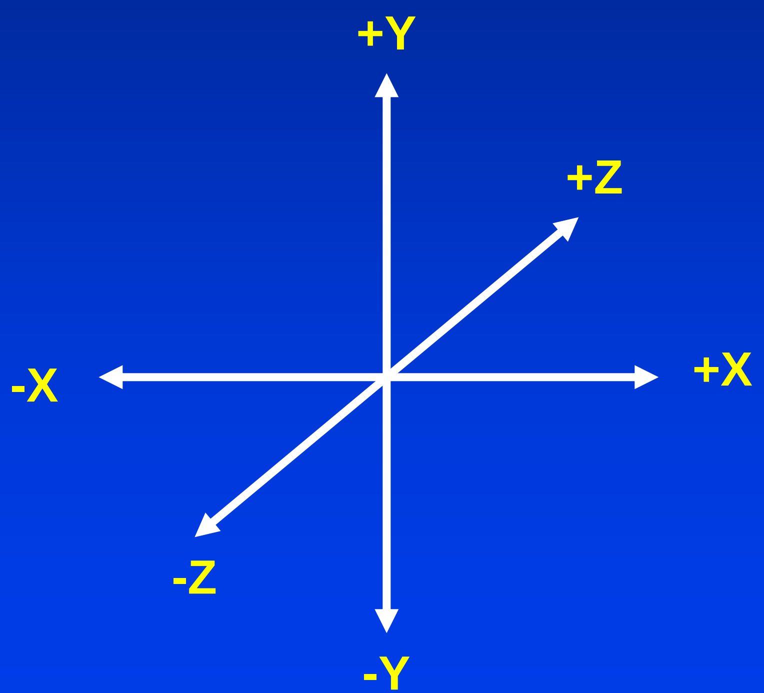
CSE328: Fundamentals of Computer Graphics

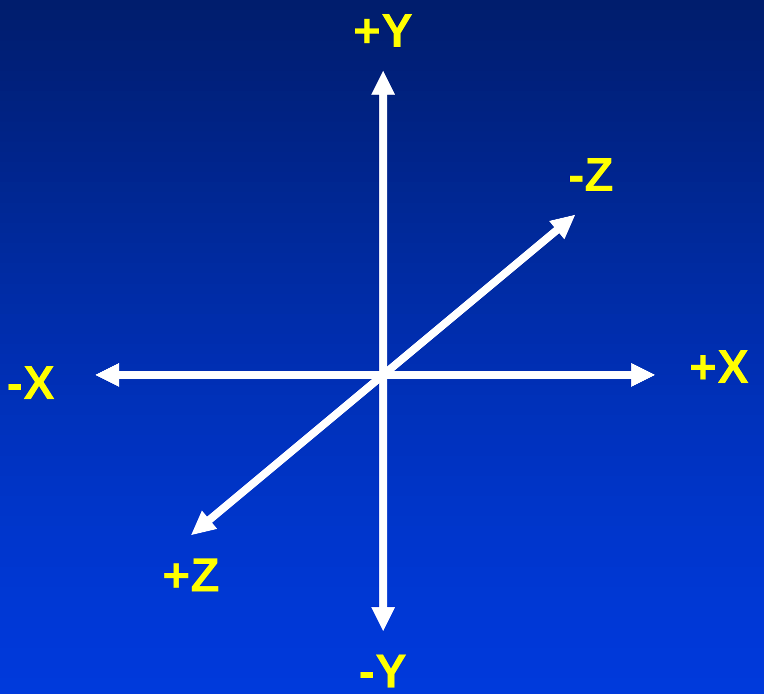
Hong Qin
Department of Computer Science
State University of New York at Stony Brook (Stony Brook University)
Stony Brook, New York 11794-2424
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.stonybrook.edu; or qin@cs.sunysb.edu
http://www.cs.sunysb.edu/~qin

Transformation and Viewing

Cartesian Coordinate System

Left-handed





Right-handed

3D Graphics Concepts

- 3D coordinate system
  - x, y, and z values
  - Depth information
- Geometric modeling of various 3D objects
  - Point, line, polygon (residing on a 3D plane)
  - Curve, surface, solid

3D Graphics Concepts

- Geometric transformation
- 3D viewing
  - Parallel projection
  - Perspective projection
- Display methods of 3D objects
  - Wireframe
  - Shaded objects
  - Visible object identification
  - Photo-realistic rendering techniques
  - 3D stereoscopic viewing

Euclidean Space

- Scalar value
- Points: P = (x,y,z)
- Vectors: **V** = [x,y,z]
  - Magnitude or distance $\|V\| = \sqrt{(x^2+y^2+z^2)}$
  - Direction
  - No position
- Position vector
  - Think of as magnitude and distance relative to a point, usually the origin of the coordinate system

Review of Common Vector Operations in 3D

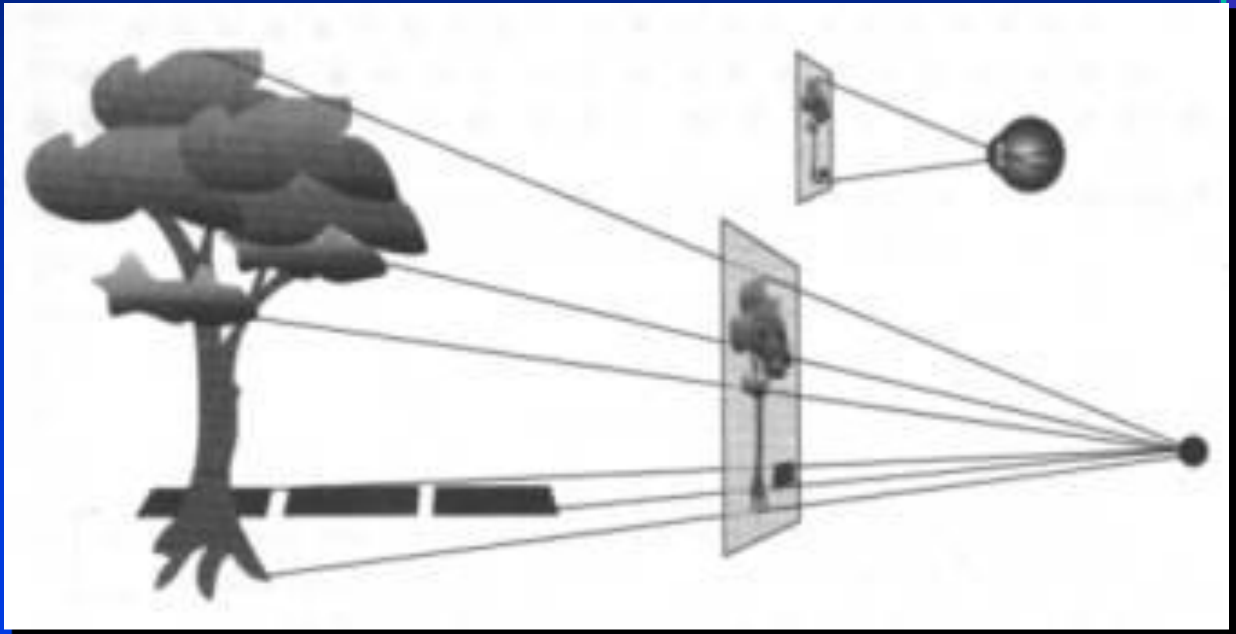
- Addition of vectors
  - $\mathbf{V_1}+\mathbf{V_2} = [x_1,y_1,z_1] + [x_2,y_2,z_2] = [x_1+x_2,\ y_1+y_2,\ z_1+z_2]$
- Multiply a scalar with a vector
  - $s\mathbf{V} = s[x,y,z] = [sx,sy,sz]$
- Dot product
  - $\mathbf{V_1}\bullet\mathbf{V_2} = [x_1,y_1,z_1]\bullet[x_2,y_2,z_2] = [x_1x_2+y_1y_2+z_1z_2]$
  - $\mathbf{V_1}\bullet\mathbf{V_2} = \|V_1\|\ \|v_2\| \cos\beta$ where β is the angle between V_1 and V_2
- Cross product of two vectors
  - $\mathbf{V_1}\times\mathbf{V_2} = [x_1,y_1,z_1]\times[x_2,y_2,z_2] = [y_1z_2\text{-}y_2z_1,\ x_2z_1\text{-}x_1z_2,\ x_1y_2\text{-}x_2y_1]$
    $= -\ V_2\times V_1$
  - Results in a vector that is orthogonal to the plane defined by V_1 and V_2

Perspective Projection

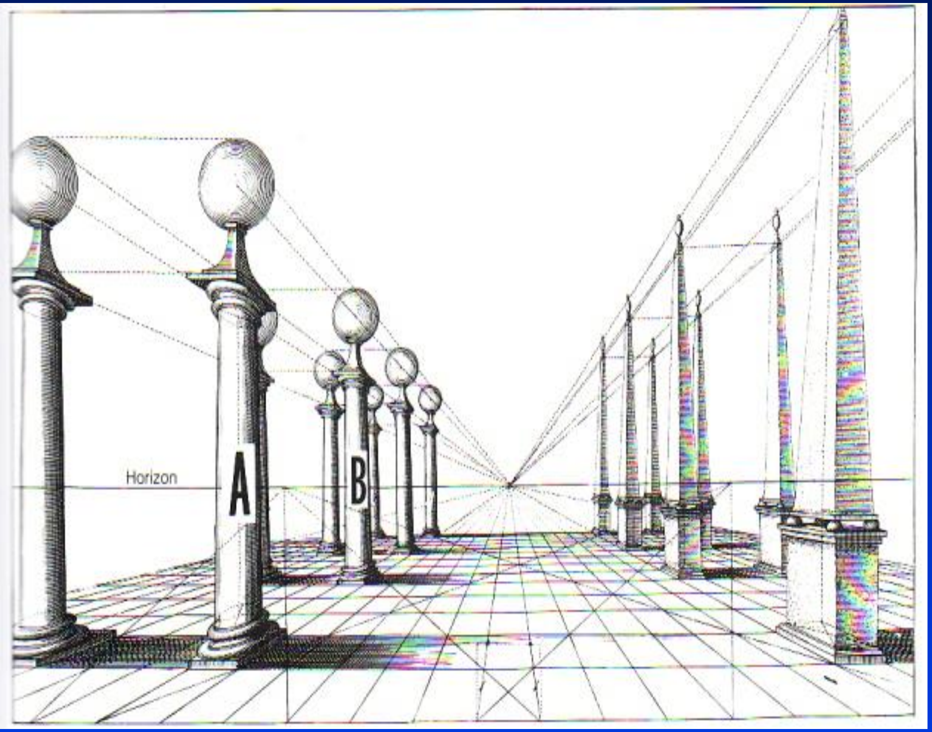
Parallel lines converge

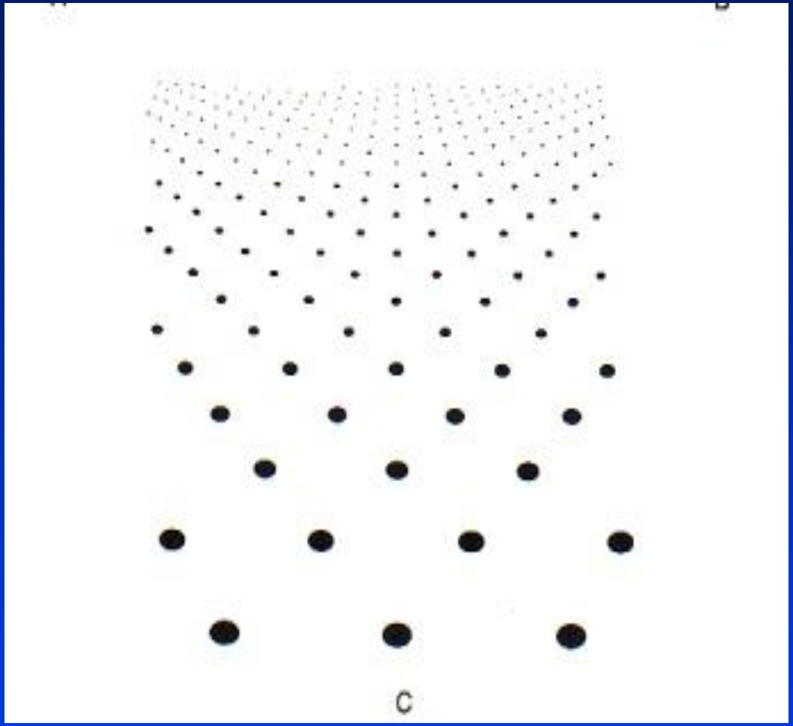
Distant objects appear smaller

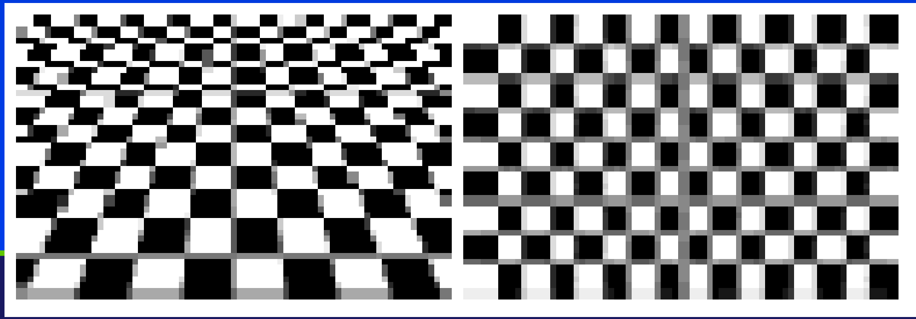
Textured elements become smaller with distance



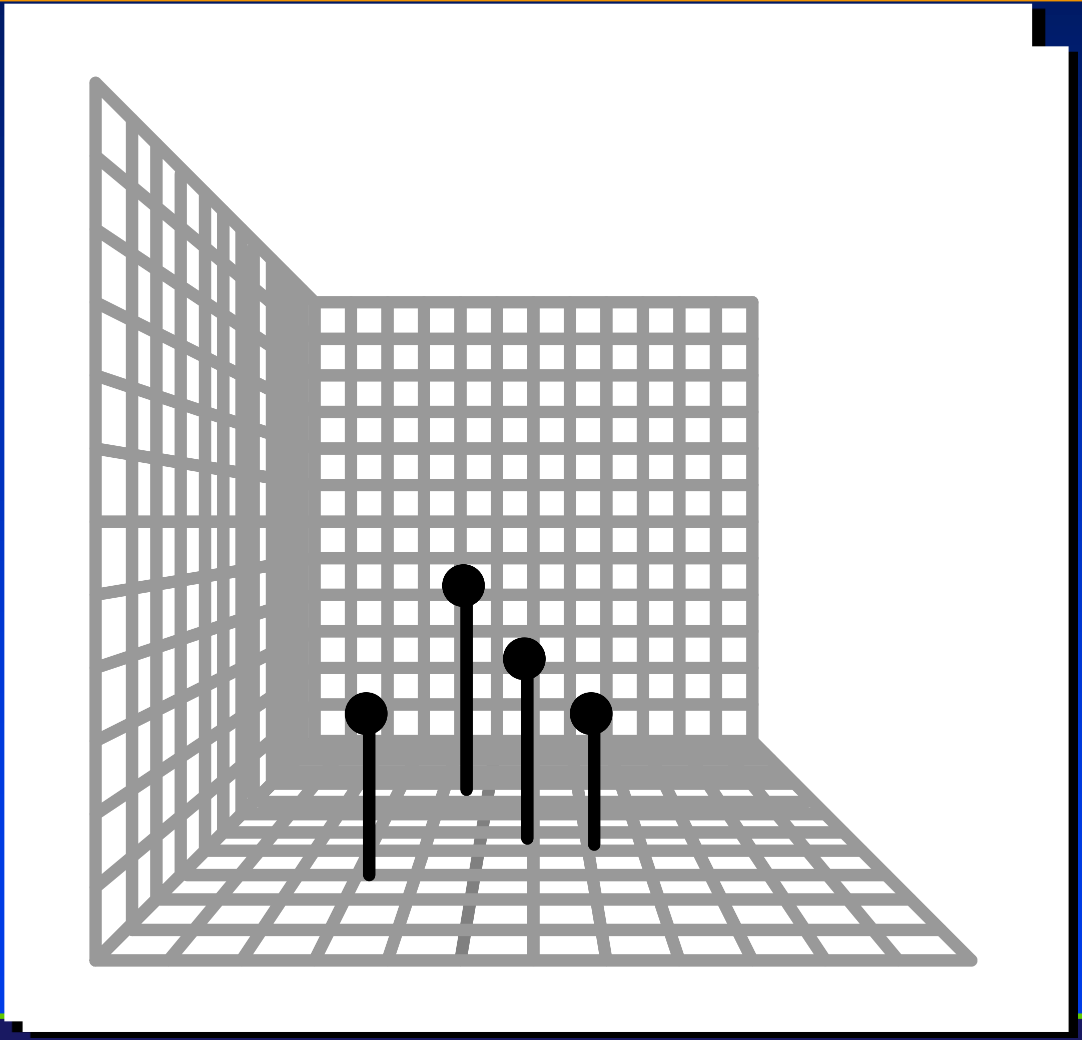
Perspective Cues



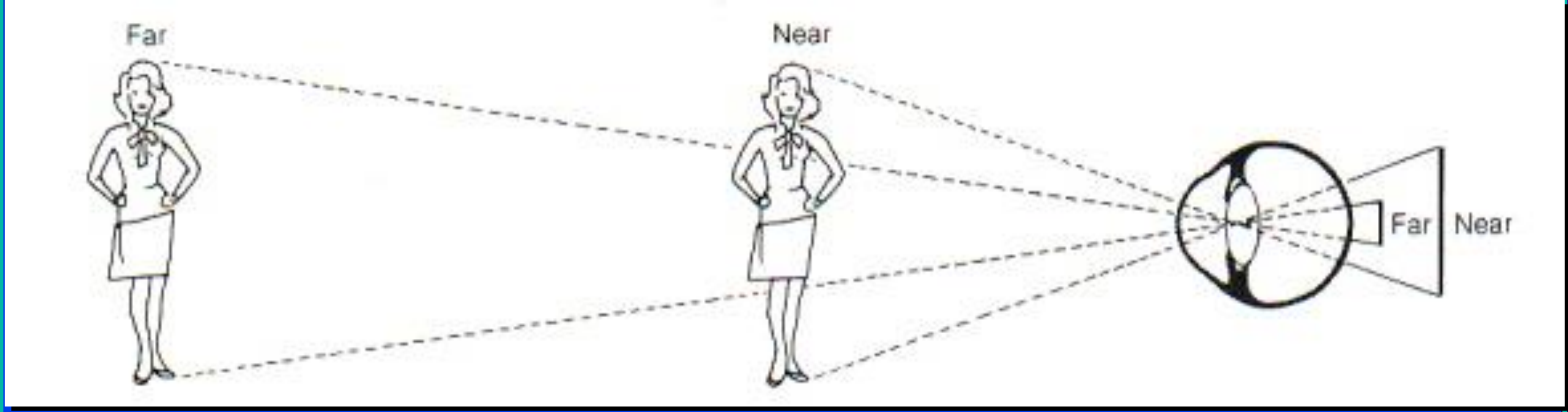




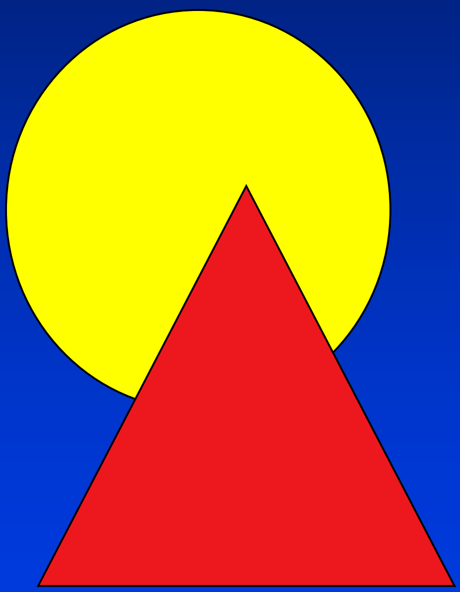
Perspective Cues

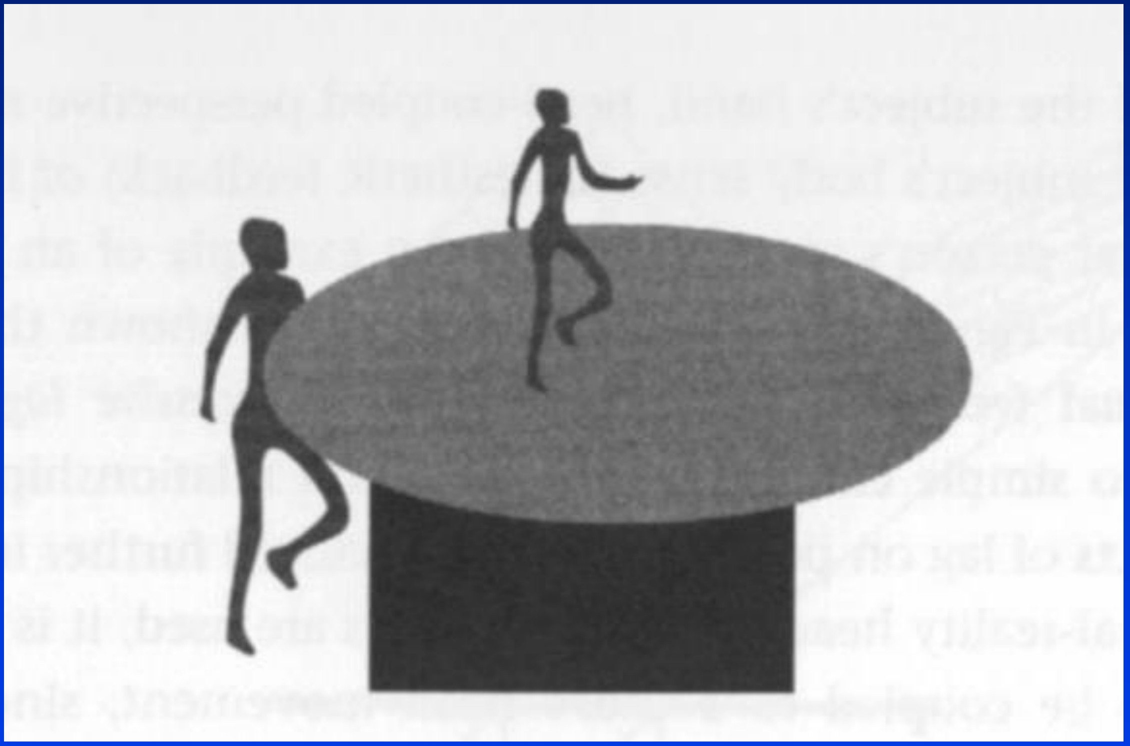


Perspective Cues

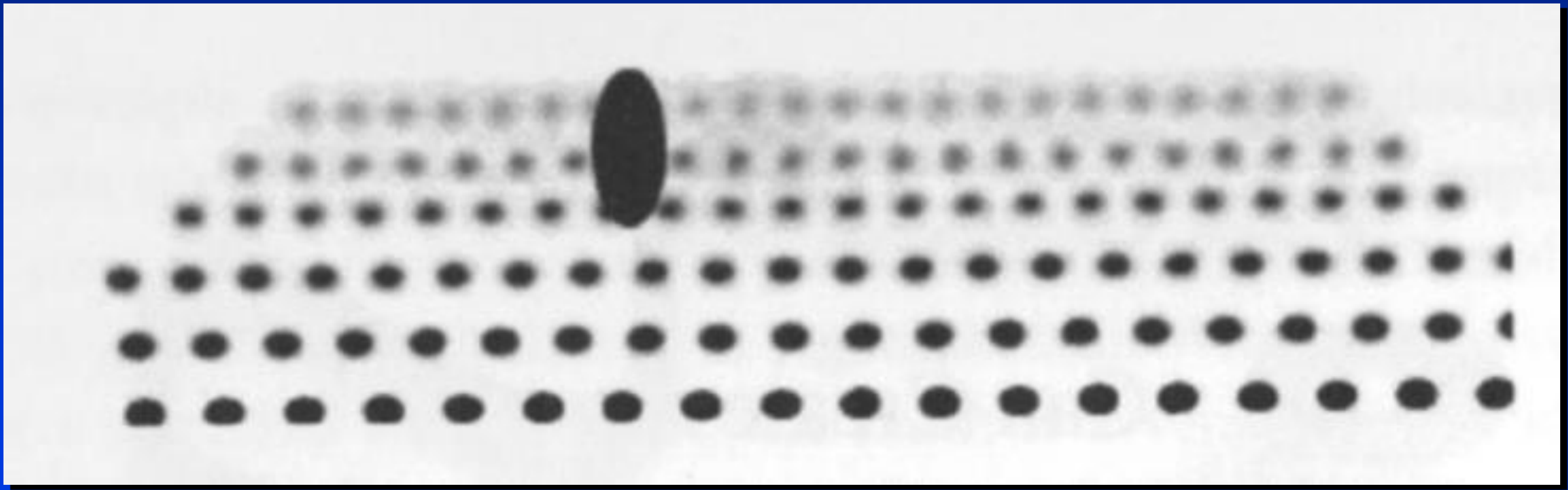


Depth Cue via Occlusion

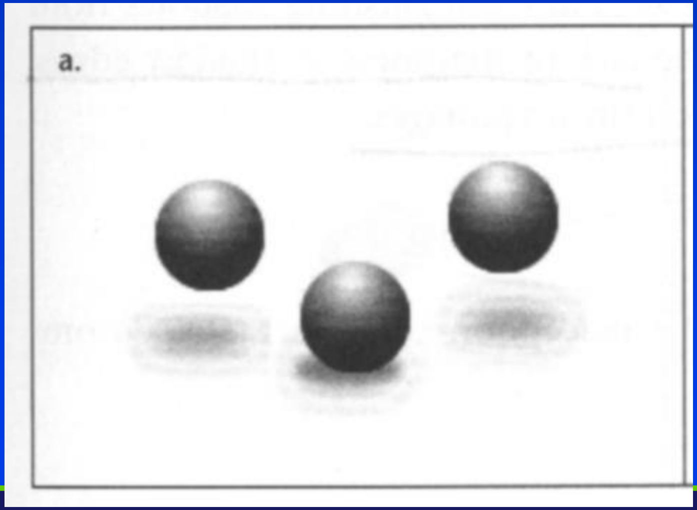


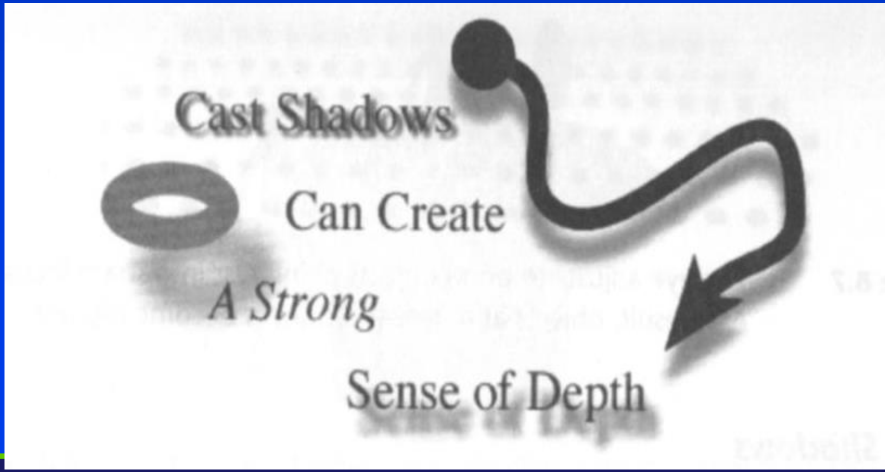


Depth Cue: Depth of Focus



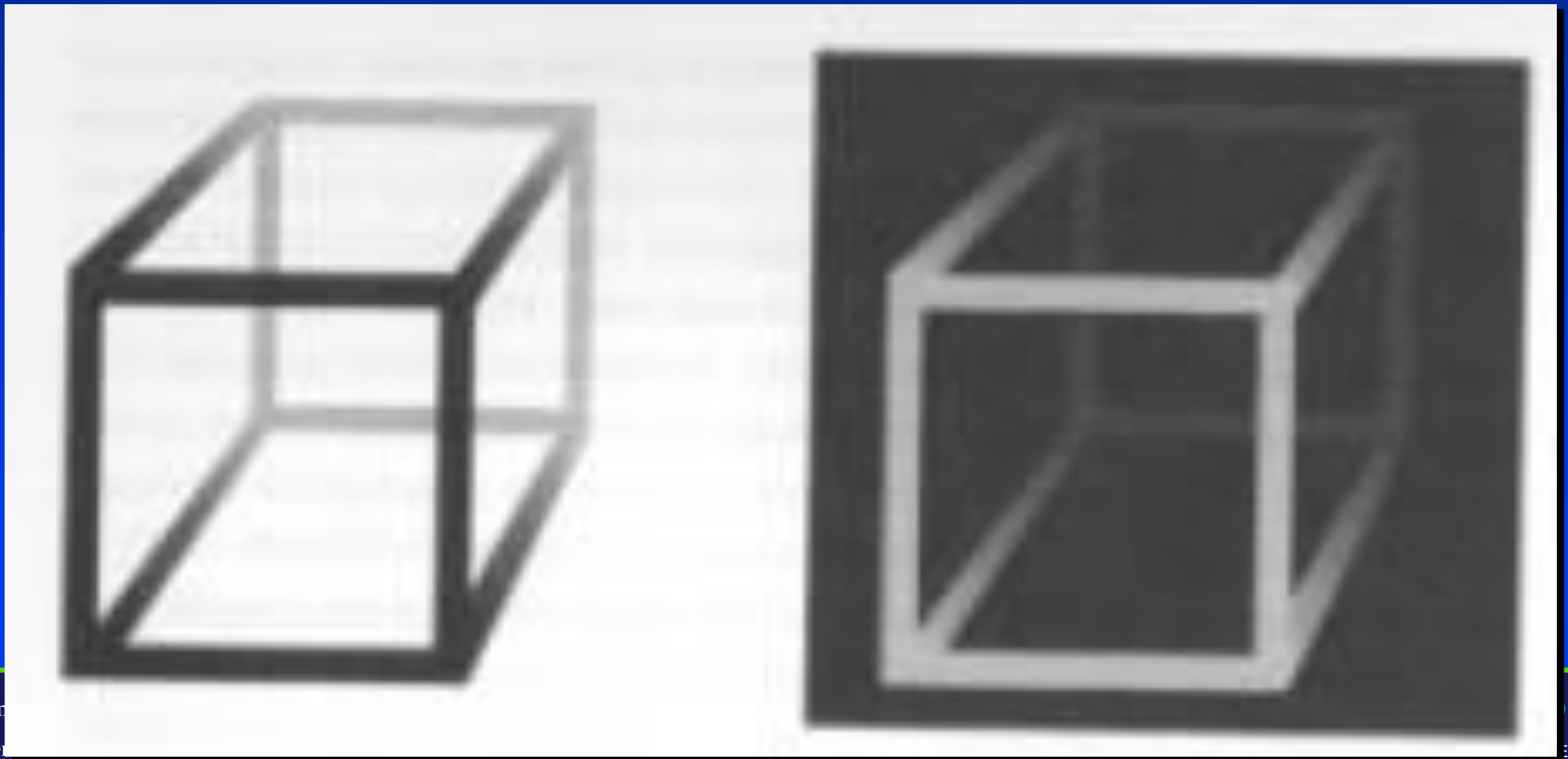
Depth Cue: Cast Shadows



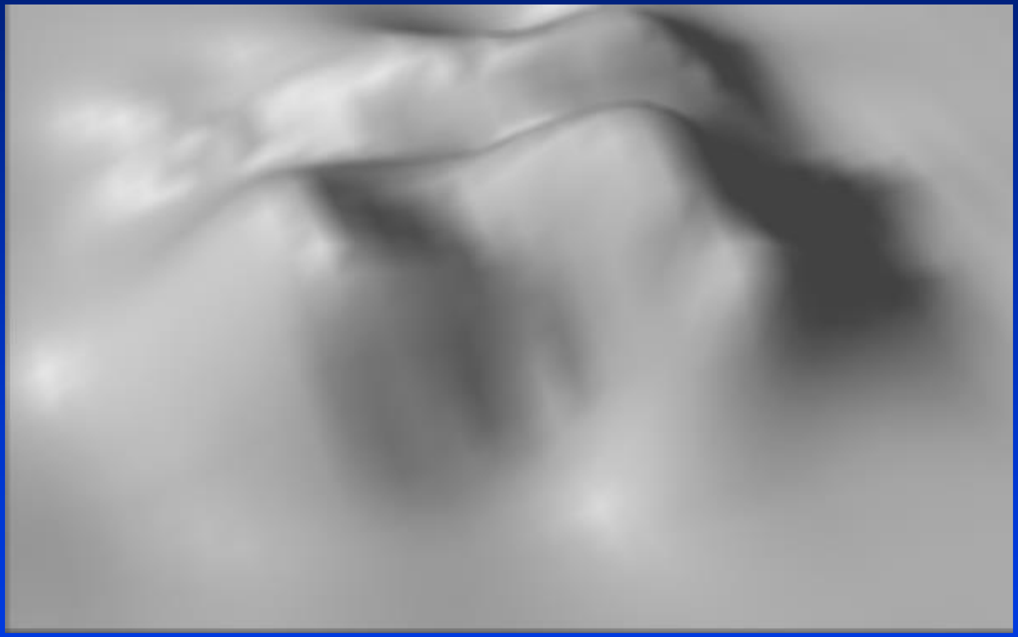


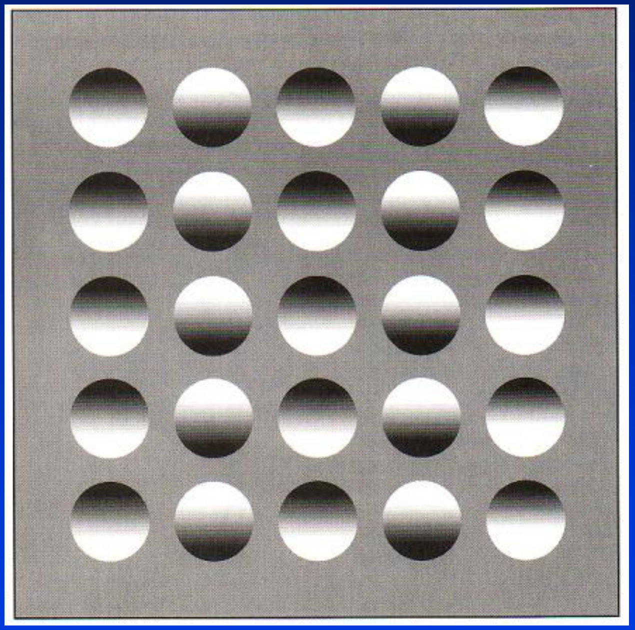
Atmospheric Depth

Reduction in contrast of distant objects

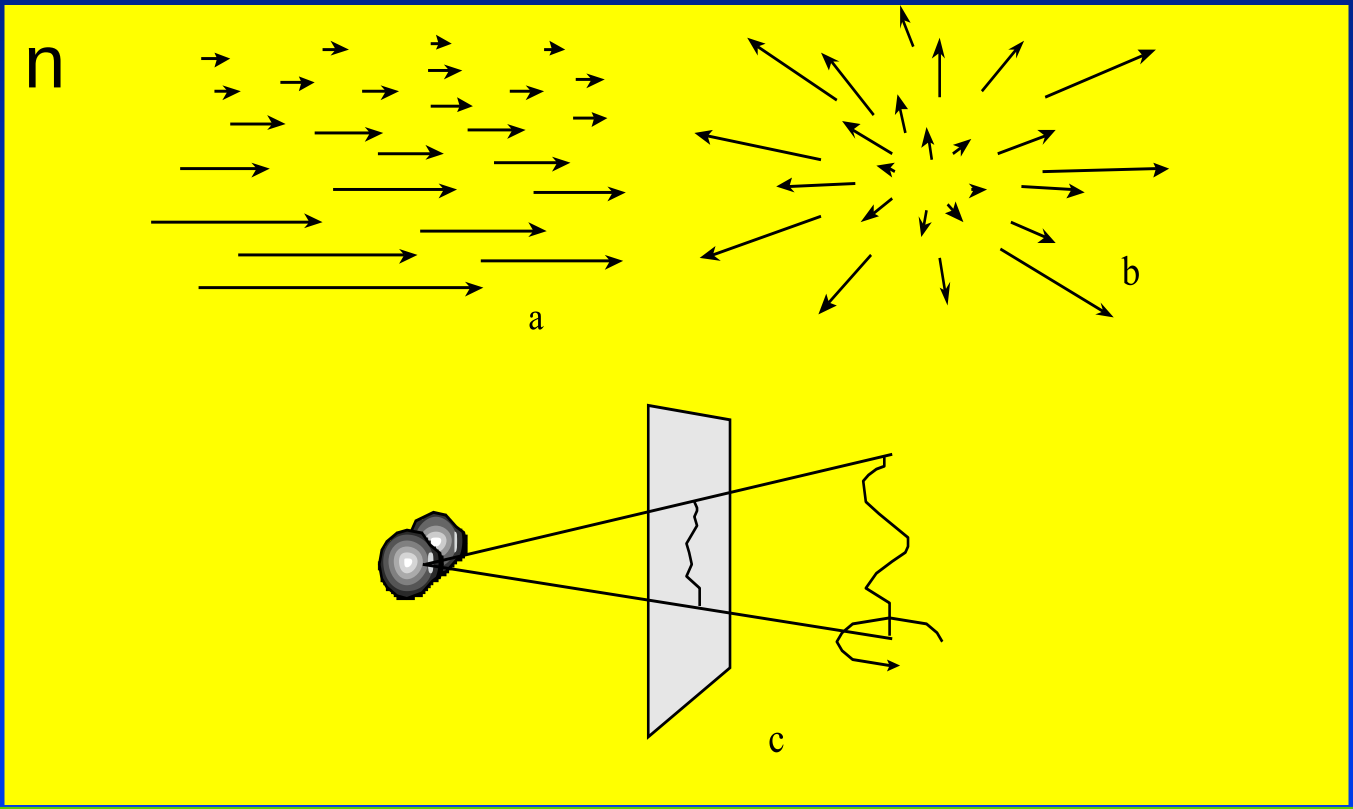


Shape from Shading



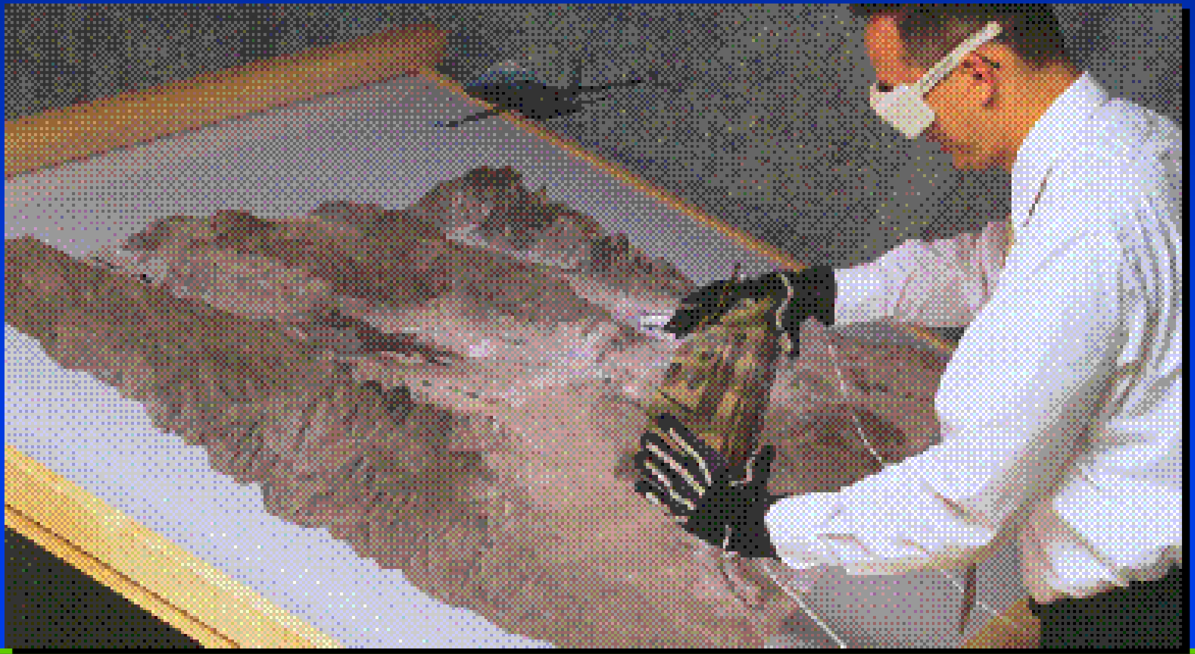


Structure from Motion



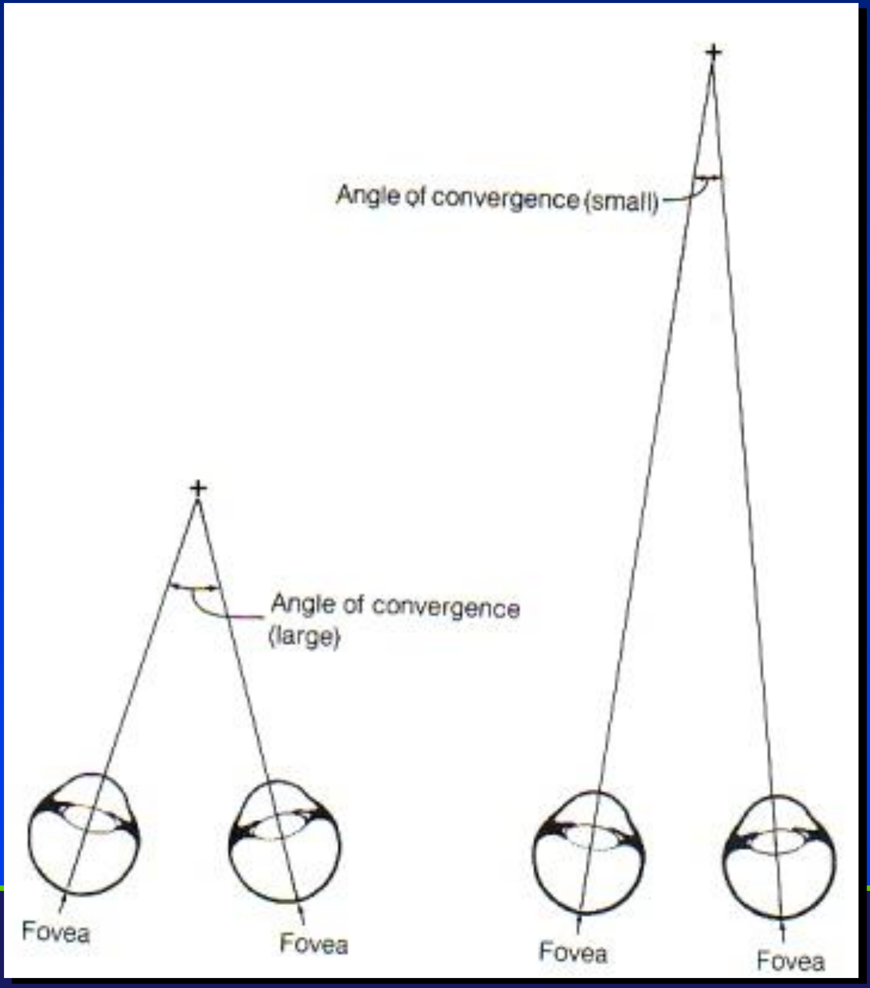
Perspective Cues

- Placement of virtual hand or object
- Need for head-coupled perspective

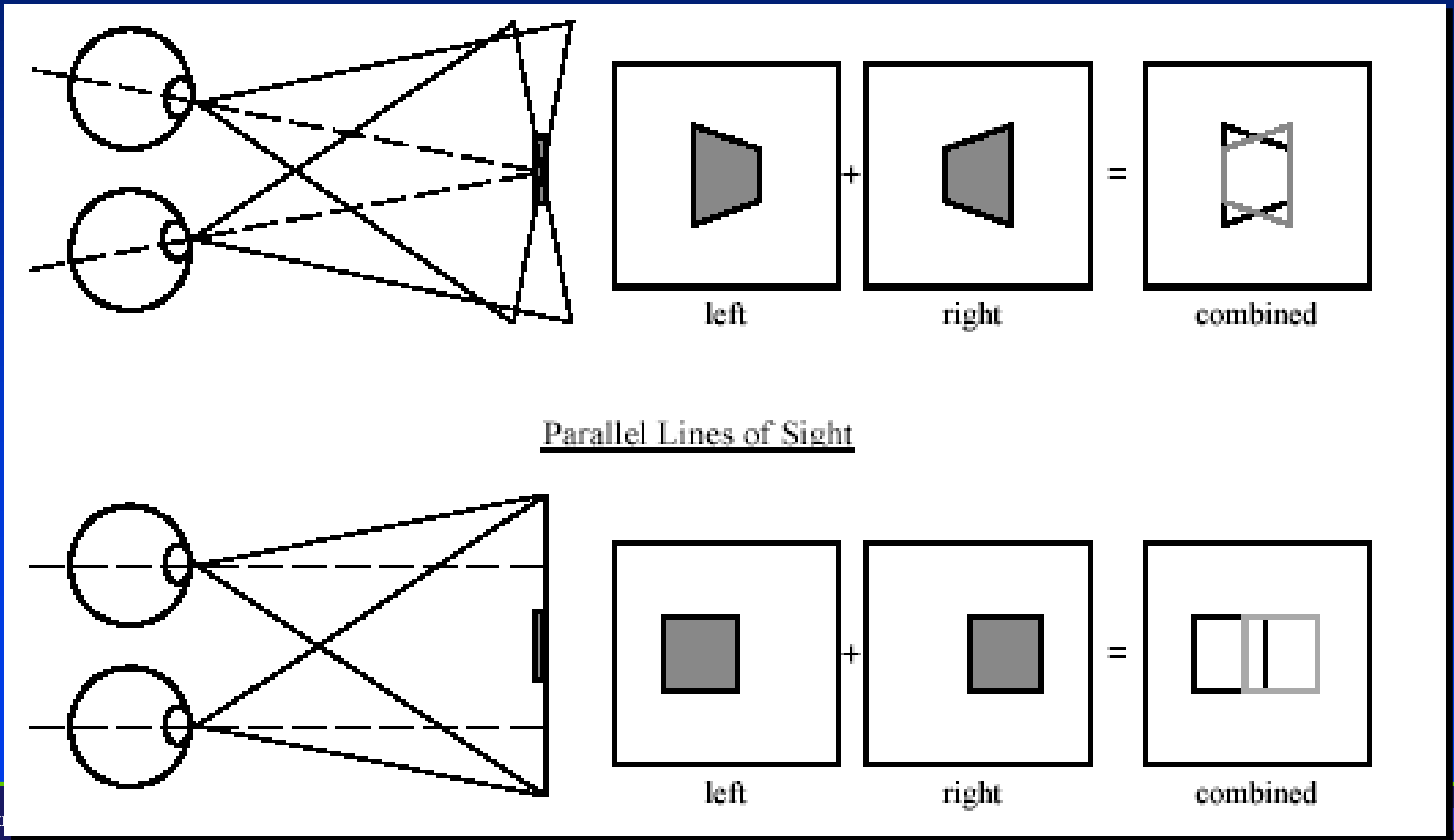


Eye Convergence

Better for relative depth than for absolute depth



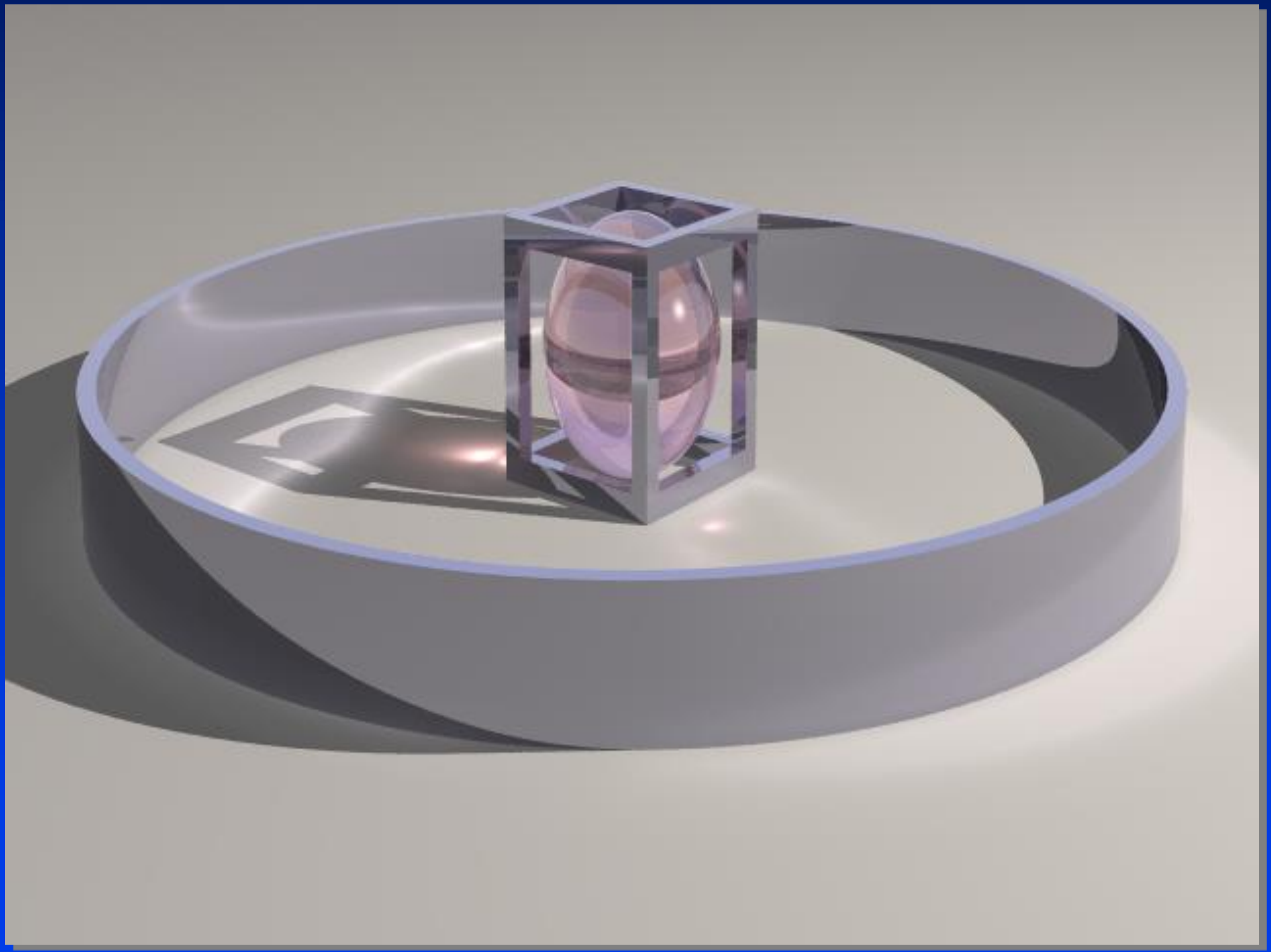
Stereoscopic Depth



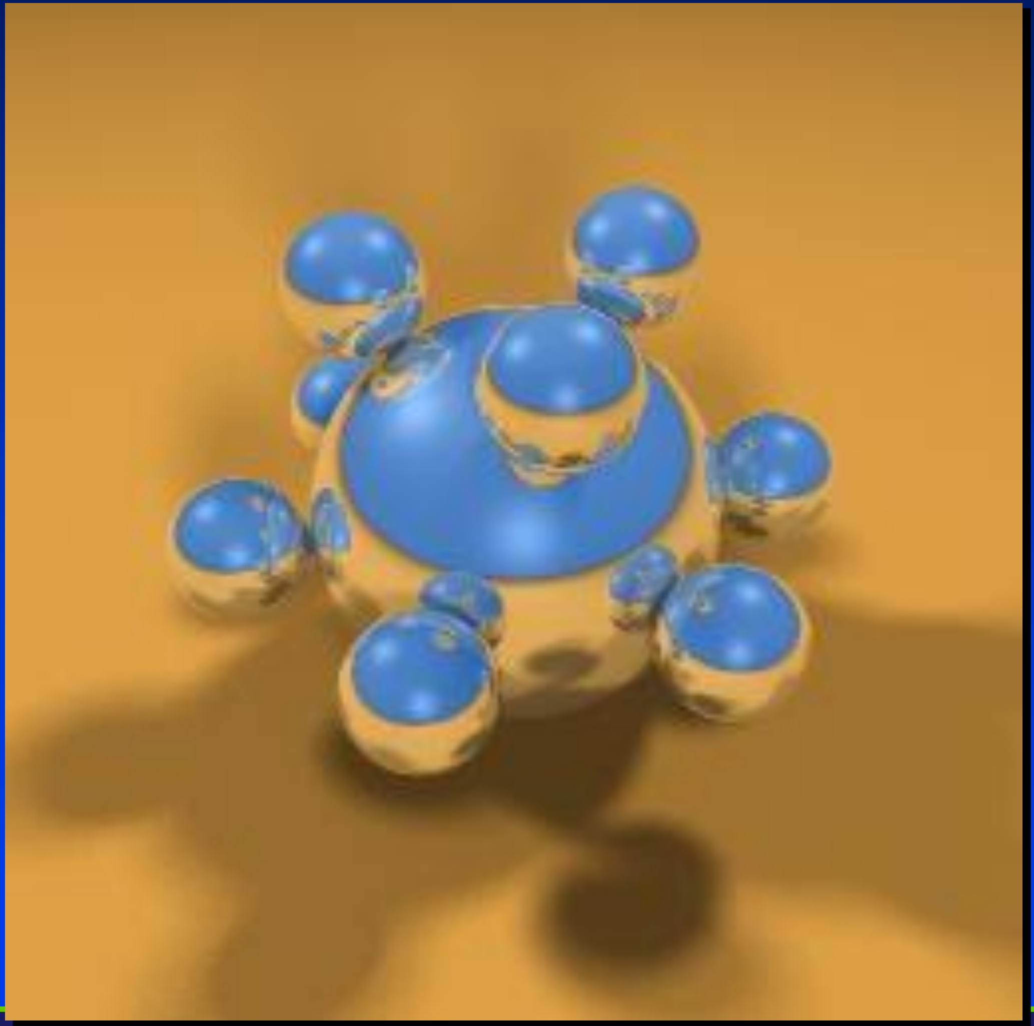
Texture Mapping

Environment Mapping

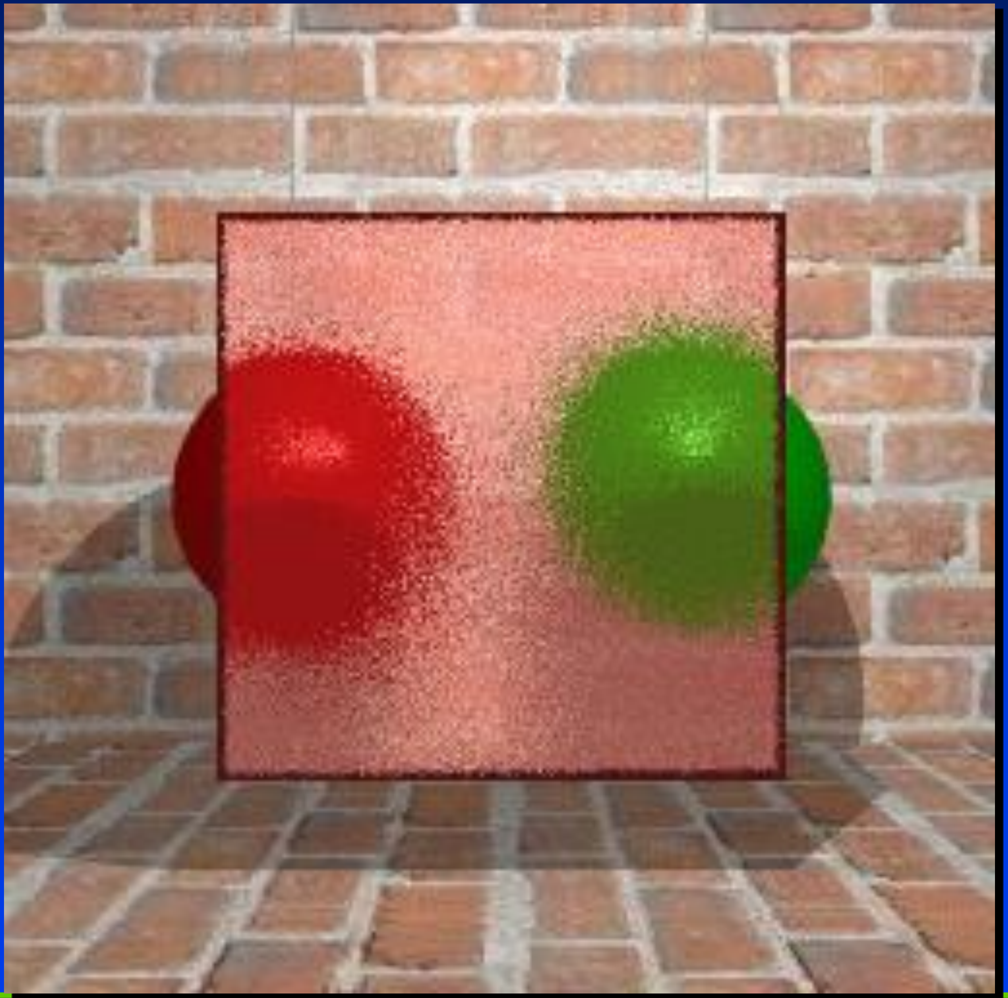
Interaction with Light



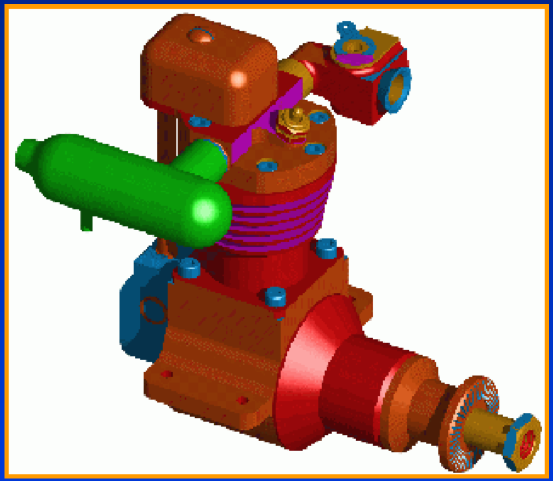
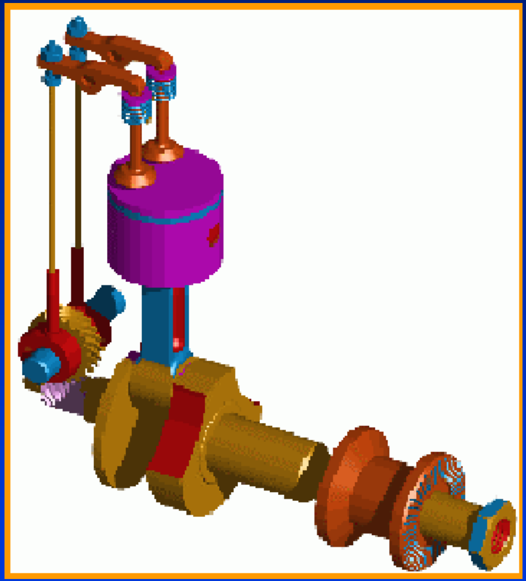
Shadowing Effects



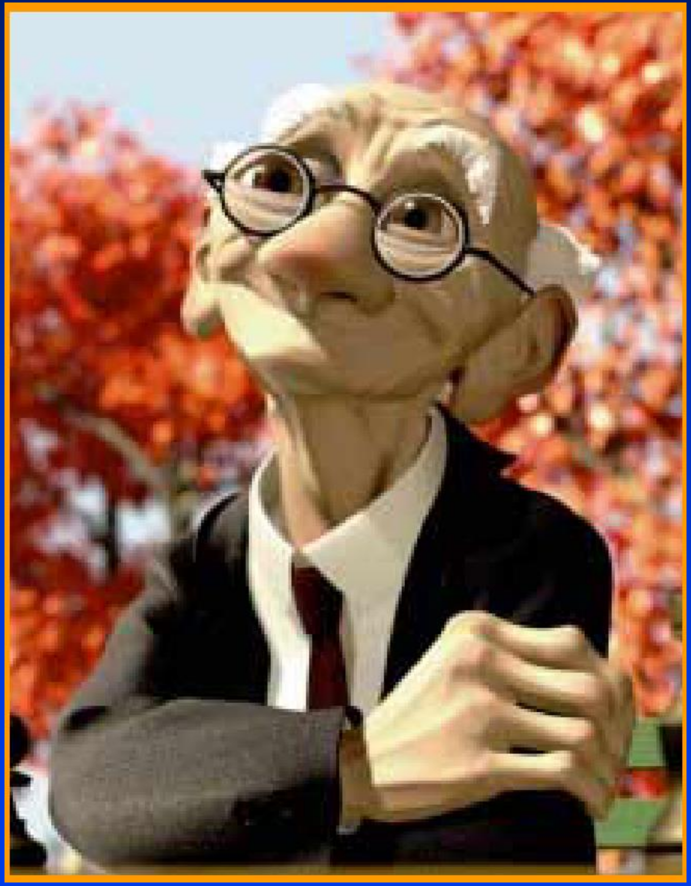
Transparency



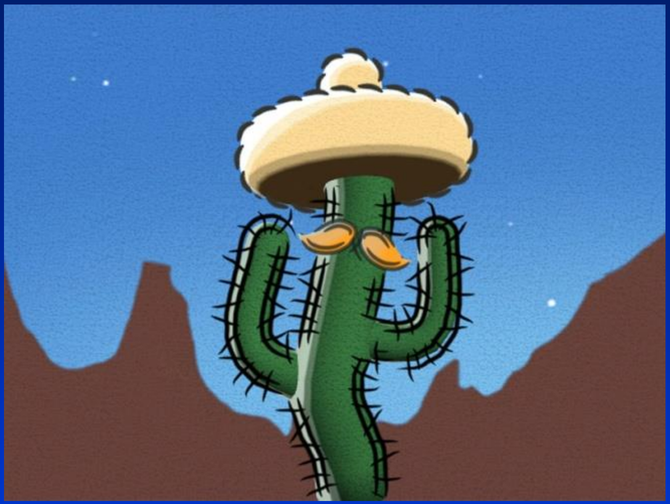
Surface Graphics

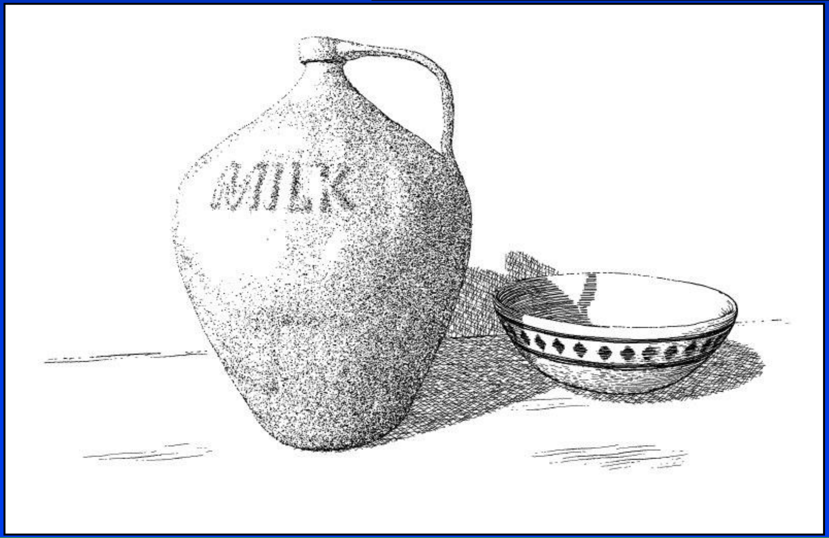




Surface Graphics

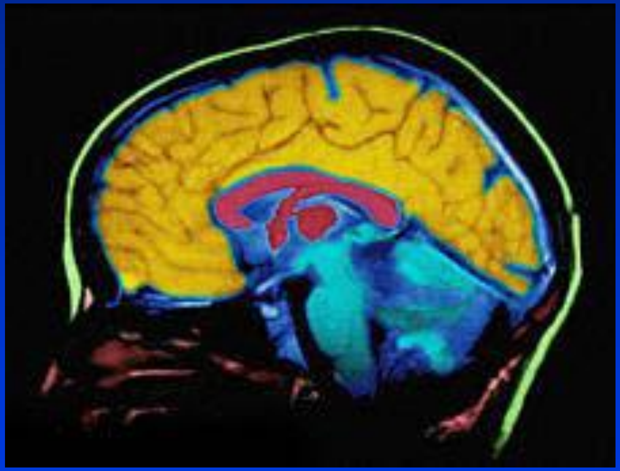


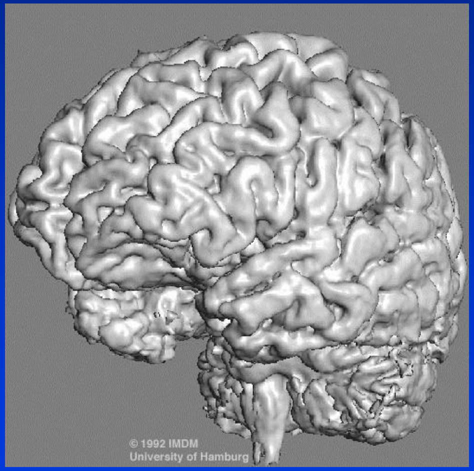
Surface Graphics





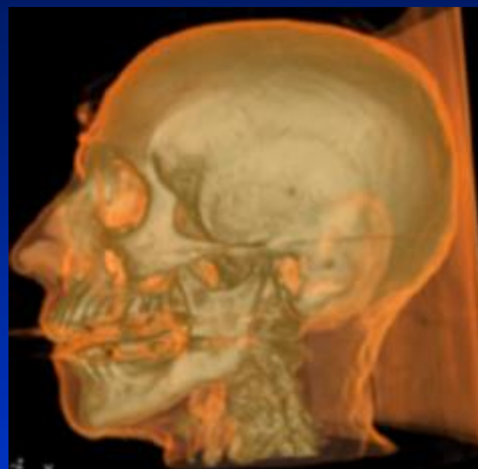
Volume Graphics

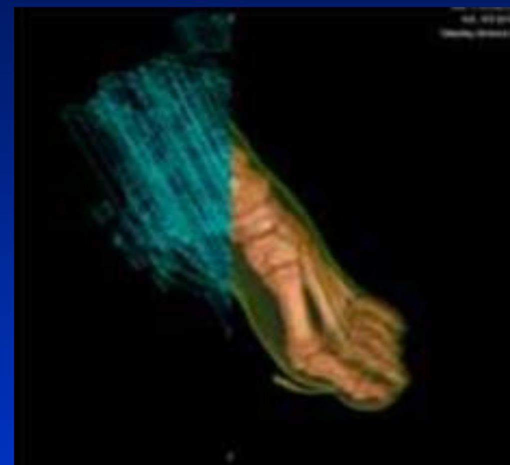


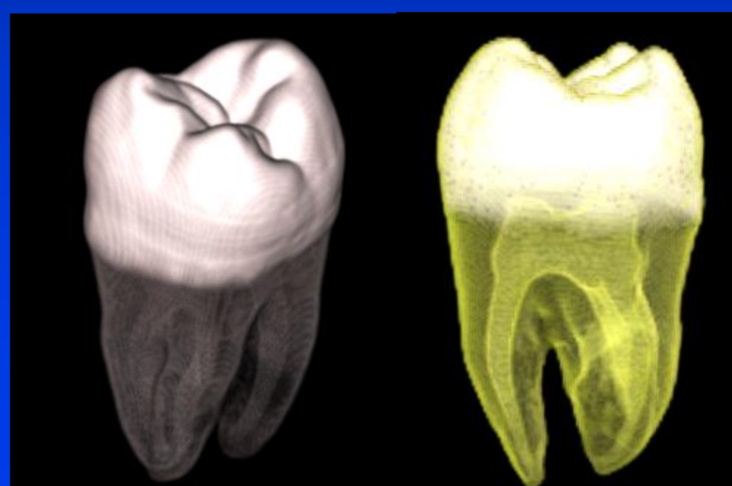


Visualization (Isosurfaces)

Visualization (Volume Rendering)



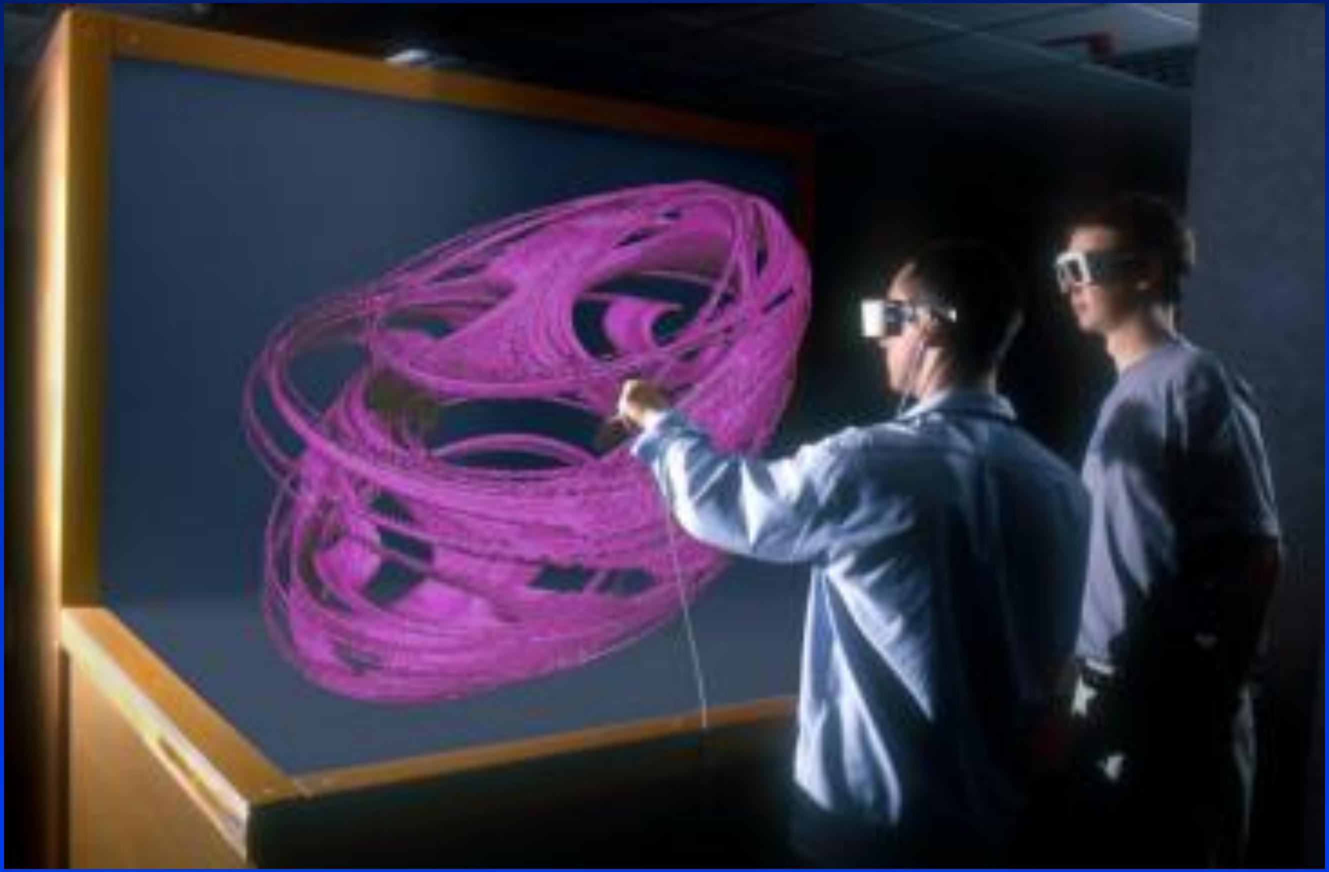




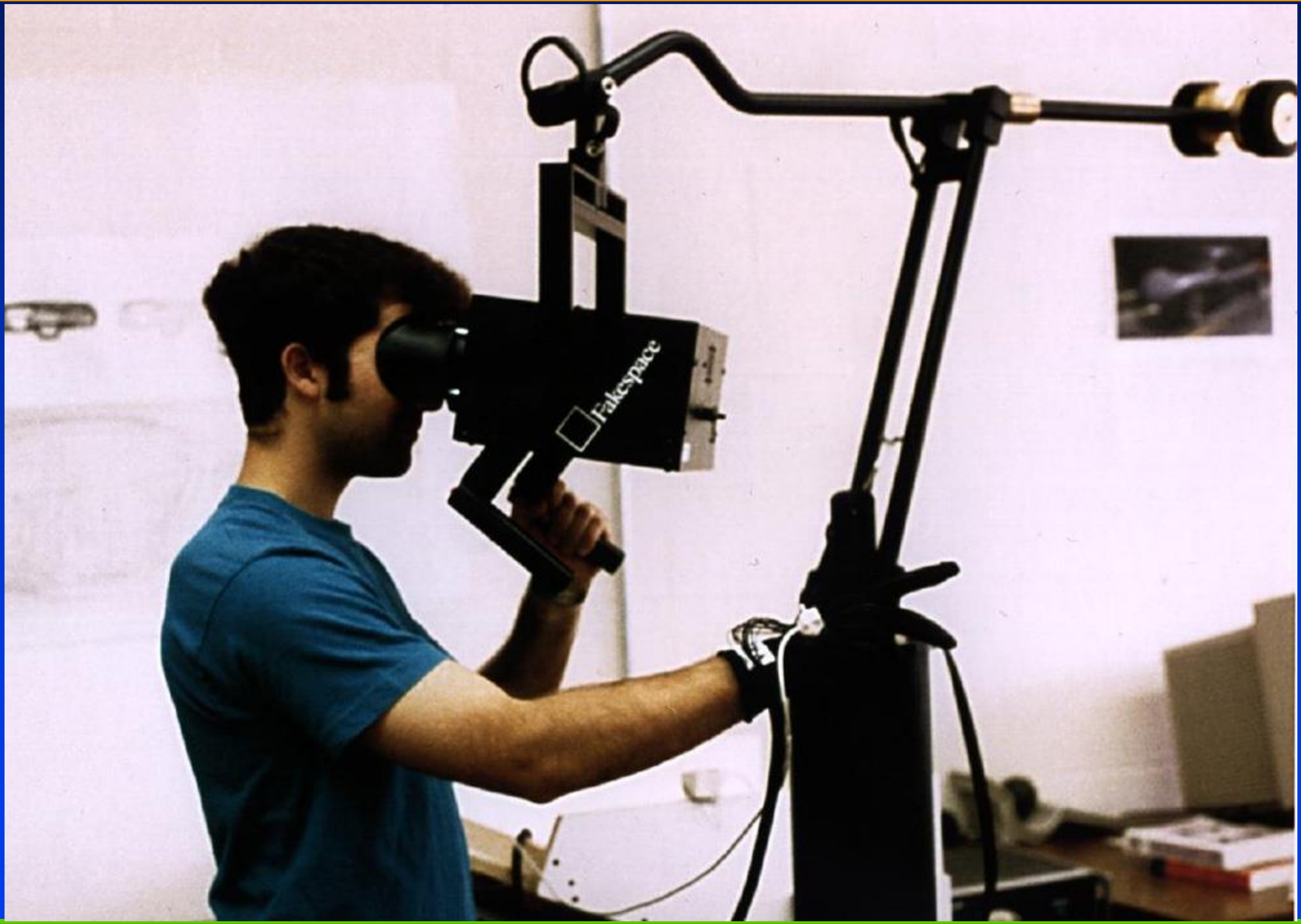
Volume Graphics

Graphics Hardware

Virtual Reality Systems

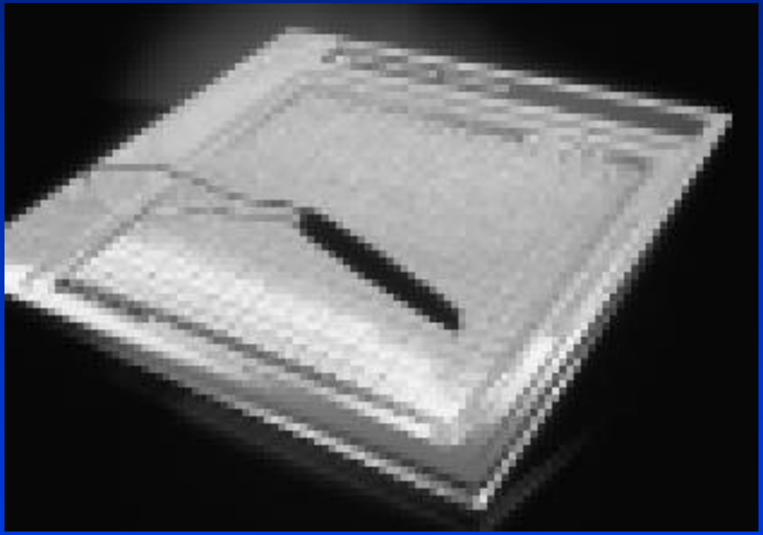


Virtual Reality Systems

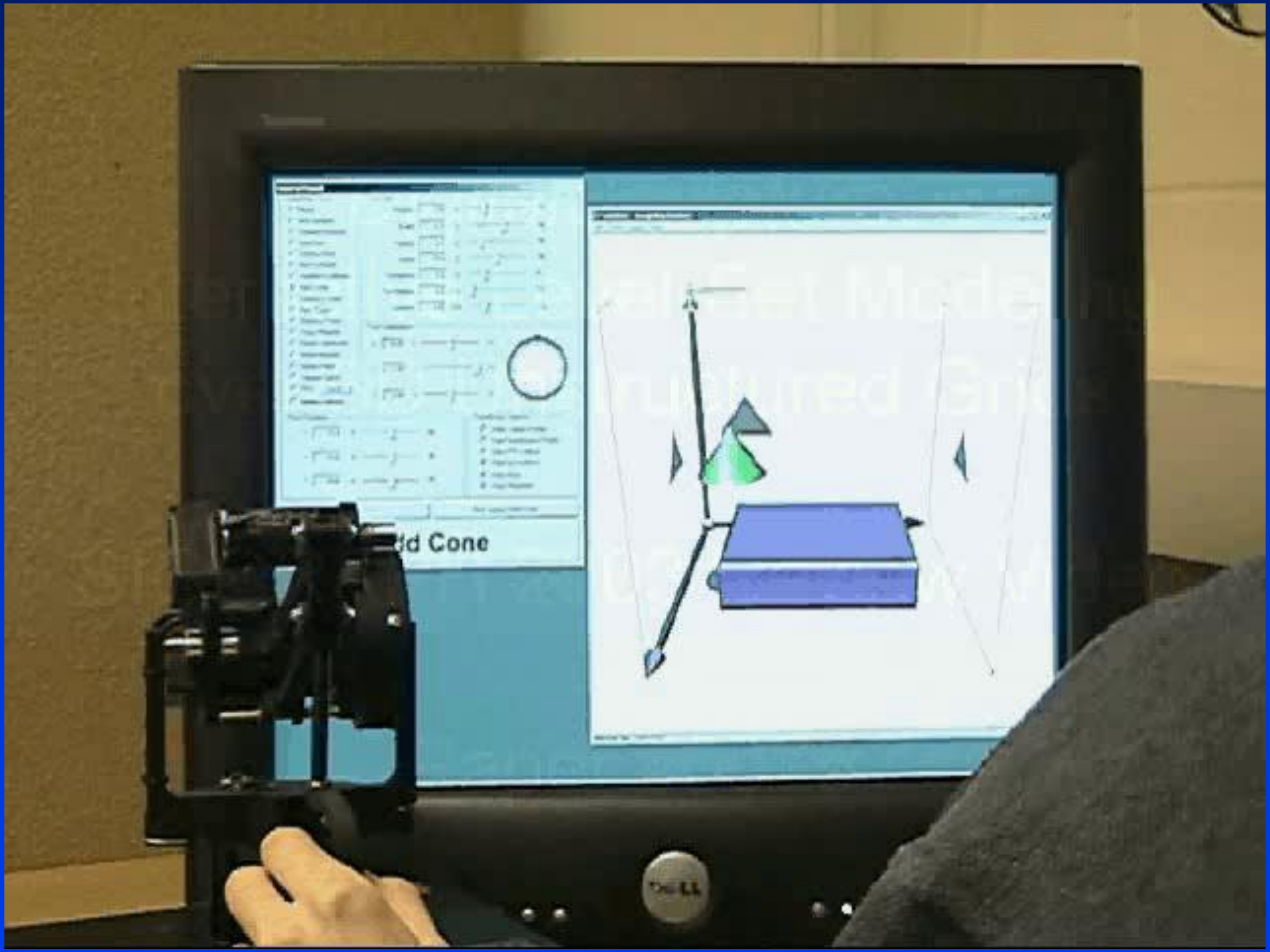


Virtual Reality Systems

Trackball, Joystick, Touch Pad

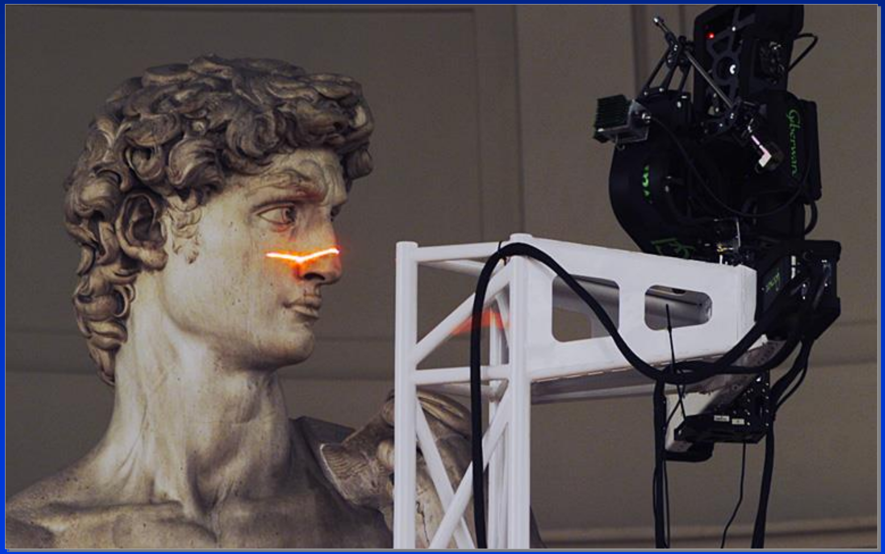


Haptics Device (Phantom 1.0)



3D Laser Range Scanner

3D Laser Range Scanner



3D Camera

Plane Equation

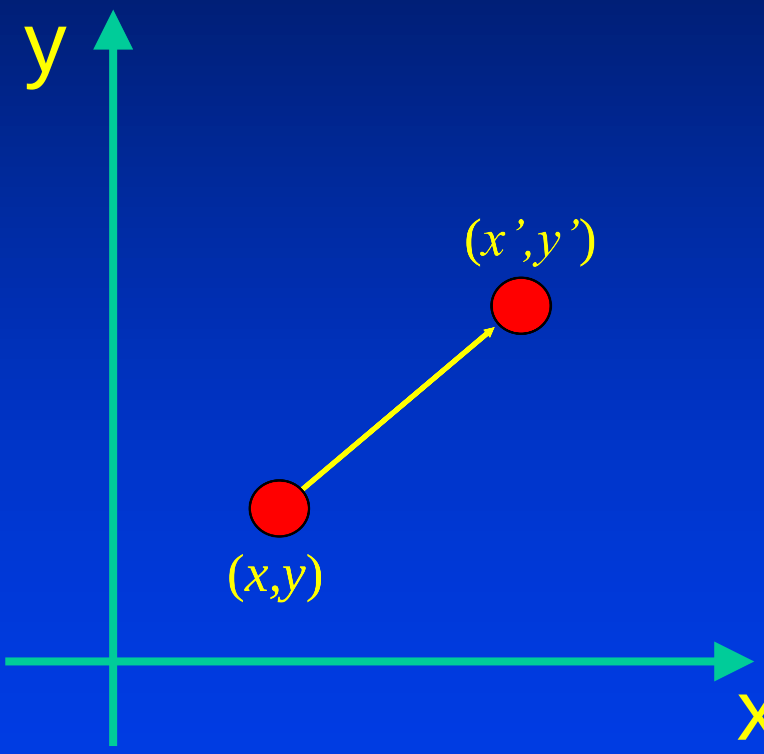
2D Geometric Transformations (A Quick Review)

- Translation
- Rotation
- Scaling
- Shear
- Homogenous Coordinates
- Matrix Representations
- Composite Transformations

Translation

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

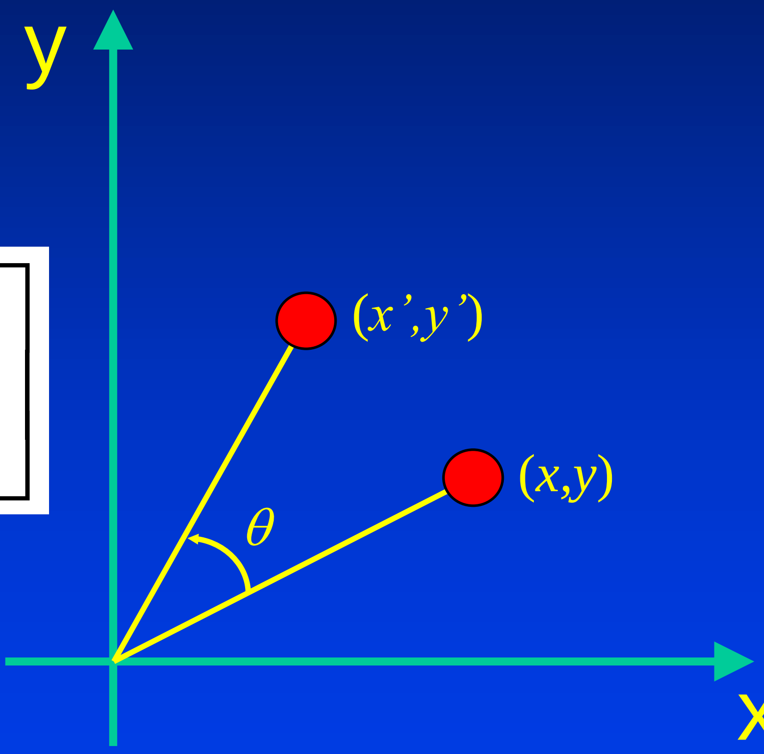
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$



Rotation

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

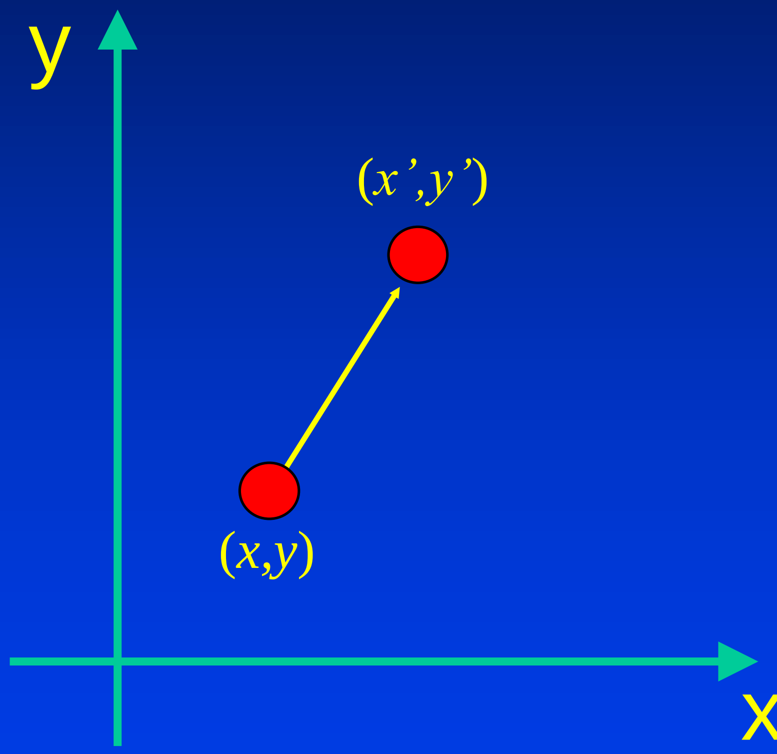
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Scaling

- $x' = S_x \cdot x$
- $y' = S_y \cdot y$

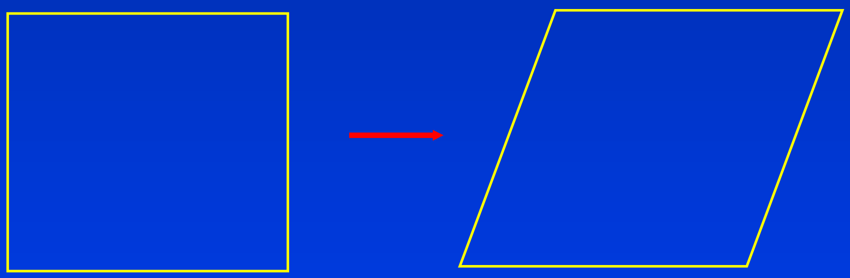
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Shear

- $x' = x + h_x \cdot y$
- $y' = y$

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



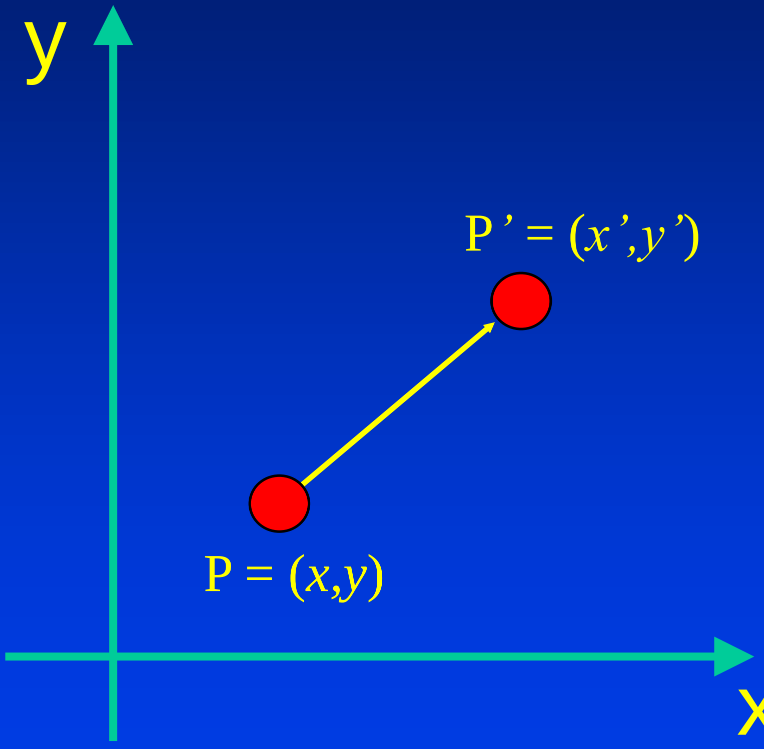
Homogenous Coordinates: Geometric Intuition

- Each position (x, y) is represented as $(x, y, 1)$.
- All transformations can be represented as matrix multiplication.
- Composite transformation becomes easier.

Translation in Homogenous Coordinates

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

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

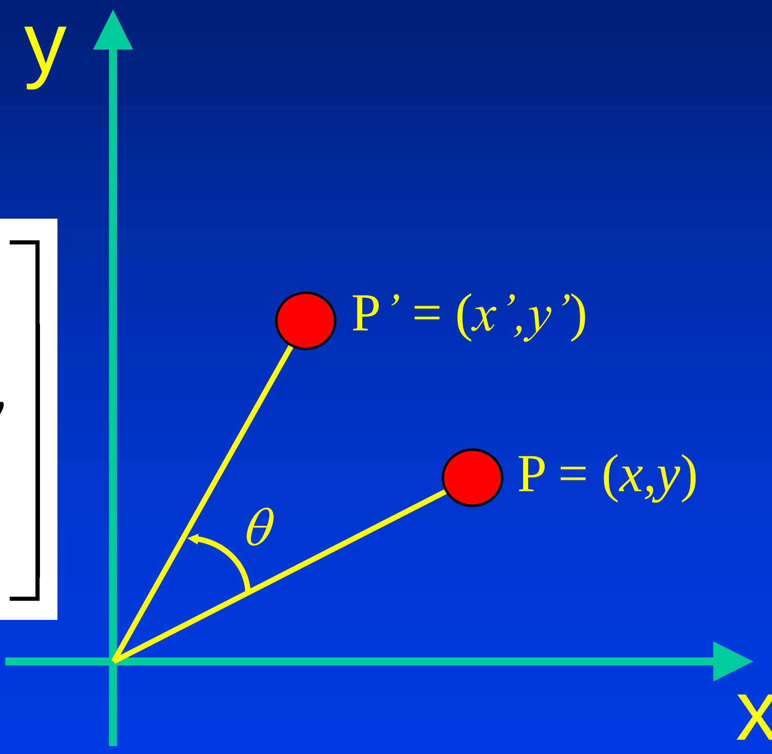


$$\mathbf{P'} = \mathbf{T}(t_x, t_y) \bullet \mathbf{P}$$

Rotation in Homogenous Coordinates

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

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



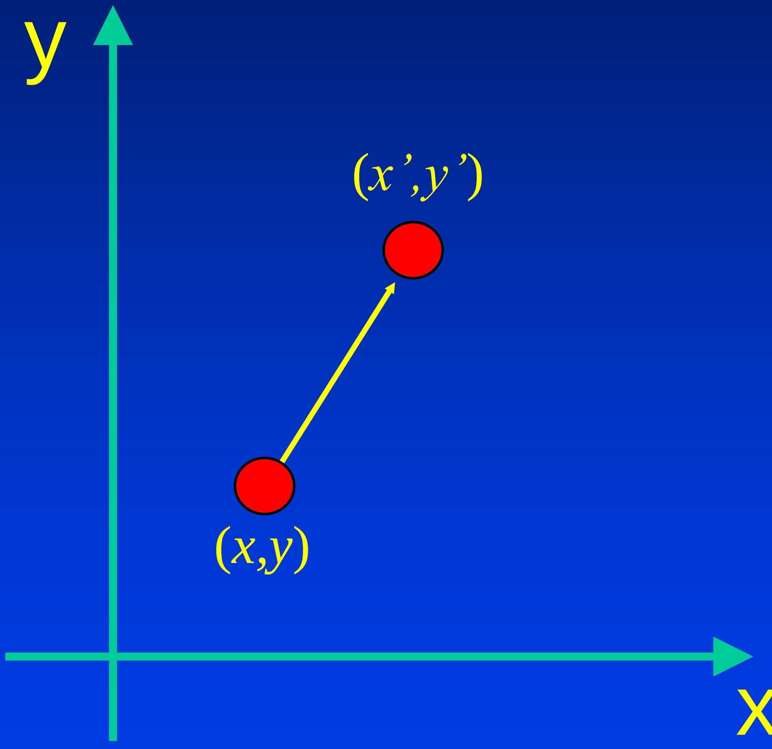
$$\mathbf{P'} = \mathbf{R}(\theta) \cdot \mathbf{P}$$

Scaling in Homogenous Coordinates

- $x' = s_x \cdot x$
- $y' = s_y \cdot y$

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

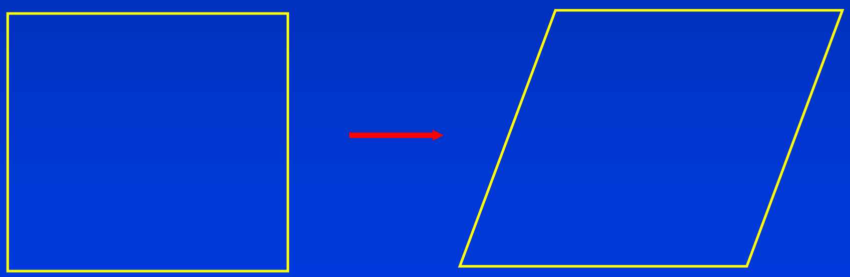
$$\mathbf{P'} = \mathbf{S}(s_x, s_y) \cdot \mathbf{P}$$



Shear in Homogenous Coordinates

- $x' = x + h_x \cdot y$
- $y' = y$

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



$$\mathbf{P}' = \mathbf{SH}_x \cdot \mathbf{P}$$

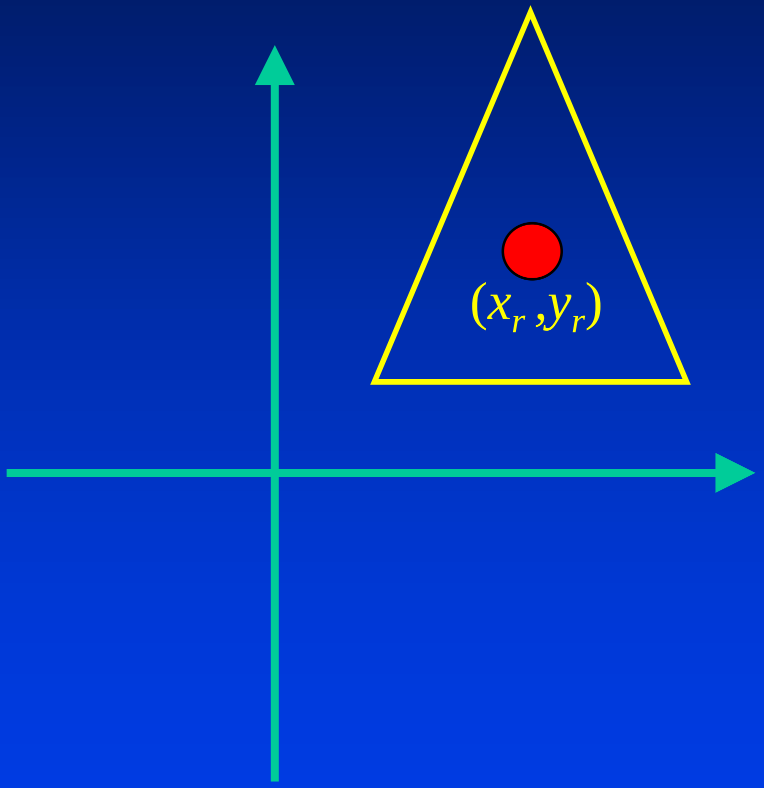
2D Geometric Transformations

- Translation
- Rotation
- Scaling
- Shear
- Homogenous Coordinates
- Composite Transformations

2D Geometric Transformations

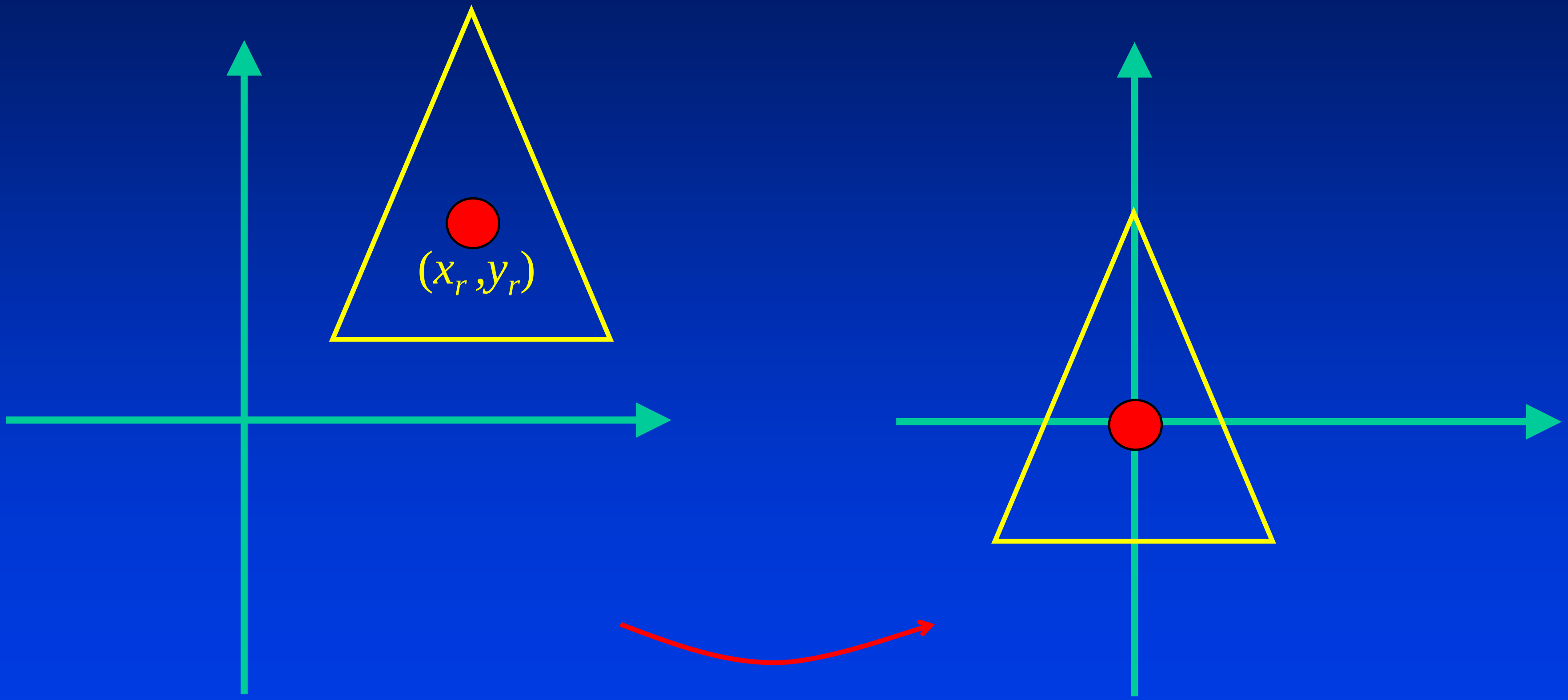
- Translation
- Rotation
- Scaling
- Shear
- Homogenous Coordinates
- Composite Transformations
  - Rotation about a fixed point

Rotation About a Fixed Point



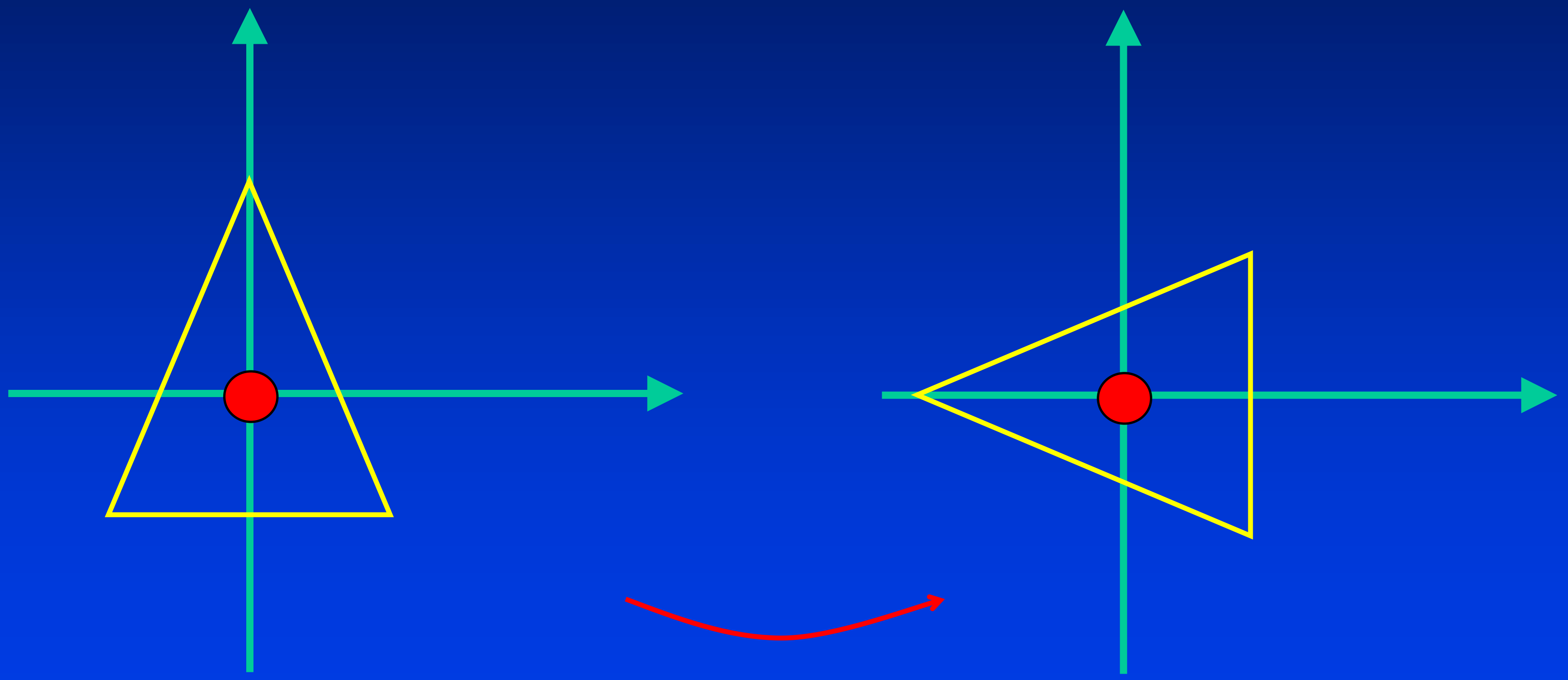
1. Translate the object to the origin.
2. Rotate around the origin.
3. Translate the object back.

Rotation About a Fixed Point



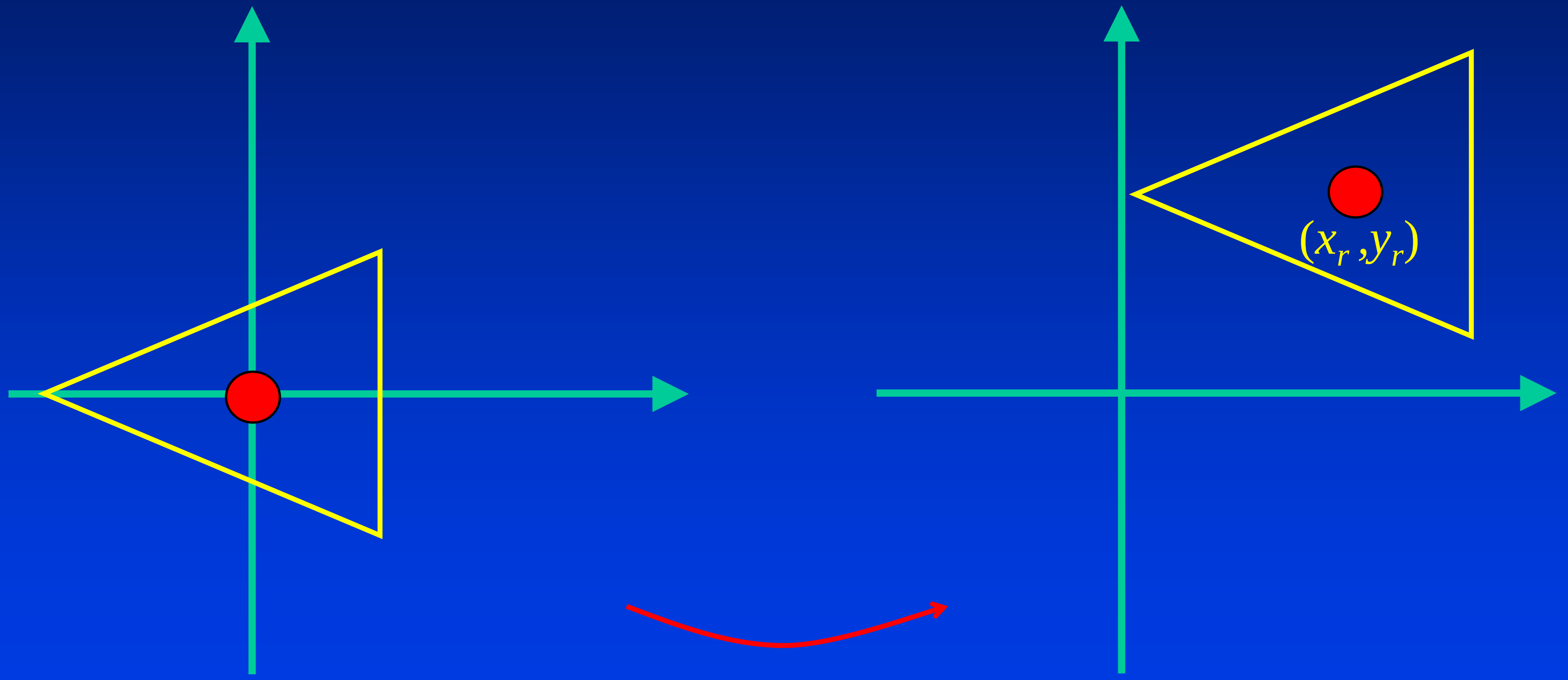
1. Translate the object to the origin

Rotation About a Fixed Point



2. Rotate about origin

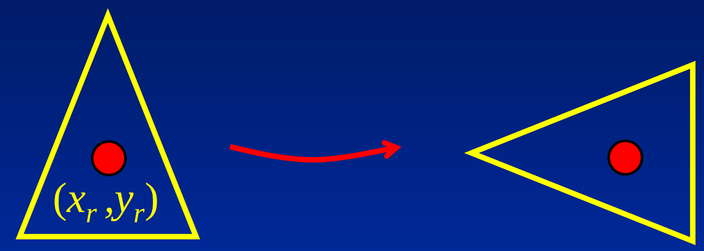
Rotation About a Fixed Point



3. Translate the object back

Rotation About a Fixed Point

1. **Translate the object to the origin.**
2. **Rotate around the origin.**
3. **Translate the object back.**



$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}(x_r, y_r) \bullet \mathbf{R}(\theta) \bullet \mathbf{T}(-x_r, -y_r)$$

$$\mathbf{P'} = \mathbf{T}(x_r, y_r) \bullet \mathbf{R}(\theta) \bullet \mathbf{T}(-x_r, -y_r) \bullet \mathbf{P}$$

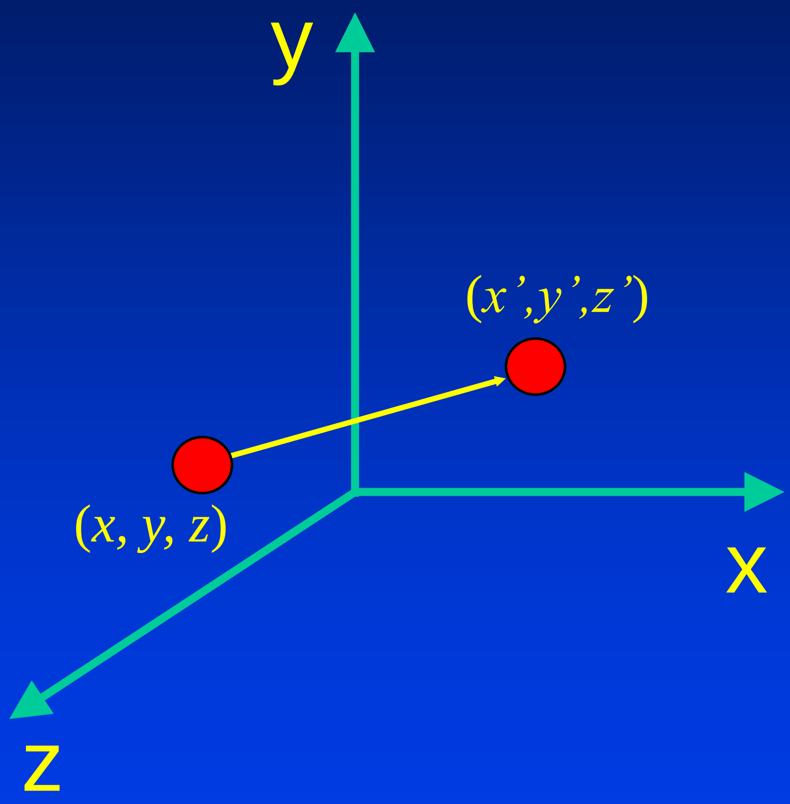
3D Geometric Transformations

- Basic 3D Transformations
  - Translation
  - Rotation
  - Scaling
  - Shear
- Composite 3D Transformations
- Change of Coordinate systems

Translation

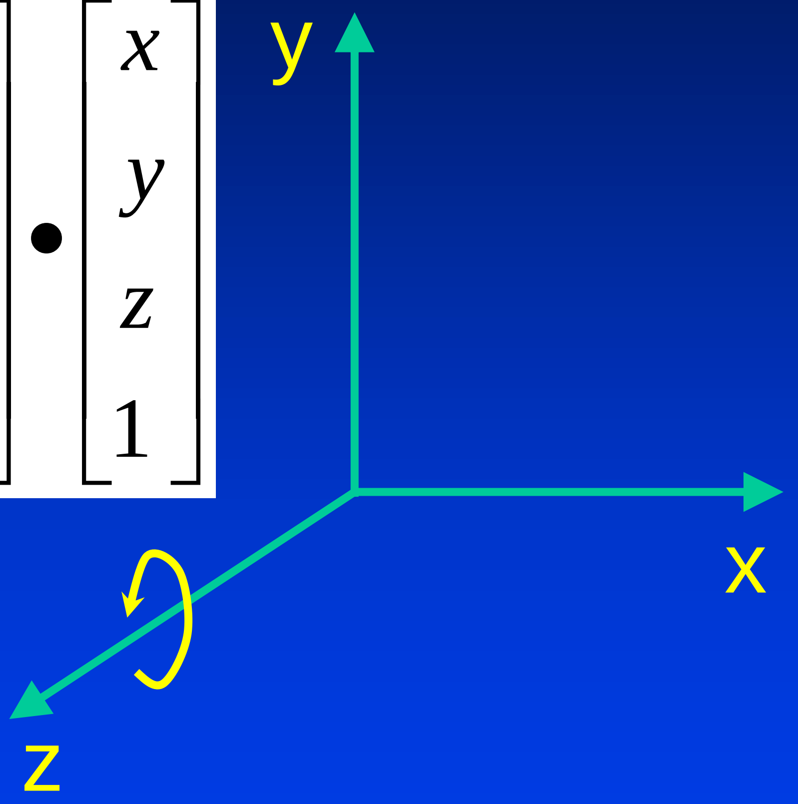
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

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



Rotation about z-axis

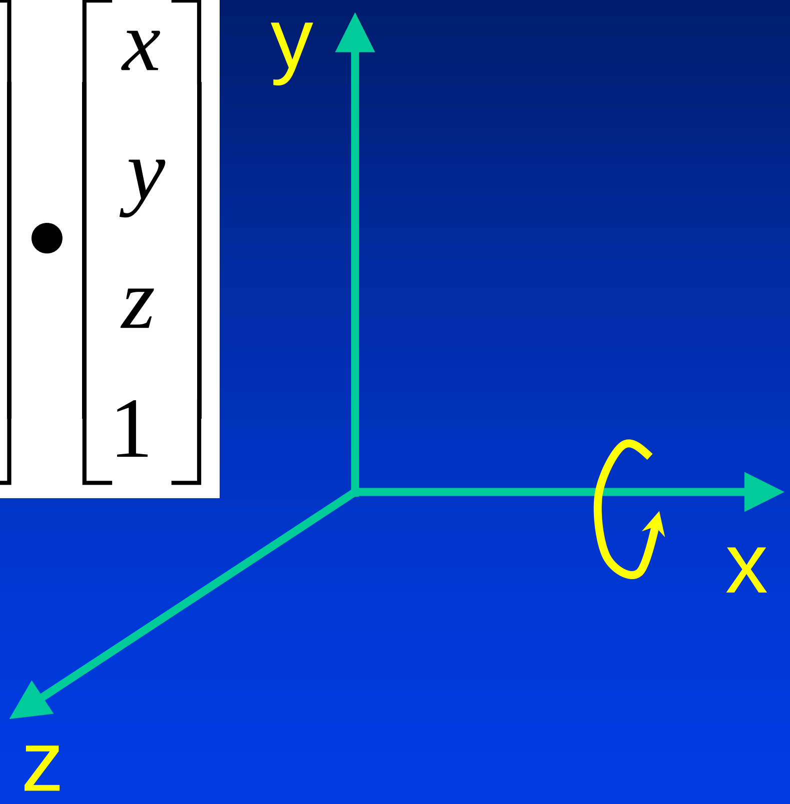
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



$$\mathbf{P'} = \mathbf{R_z(\theta) \cdot P}$$

Rotation about x-axis

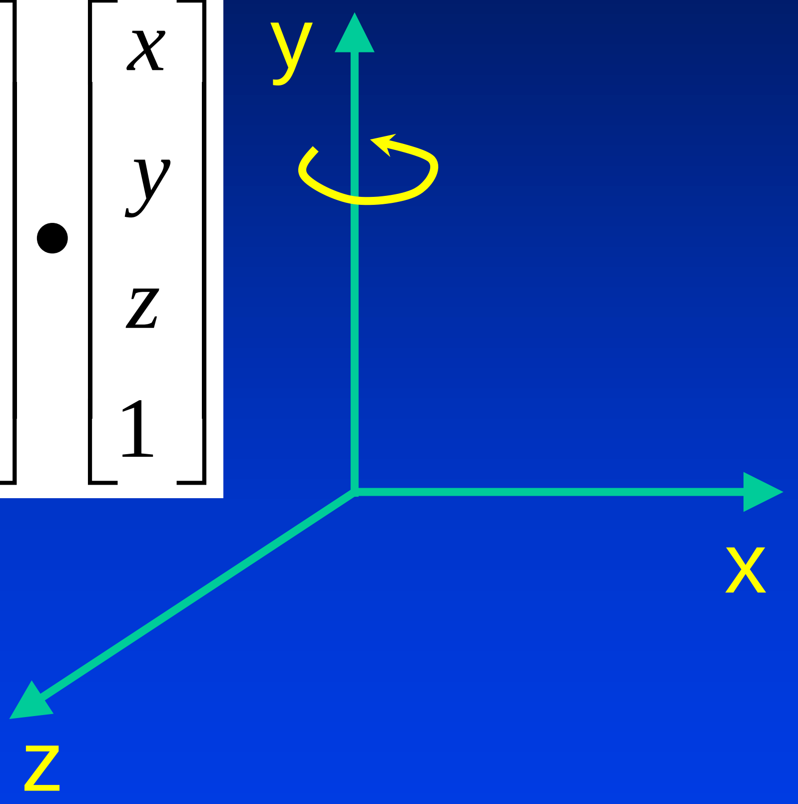
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



$$\mathbf{P}' = \mathbf{R}_x(\theta) \bullet \mathbf{P}$$

Rotation about y-axis

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



$$\mathbf{P}' = \mathbf{R}_y(\theta) \cdot \mathbf{P}$$

Rotation About a Fixed Point

1. **Translate the object to the origin.**
2. **Rotate about the three axis, respectively.**
3. **Translate the object back.**

$$\mathbf{P}' = \mathbf{T}\,(x_r\ ,\, y_{r,}\, ,\, z_r) \bullet \mathbf{R1} * \mathbf{R2} * \mathbf{R3} \bullet \mathbf{T}\,(-x_r\ ,\, -y_{r,}\, ,\, -z_r) \bullet \mathbf{P}$$

$$\mathbf{Ri} = \mathbf{R}_{\mathbf{x}}(\theta_{\mathbf{x,i}}) \bullet \mathbf{R}_{\mathbf{y}}(\theta_{\mathbf{y,i}}) \bullet \mathbf{R}_{\mathbf{z}}(\theta_{\mathbf{z,i}})$$

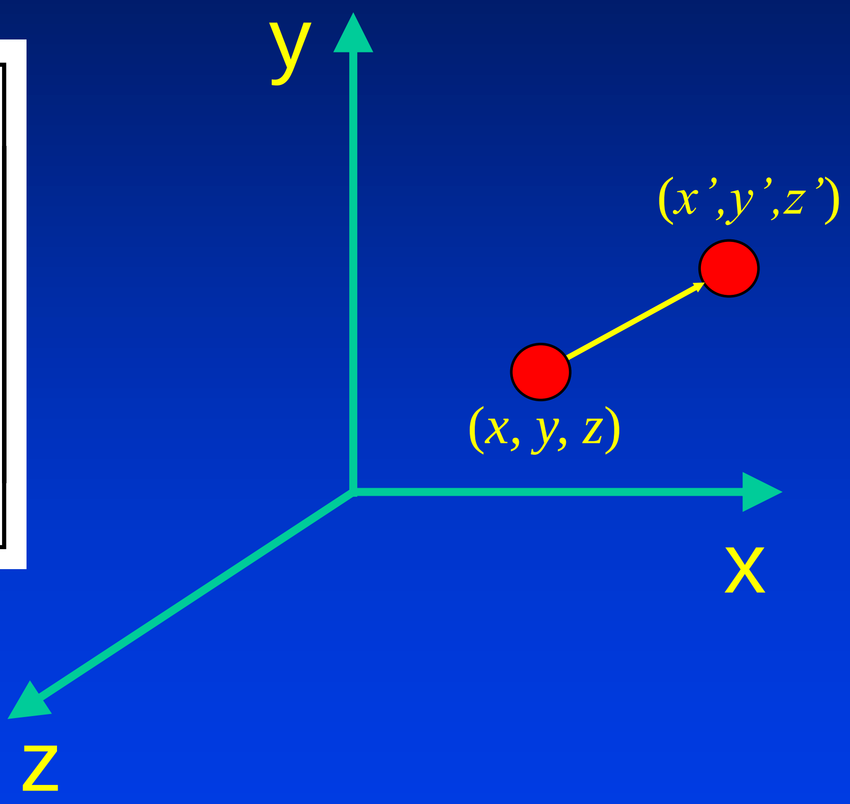
Rotation with Arbitrary Direction

1. We will have to translate an arbitrary vector so that its starting point starts from the origin
2. We will have to rotate w.r.t. x-axis so that this vector stays on x-z plane
3. We will then rotate w.r.t. y-axis so that this vector aligns with z-axis
4. We will then rotate w.r.t. z-axis
5. Reverse (3), (2), and (1)

Scaling

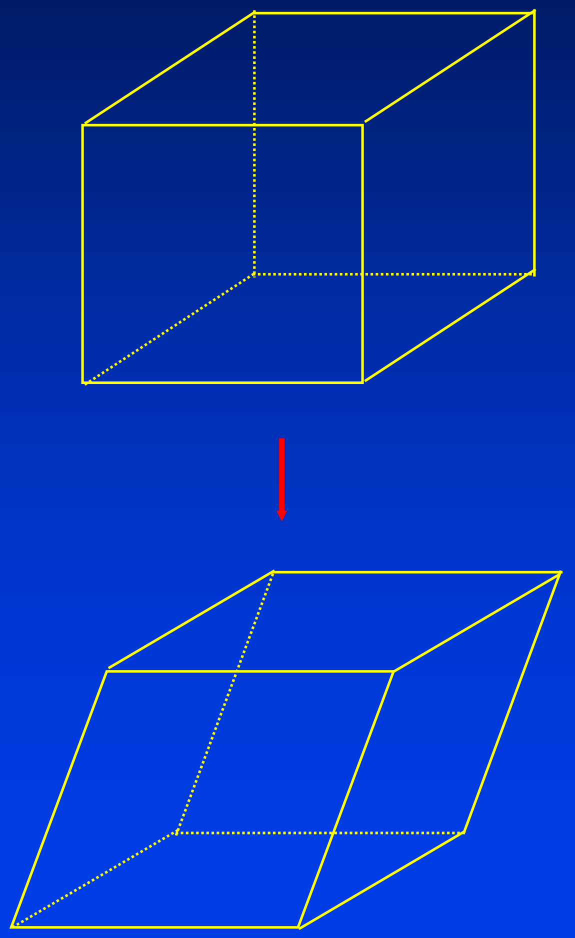
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

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



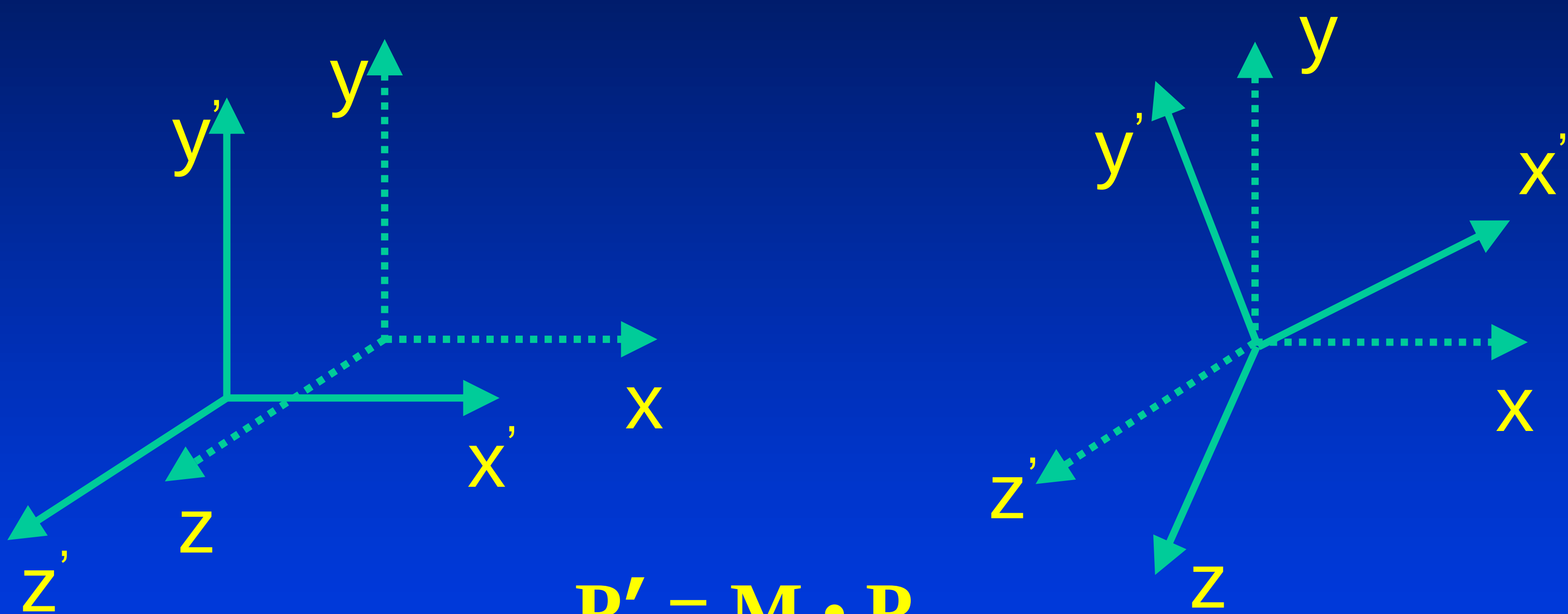
Shear

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



$$\mathbf{P}' = \mathbf{SH}_{xy} \bullet \mathbf{P}$$

Change in Coordinate Systems



$$P' = M \cdot P$$

M can be a combination of translation, rotation and scaling.

Taking a Picture with a Camera

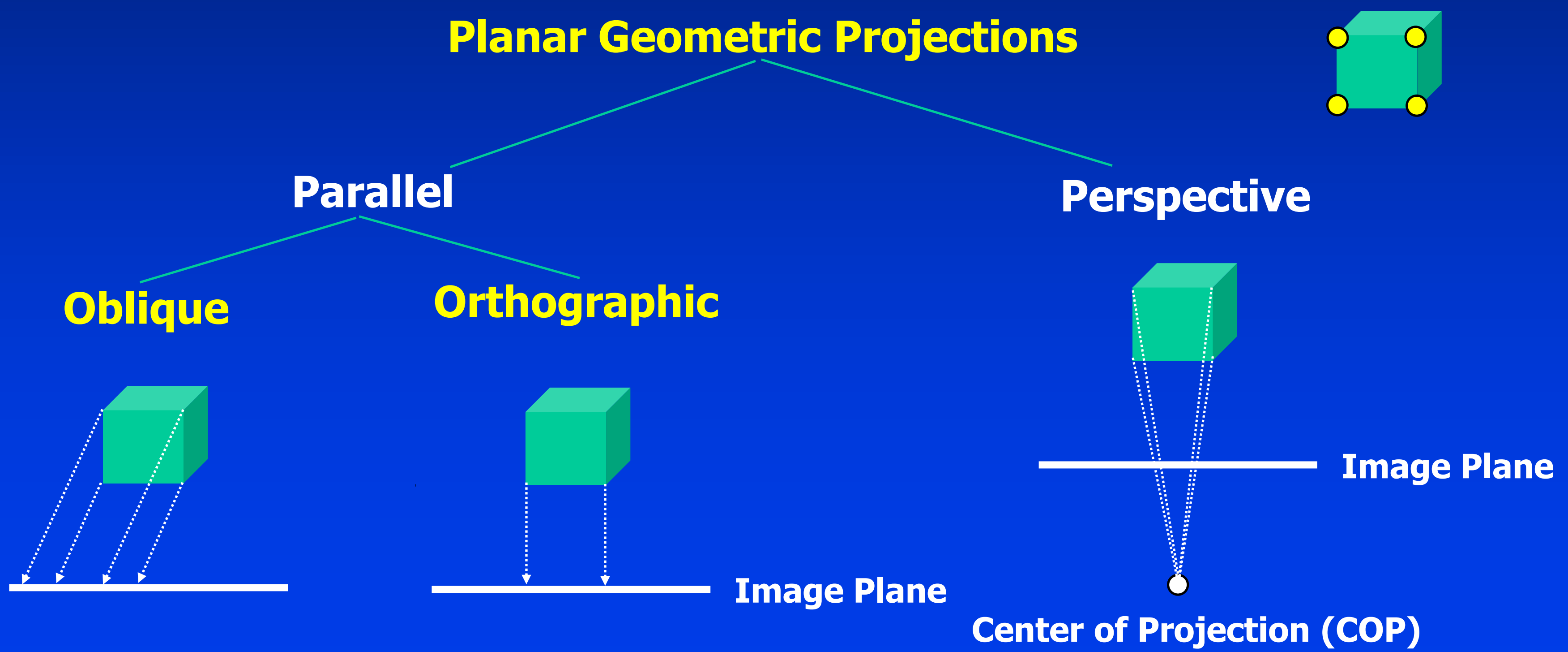
- Geometric Coordinate Systems: Local, World, Viewing
- Graphics Rendering Pipeline
- ModelView
  - Matrix operations on models
- World coordinates to Viewing coordinates
  - Matrix operations (models or cameras)
- Projection with a camera

Viewing in 3D

- Planar Geometric Projections
- Parallel Orthographic Projections
- Perspective Projections
- Projections in OpenGL
- Clipping

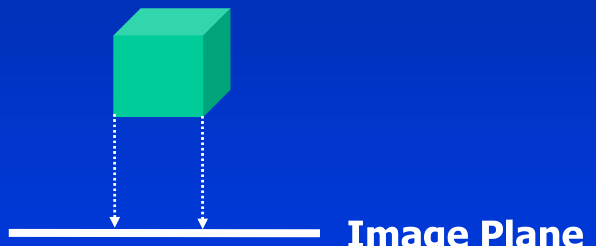
Planar Geometric Projections

- Maps points from camera coordinate system to the screen (image plane of the virtual camera).



Parallel Orthographic Projection

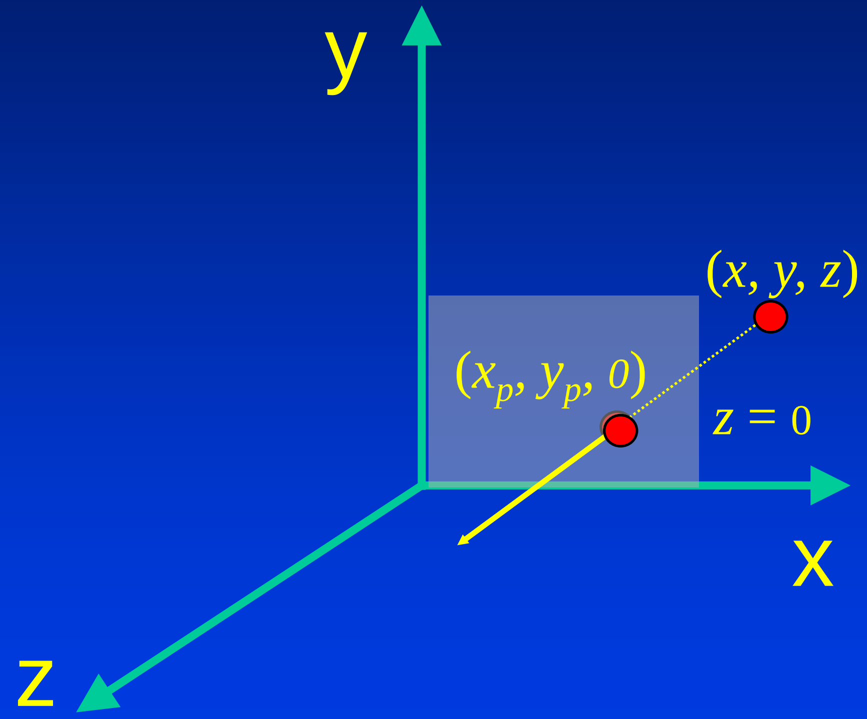
- Preserves X and Y coordinates.
- Preserves both distances and angles.



Parallel Orthographic Projection

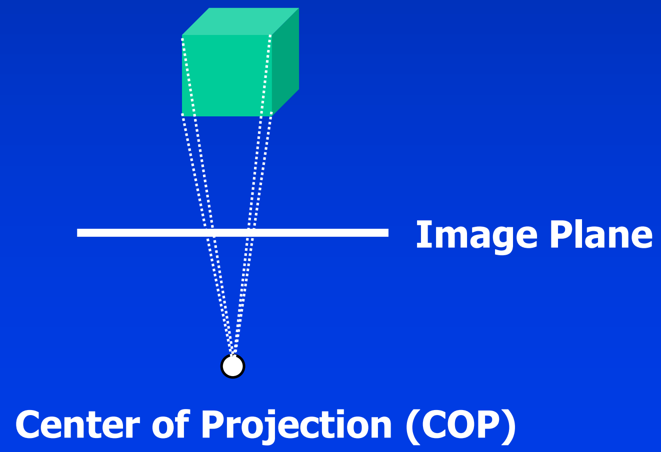
- $x_p = x$
- $y_p = y$
- $z_p = 0$

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



Perspective Projection

- Only preserves parallel lines that are parallel to the image plane.
- Line segments are shorten by distance.

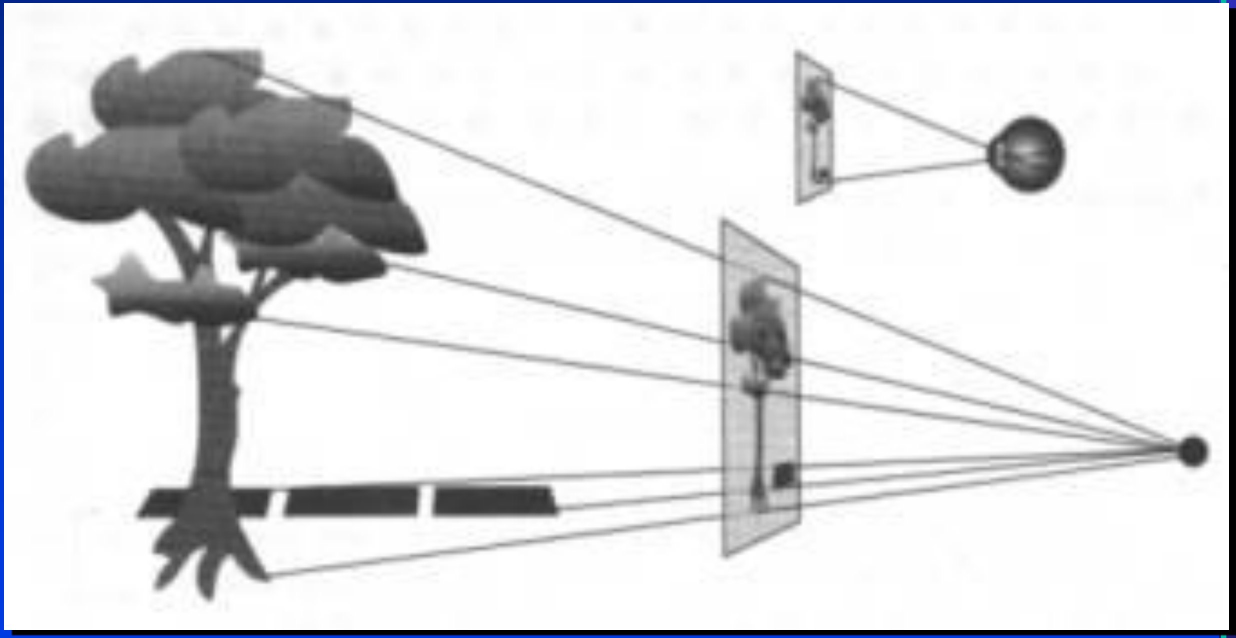


Perspective Projection

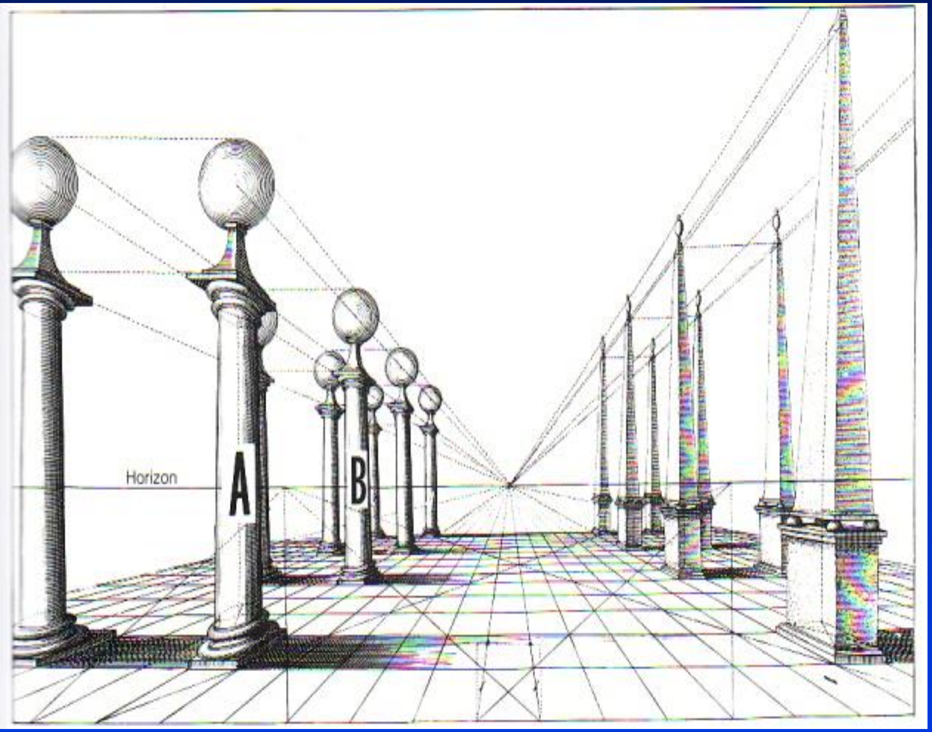
Parallel lines converge

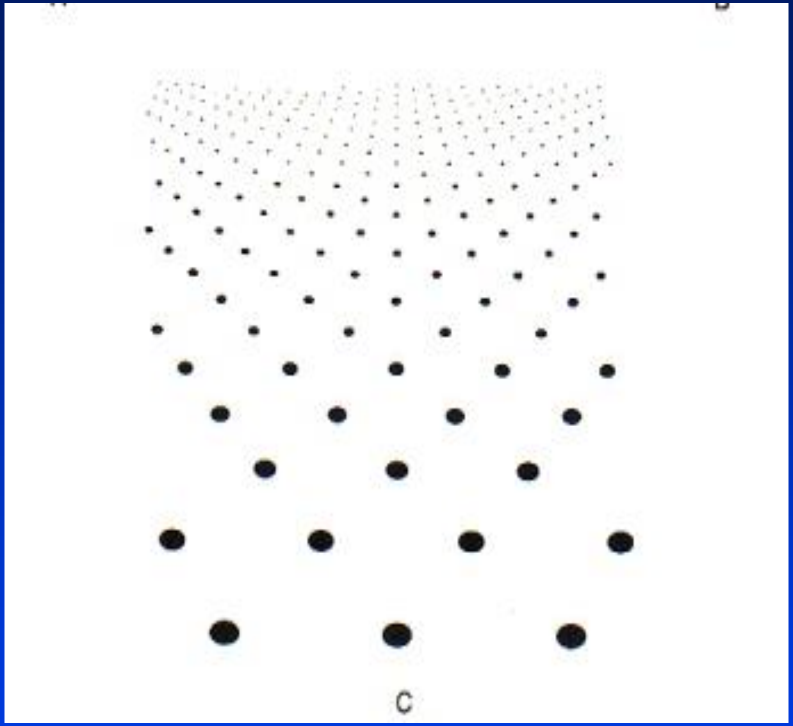
Distant objects appear smaller

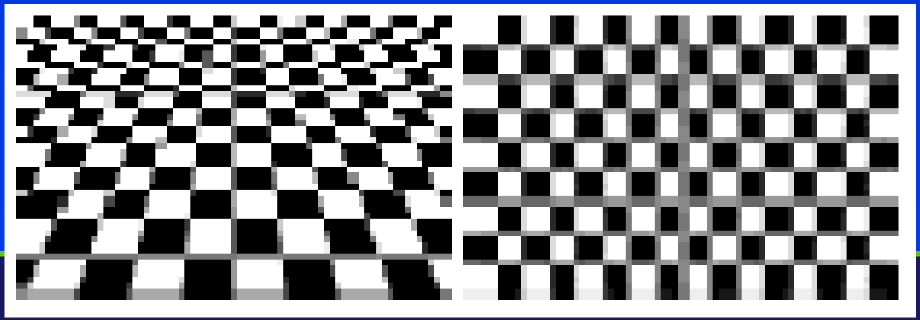
Textured elements become smaller with distance



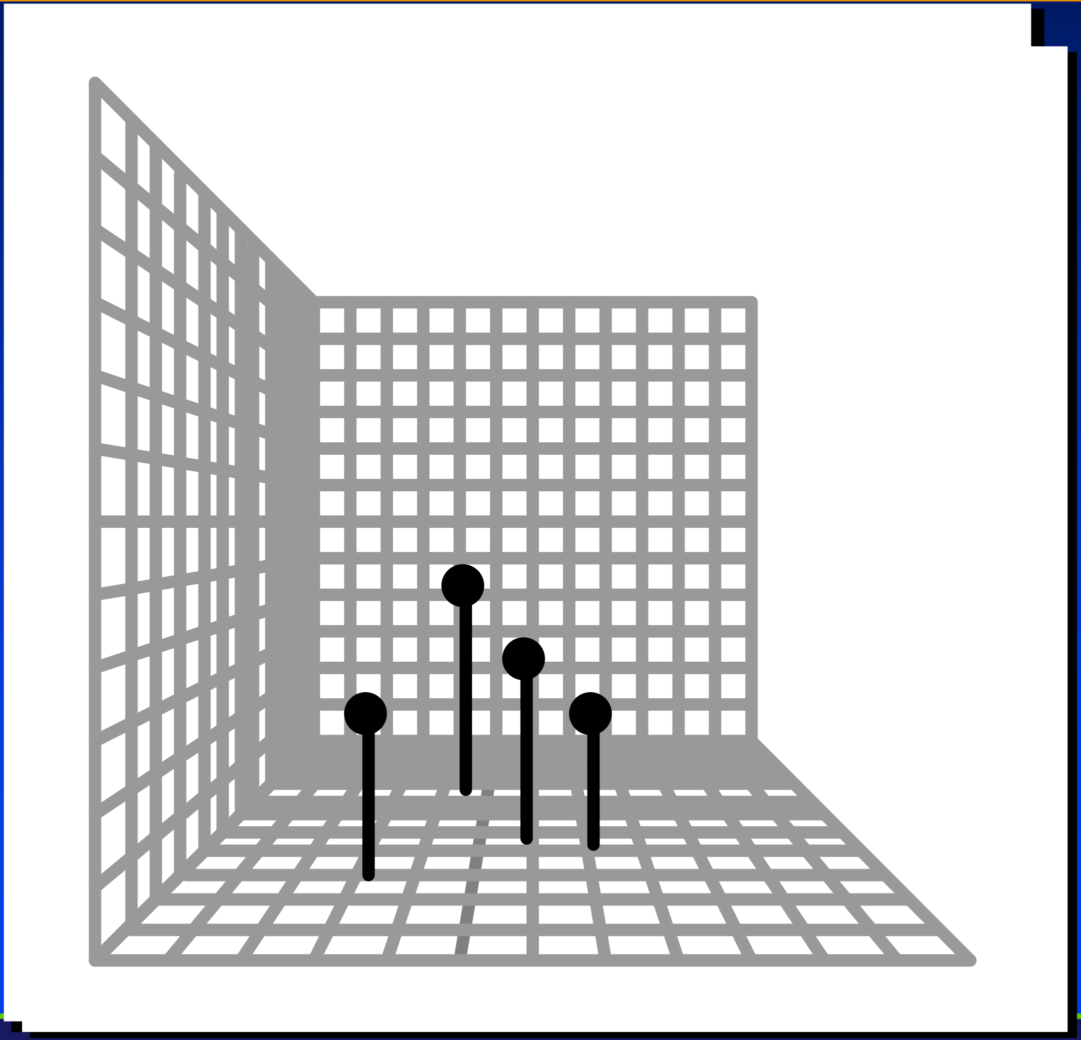
Perspective Cues



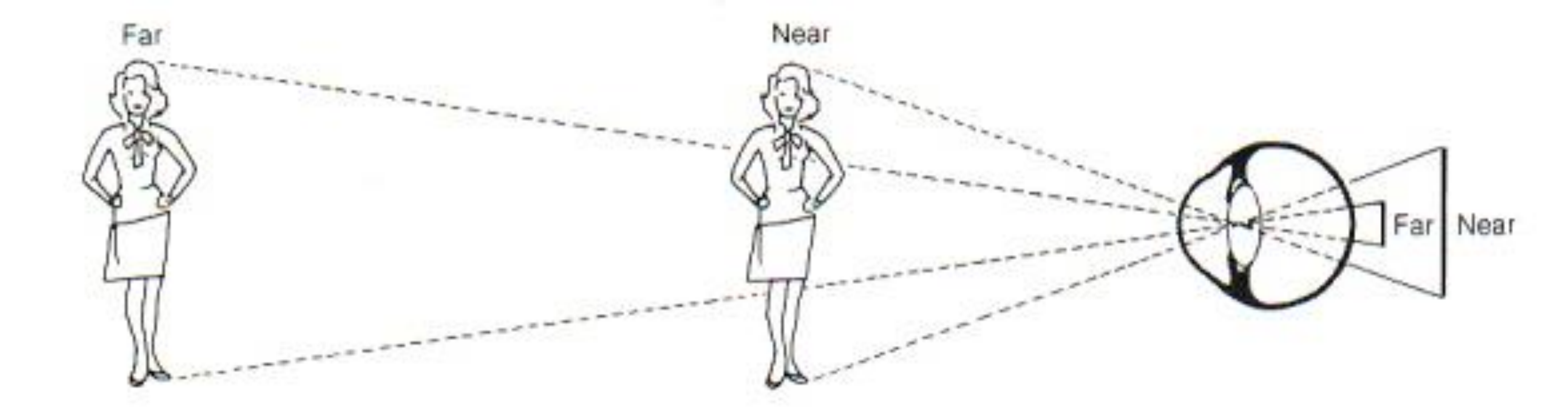




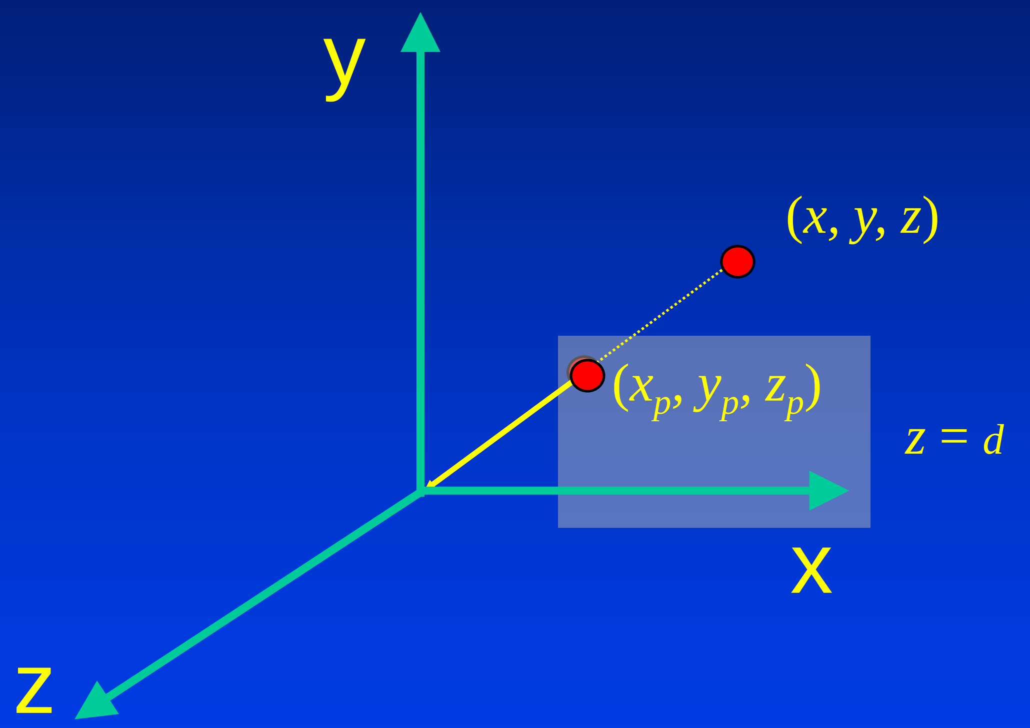
Perspective Cues



Perspective Cues

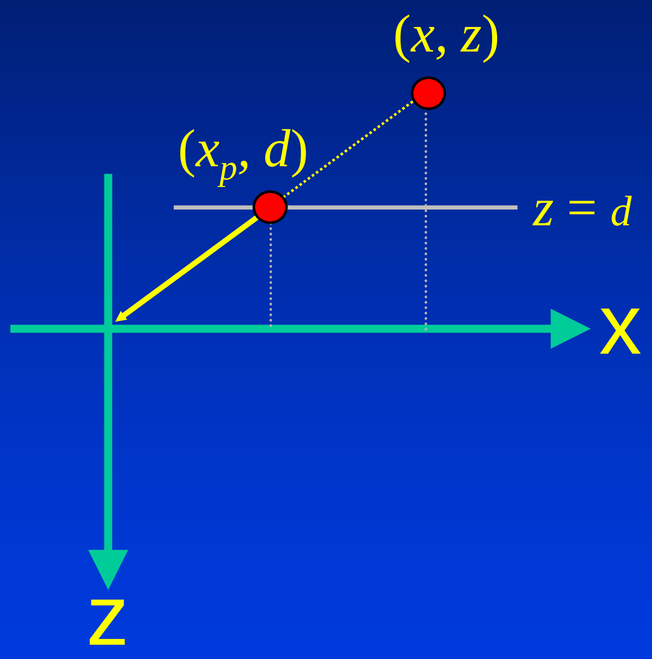


Perspective Projection



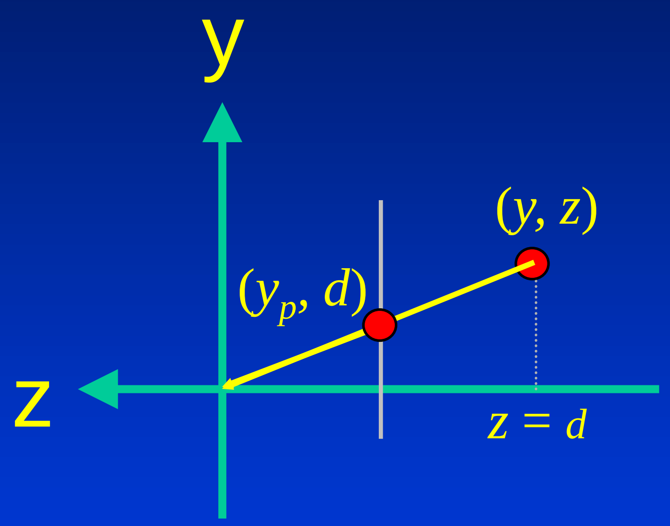
Perspective Projection

- $z_p = d$
- $x_p = (x \cdot d) / z$



Perspective Projection

- $z_p = d$
- $y_p = (y \cdot d) / z$



Perspective Projection

- $x_p = (x \cdot d) / z = x/(z/d)$
- $y_p = (y \cdot d) / z = y/(z/d)$
- $z_p = d \qquad = z/(z/d)$

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1/h & 0 & 0 & 0 \\ 0 & 1/h & 0 & 0 \\ 0 & 0 & 1/h & 0 \\ 0 & 0 & 0 & 1/h \end{bmatrix} \begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix}$$

Viewing in 3D

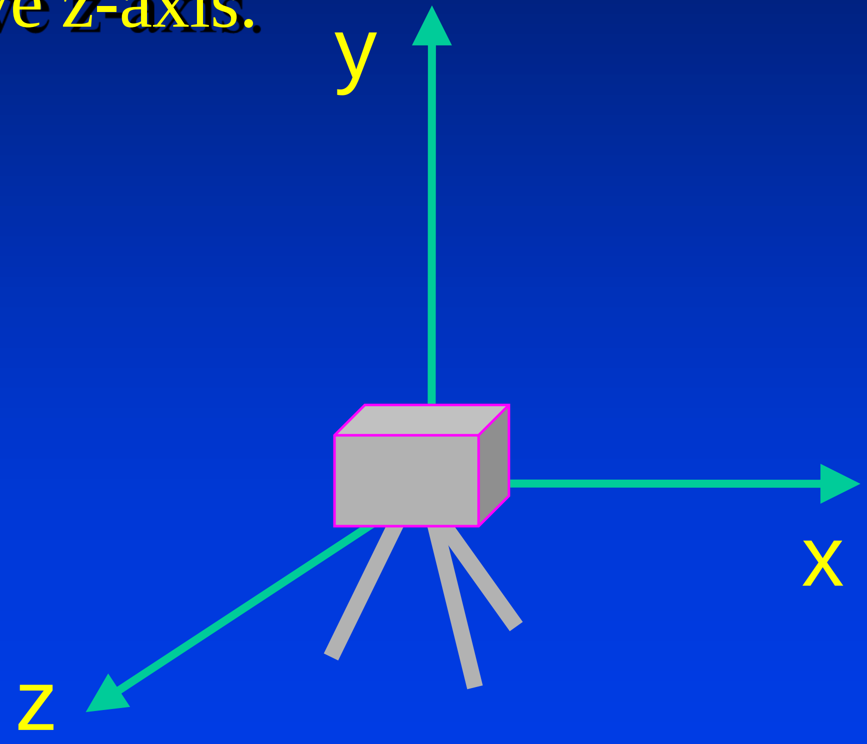
- Planar Geometric Projections
- Parallel Orthographic Projections
- Perspective Projections
- Projections in OpenGL

Viewing in 3D

- Planar Geometric Projections
- Parallel Orthographic Projections
- Perspective Projections
- Projections in OpenGL
  - Positioning of the Camera
  - Define the view volume

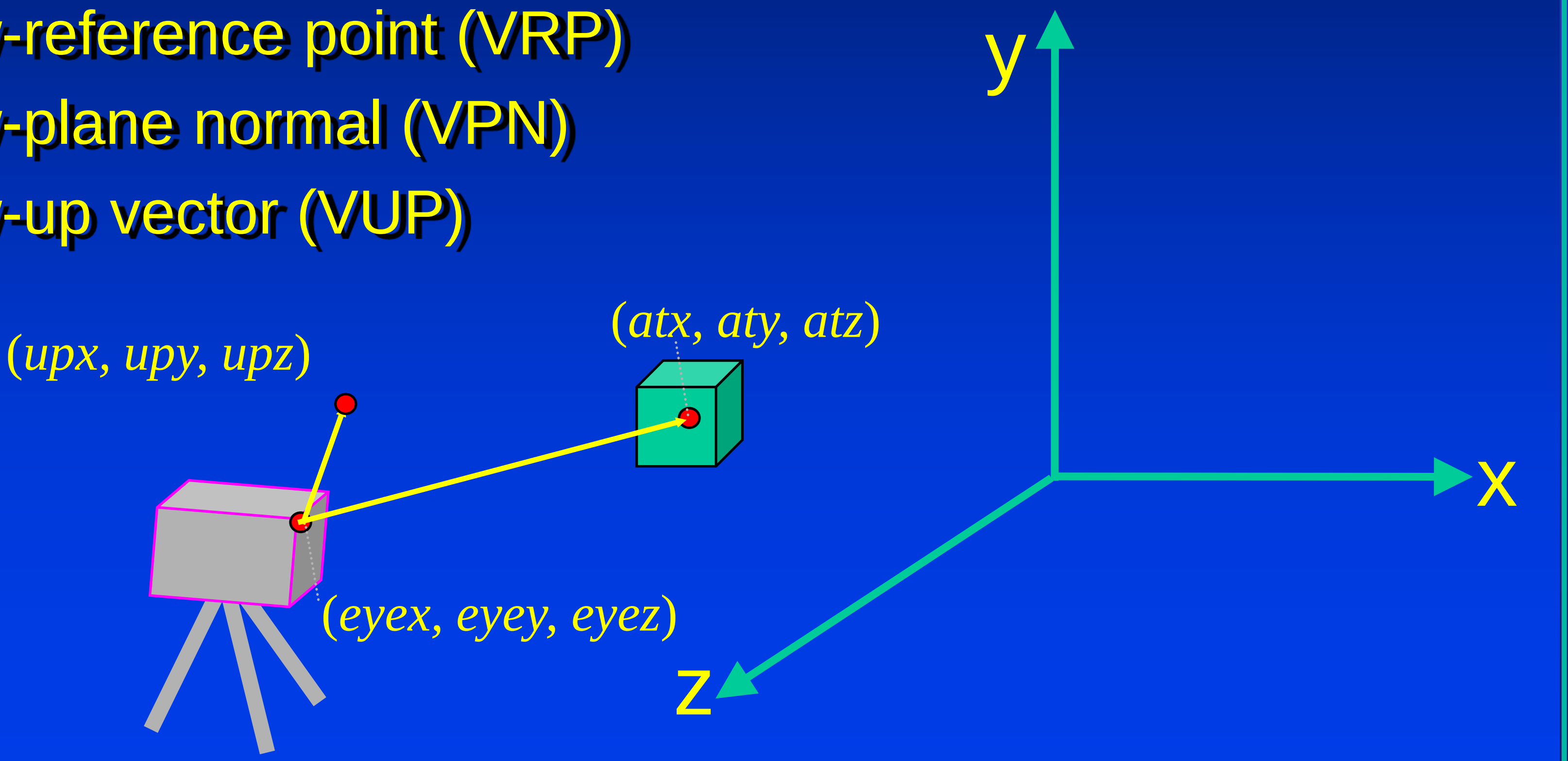
Positioning the Camera

- By default, the camera is placed at the origin pointing towards the negative z-axis.



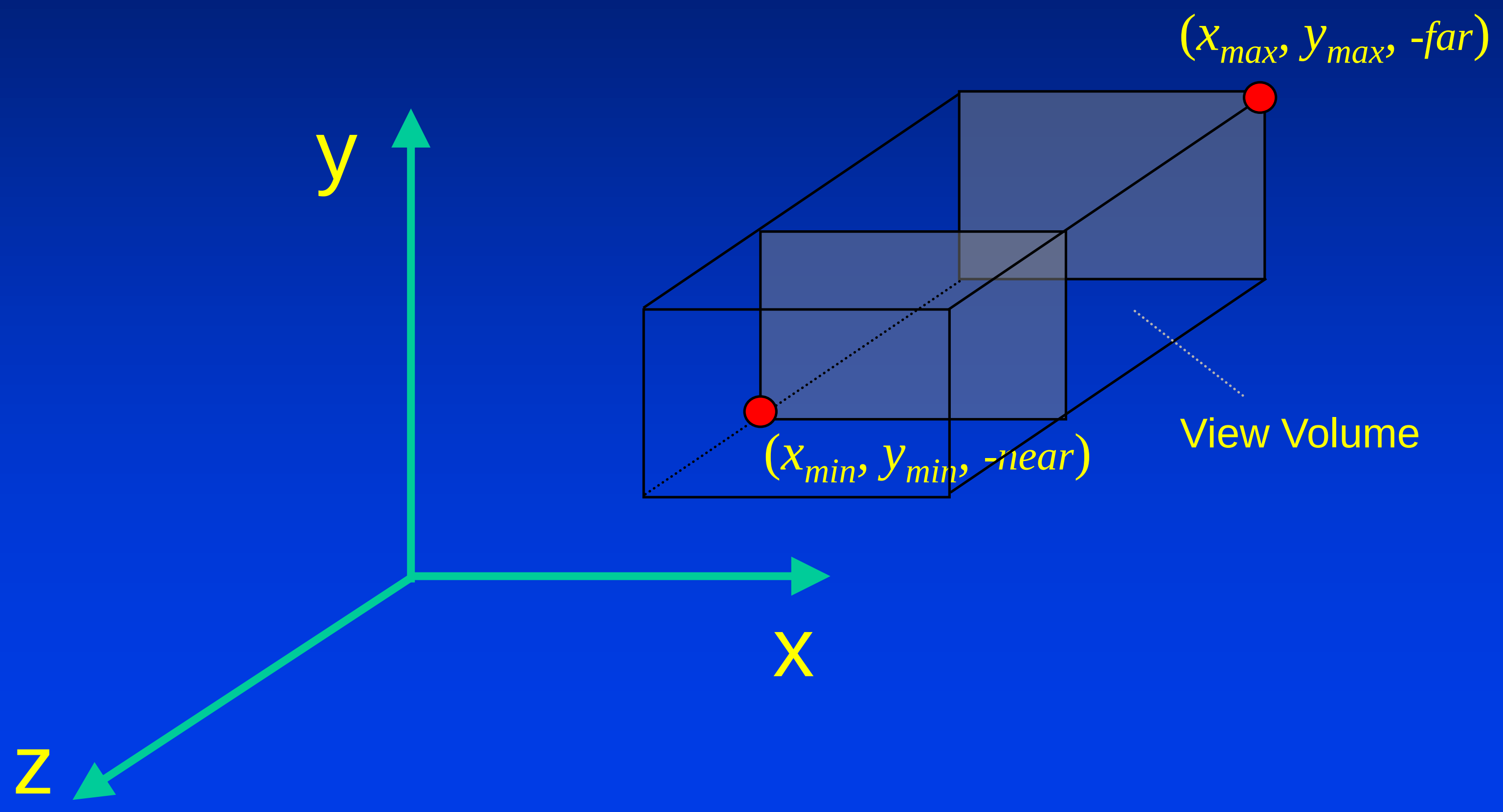
Positioning the Camera

- OpenGL Look-At Function
  gluLookAt(eyex, eyey, eyez, atx, aty, atz, upx, upy, upz)
- View-reference point (VRP)
- View-plane normal (VPN)
- View-up vector (VUP)



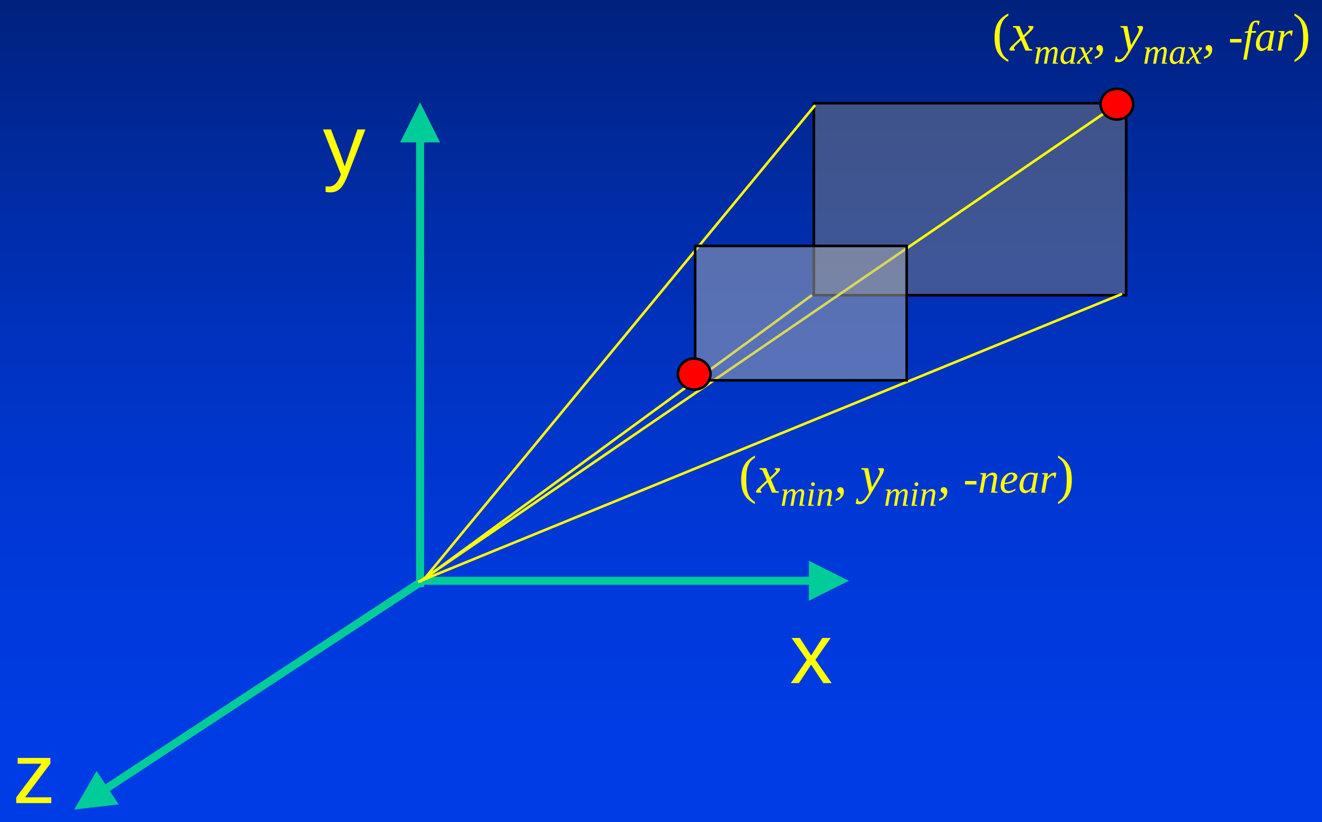
Defining the Parallel View Volume

glOrtho(xmin, xmax, ymin, ymax, near, far)



Defining the Perspective View Volume

glFrustum(left, right, bottom, top, near, far)



Defining the Perspective View Volume

gluPerspective(fovy, aspect, near, far)