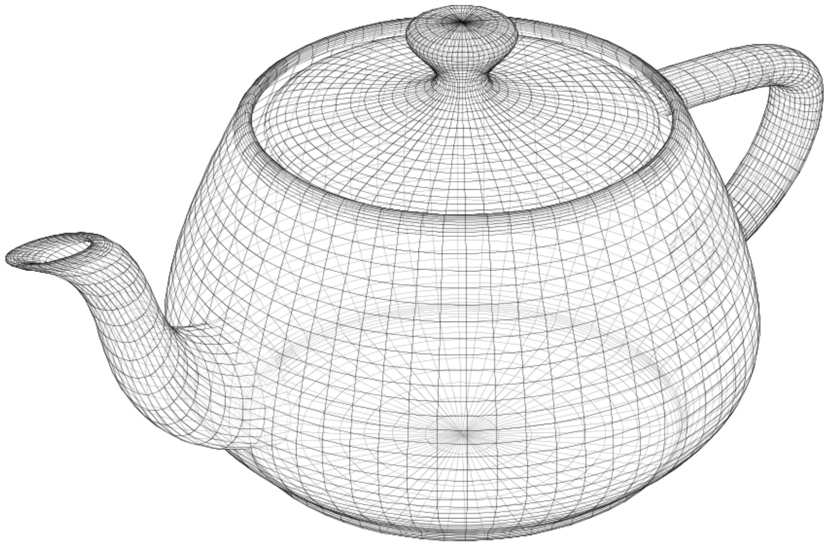


Affine Transformations
Rotations
3D Transformations

CS 418: Interactive Computer Graphics

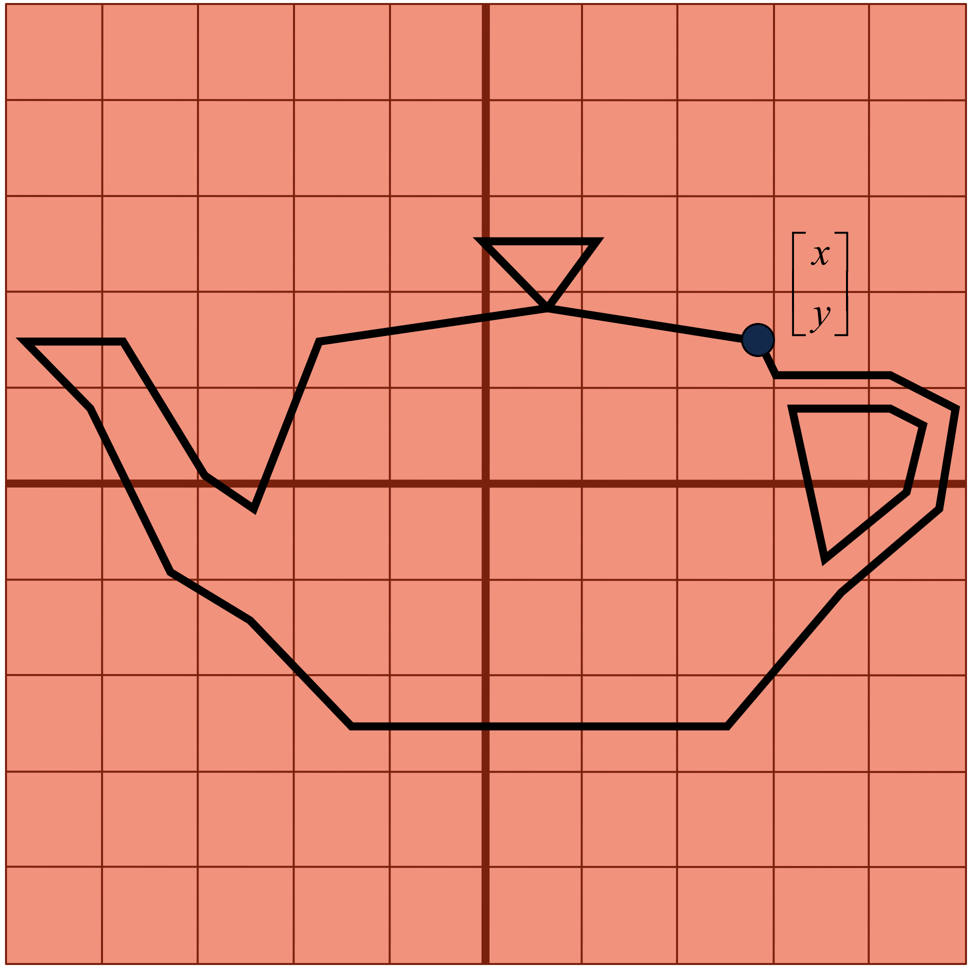
Professor Eric Shaffer

Slides courtesy of Professor John Hart

ILLINOIS

2-D Rotation

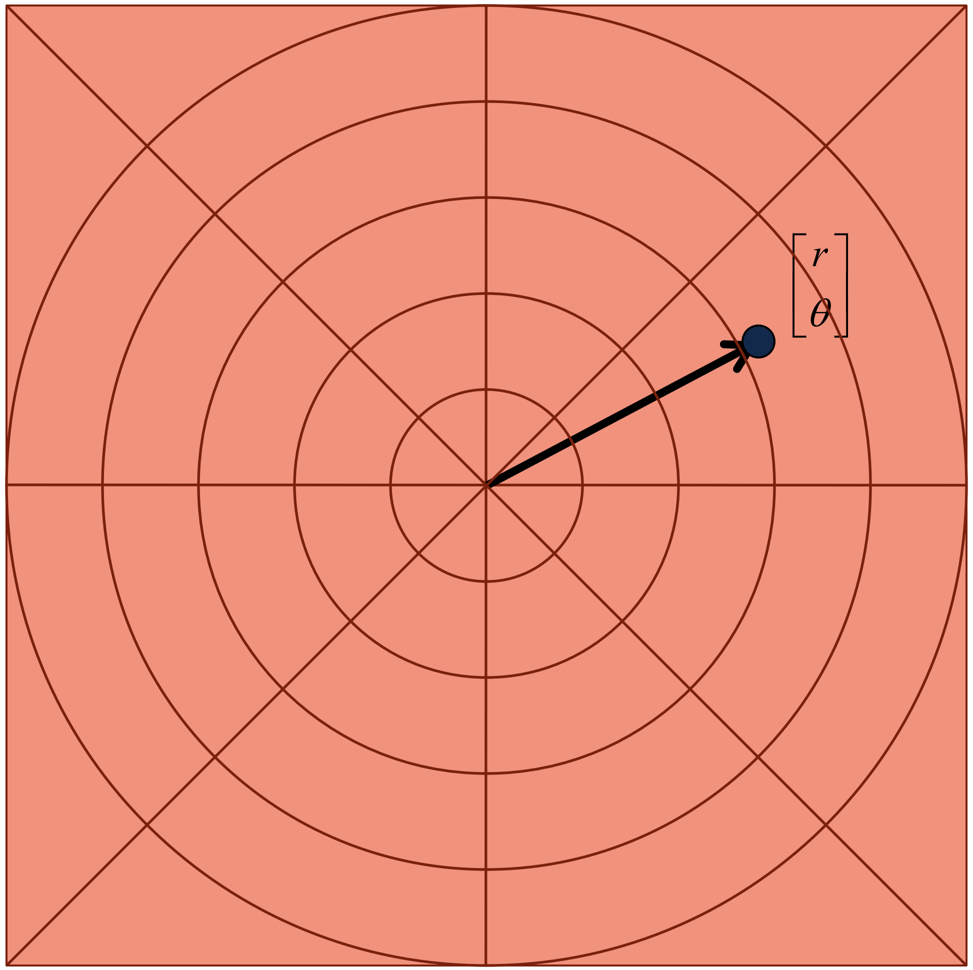
- Pick a point (x,y)



2-D Rotation

- Pick a point (x,y)
- Assume polar coords

  $x = r \cos \theta, y = r \sin \theta$

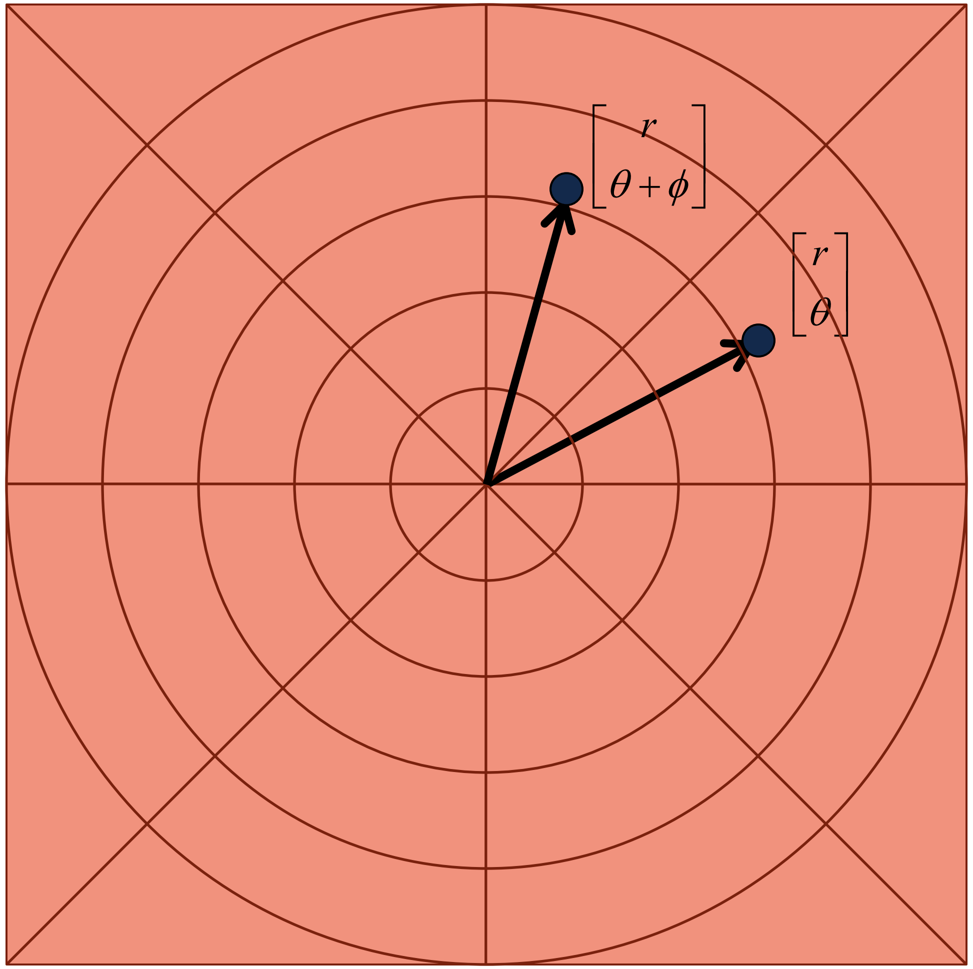


2-D Rotation

- Pick a point (x,y)
- Assume polar coords

  $x = r\cos\theta,\ y = r\sin\theta$
- Rotate about origin by ϕ

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



2-D Rotation

- Pick a point (x,y)
- Assume polar coords

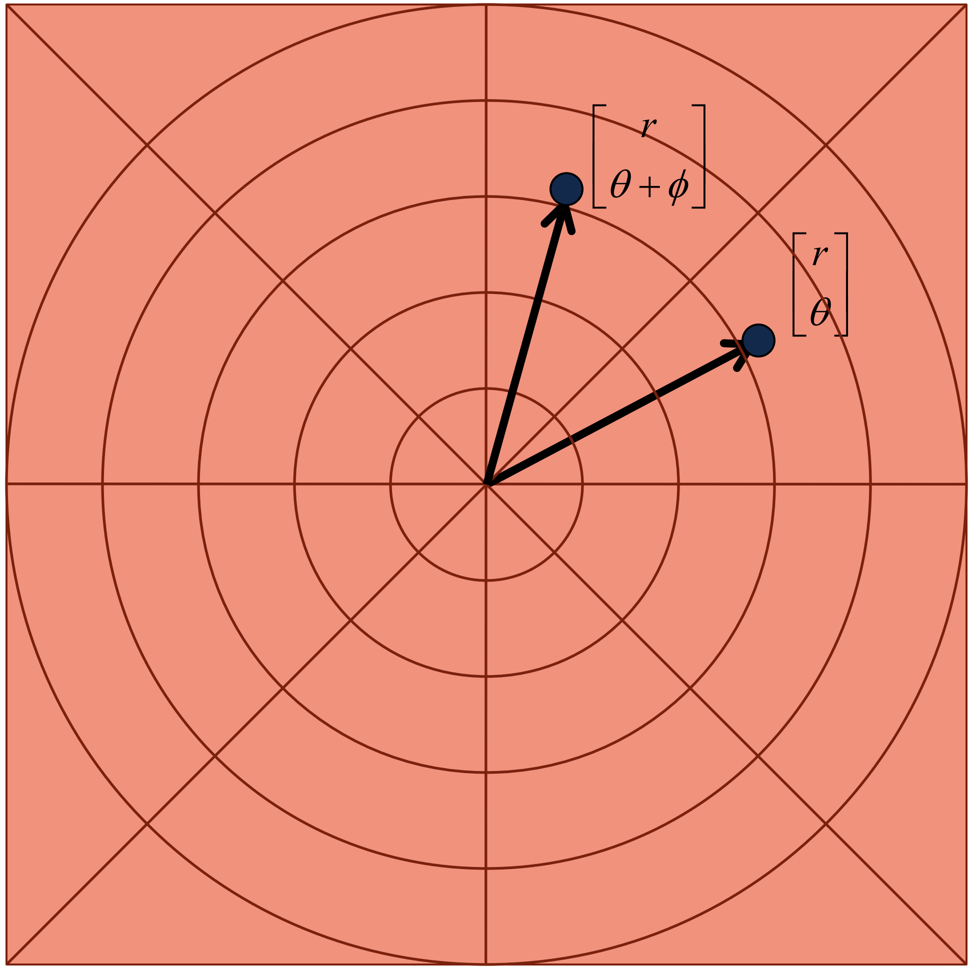
  $x = r \cos\theta,\ y = r \sin\theta$

- Rotate about origin by ϕ

  $x' = r \cos\theta+\phi,\ y' = r \sin\theta+\phi$

- Recall trig. identities

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



2-D Rotation

- Pick a point (x,y)
- Assume polar coords

  $x = r\cos\theta,\ y = r\sin\theta$

- Rotate about origin by ϕ

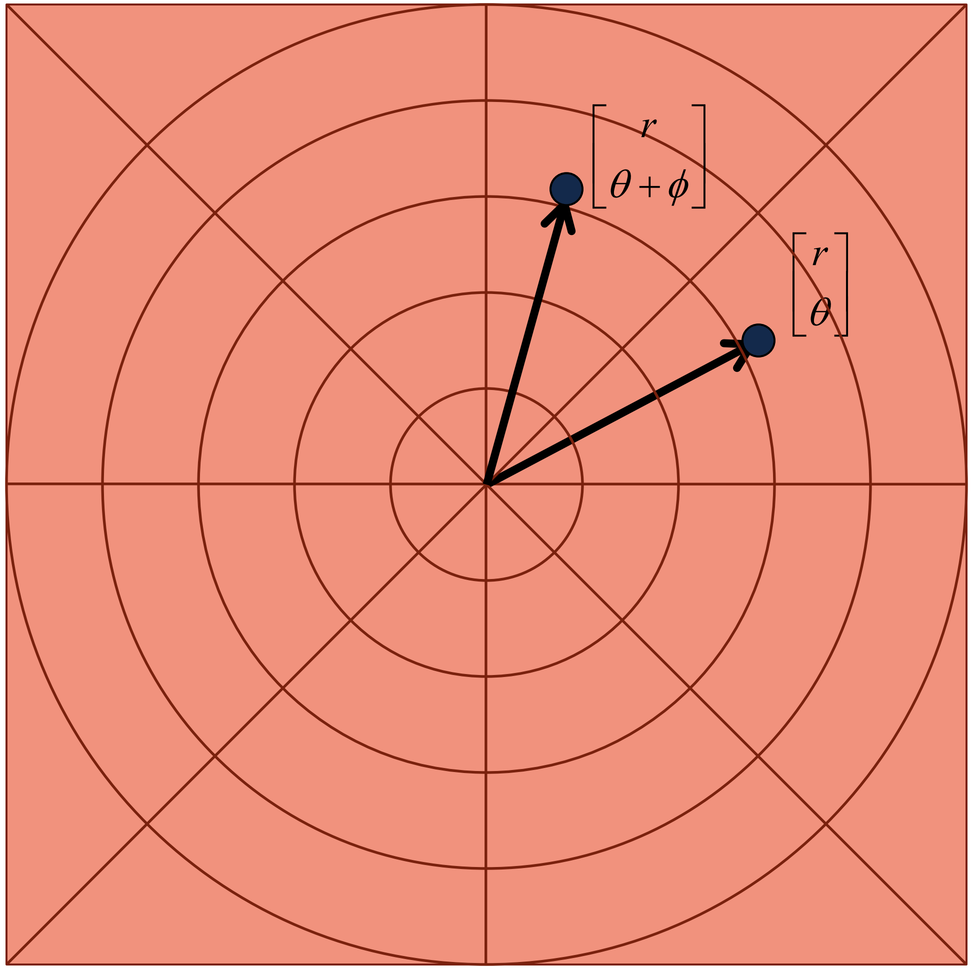
  $x' = r\cos\theta+\phi,\ y' = r\sin\theta+\phi$

- Recall trig. identities

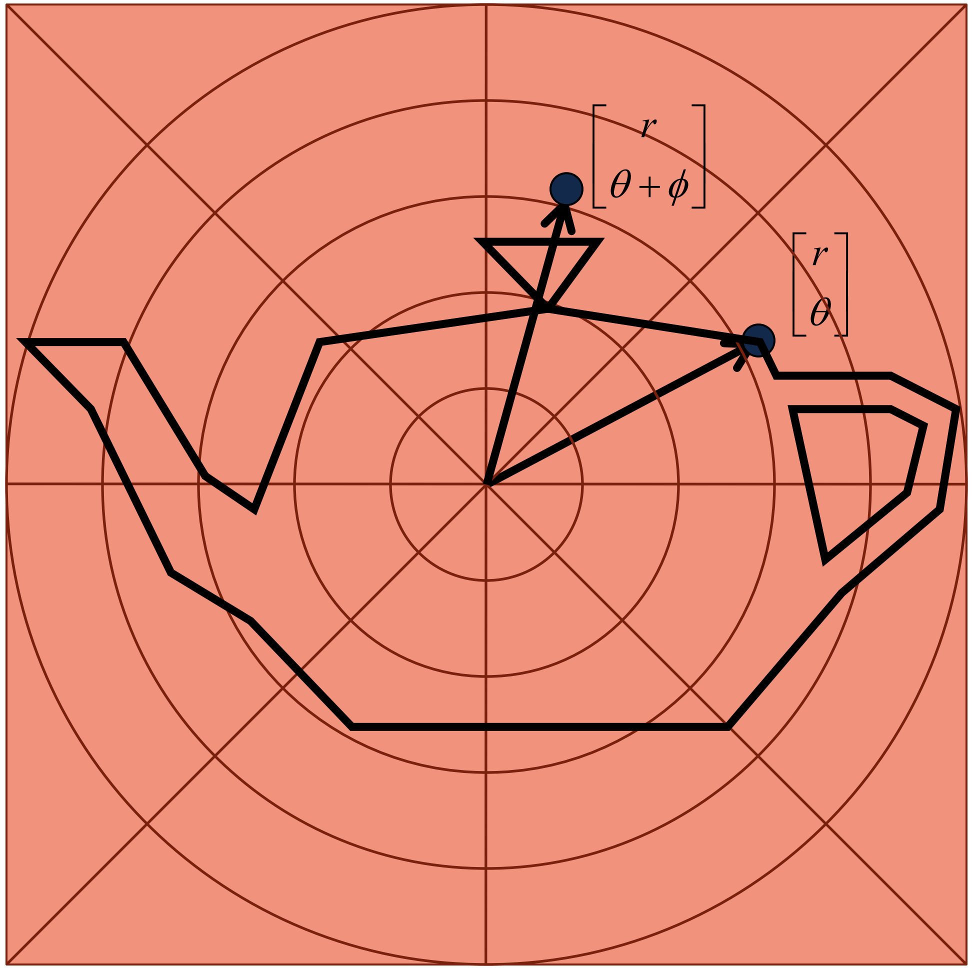
  $x' = r\,(\cos\theta\cos\phi - \sin\theta\sin\phi)$
  $y' = r\,(\sin\theta\cos\phi + \cos\theta\sin\phi)$

- Rearrange terms

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



2-D Rotation



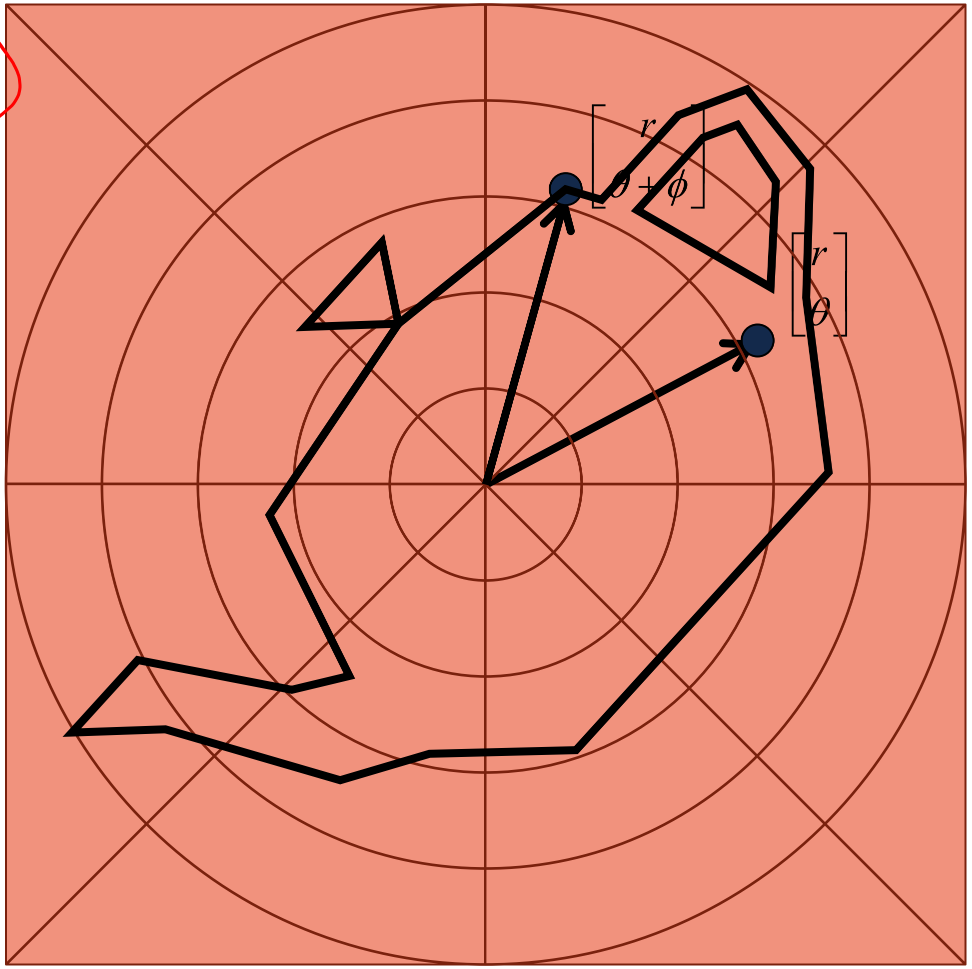
$$x' = \cos\phi\,(r\cos\theta) - \sin\phi\,(r\sin\theta)$$
$$y' = (r\cos\theta)\sin\phi + (r\sin\theta)\cos\phi$$

2-D Rotation

$$x' = \cos\phi\,\overbrace{(r\cos\theta)}^{x} - \sin\phi\,\overbrace{(r\sin\theta)}^{y}$$

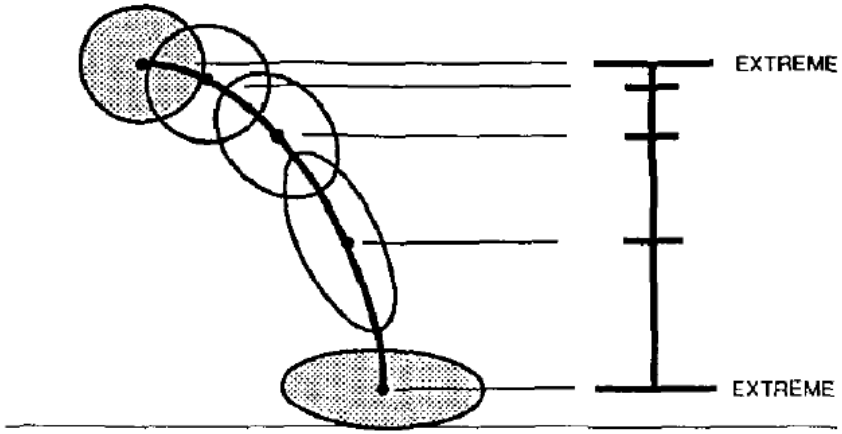
$$y' = \underbrace{(r\cos\theta)}_{x}\,\sin\phi + \underbrace{(r\sin\theta)}_{y}\,\cos\phi$$

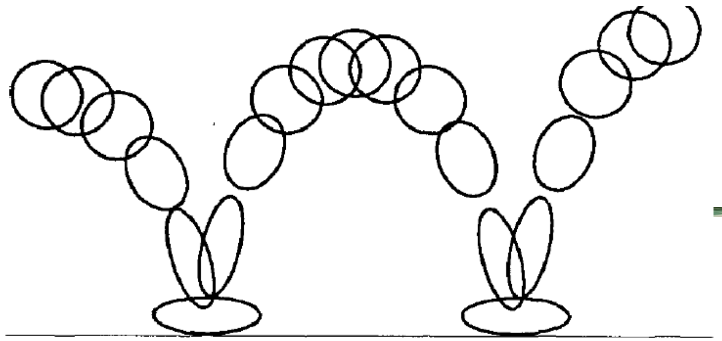
$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

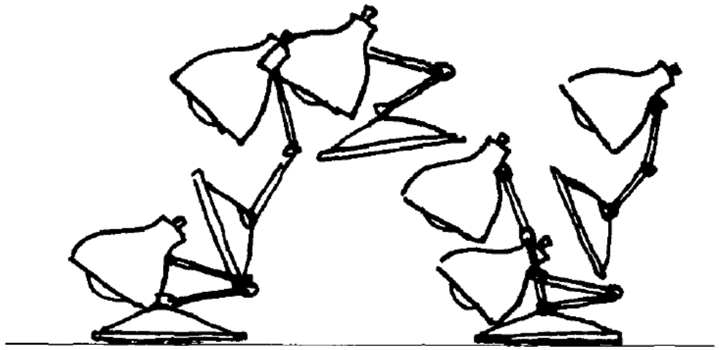


Squash & Stretch

From: *John Lasseter: "Principles of Traditional Animation Applied to 3-D Computer Animation" Proc. SIGGRAPH 87*







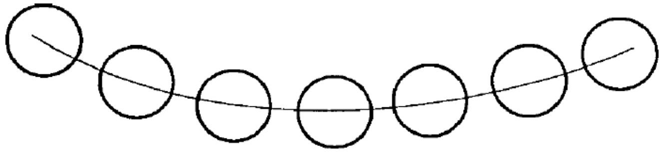


FIGURE 4b. Strobing occurs in a faster action when the object's positions do not overlap and the eye perceives seperate images.

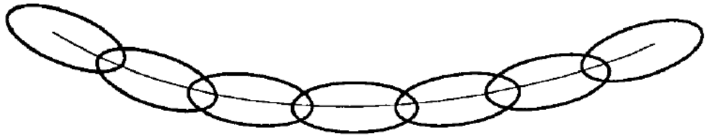


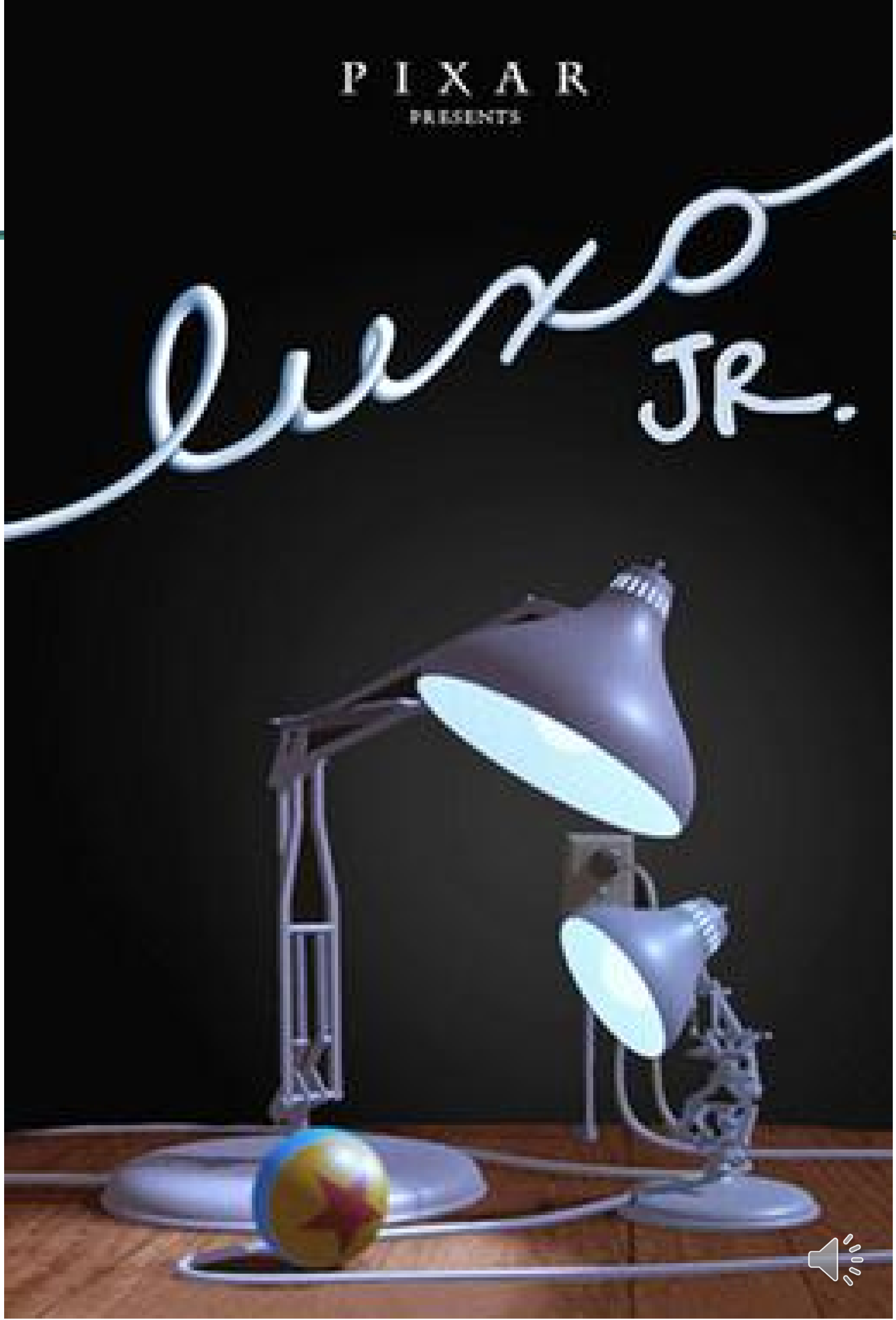
FIGURE 4c. Stretching the object so that it's positions overlap again will relieve the strobing effect.

Luxo Jr.

Directed by	John Lasseter
Written by	John Lasseter
Production company	Pixar
Release date	Aug. 17, 1986 (SIGGRAPH)

Luxo Jr.

- Breakthrough demonstration of computer animation in 1987
- Demonstrates shadow maps and Renderman surface shaders
    - Lamp and ball behavior implemented using affine transformations
- Took about 5 months of work from Pixar’s (then) tiny animation team
- It was the first CGI film nominated for an Academy Award.
- In 2014, *Luxo Jr.* was deemed "culturally, historically, or aesthetically significant" by the Library of Congress and selected for preservation in the National Film Registry.



3-D Affine Transformations

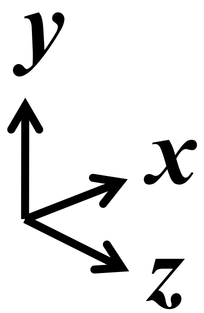
General Form (with homogenous coordinates)

$$\begin{bmatrix} d & e & f & a \\ g & h & i & b \\ j & k & l & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} dx + ey + fz + a \\ gx + hy + iz + b \\ jx + ky + lz + c \\ 1 \end{bmatrix}$$

Translation

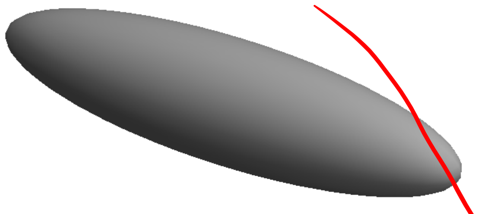
$$\begin{bmatrix} 1 & & & a \\ & 1 & & b \\ & & 1 & c \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + a \\ y + b \\ z + c \\ 1 \end{bmatrix}$$

Scale

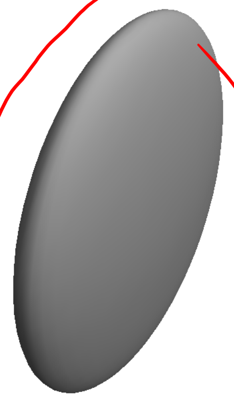


$$\begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} ax \\ by \\ cz \\ 1 \end{bmatrix}$$

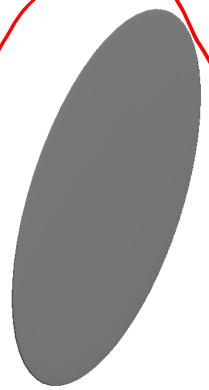
Uniform Scale
$a = b = c = ¼$



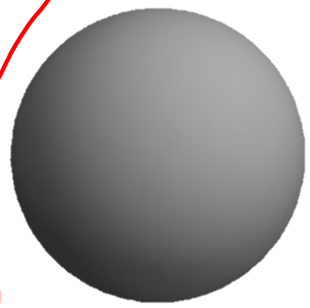
Stretch
$a = b = 1$, $c = 4$



Squash
$a = b = 1$, $c = ¼$



Project
$a = b = 1$, $c = 0$



Invert
$a = b = 1$, $c = -1$

3-D Rotations

- About *x*-axis
  - rotates *y* → *z*

$$\begin{bmatrix} 1 & & & \\ & \cos\theta & -\sin\theta & \\ & \sin\theta & \cos\theta & \\ & & & 1 \end{bmatrix}$$

- About *y*-axis
  - rotates *z* → *x*

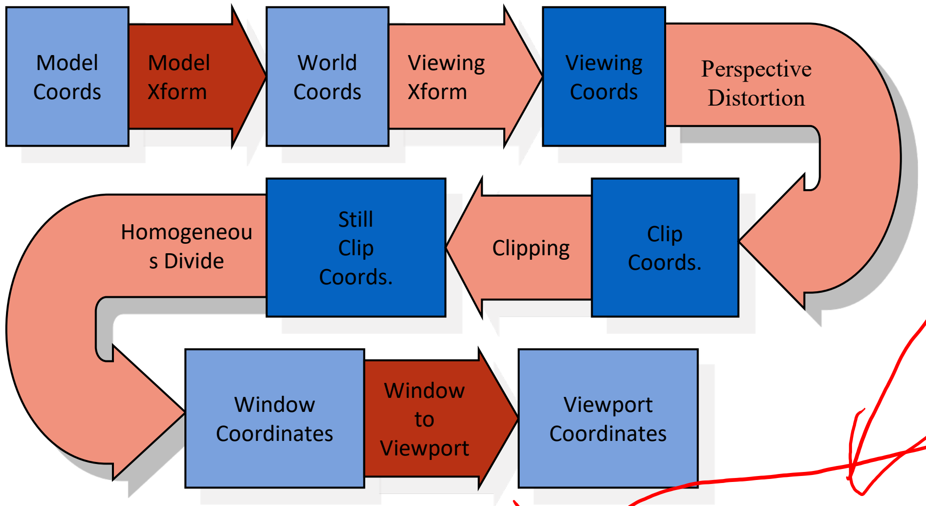
$$\begin{bmatrix} \cos\theta & & \sin\theta & \\ & 1 & & \\ -\sin\theta & & \cos\theta & \\ & & & 1 \end{bmatrix}$$

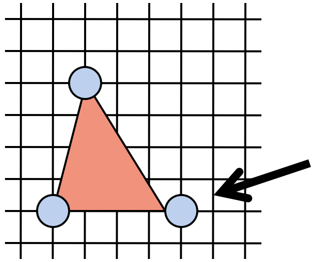
- About *z*-axis
  - rotates *x* → *y*

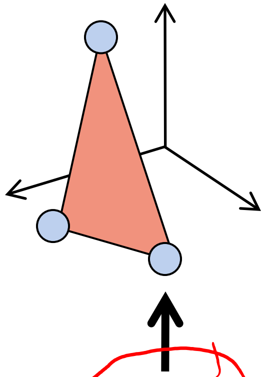
$$\begin{bmatrix} \cos\theta & -\sin\theta & & \\ \sin\theta & \cos\theta & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

- Rotations do not commute!

Graphics Pipeline







$$\begin{bmatrix} x_s \\ y_s \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \text{W2V} \end{bmatrix} \begin{bmatrix} \text{Persp} \end{bmatrix} \begin{bmatrix} \text{View} \end{bmatrix} \begin{bmatrix} \text{Model} \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix}$$