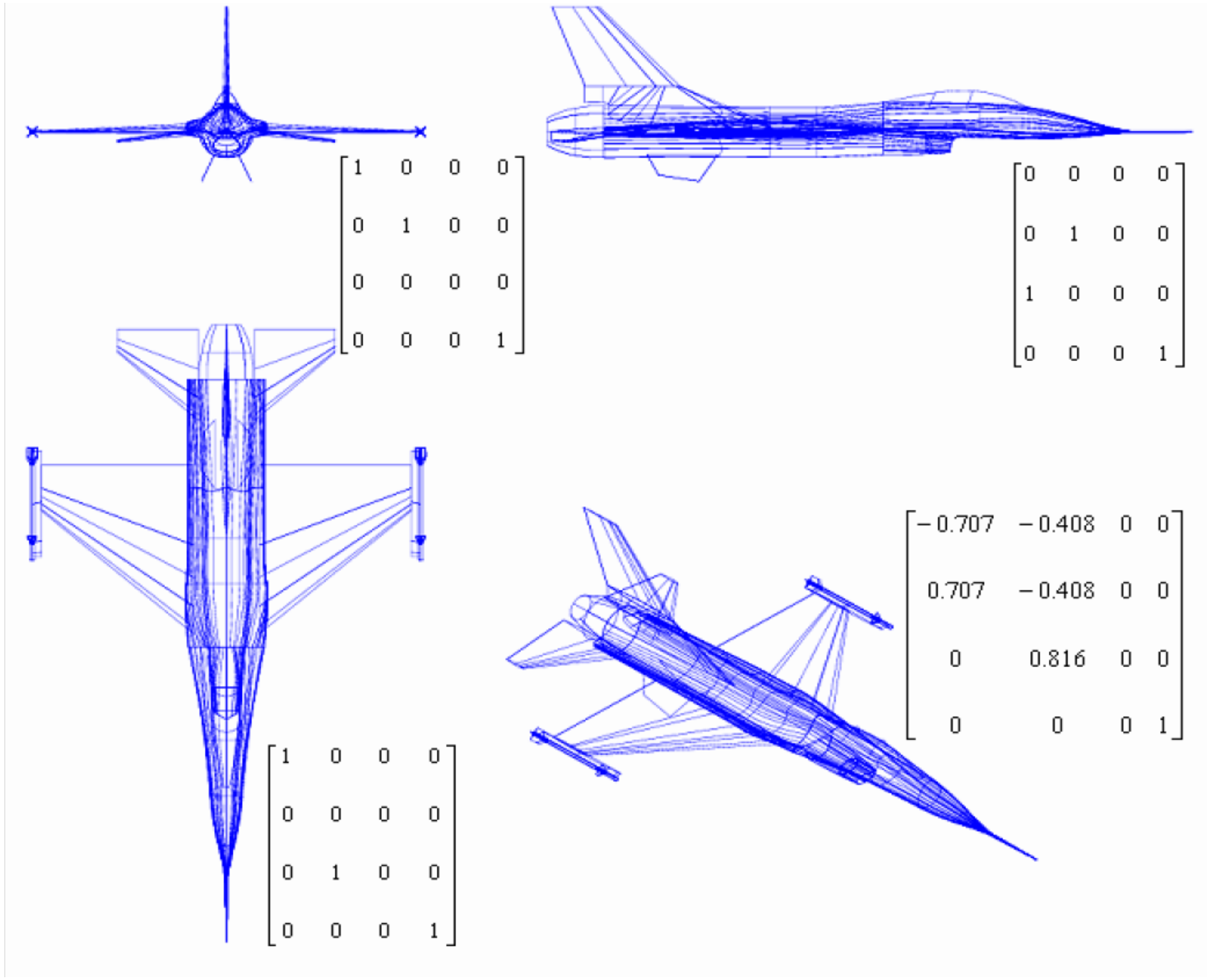
http://www.nikonweb.com/fisheye/
http://etc.usf.edu/clipart/52100/52103/52103_chair_o-p.htm
http://en.wikipedia.org/wiki/File:One_point_perspective.jpg

(Thanks Justin the almighty)

Projective Transformations

- Drop one axis?



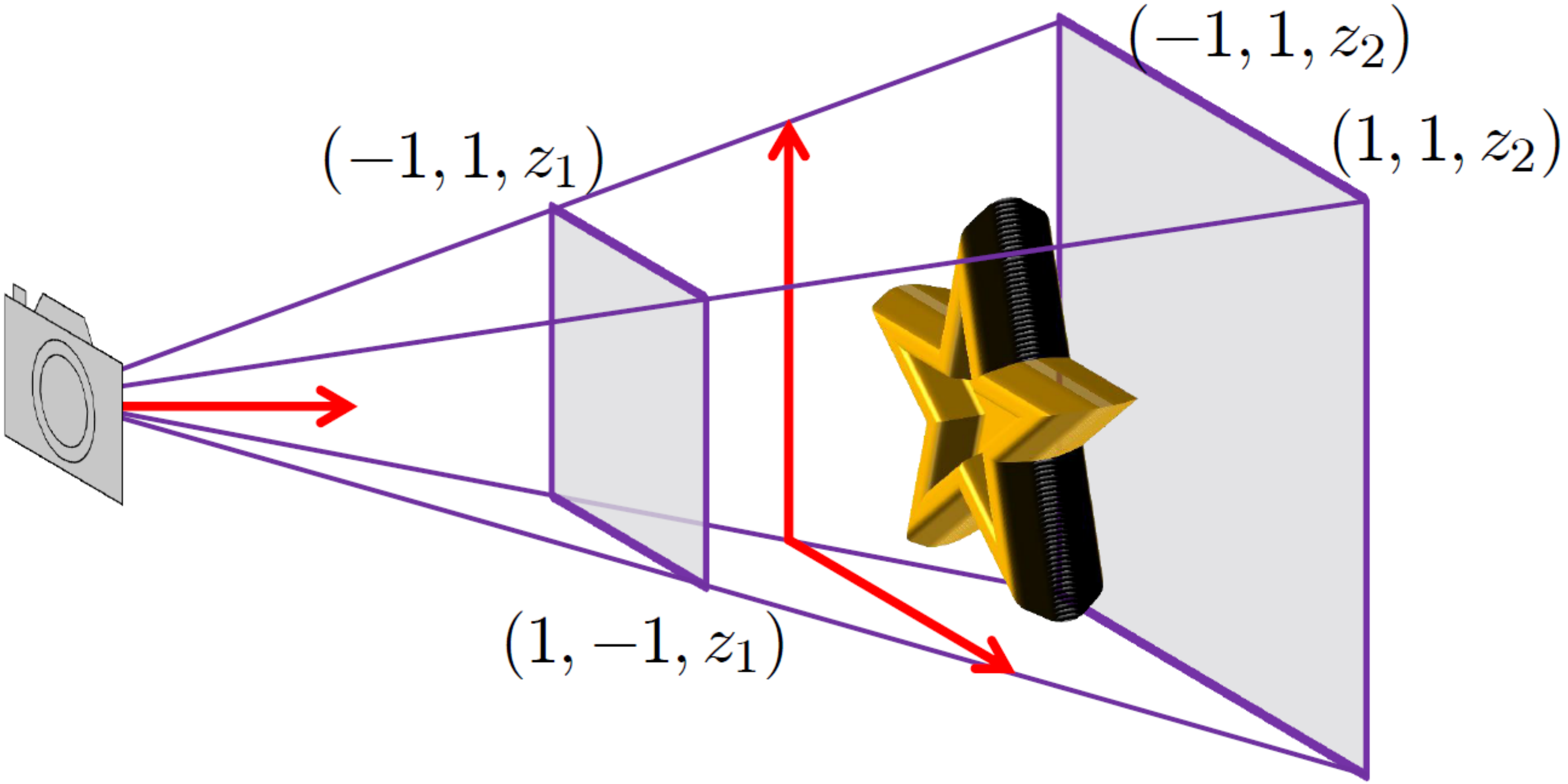


Projective Transformations

- What is wrong with this picture?

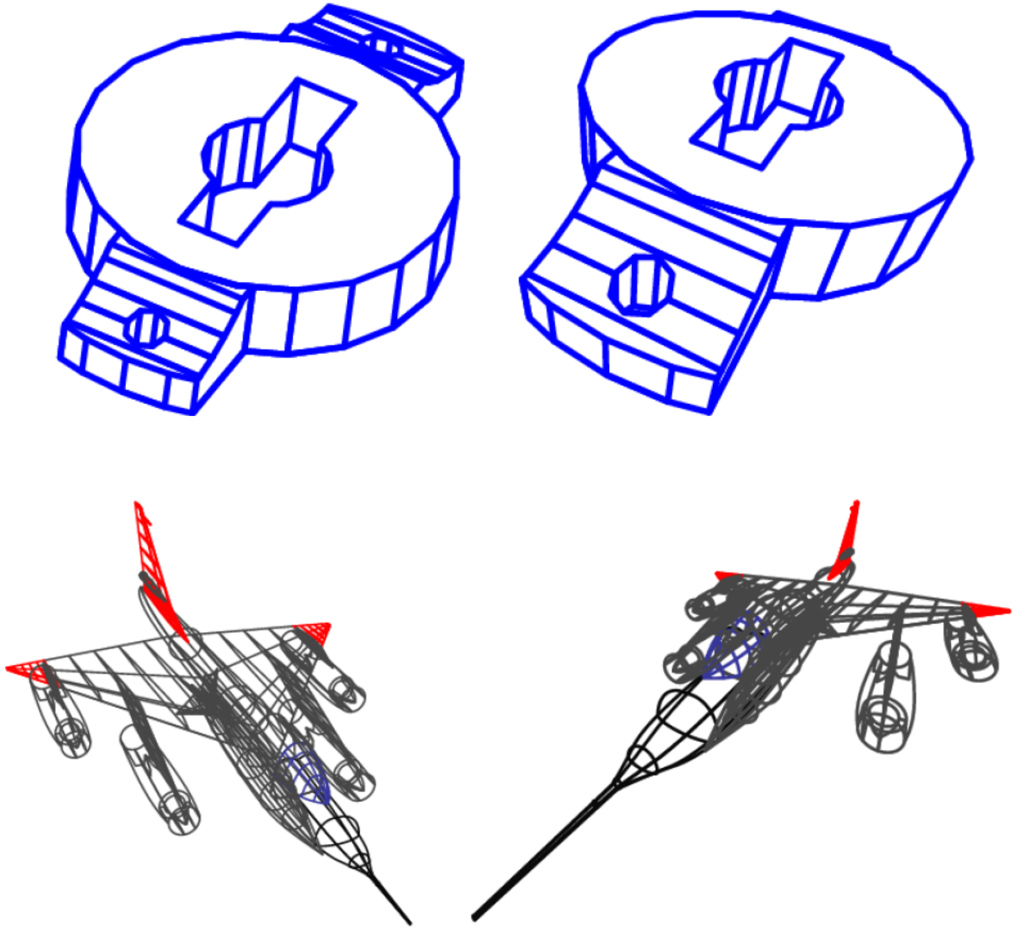
Projective Transformations

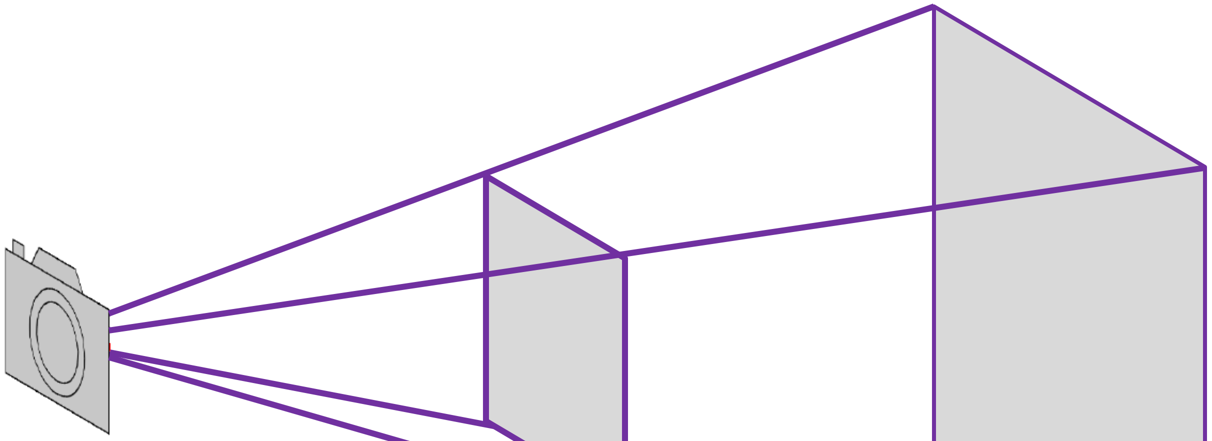
- Pinhole camera model



Projective Transformations

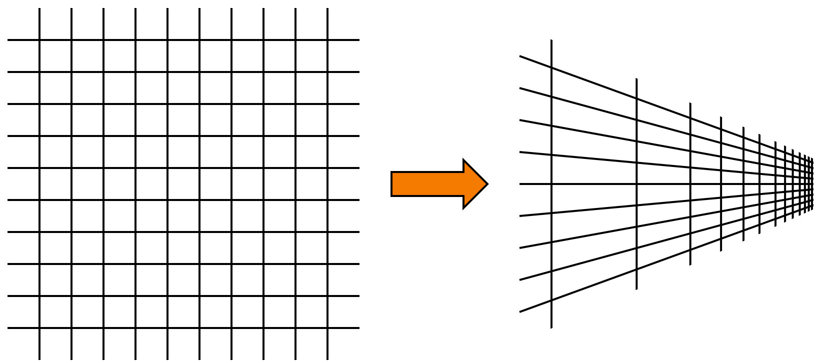
- Perspective Warp!





Projective Transformations

- Will be useful to model (pinhole) cameras:
  can represent camera projection in same framework as modeling transformations



Projective Transformations

- Projective transformations (homographies):
  - 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
  - Point at infinity may map to finite point
  - Lines map to lines
  - Parallel lines do not necessarily remain parallel
  - Ratios are not preserved
  - Closed under composition

Projective Transformations

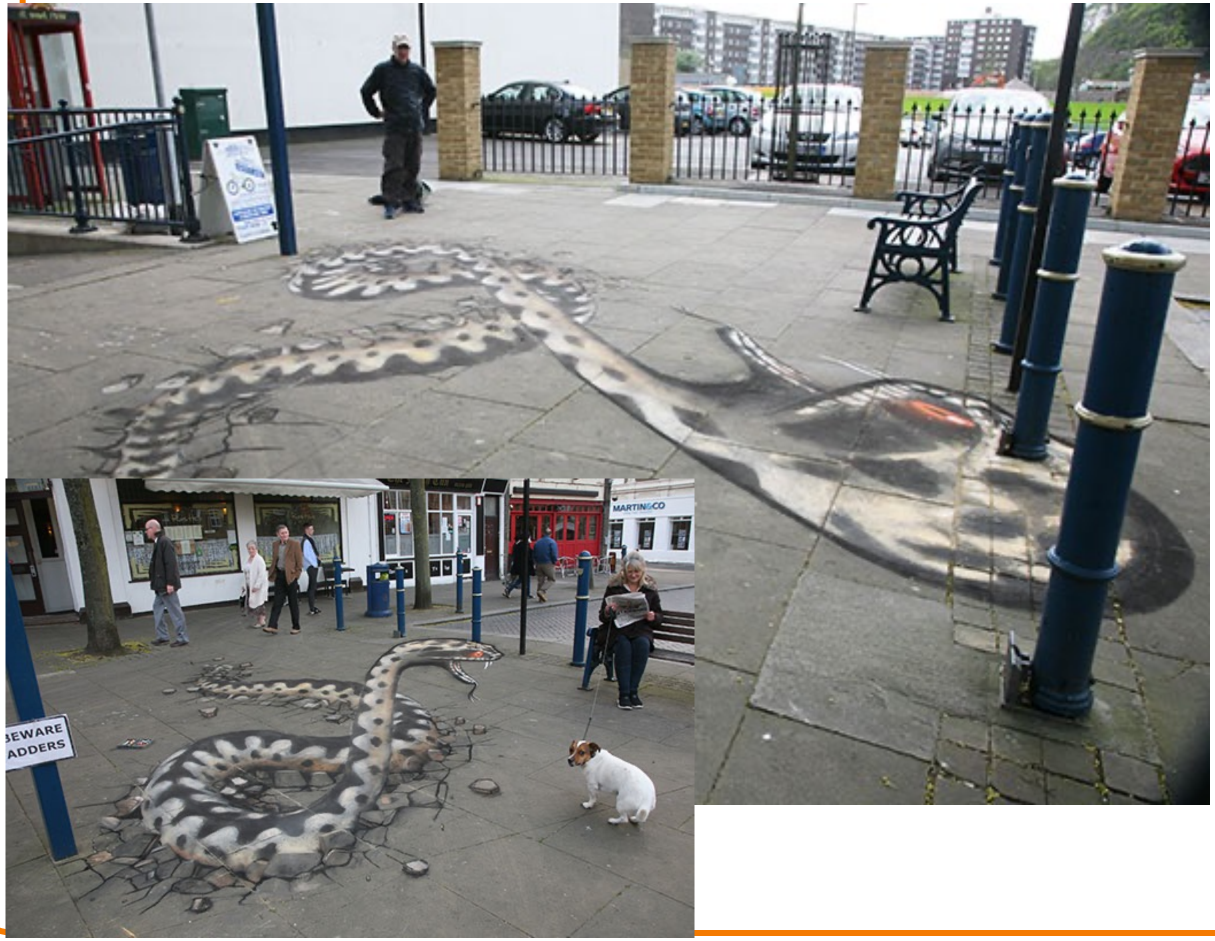
- Perspective warp in art
  - Julian Beever





Projective Transformations

- Perspective warp in art
  - Julian Beever



Overview

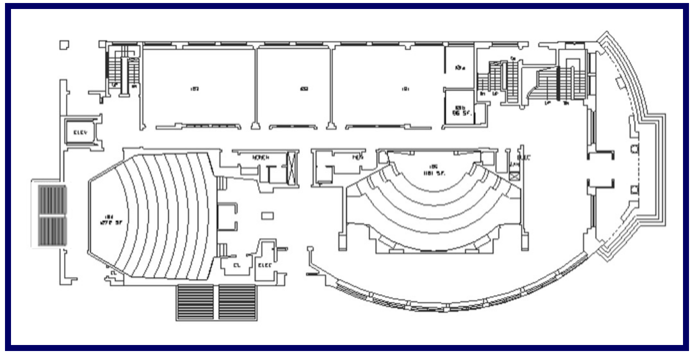
- Scene graphs
  - Geometry & attributes
  - Transformations
  - Bounding volumes
- Transformations
  - Basic 2D transformations
  - Matrix representation
  - Matrix composition
  - 3D transformations

Matrix Composition

- Transformations can be combined by matrix multiplication

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

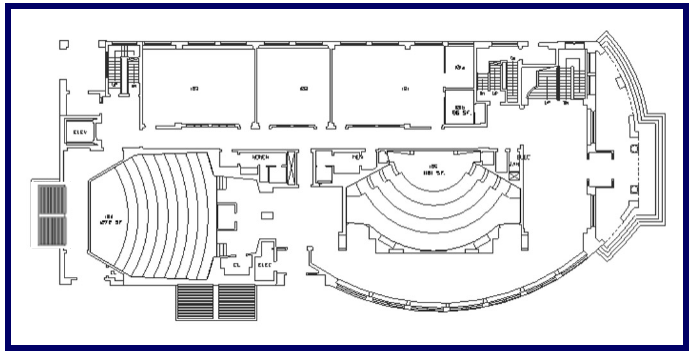
p' = T(tx,ty) R(Θ) S(sx,sy) **p**



Matrix Composition

- Matrices are a convenient and efficient way to represent a sequence of transformations
  - General purpose representation
  - Hardware matrix multiply
  - Efficiency with premultiplication
    - » Matrix multiplication is associative

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

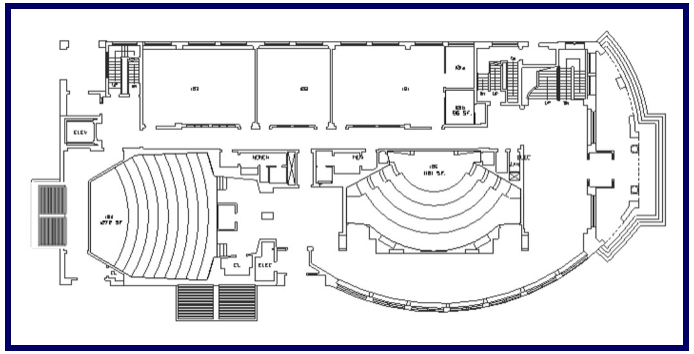


Matrix Composition

- Be aware: order of transformations matters
  - » Matrix multiplication is **not** commutative

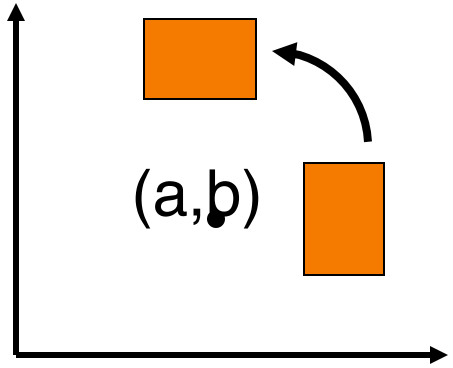
$$\mathbf{p}' = T * R * S * \mathbf{p}$$

"Global" ⟷ "Local"

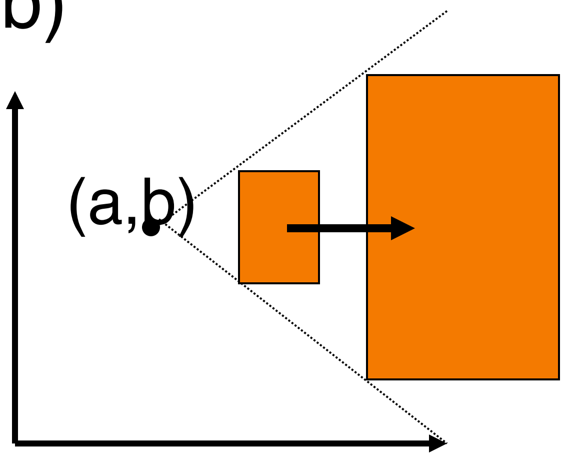


Matrix Composition

- Rotate by Θ around arbitrary point (a,b)



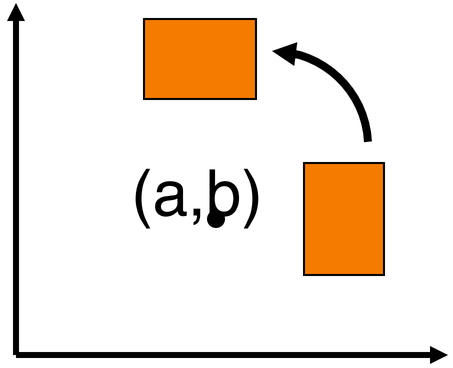
- Scale by sx,sy around arbitrary point (a,b)



Matrix Composition

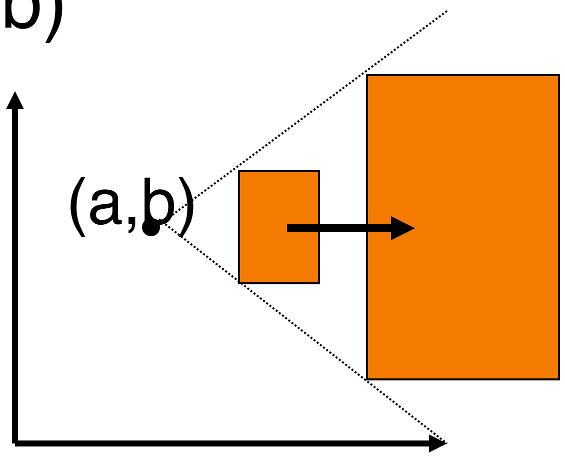
- Rotate by Θ around arbitrary point (a,b)
  - $M=T(a,b) * R(\Theta) * T(-a,-b)$

The trick:
First, translate (a,b) to the origin.
Next, do the rotation about origin.
Finally, translate back.



- Scale by sx,sy around arbitrary point (a,b)
  - $M=T(a,b) * S(sx,sy) * T(-a,-b)$

(Use the same trick.)



Overview

- Scene graphs
  - Geometry & attributes
  - Transformations
  - Bounding volumes
- Transformations
  - Basic 2D transformations
  - Matrix representation
  - Matrix composition
  - 3D transformations

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

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

Scale

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

Translation

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

Mirror over X axis

Rotations become more tricky

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:

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

Helpful hint:

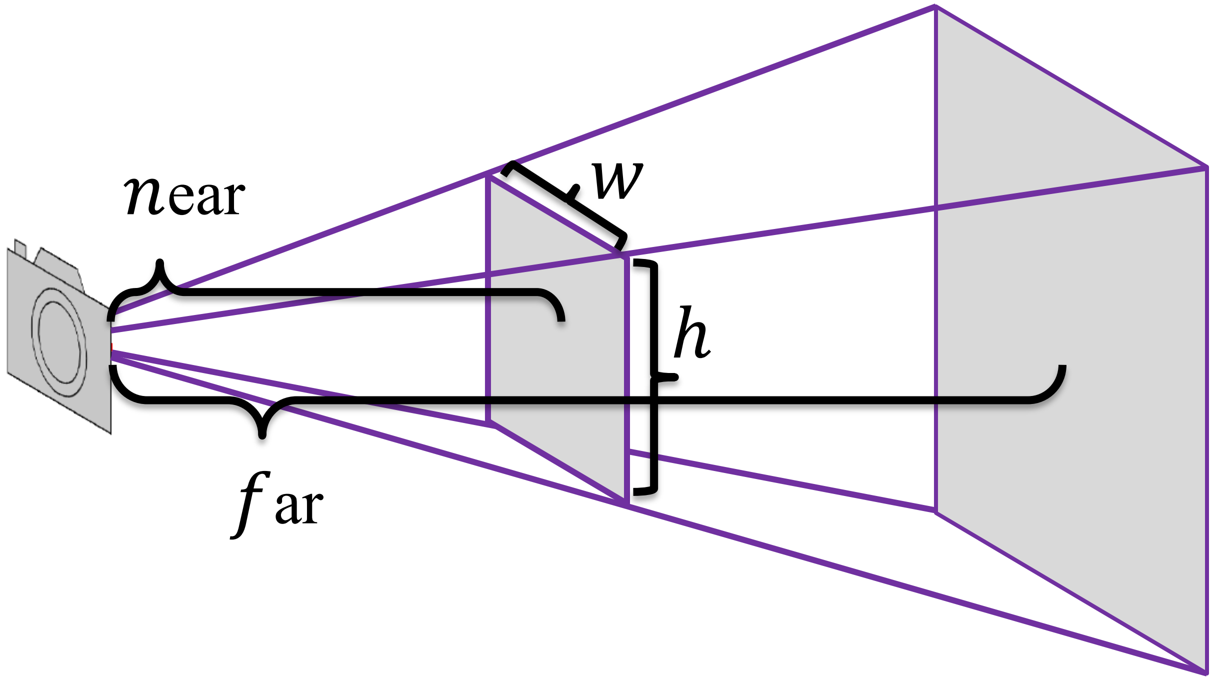
Never try to write out the full matrix for a general 3D rotation. Always implement the simple rotations and multiply the matrices. Otherwise, you're guaranteed to get it wrong.

You're welcome.

Projective Transformations

- OpenGL's version

$$\begin{pmatrix} \frac{2n}{w} & 0 & 0 & 0 \\ 0 & \frac{2n}{h} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

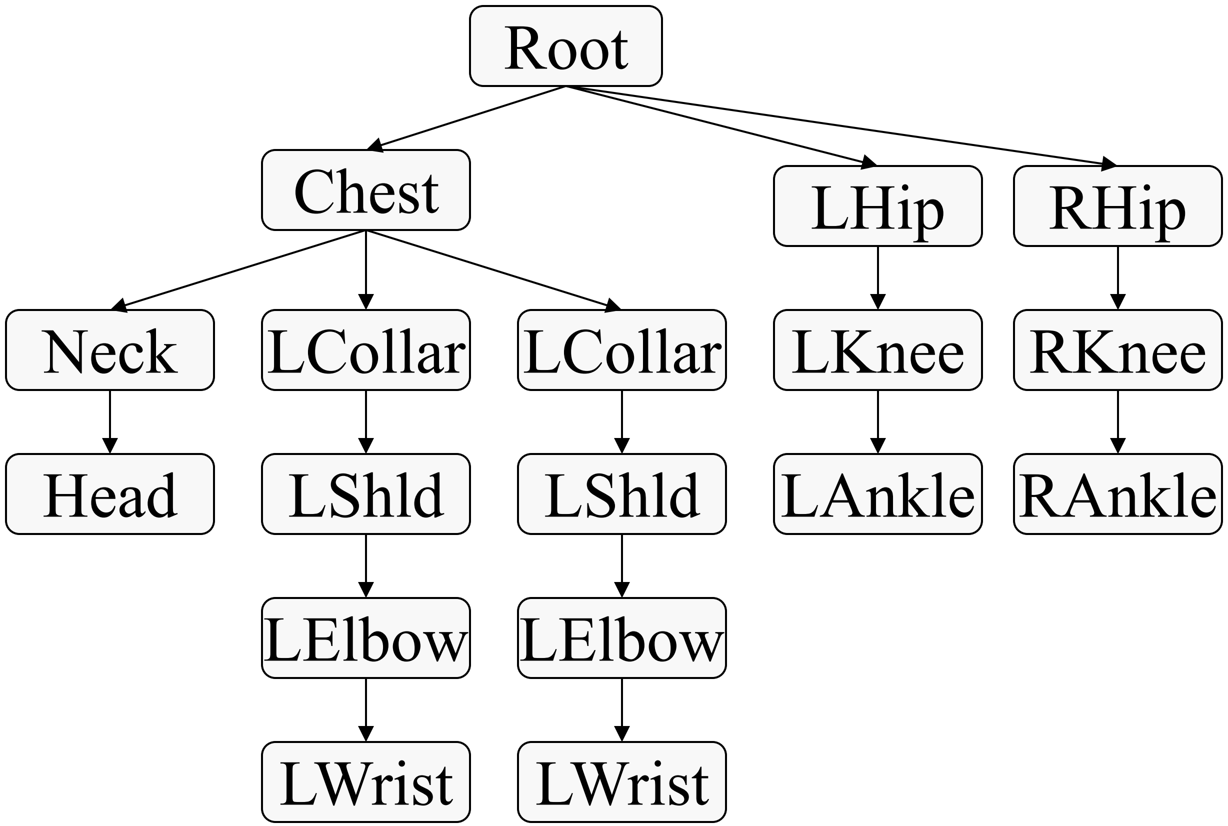


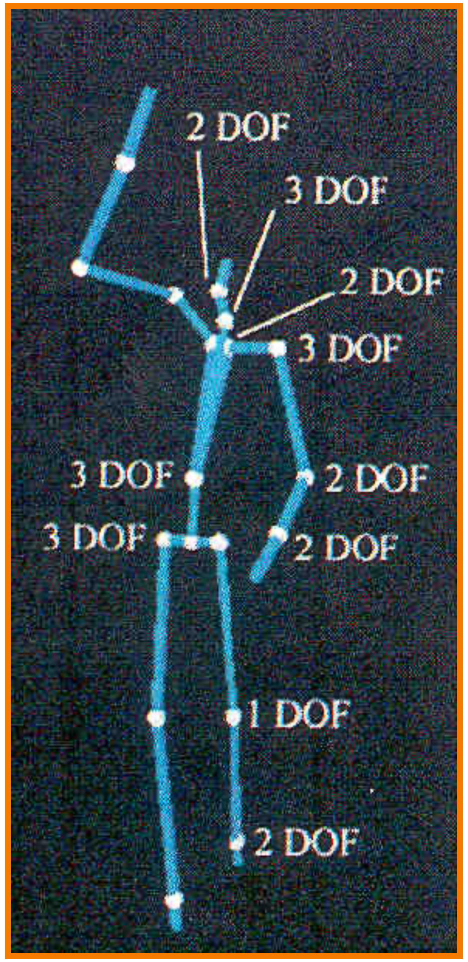
Demo: http://www.songho.ca/opengl/gl_transform.html

+

http://www.songho.ca/opengl/gl_projectionmatrix.html

Transformations in Scene Graphs





Rose et al. `96

Summary

- Scene graphs
  - Hierarchical
  - Modeling transformations
  - Bounding volumes
- Coordinate systems
  - World coordinates
  - Modeling coordinates
- 3D modeling transformations
  - Represent most transformations by 4x4 matrices
  - Composite with matrix multiplication (order matters)